

SH30F9x71/SH30F9x20/SH30F9x21

32-bit MCU Based on ARM® Cortex™-M0+ Core

Version: 2.2

2024.05



1. Features

- ARM 32-bit Cortex-M0+ CPU core
 - With frequency up to 48MHz
 - Support hardware multiplication
 - High-speed memory and instruction acceleration to improve execution efficiency
- Memories
 - Up to 256K/128K bytes of FLASH
 - Maximum 16K bytes of SRAM
 - Maximum 4K bytes built-in EEPROM-like area + 8K bytes BOOT ROM
 - 1024 bytes of OTP(One Time Programming) area
- Clock, reset and power management
 - 2.0V - 5.5V power supply (VDD) and I/O pin
 - Power-on reset (POR), brownout detection (BOD), low-voltage reset (LVR)
 - Independent watchdog timer (IWDT), window watchdog timer (WWDT)
 - 4M - 12MHz crystal oscillator/ceramic oscillator
 - 32kHz crystal oscillator
 - Internal high frequency RC oscillator: 24MHz (1% accuracy under full temperature range)
 - Internal low frequency RC oscillator:128kHz
 - Integrate PLL generates high frequency clock
 - Clock Security Monitor (CSM)
- Low power modes
 - Sleep mode(only stop CPU)
 - Stop mode(CPU stops, and peripheral clock is cut off; after exiting shutdown, system clock returns to HSI or LSI)
- Maximum 28 channels of 12-bit ADC
- DMA controller
 - 4 independent DMA channels
 - Any combination of memory and peripheral transfers
- Up to 61 fast GPIO ports
 - Up to 61 available GPIO ports
 - Internal pull-up resistors for all I/O
 - 8 pins provide strong capacity that sinking current
- 16 external interrupt input channels (EXTI0~EXTI15)
 - Up to 16 channels, all I/O mappable on external interrupt vectors
 - Support edge trigger, level trigger, and software trigger
- Interrupt sources
 - 16 Cortex-M0+ interruptions
 - TIM0~3, PCA0~3, PWM0~3
 - EXTI0~15, UART0~3, SPI0/1, TWI
 - ADC, LED, Touch Key
 - CSM, BOD, DMA, CRC
 - Maximum 32 channel Touch Key
 - LED driver
 - Support 8x8 row and column matrix drive
 - Support 7x8, 6x7, 5x6, 4x5 LED serial dot matrix drive
 - Maximum support 8x8+7x8 dot matrix
 - Support constant current drive
 - LCD driver
 - 8X36 dots (1/8 duty, 1/4 bias)
 - 6X38 dots (1/6 duty, 1/4 bias or 1/3 bias)
 - 5X39 dots (1/5 duty, 1/3 bias)
 - 4X40 dots (1/4 duty, 1/3 bias)
 - Four 16-bit PCA0-3 with two compare/capture modules
 - Capable of cascade-connected as two 32-bit PCAs
 - Four 16-bit PWM timers PWM0/1/2/3
 - Three 16-bit timers/counters(TIM0/TIM1/TIM2)
 - One 32-bit timers/counters(TIM3)
 - One 24-bit self-reducing system timer (SYSTICK)
 - Multiple serial communication interfaces
 - 4 universal asynchronous receiver transmitter (UART0/1/2/3)
 - 2 serial peripheral interfaces (SPI0/1)
 - 1 two-wire serial interfaces (TWI)
 - CRC hardware module, applicable for code verification, data verification
 - SRAM's March-C/March-X detection algorithm
 - 96-bit unique ID
 - Support 2-wire SWD interface
 - Operating temperature: -40~105°C
 - Package:
 - LQFP64(10X10)
 - LQFP64(7X7)
 - TQFP48
 - LQFP44
 - LQFP32
 - QFN32(5X5)



2. Overview

SH30F9/SA0 series is a high-performance 32-bit microcontroller based on ARM Cortex-M0+ core, operating at a frequency of up to 48MHz. Due to the memory instruction acceleration structure used, it can achieve a performance equivalent to about 0 wait state program execution from flash memory at top frequency.

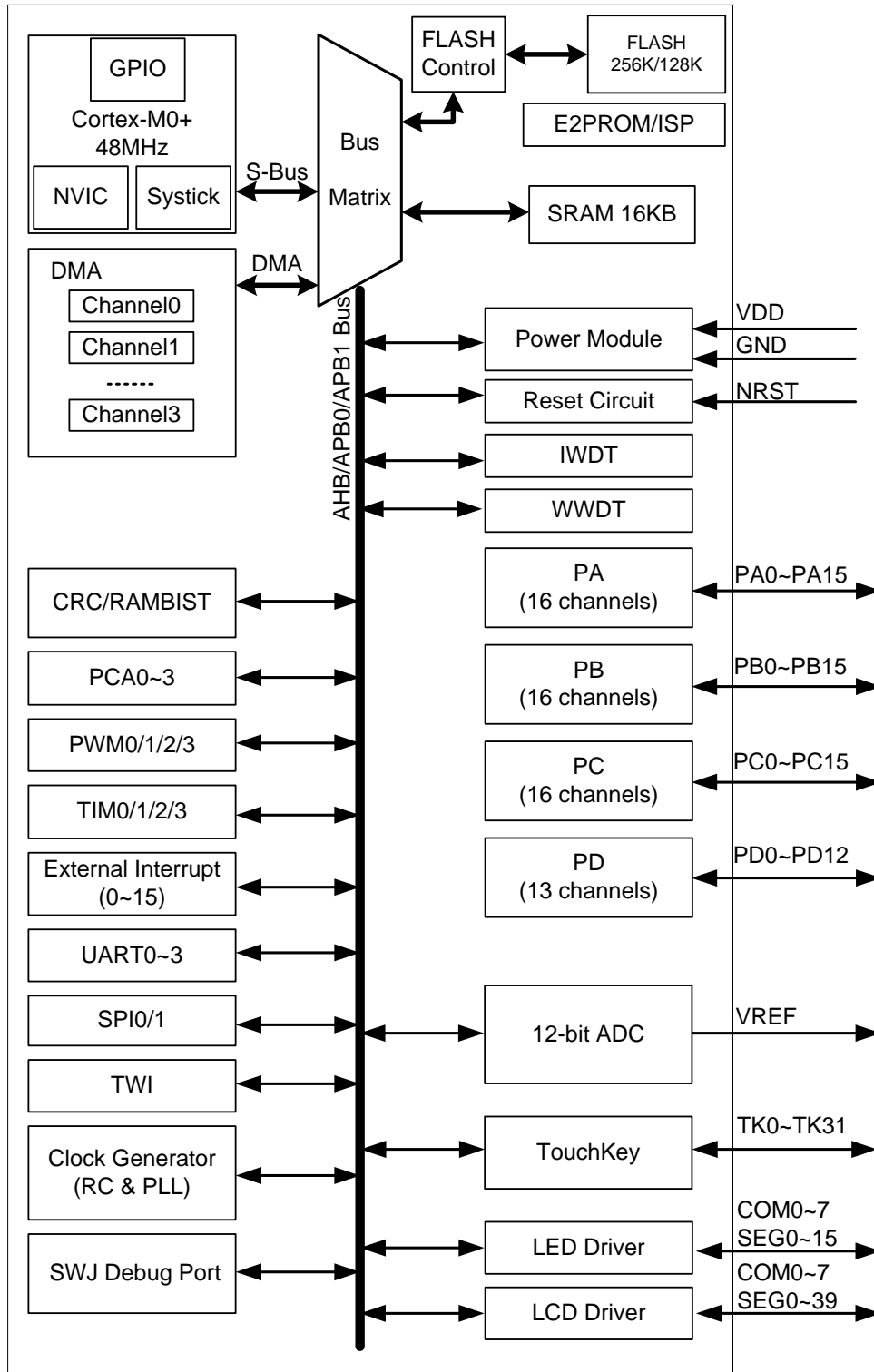
The Cortex-M0+ core is mature and reliable, has a complete ecosystem, is able to give efficient instructions and supports the traditional control system to move forward to the network intelligent control system. SH30F9/SA0 series fully exploits the advantages of the Cortex-M0+ core and focuses on providing highly integrated, highly reliable monolithic integrated circuits and providing highly cost-effective system solutions of microcontroller. It is especially suitable for the development of smart home and major appliance control solutions.

On the aspect of calculation, the Cortex-M0+ core has built-in a hardware 32-bit multiplier, which has powerful computing power. Additionally, SH30F9/SA0 series has built-in four programmable counter array (PCA) and four general 16-bit PWM timer, and integrated a 12-bit multi-channel high-speed ADC, which makes its application more extensive.

SH30F9/SA0 series supports low power application mode. It provides 8/16/32-bit CRC code and data verification, SRAM self-test unit (RAMBIST), IWDT, WWDT, low voltage reset circuit, brownout detection circuit, and other auxiliary modules, which can effectively improve code security and system reliability, and can be applied to occasions with high safety requirements.



3. Block Diagram





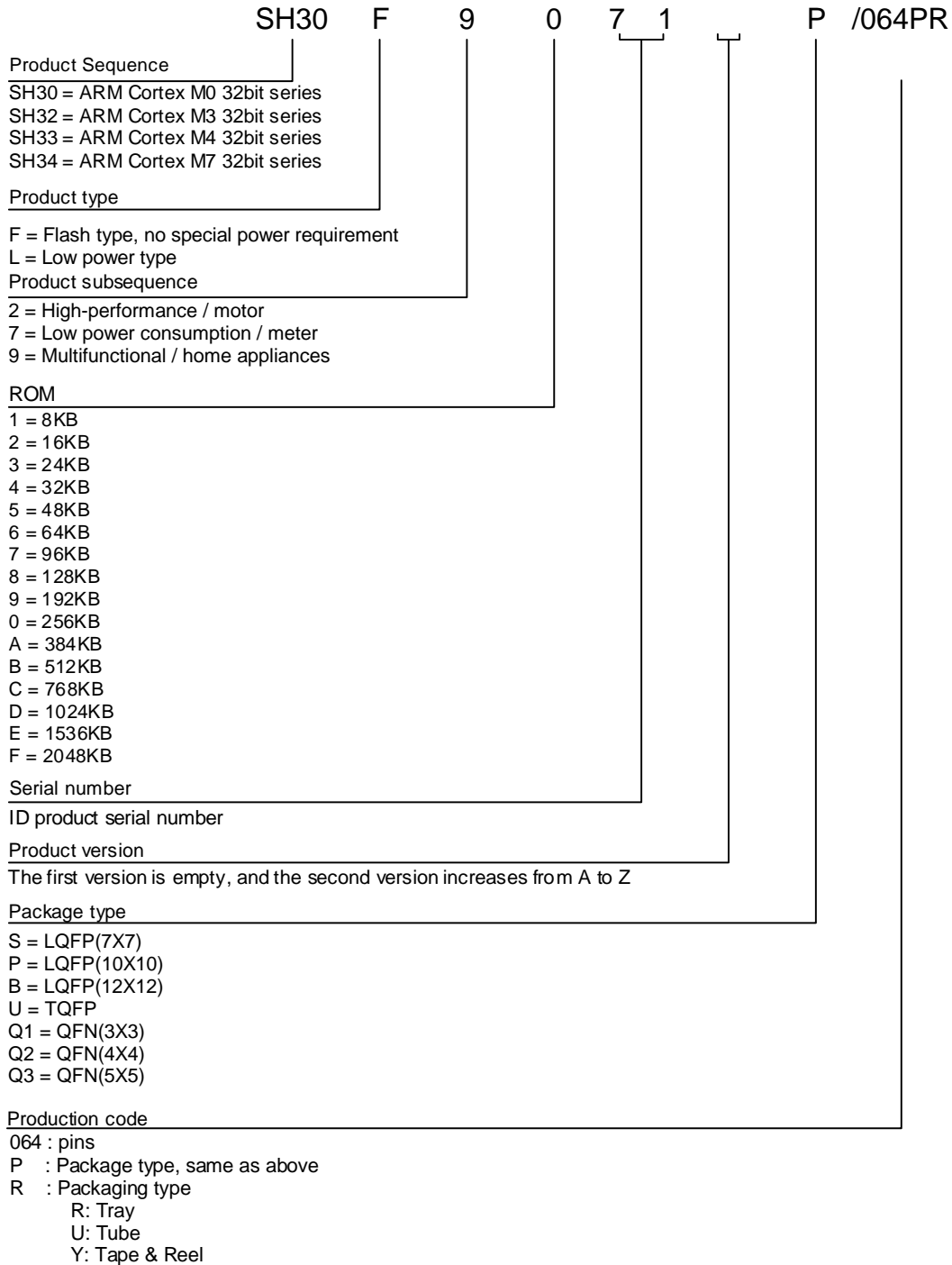
SH30F9/SA0 Series

4. Product information

| Product Number | RAM (KB) | Flash (KB) | E2 (KB) | ISP (KB) | PWM (units/chls) | ADC (chls) | UART/SPI /TWI(unit) | PCA (units/chls) | LED | LCD | TK (chls) | IO | Package |
|--------------------|----------|------------|---------|----------|------------------|------------|---------------------|------------------|---------|------------------------|-----------|----|----------------|
| SH30F9871P/064PR | 16 | 128 | 4 | 8 | 4/2 | 28 | 4/2/1 | 4/2 | 8x8+7x8 | 8x36,6x38 5x39,4x40 | 32 | 61 | LQFP64 (10X10) |
| SH30F9871P/044PR | 16 | 128 | 4 | 8 | 4/2 | 22 | 4/2/1 | 4/2 | 8x8+7x8 | - | 27 | 41 | LQFP44 |
| SH30F9071P/064PR | 16 | 256 | 4 | 8 | 4/2 | 28 | 4/2/1 | 4/2 | 8x8+7x8 | 8x36,6x38 5x39,4x40 | 32 | 61 | LQFP64 (10X10) |
| SH30F9071U/048UR | 16 | 256 | 4 | 8 | 4/2 | 23 | 4/2/1 | 4/2 | 6x3+5x6 | 5x30,4x31 | 26 | 46 | TQFP48 |
| SH30F9071P/044PR | 16 | 256 | 4 | 8 | 4/2 | 22 | 4/2/1 | 4/2 | 8x8+7x8 | - | 27 | 41 | LQFP44 |
| SH30F9820P/064PR | 16 | 128 | 4 | 8 | 4/2 | 28 | 4/2/1 | 4/2 | 8x8+7x8 | 8x36,6x38 5x39,4x40 | - | 61 | LQFP64 (10X10) |
| SH30F9820U/048UR | 16 | 128 | 4 | 8 | 4/2 | 23 | 4/2/1 | 4/2 | 6x3+5x6 | 5x30,4x31 | - | 46 | TQFP48 |
| SH30F9820P/044PR | 16 | 128 | 4 | 8 | 4/2 | 22 | 4/2/1 | 4/2 | 4x3+3x4 | 5x27,4x28 | - | 42 | LQFP44 |
| SH30F9020P/064PR | 16 | 256 | 4 | 8 | 4/2 | 28 | 4/2/1 | 4/2 | 8x8+7x8 | 8x36,6x38 5x39,4x40 | - | 61 | LQFP64 (10X10) |
| SH30F9020S/064SR | 16 | 256 | 4 | 8 | 4/2 | 28 | 4/2/1 | 4/2 | 8x8+7x8 | 8x36,6x38 5x39,4x40 | - | 61 | LQFP64 (7X7) |
| SH30F9020U/048UR | 16 | 256 | 4 | 8 | 4/2 | 23 | 4/2/1 | 4/2 | 6x3+5x6 | 5x30,4x31 | - | 46 | TQFP48 |
| SH30F9020P/044PR | 16 | 256 | 4 | 8 | 4/2 | 22 | 4/2/1 | 4/2 | 4x3+3x4 | 5x27,4x28 | - | 42 | LQFP44 |
| SH30F9821P/044PR | 16 | 128 | 4 | 8 | 4/2 | 24 | 4/2/1 | 4/2 | - | - | - | 42 | LQFP44 |
| SH30F9821P/032PR | 16 | 128 | 4 | 8 | 4/2 | 18 | 4/2/1 | 4/2 | - | - | - | 30 | LQFP32 |
| SH30F9021P/044PR | 16 | 256 | 4 | 8 | 4/2 | 24 | 4/2/1 | 4/2 | - | - | - | 42 | LQFP44 |
| SH30F9021P/032PR | 16 | 256 | 4 | 8 | 4/2 | 18 | 4/2/1 | 4/2 | - | - | - | 30 | LQFP32 |
| SH30F9621S/032SR | 16 | 64 | 4 | 8 | 4/2 | 18 | 4/2/1 | 4/2 | - | - | - | 30 | LQFP32 |
| SH30F9621Q3/032Q3Y | 16 | 64 | 4 | 8 | 4/2 | 18 | 4/2/1 | 4/2 | - | - | - | 30 | QFN32 (5X5) |



SH30F9/SA0 Series

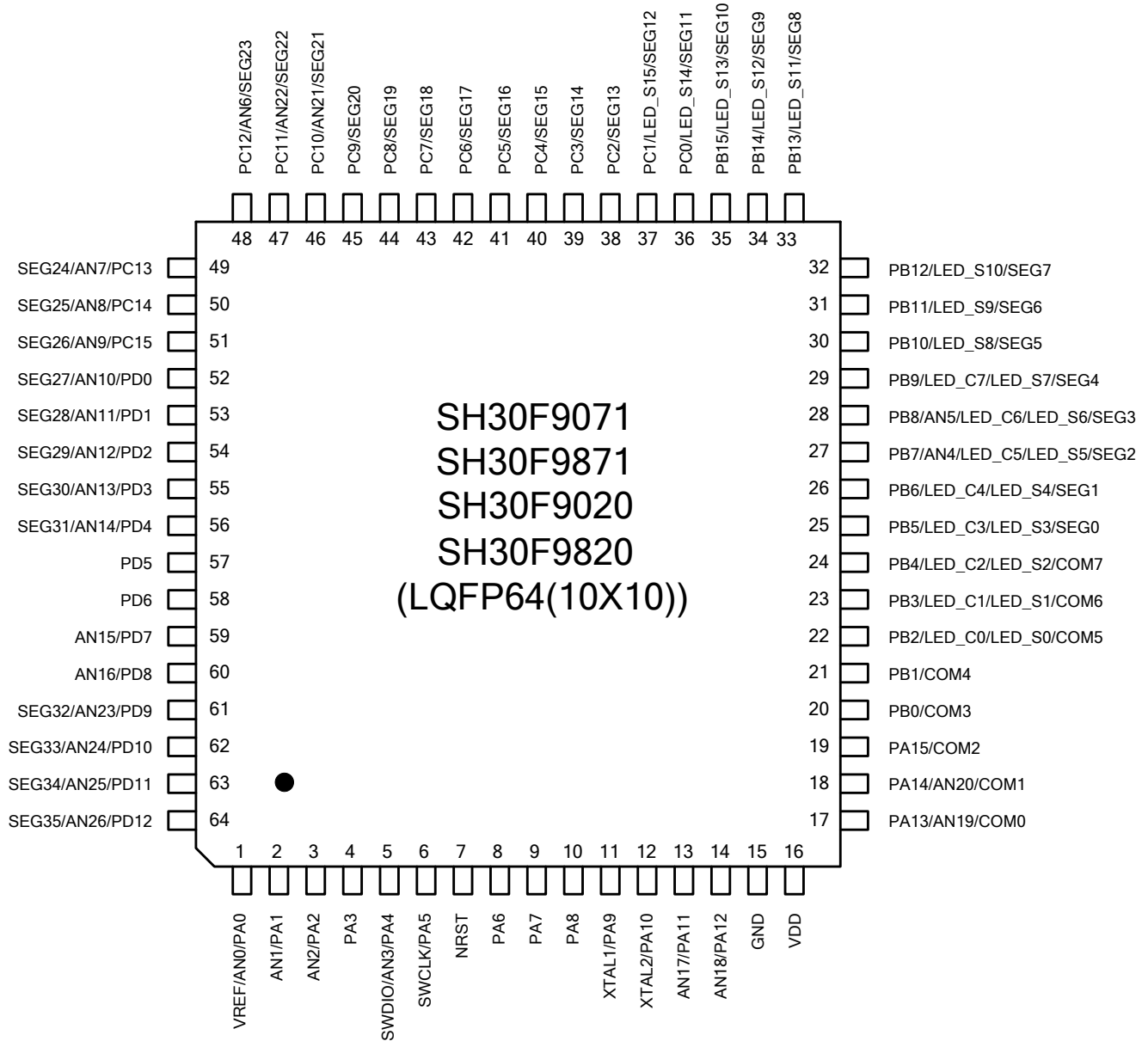




SH30F9/SA0 Series

5. Definitions Pins

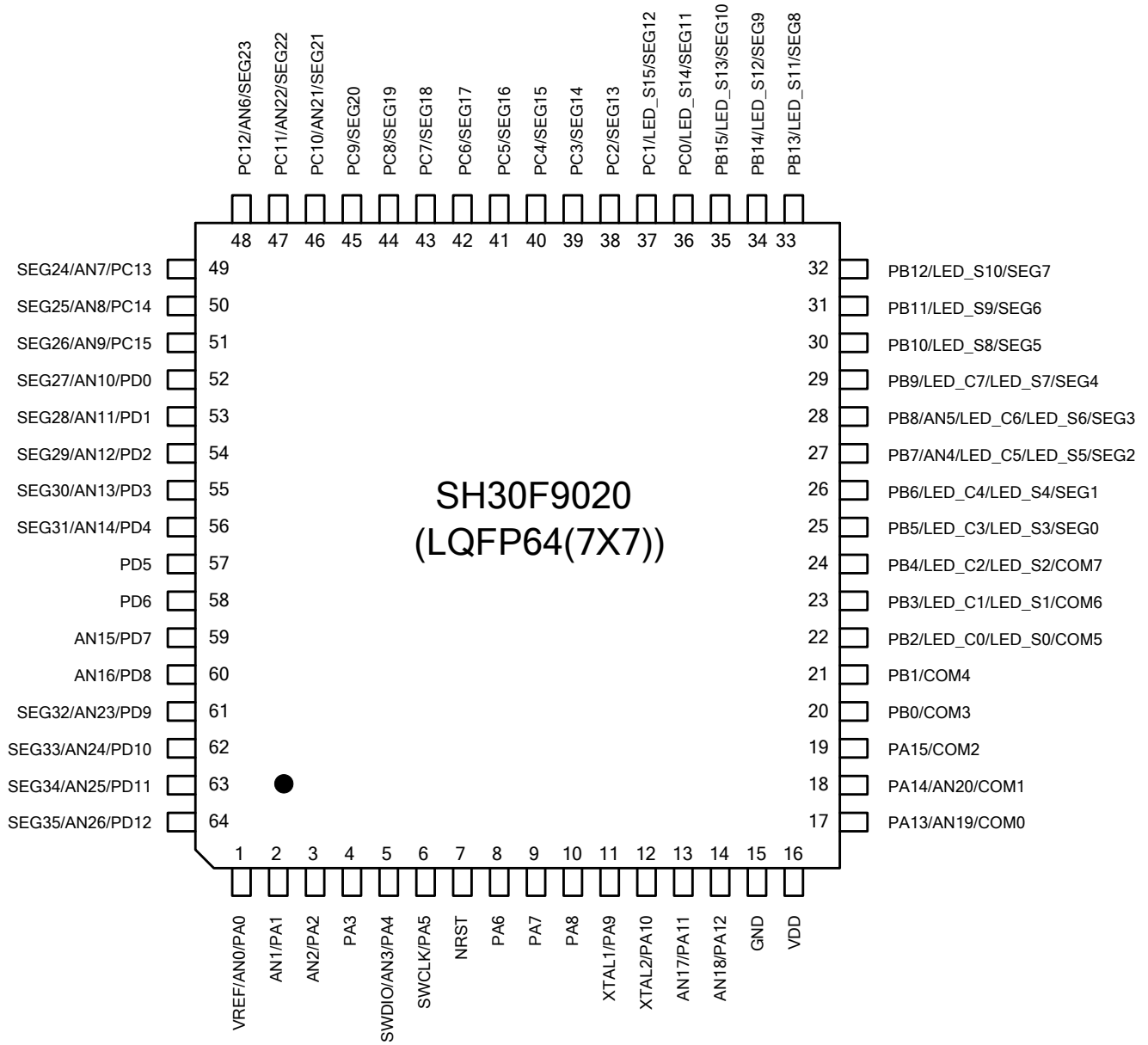
LQFP 64 pins (10X10) package





SH30F9/SA0 Series

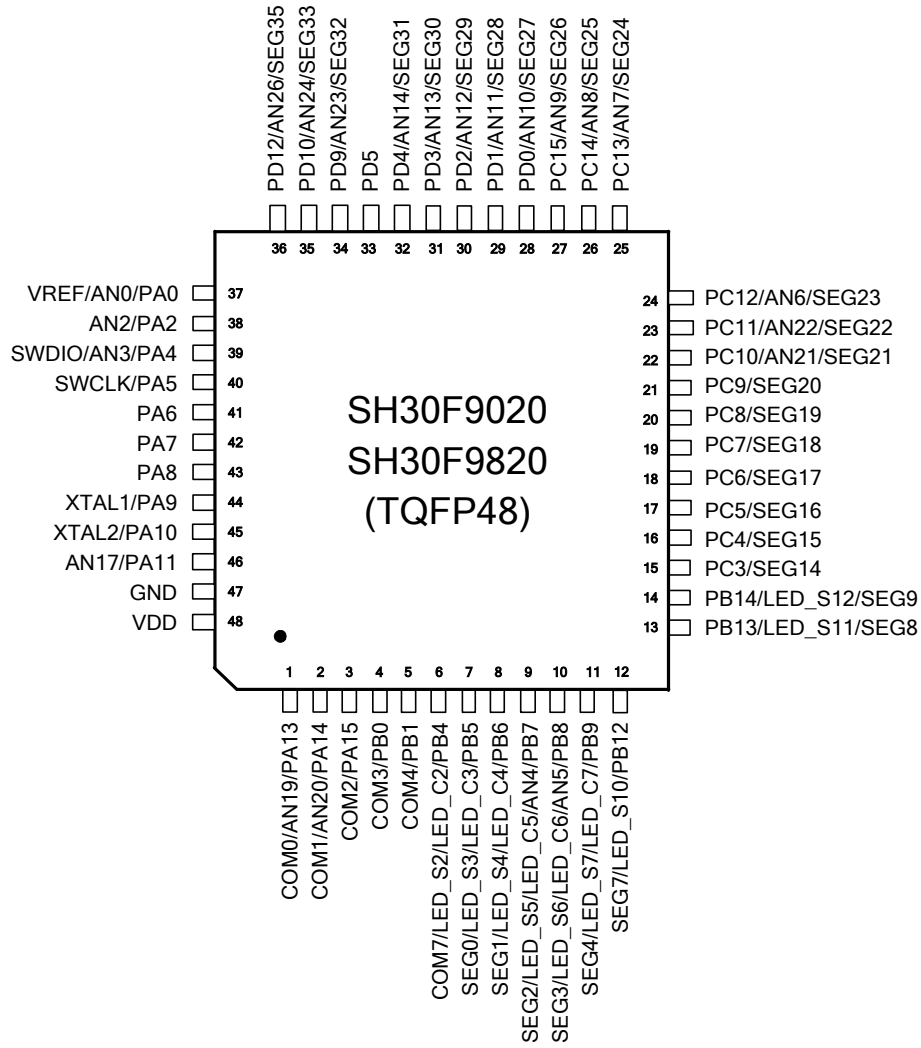
LQFP 64 pins (7X7) package





SH30F9/SA0 Series

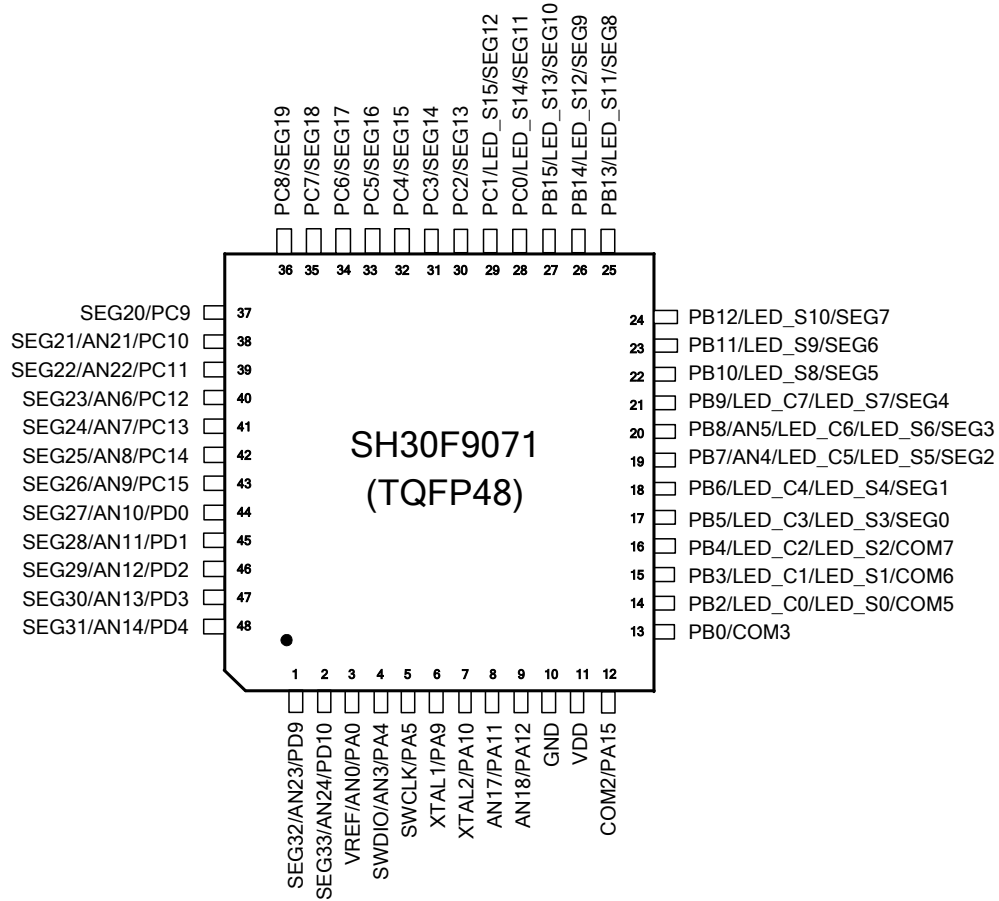
TQFP 48 pins package





SH30F9/SA0 Series

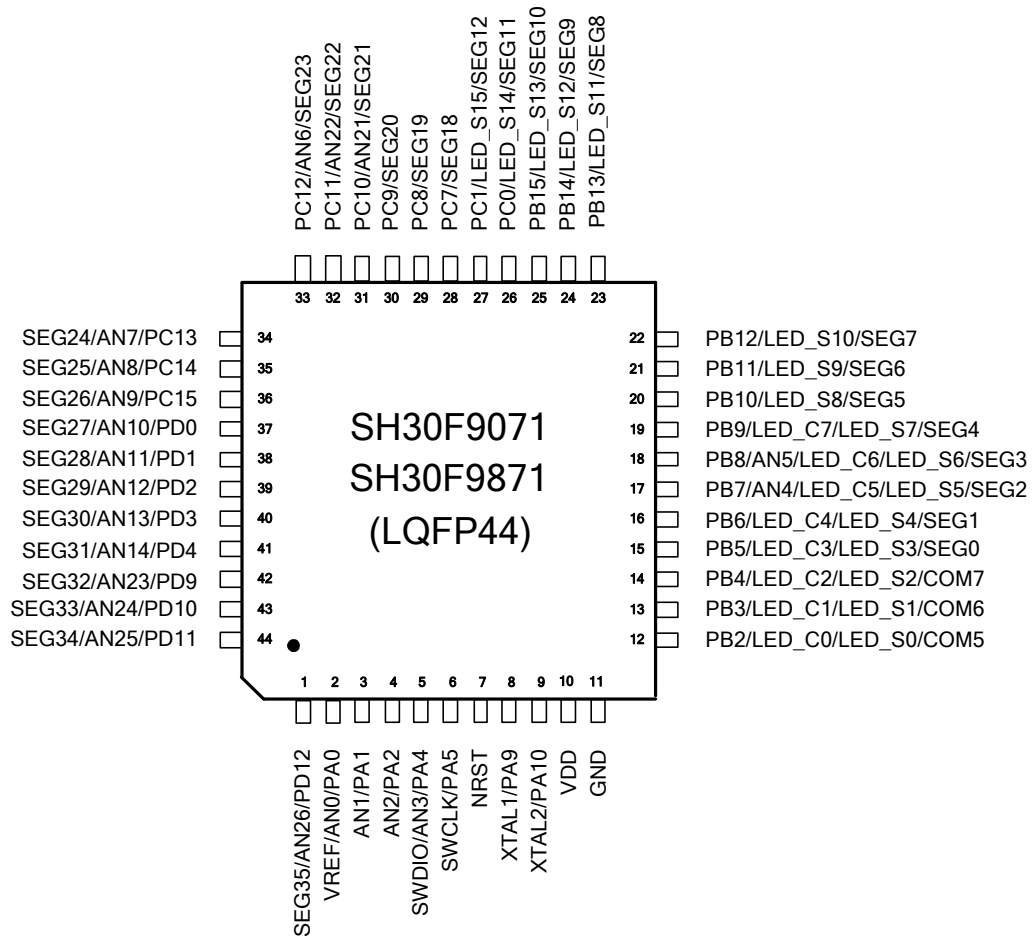
TQFP 48 pins package





SH30F9/SA0 Series

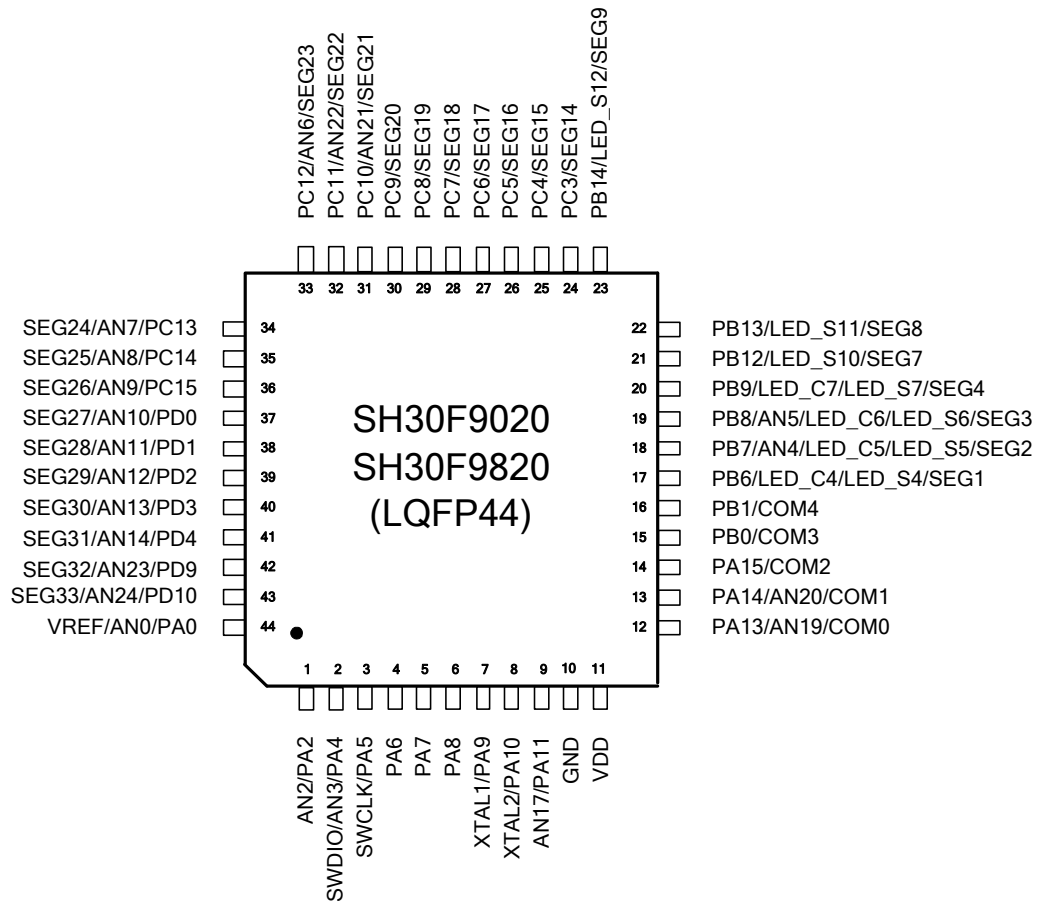
LQFP 44 pins package





SH30F9/SA0 Series

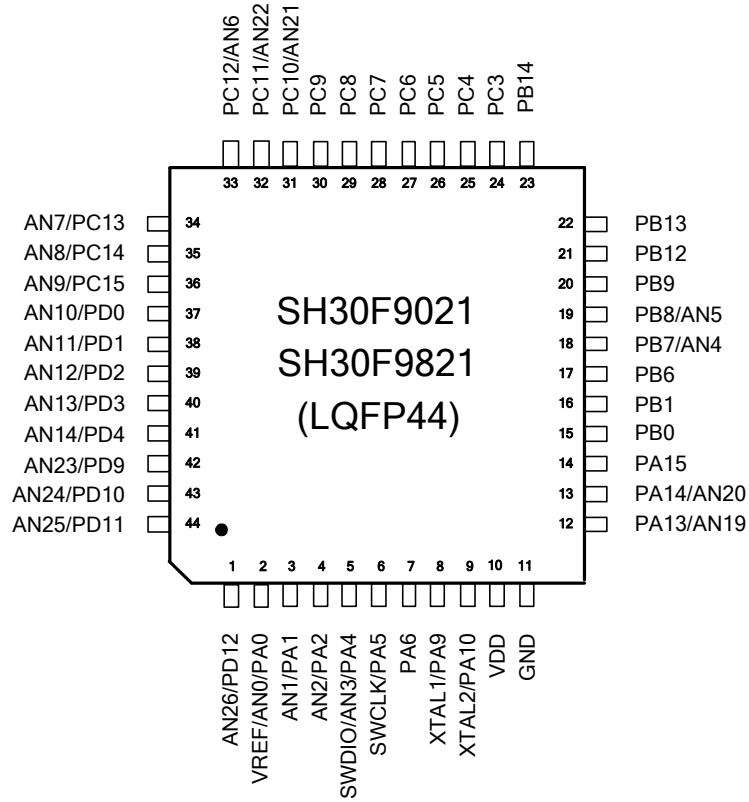
LQFP 44 pins package





SH30F9/SA0 Series

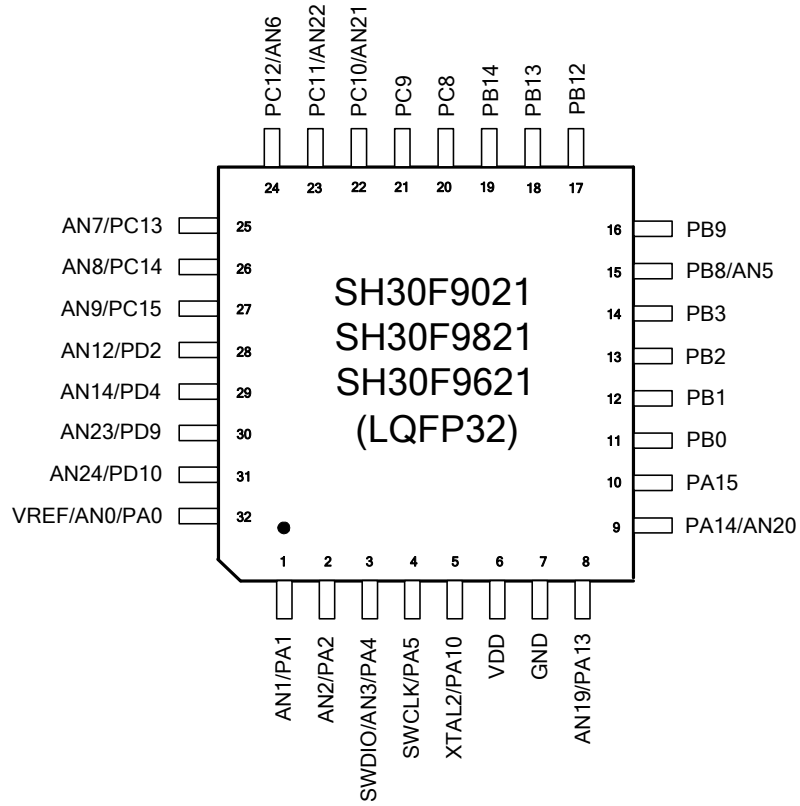
LQFP 44 pins package





SH30F9/SA0 Series

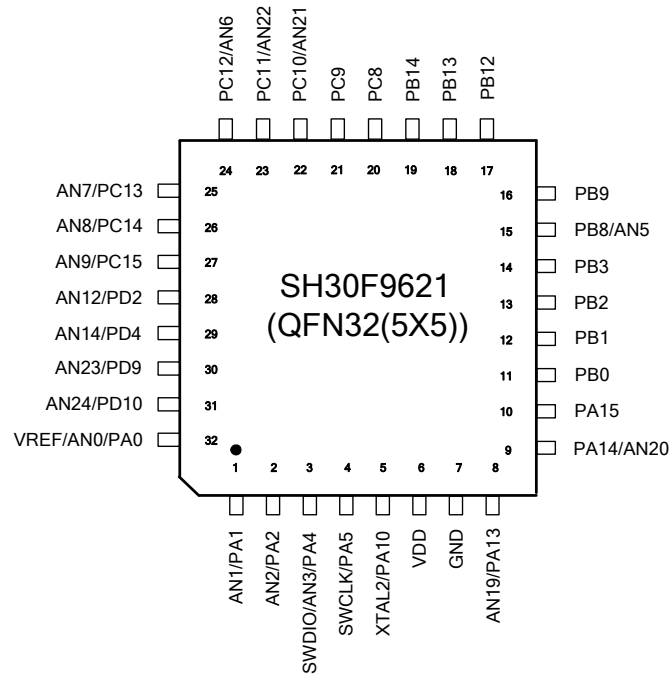
LQFP 32 pins package





SH30F9/SA0 Series

QFN 32 pins (5X5) package



Note:

Except for power and some system pins, all other pins have alternate functions. The alternate functions are selected and set by the Alternate Function Register (AFRL/AFRH). Except for the SWD port, all other ports are mapping to AF0 (high-resistance state) by default during system power-on.

The SWD port and XTAL port can also alternate as other functions, which are set in SWJCFG and OSCCFG @SYSCFG_SAFR.



SH30F9/SA0 Series

Alternate Function Mapping Table

| AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 | AF8 | AF9 | AF10 |
|------|---------|---------|---------|---------|---------|---------|-------|------|------|
| PA0 | AN0 | VREF | PCA0A | T0 | | | | | |
| PA1 | AN1 | PCA0B | T1 | FLT2 | | | | | |
| PA2 | AN2 | PCA3B | T2 | FLT1 | | | | | |
| PA3 | T3 | | | | | | | | |
| PA4 | AN3 | PCA0ECI | T0 | PWM1A | SCL0 | SDA0 | | | |
| PA5 | PCA3ECI | PWM1B | SCL0 | SDA0 | | | | | |
| PA6 | PCA3A | FLT0 | | | | | | | |
| PA7 | PWM0A | | | | | | | | |
| PA8 | PWM0B | | | | | | | | |
| PA9 | PCA3A | PCA3ECI | TXD1 | RXD1 | | | | | |
| PA10 | PCA3B | TXD1 | RXD1 | | | | | | |
| PA11 | AN17 | T1 | | | | | | | |
| PA12 | AN18 | T2 | | | | | | | |
| PA13 | AN19 | COM0 | PWM3A | SCL0 | | | | | |
| PA14 | AN20 | COM1 | PWM3B | SDA0 | | | | | |
| PA15 | COM2 | PWM2A | TXD0 | RXD0 | | | | | |
| PB0 | COM3 | PWM2B | TXD0 | RXD0 | | | | | |
| PB1 | COM4 | PCA1A | T3 | FLT2 | | | | | |
| PB2 | LED_C0 | LED_S0 | COM5 | T0 | TXD3 | | | | |
| PB3 | LED_C1 | LED_S1 | COM6 | T1 | RXD3 | SS1 | | | |
| PB4 | LED_C2 | LED_S2 | COM7 | PCA1A | SCK1 | SCL0 | | | |
| PB5 | LED_C3 | LED_S3 | SEG0 | PCA1B | MOSI1 | SDA0 | TK0 | | |
| PB6 | LED_C4 | LED_S4 | SEG1 | PCA0A | PCA1ECI | MISO1 | TK1 | | |
| PB7 | AN4 | LED_C5 | LED_S5 | SEG2 | PCA0B | PCA0ECI | MISO0 | SS0 | TK2 |
| PB8 | AN5 | LED_C6 | LED_S6 | SEG3 | PCA0B | PCA1ECI | MOSI0 | SDA0 | TK3 |
| PB9 | LED_C7 | LED_S7 | SEG4 | PCA1B | FLT0 | SCK0 | SCL0 | TK4 | |
| PB10 | LED_S8 | SEG5 | T2 | SS0 | TK5 | | | | |
| PB11 | LED_S9 | SEG6 | T3 | SS1 | TK6 | | | | |
| PB12 | LED_S10 | SEG7 | PWM0A | TXD2 | SCK1 | MISO0 | SCL0 | SDA0 | TK7 |
| PB13 | LED_S11 | SEG8 | PWM0B | RXD2 | MOSI1 | MOSI0 | SCL0 | SDA0 | TK8 |
| PB14 | LED_S12 | SEG9 | PWM0A | TXD2 | MISO1 | TK9 | | | |
| PB15 | LED_S13 | SEG10 | T3 | SS1 | TK10 | | | | |
| PC0 | LED_S14 | SEG11 | T2 | FLT3 | TK11 | | | | |
| PC1 | LED_S15 | SEG12 | PWM3A | SS0 | TK12 | | | | |
| PC2 | SEG13 | PWM3B | SCK0 | TK13 | | | | | |
| PC3 | SEG14 | T1 | PWM1A | TXD0 | RXD0 | MISO0 | TK14 | | |
| PC4 | SEG15 | T0 | PWM2A | TXD0 | RXD0 | MOSI0 | TK15 | | |
| PC5 | SEG16 | PCA0A | PCA0ECI | TXD3 | RXD3 | SCK1 | SCL0 | TK16 | |
| PC6 | SEG17 | PCA2B | T0 | TXD3 | RXD3 | MOSI1 | SCL0 | SDA0 | TK17 |
| PC7 | SEG18 | PCA3ECI | PWM2B | TXD3 | MISO1 | SDA0 | TK18 | | |
| PC8 | SEG19 | PCA3B | T1 | FLT0 | TXD1 | RXD1 | MISO0 | TK19 | |
| PC9 | SEG20 | PCA3A | T2 | TXD1 | RXD1 | MOSI0 | SDA0 | TK20 | |
| PC10 | AN21 | SEG21 | PCA2A | T3 | TXD2 | RXD2 | SCK0 | SCL0 | TK21 |
| PC11 | AN22 | SEG22 | PCA2B | PCA2ECI | TXD2 | RXD2 | SS0 | TK22 | |
| PC12 | AN6 | SEG23 | PCA2A | PCA2ECI | SCL0 | SDA0 | TK23 | | |
| PC13 | AN7 | SEG24 | PCA1A | FLT2 | SCL0 | SDA0 | TK24 | | |



SH30F9/SA0 Series

| | | | | | | | | | |
|------|------|-------|---------|-------|------|------|------|--|--|
| PC14 | AN8 | SEG25 | PCA1B | FLT1 | SCL0 | SDA0 | TK25 | | |
| PC15 | AN9 | SEG26 | PCA1ECI | PWM1A | SCL0 | SDA0 | TK26 | | |
| PD0 | AN10 | SEG27 | PWM1B | TXD0 | RXD0 | TK27 | | | |
| PD1 | AN11 | SEG28 | PCA2ECI | PWM3A | TXD0 | RXD0 | TK28 | | |
| PD2 | AN12 | SEG29 | PCA2A | FLT1 | TXD1 | RXD1 | TK29 | | |
| PD3 | AN13 | SEG30 | PCA2B | FLT3 | TXD1 | RXD1 | TK30 | | |
| PD4 | AN14 | SEG31 | PWM0B | TK31 | | | | | |
| PD5 | T0 | FLT3 | SS1 | | | | | | |
| PD6 | T1 | SCK1 | SCL0 | | | | | | |
| PD7 | AN15 | T2 | RXD2 | MOSI1 | SDA0 | | | | |
| PD8 | AN16 | T3 | TXD2 | MISO1 | | | | | |
| PD9 | AN23 | SEG32 | PWM1B | RXD2 | | | | | |
| PD10 | AN24 | SEG33 | PWM2B | TXD2 | | | | | |
| PD11 | AN25 | SEG34 | PWM3B | | | | | | |
| PD12 | AN26 | SEG35 | PWM2A | | | | | | |

Note: AF0 is unselected function (I/O is high-resistance state, only pull-up is valid), AF1 is GPIO, AF2-AF10 is digital/analog signal. In principle, it is not recommended to configure a function port on different I/O pins.



6. Descriptions of Pins

| Pin No. | Type | Description |
|---|------|---|
| I/O Port | | |
| PA.0 – PA.15 | I/O | 16-bit bidirectional I/O port |
| PB.0 – PB.15 | I/O | 16-bit bidirectional I/O port |
| PC.0 – PC.15 | I/O | 16-bit bidirectional I/O port |
| PD.0 – PD.12 | I/O | 13-bit bidirectional I/O port |
| General purpose PWM timer (PWM0~PWM3, support two-channel complementary output with dead zone) | | |
| PWM0A– PWM3A | O | 16-bit PWM0~PWM3 channel A pulse output |
| PWM0B– PWM3B | O | 16-bit PWM0~PWM3 channel B pulse output |
| FLT0 – 3 | I | PWM0~PWM3 fault detection input |
| PCA controller (PCA0~PCA3,16-bit programmable counter array) | | |
| PCA0A~PCA3A | I/O | PCA0~PCA3 module 0 input/output |
| PCA0B~PCA3B | I/O | PCA0~PCA3 module 1 input/output |
| PCA0ECI~PCA3ECI | I | PCA0~PCA3 external clock input |
| Basic timer (TIM0~TIM3) | | |
| T0 – T3 | I/O | TIM0~TIM3 External clock input/clock compare output |
| ADC (28 channels analog-to-digital converter) | | |
| AN0 – AN26 | I | ADC input channel |
| VREF | I/O | ADC external reference voltage input interface |
| Reset, clock and power | | |
| NRST | I | CPU will reset when this pin remains at low level for more than 10μs. Because there is internal 30kΩ pull-up resistor connected to V _{DD} , a reset pulse can be generated by connecting only one external capacitor |
| BOOT* | I | Reserved for use as a boot mode select pin. BOOT=0 means normal booting from flash memory and BOOT=1 means booting from system memory(ISP). |
| XTAL1 | I | Resonator input |
| XTAL2 | O | Resonator output |
| GND | P | Grounding |
| V _{DD} | P | Power (2.0 - 5.5V) |
| UART (UART0-3, general purpose synchronous-asynchronous serial port) | | |
| RXD0 – 3 | I/O | Serial port 0-3 data input |
| TXD0 – 3 | O | Serial port 0-3 data output |



SH30F9/SA0 Series

| SPI (SPI0/1, serial peripheral interface) | | |
|---|-----|--|
| MOSI0/1 | I/O | SPI0/1 master output slave input pin |
| MISO0/1 | I/O | SPI0/1 master input slave output pin |
| SCK0/1 | I/O | SPI0/1 serial clock pin |
| SS0/1 | I | SPI0/1 slave device select pin |
| TWI (TWI0, two-wire serial interface) | | |
| SCL0 | I/O | TWI0 clock signal (supporting master/slave mode) |
| SDA0 | I/O | TWI0 data input and output |
| LCD | | |
| COM0~COM7 | O | LCD output (COM port) |
| SEG0~SEG35 | O | LCD output (Segment port) |
| LED | | |
| LED_C0 ~LED_C7 | O | LED output (COM port) |
| LED_S0~LED_S15 | O | LED output (Segment port) |
| Touch Key | | |
| TK0~TK31 | I | Touch Key channel port |
| Debug interface (SWDP, serial two-wire debug interface) | | |
| SWDIO | I/O | SWDP Data input and output |
| SWCLK | I | SWDP Clock input |
| Note: <i>The two interfaces of system power-on are debug interfaces by default.</i> | | |

*Note: BOOT pin is actually a common GPIO port, and its function is defined in the ISP program.



7. Core

7.1 ARM® Cortex™-M0+ Core

The ARM®Cortex™-M0+ processor is the leading 32-bit RISC processor core of the industry, which applies to real-time applications with high determinism. It is specifically developed for a wide range of high-performance, low-cost platform applications (including microcontrollers, industrial control systems as well as wireless networks and sensors). The Cortex™-M0+ processor delivers outstanding computing performance and superior system responsiveness to events. This processor provides additional code efficiency while meeting the real practical challenges of low dynamic and static power requirements. The Cortex™-M0+ features a Harvard architecture, and has separate instruction bus and data bus, which allow fetching and data access to be parallel, thus improving performance.

SH30F9/SA0 series uses the latest r0p1 version of the ARM® Cortex™-M0+.

For more information about the core, please refer to "Cortex-M0+ Technical Reference Manual".

7.2 Register Group

Cortex-M0+ processor has a register group of R0-R15, of which R13 is the stack pointer SP. There are two SPs, but only one is available at the same time (banked register).

■ R0-R12: General Purpose Registers

R0-R12 are all 32-bit general purpose registers for data operation.

Note: Most 16-bit Thumb instructions can only access R0-R7, while 32-bit Thumb-2 instructions can access all registers.

■ Banked R13: two stack pointers

Cortex-M0+ has two grouped stack pointers, and only one of them can be used at any time.

- Main stack pointer (MSP): The main stack pointer used by default after reset is used for the operating system kernel and exception handling routines (including interrupt service routines).
- Process Stack Pointer (PSP): Used by user's application code.

The lowest two bits of the stack pointer are always 0, which means that the stack is always 4-byte aligned.

In the field of ARM programming, any event that interrupts the sequential execution of a program is called an exception. Except for external interrupts, the execution of programs may be interrupted when an instruction performs an "illegal operation" or accesses prohibited memory interval, or fault occurs due to various errors, or a non-maskable interrupt occurs. These conditions above are collectively referred to as exceptions. Exceptions and interrupts can also be mixed in less strict context. In addition, the program code can also actively request to enter the exception state (usually used for system calls).

■ R14: Link Register

When a subroutine is called, the returned address is stored by R14.

Different from other processors' cores, ARM stores the returned address directly in the register in order to reduce the number of accesses to memories. This is enough to let a lot of code that is only called by level 1 subroutines without accessing memories (stack memories), which improves the efficiency of subroutine calls. If the level is more than 1, pushing R14 value of the previous level onto the stack is required. When programming on ARM, try to use only registers to save intermediate results and reduce access to memories.

■ R15: Program Count Register

R15 is the program count register which points to the current program address. On modifying its value, the execution flow of the program can be changed.

■ Special Function Registers

Cortex-M0+ also has several special function registers on core level, as shown in the following table:

- Program Status Register (PSRs)
- Interrupt Mask Register (PRIMASK)
- Control Register (CONTROL)

Table 7-1 Cortex-M0+ Special Function Registers and Their Functions

| Register | Function |
|----------|---|
| xPSR | Record ALU flags (0 flag, carry flag, negative flag, overflow flag), execution status, interrupt number currently serving |
| PRIMASK | Disable all maskable interrupts - NMI can respond |
| CONTROL | Define the privilege status and decide which stack pointer to use |



8. Memories and Bus Architecture

The bus interface of ARM® Cortex®-M0+ is based on the AHB-Lite and APB protocols, and their rules refer to "AMBA 3 Specification (3rd Edition)".

8.1 System Architecture

The SH30F9/SA0 series main system is interconnected and driven by a multi-layer AHB bus matrix, with 2 host interfaces and 5 slave interfaces, as shown in the figure below:

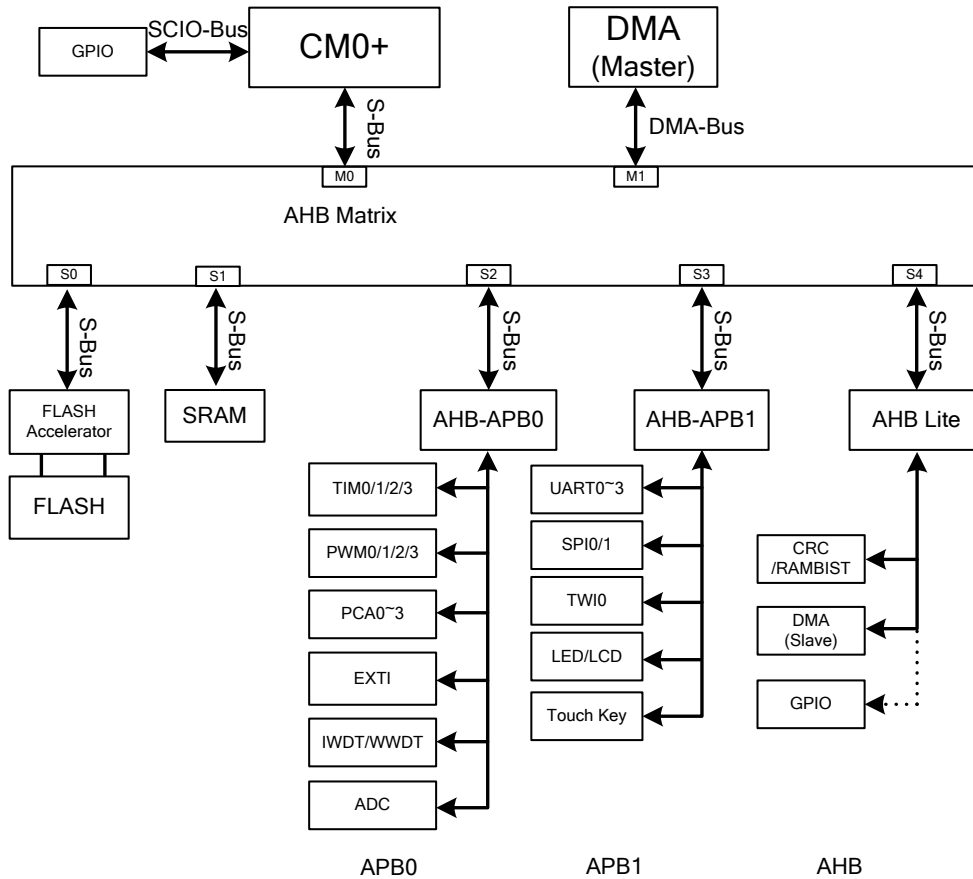


Figure 8-1 SH30F9/SA0 series system architecture block diagram

System Bus

This bus connects System bus (Peripheral Bus) of Cortex™-M0+ core with the bus matrix, the bus matrix coordinates the access between the core and DMA.

DMA Bus

This bus connects AHB of DMA with bus matrix, the bus matrix coordinates the access between CPU/DMA and SRAM, Flash, Peripheral device.

Note: This figure shows that DMA accesses peripherals through the bus matrix, and independent buses can also be used according to design requirements.

Bus Matrix

The bus matrix coordinates the access arbitration between the core system bus and the DMA master bus. The bus matrix contains 2 host interfaces (system bus and DMA bus) and 5 slave interfaces (1 flash memory interface, 1 data SRAM interface and 3 AHB peripheral interfaces). The AHB peripheral is connected to the system bus through the bus matrix, allowing DMA access.

Note: Both the memory and peripherals in the DMA controller use the DMA bus.



AHB Peripheral Bus

The AHB peripheral bus is mounted on the bus matrix, and some peripherals that require high speed and being tightly coupled to the system are mounted on the bus. The AHB/APB bridge is also mounted on this bus.

AHB/APB Bridge

AHB-APB bridges provide synchronous connections between the AHB and the APB buses. Both APB0 and APB1 are specialized peripheral buses, with a maximum speed of 24MHz (generally set to one-half of the AHB clock).

Single cycle IO bus (SCIOB)

GPIO module can choose to mount on SCIOB or AHB through register BUS@SYSCFG_GPIOBCR.

Before using a peripheral, the corresponding control bits of the clock registers RCC_AHBENR / RCC_APB0ENR / RCC_APB1ENR must be set to turn on that peripheral's clock. See the "Reset and Clock Control (RCC)" part for more details.

8.2 Memory Organization

The SH30F9/SA0 series follows ARM® Cortex™-M0+ basic rules for memory organization.

The basic organization of the memory is as follows:

Table 8-1 SH30F9/SA0 series basic memory organization table

| Address Range | Use | Detailed Address Range | Description |
|---------------------------|---|---------------------------|--|
| 0x0000 0000 – 0x0FFF FFFF | On-chip non-volatile code memory area (Flash) | 0x0000 0000 – 0x0001 FFFF | 128K flash on the chip |
| | | 0x0000 0000 – 0x0003 FFFF | 256K flash on the chip |
| | On-chip non-volatile system memory (System Block) | 0x0FFC 0000 – 0x0FFC 0FFF | On-chip EEPROM-Like Memory (4K in total) |
| | | 0x0FFE0000 – 0x0FFE 1FFF | On-chip Boot Rom Area (8K in total, including ISP) |
| | | 0x0FFF 8000 – 0x0FFF E3FF | On-chip Information Area (4K in total, 256 bytes of OTP for customer use) |
| 0x2000 0000 – 0x3FFF FFFF | On-chip volatile data storage area(SRAM) | 0x2000 0000 – 0x2000 3FFF | On-chip 16K SRAM |
| 0x4000 0000 – 0x5FFF FFFF | On-chip peripheral area | 0x4000 0000 – 0x4001 FFFF | On-chip APB0 peripheral area, with each peripheral allocated 1K byte address space |
| | | 0x4002 0000 – 0x4003 FFFF | On-chip APB1 peripheral area, with each peripheral allocated 1K byte address space |
| | | 0x4004 0000 – 0x400F FFFF | On-chip AHB peripheral area, with each peripheral allocated 1K byte address space |
| 0x6000 0000 – 0x9FFF FFFF | Reserved | - | Can expand external RAM devices in this area |
| 0xA000 0000 – 0xDFFF FFFF | Reserved | - | Can expand external peripheral devices in this area |
| 0xE000 0000 – 0xE00F FFFF | Cortex-M0+ private peripheral area | 0xE000 0000 – 0xE000 0FFF | Reserved |
| | | 0xE000 1000 – 0xE000 1FFF | Data observation points and tracking (DWT) |
| | | 0xE000 2000 – 0xE000 2FFF | Breakpoint Unit (BPU) |
| | | 0xE000 3000 – 0xE000 DFFF | Reserved |
| | | 0xE000 E000 – 0xE000 EFFF | System Control Area (SCS) |
| | | 0xE000 F000 – 0xE00FEFFF | Reserved |
| | | 0xE00FF000 – 0xE00FFFFF | The ARMv6-M ROM table |

8.2.1 Register Image

Table 8-2 lists the base addresses of all built-in peripherals (not including the Cortex-M0+ core device) in SH30F9/SA0 series. For the register image of each peripheral, see the register address map in the peripheral "Registers" part.

Peripheral register address: peripheral base address + register offset address.



SH30F9/SA0 Series

Table 8-2 Basic Addresses of SH30F9/SA0 series Built-in Peripherals

| Base address | Peripherals | Bus |
|---------------------------|-------------|-------------------------------------|
| 0x4000 0000 ~ 0x4000 03FF | Reserved | APB0 Bus: 0x4000 0000 - 0x4000 43FF |
| 0x4000 0400 ~ 0x4000 07FF | TIM0 | |
| 0x4000 0800 ~ 0x4000 0BFF | TIM1 | |
| 0x4000 0C00 ~ 0x4000 0FFF | TIM2 | |
| 0x4000 1000 ~ 0x4000 13FF | TIM3 | |
| 0x4000 1400 ~ 0x4000 17FF | PWM0 | |
| 0x4000 1800 ~ 0x4000 1BFF | PWM1 | |
| 0x4000 1C00 ~ 0x4000 1FFF | PWM2 | |
| 0x4000 2000 ~ 0x4000 23FF | PWM3 | |
| 0x4000 2400 ~ 0x4000 27FF | PCA0 | |
| 0x4000 2800 ~ 0x4000 2BFF | PCA1 | |
| 0x4000 2C00 ~ 0x4000 2FFF | PCA2 | |
| 0x4000 3000 ~ 0x4000 33FF | PCA3 | |
| 0x4000 3400 ~ 0x4000 37FF | EXTI | |
| 0x4000 3800 ~ 0x4000 3BFF | IWDT | |
| 0x4000 3C00 ~ 0x4000 3FFF | WWDT | |
| 0x4000 4000 ~ 0x4000 43FF | ADC | |
| 0x4002 0000 ~ 0x4002 03FF | UART0 | |
| 0x4002 0400 ~ 0x4002 07FF | UART1 | |
| 0x4002 0800 ~ 0x4002 0BFF | UART2 | |
| 0x4002 0C00 ~ 0x4002 0FFF | UART3 | |
| 0x4002 1000 ~ 0x4002 13FF | SPI0 | |
| 0x4002 1400 ~ 0x4002 17FF | SPI1 | |
| 0x4002 1800 ~ 0x4002 1BFF | TWI0 | |
| 0x4002 1C00 ~ 0x4002 1FFF | LED | |
| 0x4002 2000 ~ 0x4002 23FF | LCD | |
| 0x4002 2400 ~ 0x4002 27FF | TOUCHKEY | AHB Bus: 0x4004 0000 - 0x4004 1BFF |
| 0x4004 0000 ~ 0x4004 007F | GPIOA | |
| 0x4004 0080 ~ 0x4004 00FF | GPIOB | |
| 0x4004 0100 ~ 0x4004 017F | GPIOC | |
| 0x4004 0180 ~ 0x4004 01FF | GPIOD | |
| 0x4004 0400 ~ 0x4004 07FF | SYSCFG | |
| 0x4004 0800 ~ 0x4004 0BFF | RCC | |
| 0x4004 0C00 ~ 0x4004 0FFF | FLASH | |
| 0x4004 1000 ~ 0x4004 13FF | CRC | |
| 0x4004 1400 ~ 0x4004 17FF | RAMBIST | |
| 0x4004 1800 ~ 0x4004 1BFF | DMA | |

8.2.2 Data Memory (SRAM)

The SH30F9/SA0 series has built-in 16K data RAM (SRAM). It can be accessed in bytes, half-words (16 bits) or full words (32 bits). The starting address of the SRAM is 0x2000 0000. SRAM supports March detection algorithm.

8.2.3 Program Memory

The SH30F9/SA0 series has a built-in flash memory, which is divided into two parts: main memory block and information memory block:

- The main memory block is 256K/128K bytes and is divided into 256/128 sectors (1K bytes per sector).
- The information memory block is 16K bytes and is divided into 16 sectors (1K bytes per sector).

For more information about Flash program memory, please refer to the next section "Flash Program Memory".



8.3 Flash Program Memory

8.3.1 Features

- Flash main memory includes 256/128 X 1KB sectors for a total of 256/128KB
- Flash information memory includes 16 X 1KB sectors for a total of 16KB
- Flash write operation supports 32-bit & 16-bit write
- Able to execute programming and erasing operations within working voltage range
- Support overall/sector erasure and programming
- Support the function of dividing flash into blocks to facilitate online upgrade and backup
- Flash read protection to prevent illegal access
- Flash write protection to prevent accidental operation
- Program/erase: Program area: at least 100000 times
EEPROM area: at least 100000 times
- Data storage period: at least 20 years
- Low power consumption

8.3.2 Introduction

The SH30F9/SA0 series has a built-in flash memory, which is divided into two parts: main memory block and information memory block.

The SH30F9/SA0 series has a built-in 256/128K programmable main program memory block with 1024 bytes per sector and 256/128 sectors that can be erased separately.

The special information storage area of SH30F9/SA0 series includes 4KB built-in EEPROM-like storage area for storing user data, 1024 bytes per sector, 8KB boot rom block, and 1KB OTP area is used to store factory initialization data. Once the OTP area data is written, it cannot be erased.

The structure of the flash memory is shown in the figure below.

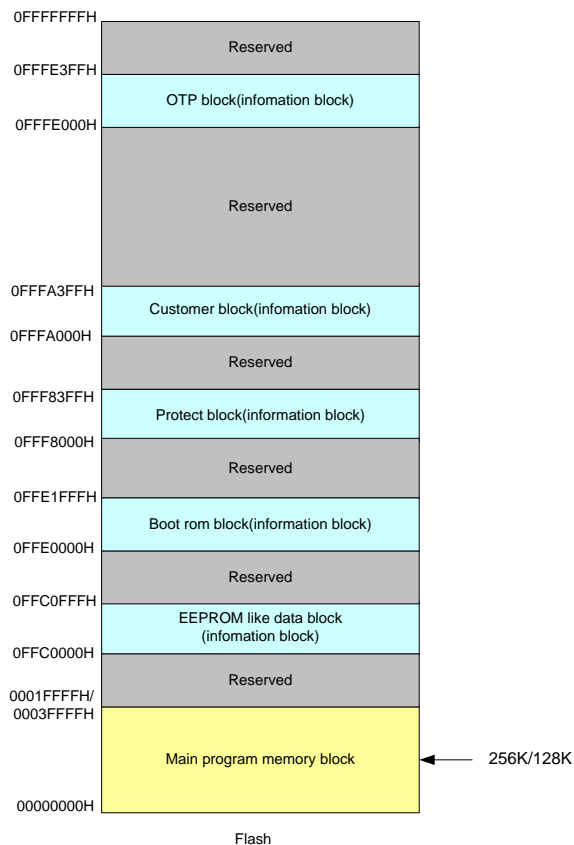


Figure 8-2 Flash Memory Structure



Table 8-3 Flash Memory Structure

| Block | Name | Base address | Size (bytes) |
|-------------------|----------------------------------|---------------------------|--------------------------|
| Main memory | Sector 0 | 0x0000 0000 - 0x0000 03FF | 1024 |
| | Sector 1 | 0x0000 0400 - 0x0000 07FF | 1024 |
| | Sector 2 | 0x0000 0800 - 0x0000 0BFF | 1024 |
| | ... | ... | ... |
| | Sector 127 | 0x0001 FC00 - 0x0001 FFFF | 1024 |
| | ... | ... | ... |
| | Sector 255 | 0x0003 FC00 - 0x0003 FFFF | 1024 |
| Information Block | EEPROM Block | 0x0FFC 0000 - 0x0FFC 0FFF | 4096 |
| | Boot Rom | 0x0FFE 0000 - 0x0FFE 1FFF | 8192 |
| | Protect Block | 0x0FFF 8000 - 0x0FFF 83FF | 1024 |
| | Customer Block | 0x0FFF A000 - 0x0FFF A3FF | 1024 |
| | OTP Block | 0x0FFF E000 - 0x0FFF E3FF | 1024 |
| | Flash memory Operation registers | FLASH_ACR | 0x40040C00 - 0x4004 0C03 |
| FLASH_MKYR | | 0x40040C04 - 0x4004 0C07 | 4 |
| FLASH_E2KYR | | 0x40040C08 - 0x4004 0C0B | 4 |
| FLASH_SR | | 0x40040C0C - 0x4004 0C0F | 4 |
| FLASH_CR | | 0x40040C10 - 0x4004 0C13 | 4 |
| FLASH_CR1 | | 0x40040C14 - 0x4004 0C17 | 4 |
| FLASH_RPR | | 0x4004 0C1C - 0x4004 0C1F | 4 |
| FLASH_WRPR | | 0x4004 0C20 - 0x4004 0C23 | 4 |
| FLASH_CNTR | | 0x4004 0C28 - 0x4004 0C2B | 4 |
| FLASH_UPCNTR | | 0x4004 0C2C - 0x4004 0C2F | 4 |
| FLASH_CNTR | | 0x4004 0C30 - 0x4004 0C33 | 4 |
| FLASH_IKYR | 0x4004 0C34 - 0x4004 0C37 | 4 | |

8.3.3 Flash Memory Function Description

8.3.3.1 Flash Erasure and Programming

After the system is reset, the flash memory operation register is locked, so the flash memory operation register needs to be unlocked before performing erasure programming operation on flash. Different flash memory areas have different unlock registers, and corresponding registers need to be unlocked to execute operations to different areas. There are two cases for unlocking flash operations. One is to perform a single operation on flash after unlocking, and the other is to perform multiple operations on flash after unlocking. If the flash memory operation register is unlocked, the flash memory areas can be erased or programmed by controlling the FLASH_CR register. However, in order to prevent the flash from being mis-operated, when accessing the flash memory control register, the count value of the Flash operation timer must be guaranteed within a valid range. Otherwise, operations on flash memory areas will be prohibited.

For flash memory, 16-bit or 32-bit data can be written in each time. After the main memory area is erased, every 16 bits or 32 bits can only be programmed once, which means if each bit of the programmed 16-bit or 32-bit data is not 0, it cannot be programmed again, otherwise the corresponding error flag is set to 1.

Whole-chip erase and sector erase of the main memory area, programming of the main memory area, sector erase and programming of the EEPROM-like memory area, sector erase and program of the customer information area, they all can be achieved by running the program in RAM, the program in Boot area, or directly accessing the flash operation register in debug mode.

Sector erase of the main memory area, programming of the main memory area, sector erase and programming of the EEPROM-like memory area, sector erase and program of the customer information area, they all can be realized through accessing flash control register by program in main program sector.

In debug mode, the sector erasure and programming of boot area can only be realized by directly accessing flash operation register.

The main program area in flash can be prevented from being illegally read by setting read protection. Similarly, write protection can be set for each sector of the flash memory area to prevent from being accidentally changed under the circumstance of program fleet. For a 256KB/128KB flash memory area, the basic unit of write protection is 8 sectors as a protection unit.

Note: before erasing and programming the flash memory, please feed the IWDT and turn off the LVR function to avoid reset during operation.



8.3.3.2 Flash Control Register Unlock

After the system is reset, the flash memory operation register is locked, so the flash memory operation register needs to be unlocked before performing erasure programming operation on flash. Different flash memory areas have different unlock registers, and corresponding registers need to be unlocked to execute operations to different areas. To unlock flash, two unlocking values are required to be written to the corresponding unlock register. Whenever writing error occurs on the first or the second key value, the corresponding flash area cannot be unlocked, and a HardFault will be generated. Unlocking the unlock register when any one area is unlocked will also generate a HardFault. There are also two cases on flash unlocking operations. One is to perform a single operation on flash after unlocking, while the other to perform multiple operations on flash after unlocking. If the flash memory operation register is in unlock state, the flash memory area can be erased and programmed by controlling FLASH_CR.

After reset, bit MNLCK@FLASH_CR is set to 1. The FLASH_CR can be unlocked by first writing the flash unlock value 0x8ACE 0246 to the FLASH_MKYR register, then writing the single operation unlock value 0xC3C3 C3C3 or the multiple operation unlock value 0xB4B4 B4B4 to the FLASH_MKYR register. After two correct unlock values writing operations are completed, bit MNLCK@FLASH_CR is cleared by hardware. The software can set the control bit MNLCK@FLASH_CR to 1 to lock the main program area. Main program area is locked by the FLASH_MKYR register. After unlocking, the main program area can be operated by running a program in RAM, running a program in the main program area, running a program in boot area, or directly accessing the flash operation register using a debugging tool.

EEPROM-like area of the special information area can be locked by the FLASH_E2KYR register. The unlocking process also contains two write operations. After flash unlock value 0x9BDF 1357 is written to FLASH_E2KYR register, and single operation unlock value 0xC3C3 C3C3 or multiple operation unlock value 0xB4B4 B4B4 is written to FLASH_E2KYR register, this area can be unlocked and the control bit E2LCK@FLASH_CR is cleared by hardware. The software can set the control bit E2LCK@FLASH_CR to 1 to lock the EEPROM-like area of the special information area. It is the EEPROM-like area of the special information area which is locked by FLASH_E2KYR register. After this area is unlocked, operations can be performed by running a program in RAM, running a program in the main program area, running a program in boot area, or directly accessing the flash operation register using a debugging tool.

Protect Block, Customer Block, OTP and Boot Rom of the special information area can be locked by the FLASH_IKYR register. The unlocking process also contains two write operations. After flash unlock value 0xABCD 5678 is written to FLASH_IKYR register, and single operation unlock value 0xC3C3 C3C3 or multiple operation unlock value 0xB4B4 B4B4 is written to FLASH_IKYR register, this area can be unlocked and the control bit INFLCK@FLASH_CR is cleared by hardware. The software can set the control bit INFLCK@FLASH_CR to 1 to lock the Protect Block, Customer Block, OTP and Boot Rom of the special information area. It is the Protect Block, Customer Block, OTP and Boot Rom of the special information area which is locked by FLASH_IKYR register. After the Protect Block, Customer Block and OTP are unlocked, operations can be performed by running a program in RAM, running a program in the main program area, running a program in boot area, or directly accessing the flash operation register using a debugging tool. The boot area can only be operated by using debugging tools to directly access flash operation register.

For flash single unlock operation, after MNLCK@FLASH_CR or E2LCK/INFLCK@FLASH_CR is cleared to 0, only one programming or erasing operation on flash is allowed. After the operation is completed, MNLCK@FLASH_CR or E2LCK/INFLCK@FLASH_CR is automatically set to 1 by hardware, locking the related flash block again.

For flash multiple unlock operation, after MNLCK@FLASH_CR or E2LCK/INFLCK@FLASH_CR is cleared to 0, only one programming or erasing operation on flash is allowed. After the operation is completed, MNLCK@FLASH_CR or E2LCK/INFLCK@FLASH_CR must be set to 1 by software, then the related flash block can be locked again.



SH30F9/SA0 Series

Table 8-4 Unlock Value and Description of Each Unlocking Register

| Register | Function | Flash Initial Unlock Value | Flash Single Operation Unlock Value | Flash Multiple Operation Unlock/Lock Value | Description |
|-------------|--|----------------------------|-------------------------------------|--|--|
| FLASH_MKYR | Unlock main program area | 0x8ACE 0246 | 0xC3C3 C3C3 | 0xB4B4 B4B4 | Directly access by running a program in RAM or a program in main program area, boot rom area or debug tool |
| FLASH_E2KYR | Unlock EEPROM-like area of special information area | 0x9BDF 1357 | 0xC3C3 C3C3 | 0xB4B4 B4B4 | Directly access by running a program in RAM or a program in main program area, boot rom area or debug tool |
| FLASH_IKYR | Unlock Protect Block , Customer Block and Boot Rom of special information area | 0xABCD 5678 | 0xC3C3 C3C3 | 0xB4B4 B4B4 | After the Protect Block, Customer Block and OTP are unlocked, operations can be performed by running a program in RAM, running a program in the main program area, running a program in boot area, or directly accessing the flash operation register using a debugging tool. The boot area can only be operated by using debugging tools to directly access flash operation register. |



8.3.3.3 Flash Operation Timer Function

The flash control module has an operation timer. The count value of the flash operation timer uses the system clock as its clock source. The count value of the flash operation timer must be less than the upper limit of the flash operation timer, and is greater than zero, then operations can be performed on flash memory, or a 32-bit/16-bit data can be written to target address. If a flash operation is initiated, which means when `STRT@FLASH_CR` bit is set to 1 or write operation to flash is initiated, the flash operation timer count value is not in the valid time window, then the flash operation cannot be performed and `PGWERR@FLASH_SR` bit is set to 1. For detailed usage of flash operation timer, see the figure below:

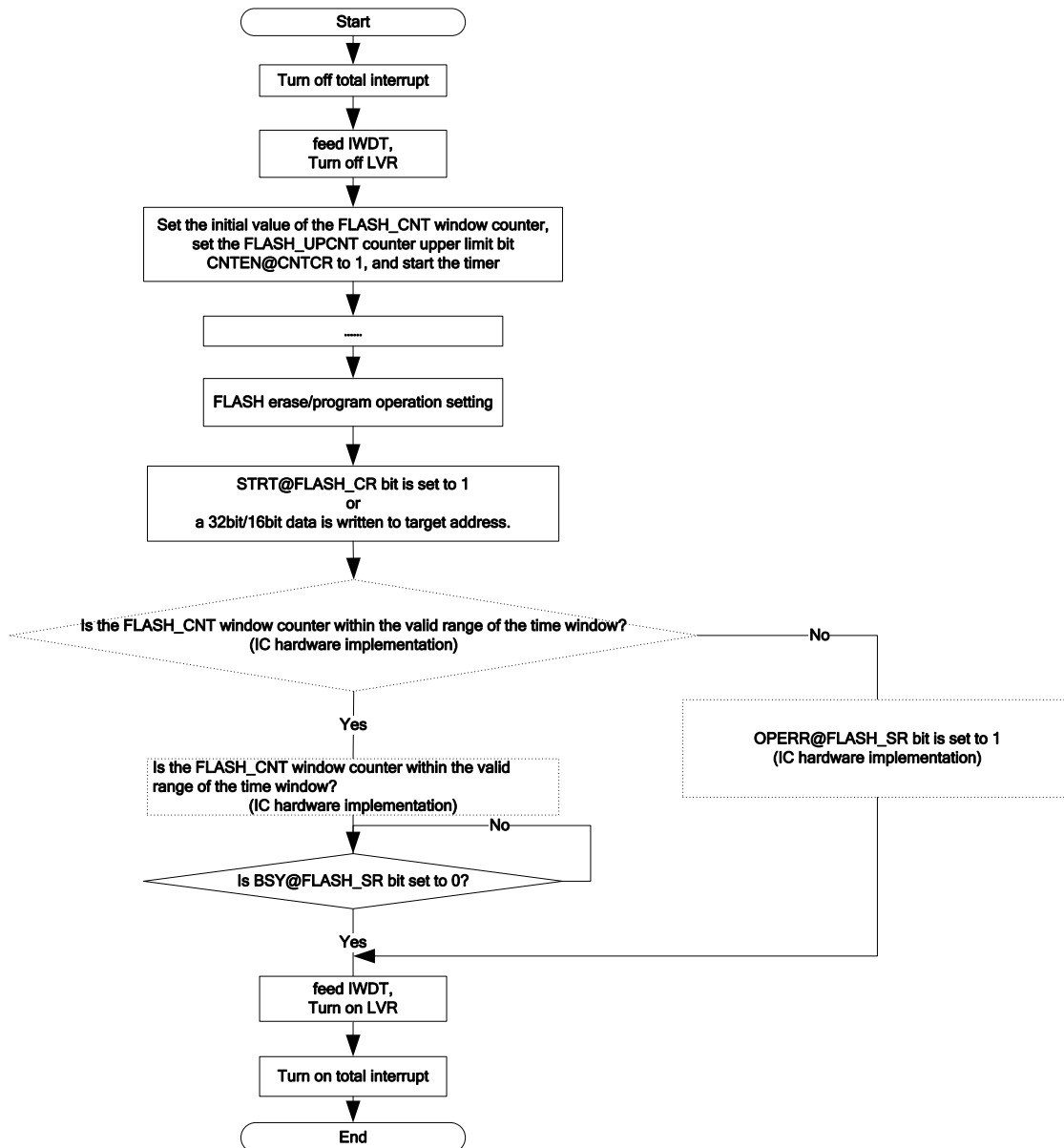


Figure 8-3 Flash Operation Timer Flowchart



8.3.3.4 Main Program Area Total Erasure

The flash operation provides a full-chip erase function that erases the contents of the main memory block area. Detailed steps are as follows:

- (1) Turn off the total interrupt.
- (2) Feed the WDT to the dog and turn off the LVR function to avoid reset during operation.
- (3) FLASH_MKYR sequentially writes the flash unlock value and the single operation unlock value to ensure that the main flash program area is not locked.
- (4) Check BSY@FLASH_SR bit to determine whether flash memory is running.
- (5) When the BSY@FLASH_SR bit is 0, write the flash main program area total erasure instruction word to the FLASH_CR register.
- (6) Full-chip erasure operation is initiated via setting STRT@FLASH_CR bit to 1.
- (7) Check BSY@FLASH_SR bit to determine whether the erasure instruction has been executed.
- (8) The operation is done when BSY@FLASH_SR bit is 0.
- (9) Feed the WDT to the dog and turn on the LVR function.
- (10) Turn on the total interrupt.

Bit EOP@FLASH_SR indicates the end of the operation. The bit 1 indicates the completion of the operation. All flash data is reset to 0x0000 0000 after being erased.

The flowchart is as follows:

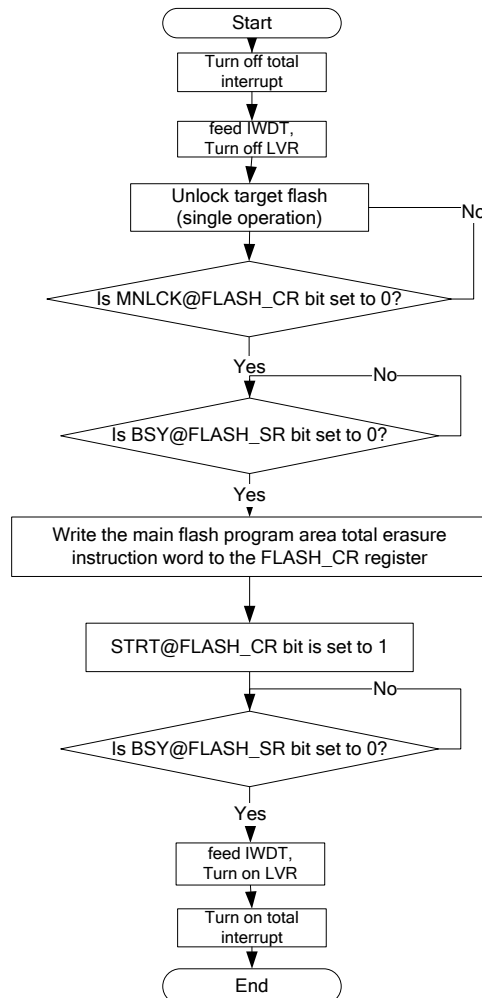


Figure 8-4 Main Program Area Total Erasure



8.3.3.5 Main Program Area Sector Erasure

Each sector of the flash memory can be erased independently without affecting other sector contents. The steps to erase the sector by the flash operation are as follows:

- (a) Erase a sector separately
 - (1) Turn off the total interrupt.
 - (2) Feed the WDT to the dog and turn off the LVR function to avoid reset during operation.
 - (3) FLASH_MKYR sequentially writes the flash unlock value and the single operation unlock value to ensure that the main flash program area is not locked.
 - (4) Check BSY@FLASH_SR bit to determine if flash memory is running.
 - (5) When the bit BSY@FLASH_SR is 0, write the main program area sector erase command word and the sector number to be erased to the FLASH_CR and FLASH_CR1 registers respectively.
 - (6) Sector erasure operation is initiated via setting STRT@FLASH_CR bit to 1.
 - (7) Check BSY@FLASH_SR bit to determine whether the erasure instruction has been executed.
 - (8) The operation is done when BSY@FLASH_SR bit is 0.
 - (9) Feed the WDT to the dog and turn on the LVR function.
 - (10) Turn on the total interrupt.
- The flowchart is as follows:

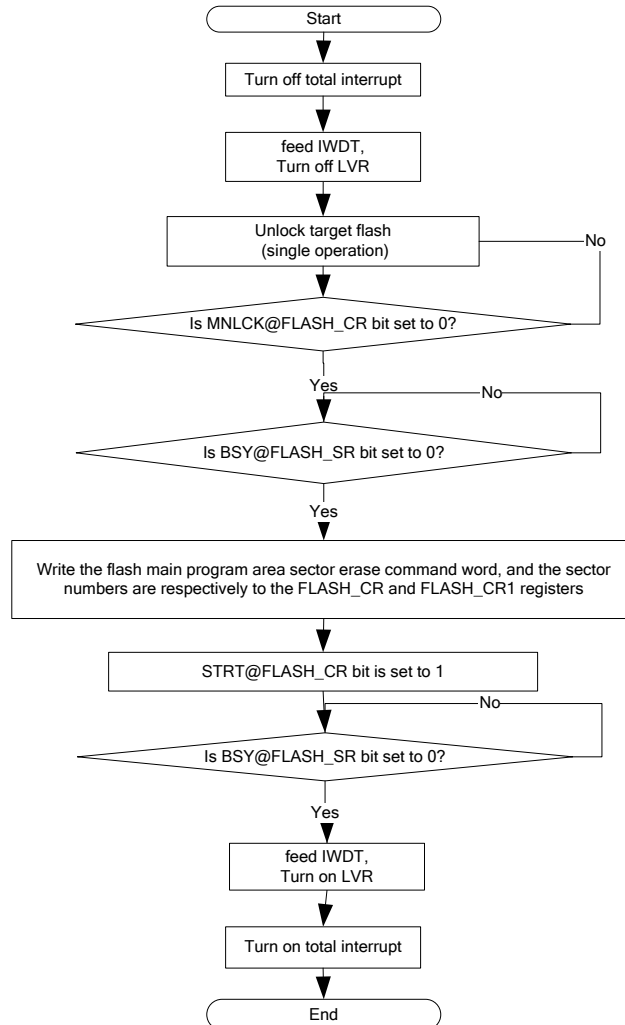


Figure 8-5 Main Program Area Sector Erasure (Single Sector)



(b) Erase multiple sectors

- (1) Turn off the total interrupt.
- (2) Feed the WDT to the dog and turn off the LVR function to avoid reset during operation.
- (3) FLASH_MKYR sequentially writes the flash unlock value and the multiple operation unlock value to ensure that the main flash program area is not locked.
- (4) Check BSY@FLASH_SR bit to determine if flash memory is running.
- (5) When the bit BSY@FLASH_SR is 0, write the sector erase command word and sector code of the main program area to the FLASH_CR and FLASH_CR1 registers respectively.
- (6) Sector erasure operation is initiated via setting STRT@FLASH_CR bit to 1.
- (7) Check BSY@FLASH_SR bit to determine whether the erasure instruction has been executed.
- (8) Repeat steps 3 to 5 to erase other sectors.
- (9) Set MNLCK@FLASH_CR bit to 1 to complete the operation.
- (10) Feed the WDT to the dog and turn on the LVR function.
- (11) Turn on the total interrupt.

The flowchart is as follows:

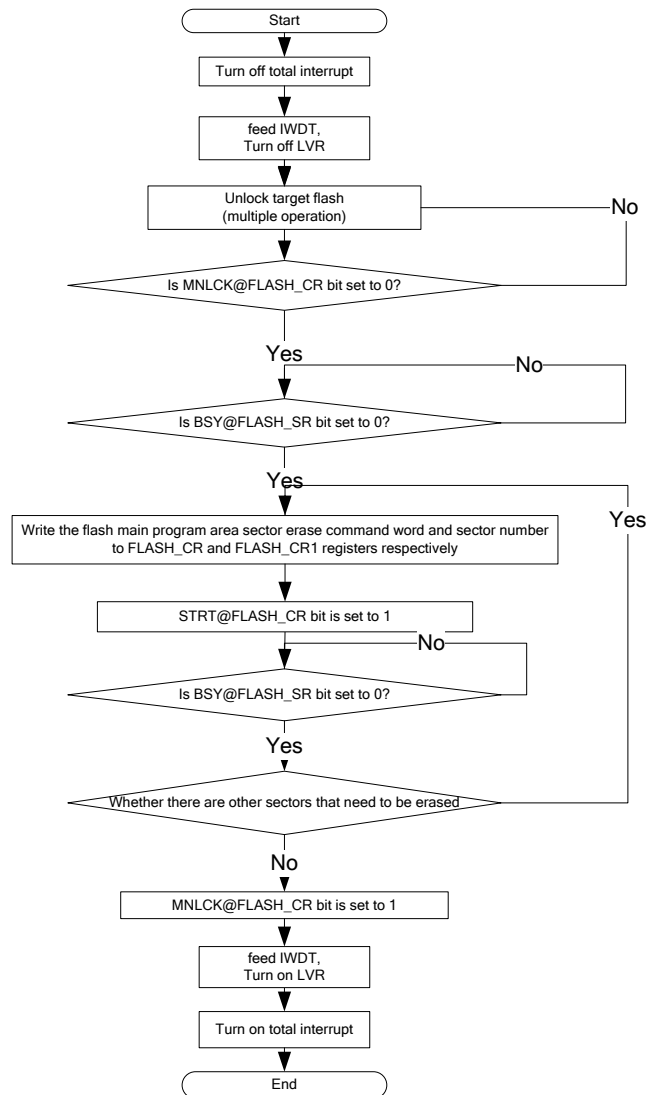


Figure 8-6 Main Program Area Sector Erasure (Multiple Sector)



When target erasure sector is used to fetch instructions or access data, the corresponding erasure operation is invalid, but the flash operation will not provide any notification, so it is necessary to ensure that the target erasure sector address is correct. EOP@FLASH_SR bit indicates the end of the operation.

8.3.3.6 Main Program Area Programming

The Flash operation provides a 32-bit word/16-bit half-word programming function to modify the contents of the flash main memory block. To program on main flash program area, 32 bits/16 bits can be written each time. The addresses must be aligned by 32 bits/16 bits. If they are not aligned, the corresponding operation is invalid and the relevant error flag is set to 1. When the FLASH_CR register operation instruction word is set to main program area programming, writing a word or half word at a flash address will initiate a programming. During the programming process (bit BSY@FLASH_SR bit is '1'), any operation that reads or writes to flash will cause CPU to pause until the end of this flash programming. The following steps show the procedure of word programming operation register.

(a) Single word or half-word programming

- (1) Turn off the total interrupt.
- (2) Feed the WDT to the dog and turn off the LVR function to avoid reset during operation.
- (3) FLASH_MKYR sequentially writes the flash unlock value and the single operation unlock value to ensure that the main flash program area is not locked.
- (4) Check BSY@FLASH_SR bit to determine if flash memory is running.
- (5) Before the BSY@FLASH_SR bit is 0, write the flash main program area programming instruction word and program operation bit width to the FLASH_CR register; configuration PSIZE [2:0] @ FLASH_CR bits, select the corresponding burn write operation bit width; set bit STRT@FLASH_CR to 1;
- (6) Write a 32-bit word/16-bit half-word into target address.
- (7) Check BSY@FLASH_SR bit to determine whether the programming instruction has been executed.
- (8) The operation is done when BSY@FLASH_SR bit is 0.
- (9) Feed the WDT to the dog and turn on the LVR function.
- (10) Turn on the total interrupt.

The operation flowchart is as follows:

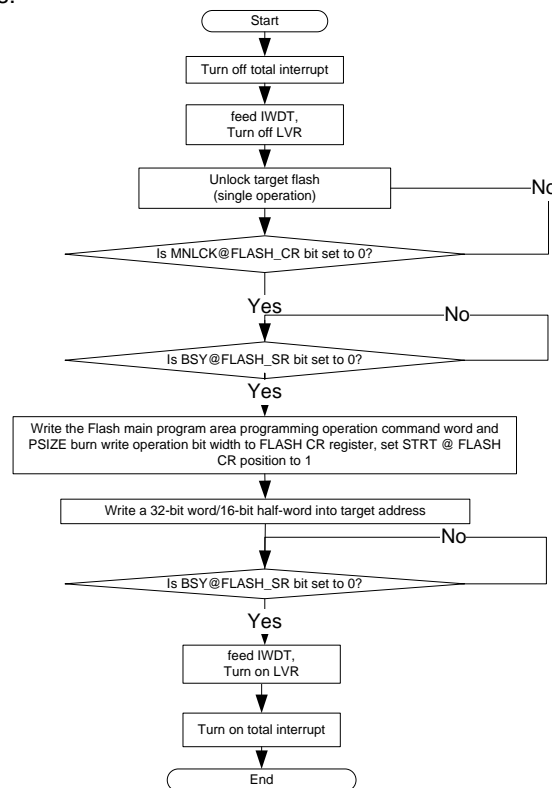


Figure 8-7 Main Program Area Programming (Single word or half word)



(b) Continuous multi-word writing programming

- (1) Turn off the total interrupt.
- (2) Feed the WDT to the dog and turn off the LVR function to avoid reset during operation.
- (3) FLASH_MKYR sequentially writes the flash unlock value and the multiple operation unlock value to ensure that the main flash program area is not locked.
- (4) Check BSY@FLASH_SR bit to determine if flash memory is running.
- (5) Before the BSY@FLASH_SR bit is 0, write the flash main program area programming instruction word and program operation bit width to the FLASH_CR register; configuration PSIZE [2:0] @ FLASH_CR bits, select the corresponding burn write operation bit width;
- (6) Set bit STRT@FLASH_CR to 1; write a 32-bit word/16-bit half-word into target address;
- (7) Check BSY@FLASH_SR bit to determine whether the programming instruction has been executed.
- (8) Repeat steps 4 to 5 until all programming are done on multiple addresses.
- (9) Set MNLCK@FLASH_CR bit to 1 to complete the operation.
- (10) Feed the WDT to the dog and turn on the LVR function.
- (11) Turn on the total interrupt.

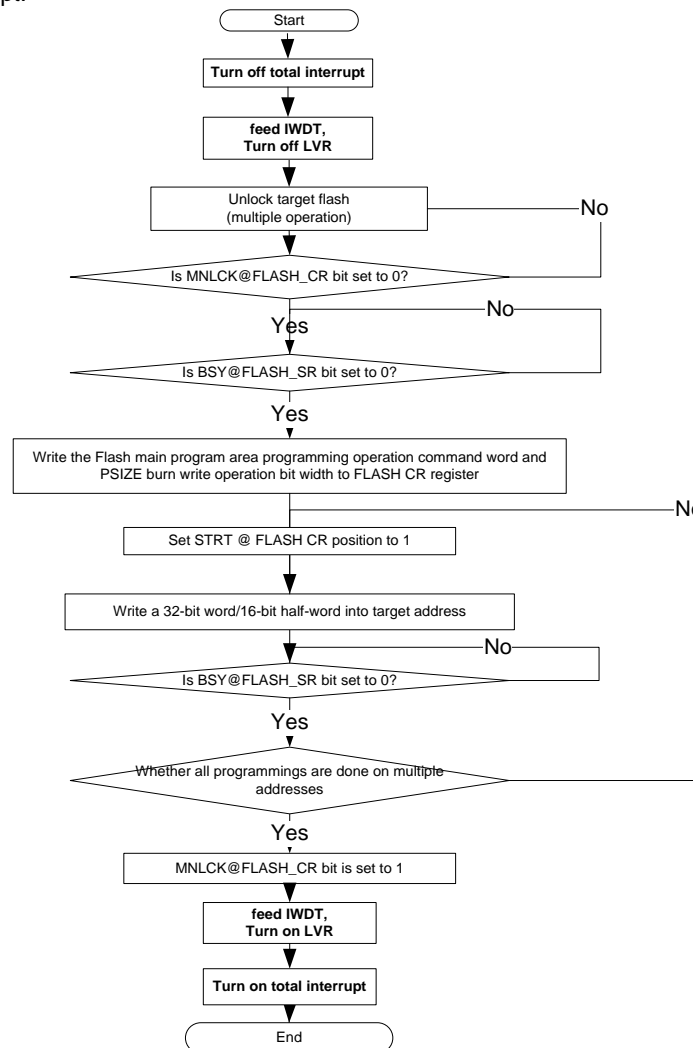


Figure 8-8 Main Program Area Programming (Continuous multi-word)



For flash memory, 16-bit or 32-bit data can be written each time. After the main memory erasure operation, of the 16-bit or 32-bit data which are going to be programmed, if any one bit of them is not 0, it cannot be programmed again, otherwise the programming error flag PGERR@FLASH_SR bit is set to 1.

If a certain sector of the main flash program area has been protected by any write protection bit, but this sector is still being written, then the write protection error flag WRPRTERR @FLASH_SR bit is set to 1.

8.3.3.7 EEPROM-like Memory Area Erasure Programming

EEPROM-like memory area erasure programming is similar to main program area sector erasure and programming. The difference lies in flash unlock registers. The operation instruction words written to FLASH_CR register are different, and the flag bits of the lock memory area are different.

Note: before erasing and programming the EEPROM-like memory, please feed the IWDT and turn off the LVR function to avoid reset during operation.

8.3.3.8 Customer information area (Customer Block) Erasure and Programming

Customer information area (Customer Block) erasure and programming are similar to main program area by sector. But the flash unlock registers are different, the operation command words written to the flash unlock registers are different, and the flag bits of the lock storage area are different.

Note: before erasing and programming the Customer information area, please feed the IWDT and turn off the LVR function to avoid reset during operation.

8.3.3.9 OTP Area Programming

OTP area can only be programmed once, and it cannot be erased after programming. The programming of the OTP area is similar to the programming of the main program area. The difference lies in flash unlock registers. The operation instruction words written to FLASH_CR register are different, and the flag bits of the lock memory area are different.

Note: Before programming the OTP area, please feed the IWDT and turn off the LVR function to avoid reset during operation.

8.3.3.10 Code Protection (Protect Block)

The code protection control area does not support erasure by sector. The programming of the code protection control area is similar to that of the main program area. But the operation command words written to the FLASH_CR register are different, and the flag bits of the lock storage area are different. (It should be noted that the flash block division function is added, and the corresponding relationship may be different between each sector number and logical/physical address.)

The main program area in flash can be prevented from being illegally read via setting read protection. Write protection can also be set to prevent illegal modification. For 256KB/128KB flash storage area, the basic unit of write protection is 8 sectors as a protection unit.

(1) Read Protection

Read protection is by setting the read protection byte, which takes effect immediately after writing.

The Flash memory provides a safe read protection function to prevent the user code in the flash memory from being illegally read out to ensure the safety of the user code. There are three protection levels for read protection, namely no read protection and low-level read protection.

No read protection: When the read protection register is set to 0xA55A, the read protection is not performed, and the Flash read, write and erase operation is not restricted.

Low-level read protection: When the read protection register is set to values other than 0xA55A and 0xC33C, the low-level read protection takes effect, and the read protection byte read at this time is 0xAAAA. In normal conditions, when booting from the user program and booting from Boot Rom, there is no restriction on the access to the Flash; In Debug mode (SWD), it is not allowed to access the Flash main memory by executing the program in RAM, that is, it is not allowed to read, write the Flash main memory, and to erase the sector (even if write protection is not selected), even if the CPU fetch instructions to read FLASH, but the whole erase is not limited. In the low level read protection state, the operation of indirectly accessing the flash main memory through DMA or any other means in Debug mode (SWD) is also prohibited.

When low-level read protection is in effect, only user code is allowed to read the Flash main program memory in non-debugging mode. In the read protection state, the code executed in the Flash main memory can program Flash to achieve functions such as IAP or data storage. It should be noted that the flash code cannot be debugged under the low-level read protection status, but the low-level read protection can be relieved by loading code to the built-in RAM and executing the code instruction in the Debug menu (SWD), or by using the SinoWealth tool (SWD/ISP) to perform the entire chip erase.

High-level read protection: When the read protection register is set to 0xC33C, the high-level read protection takes effect. Under the high-level read protection, the debugging function (SWD) is closed. In addition, starting from the Boot Rom area is also prohibited.



(2) Write Protection

The write protection bit is used for erasing/programming protection of each sector of Flash, which can prevent accidental operation of Flash memory. The WP[31:0] bit of the write protection byte is 1 to enable the write protection function of the specified sector. SH30F9/SA0 series is based on 8 sectors (1KB/sector), which is fixed to 8KB Flash storage area to provide protection. If the protected sector is programmed or erased, it will not be executed, and an operation error flag will be generated. The following table shows which sector protection is enabled by setting WP[31:0] bit in 256KB/128KB Flash main program area.

Table 8-5 Write Protection Mapping Table

| WP bit | Protected Sector | Protected unit size |
|--------|-----------------------|---------------------|
| WP[0] | Sector0 ~ Sector7 | 8KB |
| WP[1] | Sector8 ~ Sector15 | 8KB |
| WP[2] | Sector16 ~ Sector23 | 8KB |
| ... | ... | ... |
| WP[15] | Sector120 ~ Sector127 | 8KB |
| ... | ... | ... |
| WP[30] | Sector240 ~ Sector247 | 8KB |
| WP[31] | Sector248 ~ Sector255 | 8KB |

When the FLASH sub-block function is turned on, the address of the unit protected by the write-protect bit is in the logical address space.

(3) Full Chip Erase Protection

The overall erasing protection bit of the main program area is controlled by 2bit. When the 2Bit is 01b, it is allowed to execute ME0 operation in normal operation mode (Normal), otherwise it is invalid to execute ME0 operation in normal operation mode (Normal).

8.4 Startup Configuration (BootLoader)

For SH30F9/SA0 series, after power on reset or pin reset, the system hardware and the boot area cooperate through software. According to the current system startup configuration status (read protection status, whether boot area code is valid, Boot pin level status during reset), select the flash main program memory area or boot area as the startup area of the system.

Table 8-6 System Startup Configuration Table

| System configuration status | | | Status after power on reset or pin reset | | |
|-----------------------------|----------------------------|-----------------------------|--|--------------------------|-----------------------------|
| Read Protection level | Whether boot code is valid | Boot pin level during reset | System startup area | Whether debug is enabled | Whether ISP mode is entered |
| No Read Protection | Yes | high | Boot area | Yes | Yes |
| No Read Protection | Yes | low | Main Program Area | Yes | No |
| No Read Protection | No | X | Main Program Area | Yes | No |
| Low-level Read Protection | Yes | high | Boot area | Yes | Yes |
| Low-level Read Protection | Yes | low | Main Program Area | Yes | No |
| Low-level Read Protection | No | X | Main Program Area | Yes | No |
| High-level Read Protection | Yes | X | Main Program Area | No | No |
| High-level Read Protection | Yes | X | Main Program Area | No | No |
| High-level Read Protection | No | X | Main Program Area | No | No |



SH30F9/SA0 Series

When flash main program area is selected as the system startup area, MEMMODE@FLASH_MEMRMP is cleared by hardware, and the hardware automatically keeps the main program area code mapped to the address 0x0000 0000. If the boot area is selected as the system startup area, MEMMODE@ FLASH_MEMRMP is set to 1 by hardware, and the hardware will automatically remap the boot area to the address 0x0000 0000 (only the first 256 bytes of the boot area are mapped), and the content of the address after 0x0FFE0000 can still be accessed normally.

The system startup flowchart is as follows:



SH30F9/SA0 Series

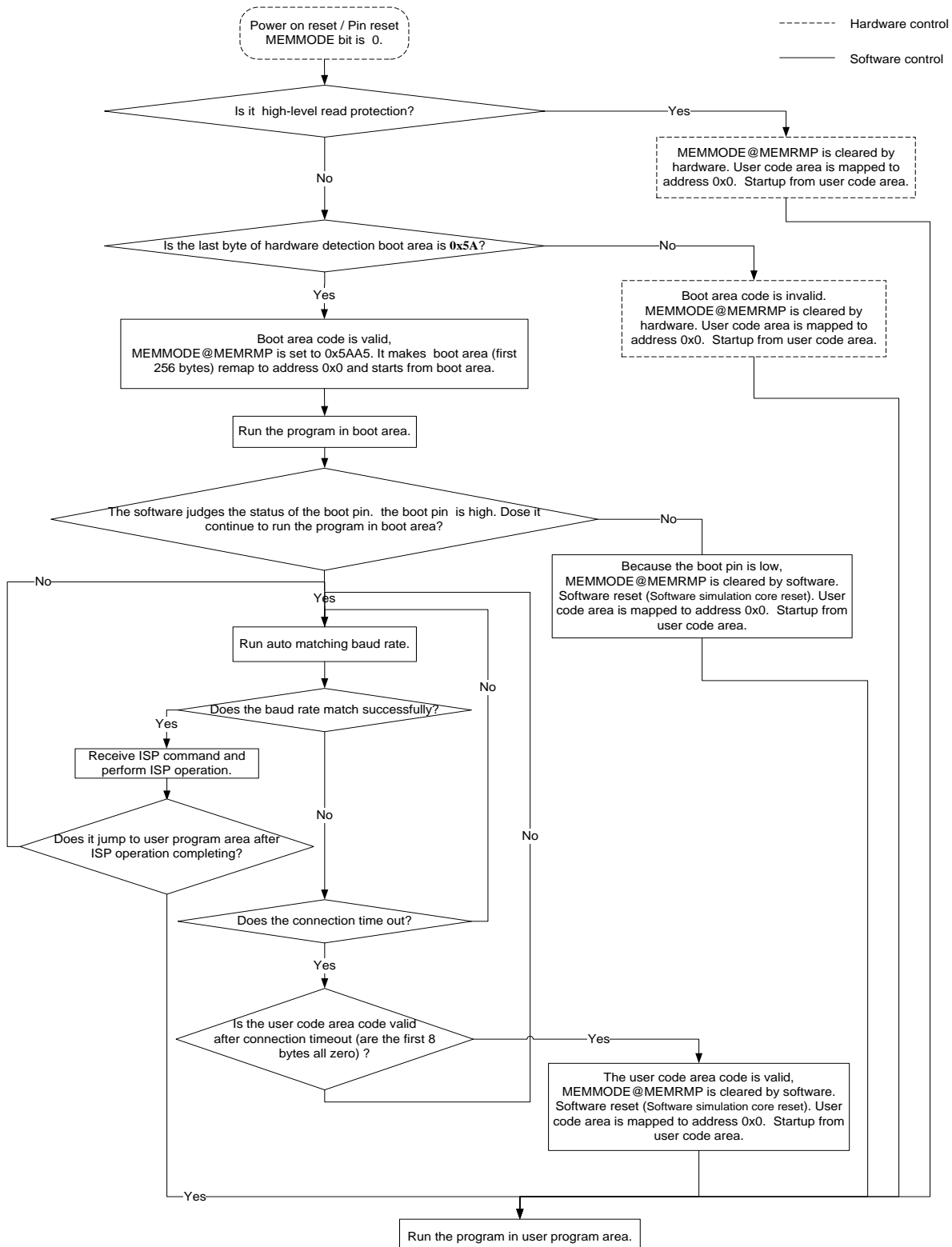


Figure 8-9 The system startup flowchart



After power on reset or pin reset, MEMMODE@FLASH_MEMRMP is cleared by hardware. The hardware automatically keeps the main program area code mapped to the address 0x0000 0000, and then reads the status of the read protection bit. If the current system is in the high-level read protection status, the main program area code (User code) is mapped to address 0x0000 0000 and startup from user code area.

If the chip is not in the high-level read protection state, the hardware detects whether the last byte of the boot area is a specific byte. If the hardware detects that the last byte of the boot area is a non-specific byte, the main program area code (User code) is mapped to address 0x0 and startup from user code area. If the last byte of the boot area is a specific byte, it indicates that the boot area program is valid. MEMMODE@ FLASH_MEMRMP is set to 1 by hardware, and the hardware will automatically remap the boot area to the address 0x0000 0000 (only the first 256 bytes of the boot area are mapped). Startup from the boot area and perform the boot area code.

Run the solidified code in the boot area. First of all, read the status of the boot pin. If the boot pin is low, MEMMODE@ FLASH_MEMRMP is cleared by software, user code area is remapped to address 0x0000 0000, then software reset(Software simulation core reset) and startup from user code area. If the boot pin is high, continue to run the boot area code, and then perform ISP operation. The ISP starts the function of auto matching baud rate first. If the baud rate is successfully matched, it can receive the ISP command and perform the ISP operation. If the baud rate matching connection times out, it can judge whether the user code is valid (the first two words of the user code are not zero). If the user code is valid, MEMMODE@ FLASH_MEMRMP is cleared by software, user code area is remapped to address 0x0000 0000, software reset(Software simulation core reset) and startup from user code area.

After entering the ISP, first match the baud rate, then enter the receiving of ISP command and the writing process of flash memory until the ISP completes. Then exit the ISP through hardware reset.

Boot pin is a general GPIO pin. Any GPIO port can be defined as a boot pin, which depends on bootloader program and is independent of hardware.

Once a GPIO port is defined as the boot pin, the function is limited when it is used as a GPIO port due to the fact that the boot pin generally needs to be added with a pull-down resistance (after bootloader completing, the boot pin can be used as a GPIO port until next power on or reset). The user should ensure that the pin level will not affect the startup mode.

8.5 In System Programming (ISP)

SH30F9/SA0 series supports ISP function. ISP code is integrated in bootloader program and written by SinoWealth in production line. ISP is used to program flash memory through communication interface.

Note: For more about ISP operations, refer to " Startup Configuration (BootLoader) "

8.6 Flash Access Acceleration Control

SH30F9/SA0 series adds a Flash access acceleration control module to balance the contradiction between the high-speed performance of Cortex-M0+ and the slow Flash memory access. For SH30F9/SA0 series, when the system runs at 48MHz main frequency, once insert a waiting time in the flash memory and to turns on the read ahead buffer, which can obtain the system performance close to zero waiting.

8.7 Flash Block Division Function

In order to ensure that the on-chip backup program and the main program can quickly switch, SH30F9/SA0 series is equipped with FLASH segmentation function.

Refer to "User Code Options" for related code options related to segmentation of FLASH blocks. among them:

FLASH_DIVIDE is the FLASH partition enable bit, used to control whether the FLASH partition function is enabled;

FLASH_SELECT is the FLASH logical address mapping switch control bit, which is only valid when FLASH_DIVIDE is '1';

In the Main programs, "Erase" and "Erase-Program" operations are allowed to FLASH_DIVIDE and FLASH_SELECT. At this time, you need to write "0x5AA5" to the LOCK register of the FLASH module to unlock it before modification. The specific operation method is the same as the "Customer Information Area" programming/erasing method.

After modifying FLASH_SELECT, software reset (SW reset) or power-on reset can be used to switch the logical address image corresponding to the PC pointer.

After power-on reset or other system reset, the hardware will automatically map the logical address of FLASH according to the read FLASH_DIVIDE and FLASH_SELECT values.

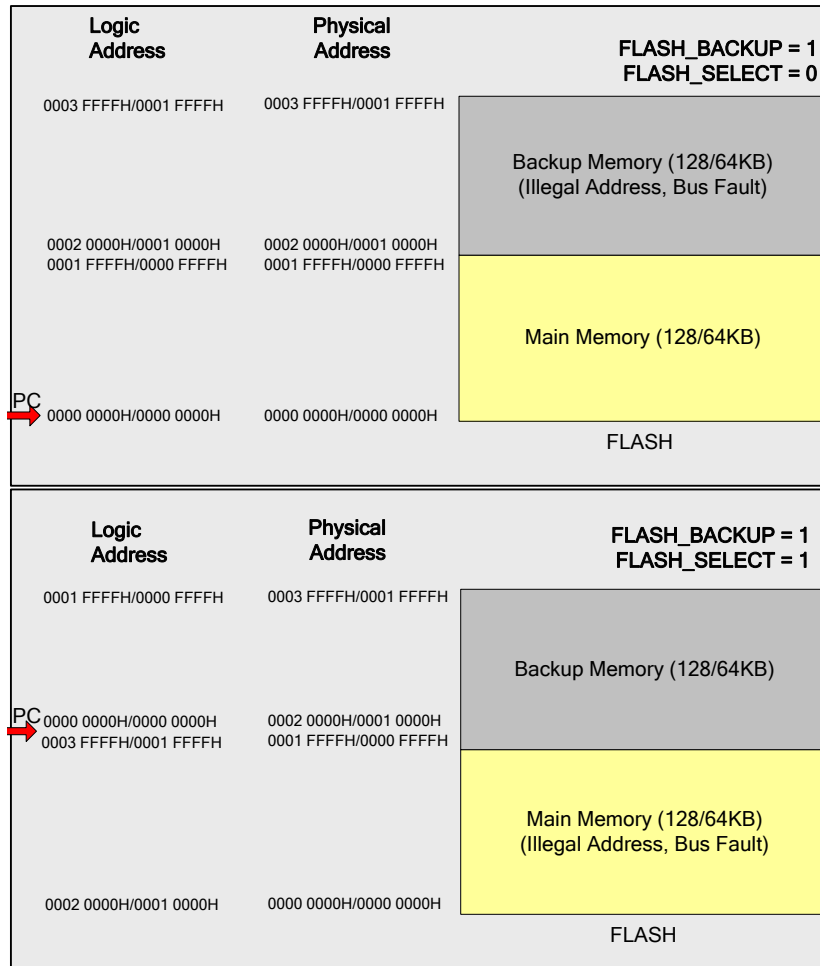


Figure 8-10 Address configuration block diagram after 256KB/128KB flash block division

Take the flash size of 128KB as an example:

(1) When FLASH_SELECT is 0, the logical address of flash area is consistent with the physical address. And after the system is reset, the content of PC pointer is that: point to the logical address 0000 0000H, physical address 0000 0000H;

(2) When FLASH_SELECT is 1, the logical address of the main memory block will swap with backup memory block in flash area. At this time:

A. Main memory block logical address = physical address + 0001 0000H;

B. Backup Memory block logical address = physical address - 0001 0000H;

And after the system is reset, the content of PC pointer is that: point to the logical address 0000 0000H, physical address 0001 0000H.

When FLASH_SELECT is 0, the PC pointer points to the main memory block. When the PC pointer exceeds the logical address range of the main memory block (0000 0000H~0000 FFFFH), it is considered that the program flies and HardFault is generated.

When FLASH_SELECT is 1, the logical address of the main memory block will swap with backup memory block in flash area. The PC pointer points to the backup memory block. When the PC pointer exceeds the logical address range of the backup memory block (0000 0000H~0000 FFFFH), it is considered that the program flies and HardFault is generated.

In a word, when the flash block division function is turned on, the corresponding block can be selected as the area accessed by PC pointer through FLASH_SELECT. If the PC pointer is beyond the range of the selected block, it is considered that the program flies and HardFault is generated. However, it should be noted that the non-selected flash block can still be read/written as ROM at this time, but the program is not allowed to run.



8.8 Registers

FLASH Module Register list (Base Address:0x4004 0C00)

| Address | Register | Description |
|-------------|----------|--|
| 0x4004 0C00 | ACR | Flash access control register |
| 0x4004 0C04 | MKYR | Flash main program storage area unlock control register |
| 0x4004 0C08 | E2KYR | EEPROM-like storage area unlock control register |
| 0x4004 0C0C | SR | Flash status and clear register |
| 0x4004 0C10 | CR | Flash control register |
| 0x4004 0C14 | CR1 | Flash control register 1 |
| 0x4004 0C1C | RPR | Flash read protection register |
| 0x4004 0C20 | WRPR | Flash write protection register |
| 0x4004 0C28 | CNTR | Flash operation timer count register |
| 0x4004 0C2C | UPCNTR | Flash operation timer upper limit |
| 0x4004 0C30 | CNTR | Flash operation timer start register |
| 0x4004 0C34 | IKYR | Flash special information storage area unlock control register |
| 0x4004 0D00 | MEMRMP | Memory map select register |

8.8.1 Flash access control register (FLASH_ACR)

Offset Address: 0x0000

Reset value: 0x0000 0002

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|----------|----------|-----|------------|----------|-----|-----|-----|-----|-----|------------------|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | CRS T | Reserved | | PRF TEN | Reserved | | | | | | LATENCY[1: 0] | |
| | | | | RW | | | RW | | | | | | | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 16 | LOCK[15:0] | Register lock bit 0x5AA5: unlock this register Other: lock this register When configuring this register, the unlock bit must be entered at the same time as 0x5AA5. After the operation is completed, it will be automatically locked. Reading this LOCK bit will always read 0x0000. |
| 15 ~ 12 | Reserved | - |
| 11 | CRST | Prefetch area reset 0: No reset in the prefetch area 1: Prefetch area reset |
| 10 ~ 9 | Reserved | - |
| 8 | PRFTEN | Prefetch Cache buffer enable 0: Close the prefetch buffer 1: Enable prefetch buffer Note: Please refer to Table 10-1 and configure accordingly based on the frequency of HCLK. |
| 7 ~ 2 | Reserved | - |



| | | |
|-------|---------------------|---|
| 1 ~ 0 | LATENCY[1:0] | <p>These bits represent the ratio of SYSCLK (system clock) cycle to Flash access time</p> <p>00: 0 wait state (system clock cycle \geq Flash access time) 01: 1 wait state (system clock cycle \geq Flash access time/2) 10: 2 wait states (system clock cycle \geq Flash access time/3) 11: 3 wait states (system clock cycle \geq Flash access time/4)</p> <p>Note: (1) Flash access time is the Flash read operation speed, please refer to the Flash characteristics section in the Electrical Characteristics chapter for specific definitions. (2) When the system main clock is faster than the Flash access speed, make sure to insert enough wait according to the specific situation. (3) Please refer to Table 10-1 and configure accordingly based on the frequency of HCLK.</p> |
|-------|---------------------|---|

Table 10-1 HCLK configurable frequencies and corresponding LACENCY [1:0], PRFTEN optional configurations

| HCLK configurable frequencies | FLASH_ACR register | |
|-------------------------------|--|--|
| | LATENCY[1:0] corresponding optional configurations | PRFTEN corresponding optional configurations |
| $\leq 12\text{MHz}$ | 00, 01, 10, 11 | 0, 1 |
| 24MHz | 10, 11 | 0, 1 |
| 48MHz | 01, 10 | 0 |

8.8.2 Flash main program storage area unlock control register (FLASH_MKYR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| MKY[31:0] | | | | | | | | | | | | | | | |
| <i>WO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| MKY[31:0] | | | | | | | | | | | | | | | |
| <i>WO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|------------------|---|
| 31 ~ 0 | MKY[31:0] | <p>Flash main program storage area unlock control bit</p> <p>Write the correct Flash unlock value and operation unlock value to MKY[31:0] in turn to unlock the operation of the Flash main program storage area. After the unlock is successful, the bit MNLCK@FLASH_CR is cleared by hardware, and MKY[31:0] is cleared by hardware. .</p> <p>For a single Flash unlock operation, when the bit MNLCK@FLASH_CR is cleared to 0, only one programming or erasing operation is allowed on the Flash. After the operation is completed, the bit MNLCK@FLASH_CR is automatically set to 1, and the Flash main program storage area is locked again.</p> <p>For multiple Flash unlock operations, when the bit MNLCK@FLASH_CR is cleared to 0, multiple programming or erasing operations are allowed on the Flash, but after the operation is completed, the bit MNLCK@FLASH_CR software must be set to 1 to lock the Flash main program again Storage area.</p> <p>These bits can only be written by software</p> |



8.8.3 EEPROM-like storage area unlock control register (FLASH_E2KYR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| E2KY[31:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| E2KY[31:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|------------|---|
| 31 ~ 0 | E2KY[31:0] | <p>Like EEPROM storage area unlock control bit</p> <p>Write the correct Flash unlock value and then operate the unlock value to E2KY[31:0] to unlock the EEPROM-like area. After unlocking is successful, the E2WRE bit in FLASH_CR is cleared by hardware, and E2KY[31:0] is cleared by hardware.</p> <p>For a single Flash unlock operation, when the unlock is successful, only one programming or erasing operation is allowed to the Flash. After the operation is completed, the E2WRE bit is automatically set to 1 by the hardware, and the Flash related block is locked again.</p> <p>For multiple Flash unlocking operations, when the unlocking is successful, multiple programming or erasing operations are allowed on the Flash, but after the operation is completed, the E2WRE bit must be set to 1 by the software to lock the Flash related blocks again.</p> <p>These bits can only be written by software</p> |

8.8.4 Flash status and clear register (FLASH_SR)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|----------|-----|-----|-----|-----|-----|-----|-----------------|-----------------|-----------------|-------------------|-----------------|--------------|------------|----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | STA ERR C | PGW ERR C | PGP ERR C | WRP RTE RRC | FLS ERR C | Rese rved | OPE RRC | EOP C |
| - | | | | | | | | WO | WO | WO | WO | WO | - | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| BSY | Reserved | | | | | | | STA ERR | PGW ERR | PGP ERR | WRP RTE RR | FLS ERR | Rese rved | OPE RR | EOP |
| RO | - | | | | | | | RO | RO | RO | RO | RO | - | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 24 | Reserved | - |
| 23 | STAERRC | <p>Flash status error flag clear bit</p> <p>0: Invalid 1: Clear</p> |
| 22 | PGWERRC | <p>Programming window error flag clear bit</p> <p>0: Invalid 1: Clear</p> |



SH30F9/SA0 Series

| | | |
|--------|------------------|---|
| 21 | PGPERRC | Programming Error Flag Clear Bit 0: Invalid 1: Clear |
| 20 | WRPRTERRC | Write protection error flag clear bit 0: Invalid 1: Clear |
| 19 | FLSERRC | Flash hardware error status clear bit 0: Invalid 1: Clear |
| 18 | Reserved | - |
| 17 | OPERRC | Flash operation error flag clear bit 0: Invalid 1: Clear |
| 16 | EOPC | Operation execution flag 0: Invalid 1: Clear |
| 15 | BSY | Busy flag This bit indicates that the Flash operation is in progress. At the beginning of the Flash operation, this bit is set to '1'; at the end of the operation or an error occurs, this bit is cleared to '0'. |
| 14 ~ 8 | Reserved | - |
| 7 | STAERR | Flash status error flag When the Flash is not unlocked normally, when attempting to erase or program the Flash, this bit is set to '1' by hardware. STAERRC@FLASH_SR writes "1" to clear this state. When STRT@FLASH_CR is not set to 1, try to program Flash, this bit is set to '1' by hardware. STAERRC@FLASH_SR writes "1" to clear this state. |
| 6 | PGWERR | Programming window error flag This bit is 1, indicating that when a Flash operation is started, that is, when the bit STRT@FLASH_CR is set to 1 or a write operation to Flash is started, the count value of the Flash operation timer is not within the valid time window, and the Flash operation is not executed. PGWERRC@FLASH_SR writes '1' to clear this state. |
| 5 | PGPERR | "Programming Error Flag Attempt to program the programmed address (a bit is not 0), and the hardware will set it to '1'. PGPERRC@FLASH_SR writes "1" to clear this state. Note: Before setting the programming operation, the bit STRT@FLASH_CR must be cleared. " |
| 4 | WRPRTERR | Write protection error flag When attempting to program/erase the write-protected Flash address, the hardware sets '1. WRPRTERRC@FLASH_SR writes 1" to clear this state. " |
| 3 | FLSERR | Flash hardware error Flash operation timed out, this error flag is set to 1. FLSERRC@FLASH_SR writes 1" to clear this state. " |
| 2 | Reserved | - |
| 1 | OPERR | Flash operation error flag Flash operation failed and the hardware is set to '1'. OPERRC@FLASH_SR writes 1" to clear this state. For specific errors, see PGWERR, PPGERR, WRPRTERR, and STAERR error flags. " |
| 0 | EOP | Operation execution flag When the Flash program/erase operation is completed, the hardware is set to '1'. EOPC@FLASH_SR writes "1" to clear this state. Note: EOP@FLASH_SR status will be set every time the program or erase is executed. |



SH30F9/SA0 Series

8.8.5 Flash control register (FLASH_CR)

Offset Address: 0x0010

Reset value: 0x0000 D000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CMD[15:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | | |
|--------|--------|-----------|---------|-----------|--------|-----------|-------|----------|----|----|----|----|----|----|----|--|--|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | |
| MNL CK | E2L CK | Rese rved | INFL CK | Rese rved | PSIZ E | Rese rved | STR T | Reserved | | | | | | | | | |
| RW1s | RW1s | - | RW1s | - | RW | - | RW | | | | | | | | | | |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 16 | CMD[15:0] | Flash operation command word 0xAA55: Overall erase of main program area (ME0) 0xE619: Main program sector erase (MSE) 0x6E91: Main program area storage unit programming (MPG) 0xB44B: Special information area E2Prom area programming (E2PG) 0x4BB4: Special information area E2Prom area is erased by sector (like E2Prom) (E2SE) 0xC33C: Special information area Customer Block area programming (CBPG) 0x3CC3: The special information area Customer Block area is erased by sector (Customer Block)(CBSE) 0xD22D: Special information area Protect area programming (SPG) 0xF00F: Special information area OTP area programming (OPG) Other: reserved value Note: This bit can only be modified when the Flash operation is unlocked. |
| 15 | MNLCK | Main program area lock flag After writing the correct Flash unlock value and operating unlock value to FLASH_MKYR in turn, this bit is cleared by the hardware. Only then can the flash main program storage area be erased and programmed. This bit can only be cleared by hardware and set by software. After an unsuccessful unlock operation, this bit cannot be changed again before the next system reset. This bit can only be cleared by hardware. |
| 14 | E2LCK | Like EEPROM storage area lock flag bit After writing the correct Flash unlock value and operating unlock value to FLASH_E2KYR in turn, this bit is cleared by the hardware. At this time, the EEPROM-like storage area is allowed to be erased and programmed. This bit can only be cleared by hardware. |
| 13 | Reserved | - |
| 12 | INFLCK | Special information area lock flag (code protection area, customer information area and OTP area) After writing the correct Flash unlock value and operating unlock value to FLASH_IKYR in turn, this bit is cleared by hardware. At this time, the special information area (code protection area, customer information area and OTP area) is allowed to be erased, Programming operation. This bit can only be cleared by hardware. |
| 11 | Reserved | - |
| 10 | PSIZE | Programming operation bit width selection bit 0: 32-bit simultaneous programming 1: 16-bit simultaneous programming Note: 1) Only works on the main program area . 2) This bit can only be modified when the Flash operation is unlocked. |
| 9 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|-----------------|---|
| 8 | STRT | Start flag When this bit is '1', an operation will be triggered. This bit can only be set to '1' by software and cleared to '0' when BSY@FLASH_SR becomes '1'. Note: This bit can only be modified when the Flash operation is unlocked. |
| 7 ~ 0 | Reserved | - |

8.8.6 Flash control register 1 (FLASH_CR1)

Offset Address: 0x0014

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|------------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | SNB[17:0] | | | | | | | |
| - | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|------------------|--|
| 31 ~ 8 | Reserved | - |
| 7 ~ 0 | SNB[17:0] | Sector erase sector selection bit: 00 00000000 00000000: Sector 0 00 00000000 00000001: Sector 1 00 00000000 00000010: Sector 2 00 00000000 11111110: Sector 254 00 00000000 11111111: Sector 255 Note: 1) This bit is only valid when the main program area is erased by sector or the EEPROM-like area is erased by sector. When the main program area is erased by sector, the size of each sector is 1024 bytes; the EEPROM area is by sector. During zone erasing, the size of each sector is 1024 bytes. 2) This bit can only be modified when the Flash operation is unlocked. |

8.8.7 Flash read protection register (FLASH_RPR)

Offset Address: 0x001C

Reset value: 0x0000 XXXX

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RDP[15:0] | | | | | | | | | | | | | | | |
| <i>RO</i> | | | | | | | | | | | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| Bit | Symbol | Description |
|---------|------------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | RDP[15:0] | Read protection This register contains the read protection option byte loaded by the code protection area loader after reset. |



8.8.8 Flash write protection register (FLASH_WRPR)

Offset Address: 0x0020

Reset value: 0xXXXX XXXX

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| WRP[31:0] | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| WRP[31:0] | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | WRP[31:0] | Write protection This register contains the write protection bytes loaded by the code protection area loader after reset. 0: Write protection failure 1: Write protection takes effect Note: These bits are read-only. |

8.8.9 Flash operation timer count register (FLASH_CNTR)

Offset Address: 0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CNT[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CNT[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | CNT[31:0] | Flash operation timer count value register The Flash operation timer based on the system clock counts down, and only after it reaches the upper limit of the Flash operation timer can the Flash control register be modified. When the Flash operation timer count value is decremented to zero, it is prohibited to modify the Flash control register. |

8.8.10 Flash operation timer upper limit (FLASH_UPCNTR)

Offset Address: 0x002C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| UPCNT[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| UPCNT[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|--------|-------------|---|
| 31 ~ 0 | UPCNT[31:0] | Flash operation timer upper limit Only after the count value of the Flash operation timer is decremented to the upper limit of the Flash operation timer can the Flash control register be modified. After the count value of the Flash operation timer is decremented to zero, modification of the Flash control register is prohibited. |

8.8.11 Flash operation timer start register (FLASH_CNTCR)

Offset Address: 0x0030

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----------------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | | CNT EN RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 1 | Reserved | - |
| 0 | CNTEN | Flash operation timer start bit 0: Flash operation timer disabled 1: Flash operation timer is enabled The flash operation timer counts down to zero, this bit is cleared to 0, and the flash operation timer is disabled. |

8.8.12 Flash special information storage area unlock control register (FLASH_IKYR)

Offset Address: 0x0034

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| INFO_KEY[31:0] WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| INFO_KEY[31:0] WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|--------|----------------|--|
| 31 ~ 0 | INFO_KEY[31:0] | <p>Flash special information storage area unlock control bit</p> <p>Write the Flash unlock value in sequence, and operate the unlock value to INFO_KEY[31:0] to unlock the Flash special information area (code protection area, customer information area and product information area). After the unlock is successful, the bit INFLCK@FLASH_CR is cleared by the hardware. 0, INFO_KEY[31:0] is cleared by hardware.</p> <p>For a single Flash unlock operation, when the unlock is successful, only one programming or erasing operation is allowed to the Flash. After the operation is completed, the INFLCK@FLASH_CR hardware is automatically set to 1, and the Flash related block is locked again.</p> <p>For multiple Flash unlocking operations, when the unlocking is successful, multiple programming or erasing operations are allowed on the Flash, but after the operation is completed, the bit INFLCK@FLASH_CR software must be set to 1 to lock the Flash related blocks again.</p> <p>These bits can only be written by software</p> |

8.8.13 Memory map select register (FLASH_MEMRMP)

Offset Address: 0x0100

Reset value: 0x0000 XXXX

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| MEMMODE[15:0] <i>RW</i> | | | | | | | | | | | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| Bit | Symbol | Description |
|---------|---------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | MEMMODE[15:0] | <p>Memory mapping selection (Memory mapping selection)</p> <p>These bits are set by hardware after reset and can be read and written by software.</p> <p>0x5AA5: Boot area is mapped to address 0x0</p> <p>Other: Main flash memory is mapped to address 0x0</p> <p>As long as 0x5AA5 is not satisfied at any time, it is mapped to the main flash memory</p> |



9. Power Control (PWR)

The system operating voltage (VDD) of SH30F9/SA0 series is 2.0 to 5.5V. An embedded linear voltage regulator is used to supply the system core power.

The SH30F9/SA0 series has a complete set of power-on reset (POR) and brownout detection (BOD) circuits, the BOD is important in some battery applications.

SH30F9/SA0 series retains the independent external reset pin (NRST).

9.1 Warm-up Timer

- Built-in power on warm-up counter to eliminate unstable state of power on
- Built-in oscillator warm-up counter to eliminate unstable state when oscillation startup

SH30F9/SA0 series has a built-in power warm-up counter. It is designed to eliminate unstable state after power on and to do some internal initial operation such as reading customer option etc.

SH30F9/SA0 series has also a built-in oscillator warm-up counter. It is designed to eliminate unstable state when oscillator starts oscillating in the following conditions: Power-on reset, Pin reset, LVR reset, Watchdog Reset and Wake up from Power-down mode.

After power-on, SH30F9/SA0 series will warm up the counting process after the power is turned on and wait for the overflow to perform the oscillator warm-up counting process. After the overflow, the program will run.

Table 9-1 Power Warm-up Time

| Power On Reset | | Low Voltage Reset/ Pin Reset/ WDT Reset (Not in Power-Down Mode) | | Low Voltage Reset/ Pin Reset/WDT Reset (Wakeup from Power-Down Mode) | | Wakeup from Power-Down Mode (Only for interrupt or event) | |
|----------------|-------------|---|-------------|---|-------------|---|-------------|
| Power Warm up | OSC Warm up | Power Warm up | OSC Warm up | Power Warm up | OSC Warm up | Power Warm up | OSC Warm up |
| ≈14ms | Yes | ≈1ms | No | ≈1ms | Yes | ≈800us | Yes |

Table 9-2 OSC Warm-up Time

| Oscillator Type | Warm-up Time |
|--------------------------------------|--|
| 32kHz Crystal (LSE) | $2^{13} \times T_{osc} \approx 256\text{ms}$ |
| High Frequency Crystal/Ceramic (HSE) | $2^{17} \times T_{osc}$ |
| Internal RC 128kHz RC (LSI) | $2^7 \times T_{osc} \approx 1\text{ms}$ |
| Internal RC 24MHz RC (HSI) | $2^7 \times T_{osc} \approx 5.4\mu\text{s}$ |
| PLL Clock | $\approx (0.5\text{ms} + 2^{13} \times T_{osc})$ |

Note: After SH30F9/SA0 series is powered on, high frequency RC (HSI) is selected as the system clock, and warm-up time is defined as $2^7 \times T_{osc}$



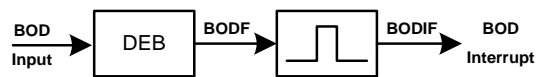
9.2 Brownout Detection (BOD)

- Enable by setting register
- Both rising and falling can generate an interrupt
- Adjustable voltage detection
- Has hysteresis to reduce the impact of voltage fluctuations
- Hardware debounce, debounce time is about 40~80us (Typical value is 60us)

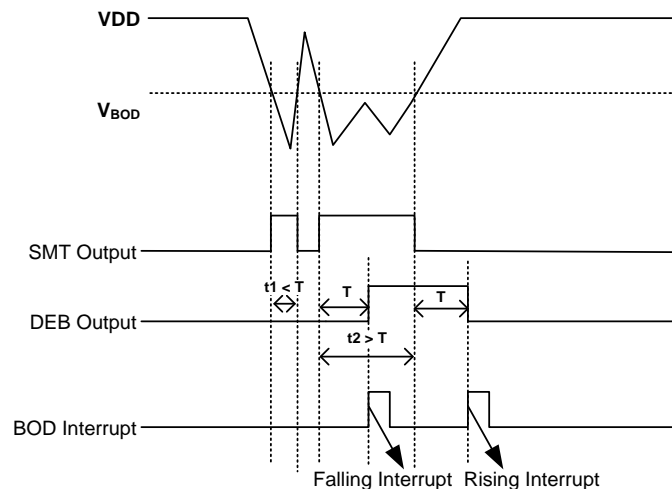
Brownout detection is used to monitor the supply voltage. If the voltage is above or below the specified value, an internal flag is generated and an interrupt is generated. It is mainly used to detect changes in the voltage of power and is also used to monitor the power supply of the system. In some applications, it can be used for low-voltage detection, and the system software can take some protection measures according to it such as data backup, site retention, etc.

BOD cannot wake up stop mode.

BOD contains a comparison circuit of return difference and has hardware jitter removal function. The basic structure is as follows:



The following figure shows the working waveform of BOD under the condition of opening the rise and fall bi-directional interrupt and rise with return difference (BOD up shift voltage):



Due to the two-way detection function, the use of BOD becomes very simple. The typical use steps are as follows:

BOD interrupt starting steps:

- 1) Enable BOD module (BODEN=1);
- 2) Set BOD detection gear register VBOD[3:0];
- 3) Set BOD double edge detection BODMD[1:0]=10b;
- 4) Clear BOD interrupt request flag bit BODIF;
- 5) Set BOD interrupt allow bit BODIE (The corresponding channel of NVIC needs to be opened in cooperation).

In the BOD interrupt, do the following:

- 1) Clear BOD interrupt allow bit BODIE;
- 2) Query BODF to determine whether it is a rise or a fall;
- 3) Perform the corresponding operation according to the query result of 2) (if a one-way detection, no need to query BODF)
- 4) Clear BOD interrupt request flag bit BODIF;
- 5) Set BOD interrupt allow bit BODIE

BOD can cause an NMI interrupt, which is enabled by the IEN_BOD@SYSCFG_SAFR control bit.



9.3 Low-voltage Reset (LVR)

In AC or high-capacity battery applications, switching on a large load can easily cause the MCU power supply to temporarily fall below the set operating voltage. A low voltage reset means that the protection system generates a valid reset below the set voltage.

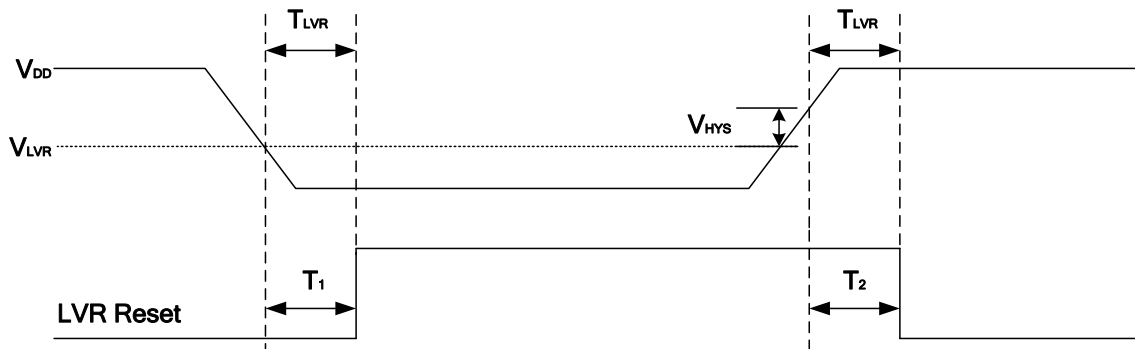
The low voltage reset (LVR) function is to monitor the supply voltage. When the supply voltage is lower than the set voltage V_{LVR} , MCU will generate internal reset.

- Can be chosen to be turned on via register, and can set LVR voltage via register.
- Debounce time of LVR(T_{LVR}) is about 40~80 μ s (Typical value is 60us).
- When power voltage is lower than the set voltage V_{LVR} , internal reset will be generated.

After LVR function is turned on, it has the following features (t indicates the duration of which the voltage is lower than the set voltage V_{LVR}):

When $V_{DD} \leq V_{LVR}$ and $t \geq T_{LVR}$, system reset is generated.

When $V_{DD} > V_{LVR}$ or $V_{DD} < V_{LVR}$, but the duration $t < T_{LVR}$, system reset will not be generated.



In the above figure, V_{DD} is the power supply voltage, V_{LVR} is LVR detection voltage, V_{HYS} is the hysteresis voltage of low voltage reset.

Note: LVR does not reset registers `LVREN@SYSCFG_PWRCR` and `VLVR[1:0]@SYSCFG_PWRCR`.

9.4 Low Power Consumption Management

At the beginning of the design, SH30F9/SA0 series focuses on low-power consumption design to reduce the power consumption of the system operation. In addition, users can also focus on the following aspects to further reduce system power consumption:

- Use low voltage power supply;
- Lower system clock;
- Turn off unused peripheral clocks on APB and AHB buses (default is off);
- Lower the current on GPIO and peripheral interface circuits;
- Avoid floating state when GPIO is input;

Users need to choose a suitable low power consumption mode according to the minimum power consumption, the fastest start time and available wake-up resources.

- Sleep Mode: basic low power mode, the MCU core stops running, but the peripherals used, including the core peripherals of the CPU, such as NVIC, SysTick, etc. are still running;
- Stop Mode: deep low power mode, in addition to shutdown of the core, most of the peripherals used are stopped, leaving only individual peripherals that are responsible for the wake-up task to keep running (clocks provided by LSI), such as IWDG, LVR, etc.



Table 9-3 Low Power Consumption Modes

| Mode | Enter | Wake Up | Delay after wake up | VDD power domain | LDO state |
|------------|--------------------------------|---|--|--|-----------------------------|
| Sleep Mode | WFI/WFE Keep SLEEPDEEP=0 | Any interrupt/wake-up event/reset signal | None | 1. Cortex-M0+ core clock stops; 2. The oscillator and power remain in the original state, and the peripheral clock and part of the analog peripheral power can be turned off by software. | Normal |
| Stop Mode | WFI/WFE Keep SLEEPDEEP=1 | Any external interrupt/reset signal | 1. HSI wake-up time; 2. HSE/PLL wake-up time (if used); 3. LDO wake-up time. | 1. HSI, HSE and PLL are all disabled, and all peripherals using this part of clock are stopped; 2. LSI can be turned on, and all peripherals using LSI can work normally at this time; 3. part of the analog peripheral power can be turned off by software. | Low Power Consumption |

9.4.1 Sleep Mode

Sleep mode can reduce system power consumption. In this mode, the program aborts and the CPU clock stops, but the peripheral clock continues to run. In sleep mode, the CPU stops in a confirmed state, and all CPU states are saved (including the system clock register) before entering sleep mode, SRAM and register contents are retained, and GPIO remains in level state as when it enters.

Enter sleep mode

Keep the SLEEPDEEP bit in the Cortex™-M0+ system control register in the reset state, perform the WFI or WFE instruction and then enter sleep mode. According to the value of SLEEPONEXIT bit in Cortex-M0+ system control register, there are two CPU options that affect entering sleep mode:

- SLEEP-NOW: If SLEEPONEXIT bit is cleared, the microcontroller immediately enters sleep mode when WFI/WFE performed;
- SLEEP-ON-EXIT: If SLEEPONEXIT bit is set, the microcontroller immediately enters sleep mode when the system exiting from the interrupt handler with the lowest priority.

Note: SLEEP-ON-EXIT is a CM0+ specific "automatic sleep" mechanism. It can be programmed to sleep immediately after returning from the interrupt service routine. In this way, all the work of the processor is to respond to the interruption and sleep at other times.

SLEEP-ON-EXIT only supports WFI instruction entry, not WFE instruction

Exit sleep mode

If a WFI instruction is executed to enter sleep mode, any peripheral interrupt that is responded to by the nested vector interrupt controller (NVIC) can wake up the system from sleep mode

If a WFE instruction is executed to enter sleep mode, the MCU will exit from sleep mode whenever a wake event occurs. Awakening events can be generated by:

- Enable an interrupt in the peripheral control register, not in NVIC (nested vector interrupt controller), and enable the SEVONPEND bit in the Cortex™-M0+ system control register. When the MCU wakes up from the WFE, the interrupt pending bit of the peripheral and the NVIC interrupt pending bit of the peripheral (in the NVIC interrupt clear pending register) must be cleared.
- Configure an external or internal EXTI as event mode. When the MCU wakes up from the WFE, it is not necessary to clear the interrupt pending bit of the peripheral or the NVIC interrupt channel pending bit of the peripheral because the pending bit corresponding to the event is not set. This mode takes the shortest amount of time to wake up, as there is no time lost on the entry or exit of interrupts.



Notes:

(1)When MCU wake up from WFI, MCU decide whether or not to wake up based on the priority of the exception system. Only when the priority of the interrupt is higher than the current priority (if WFI is used in the interrupt service routine), can the processor wake up and execute ISR. However, if the PRIMASK is set, the MCU will still be awakened and the ISR will not execute.

(2)When MCU wake up from WFE, regardless of the priority and masking conditions, as long as the SETONPEND setting, any event must occur to wake up the MCU.As for whether or not to implement ISR, It's the same with WFI.

For more details on how to exit sleep mode, please refer to Table 9-4 and Table 9-5.

Table 9-4 SLEEP-NOW mode

| SLEEP-NOW mode | Description |
|----------------|--|
| Enter | Execute a WFI or WFE instruction under the following conditions: – SLEEPDEEP = 0 and – SLEEPONEXIT = 0 Refer to Cortex™-M0+ Control Register |
| Exit | If performing WFI into sleep mode: Interrupt: refer to interrupt vector table If the WFE is executed into sleep mode: Wake up events: refer to wake up event management |
| Wake-up Delay | No |

Table 9-5 SLEEP-ON-EXIT mode

| SLEEP-ON-EXIT mode | Description |
|--------------------|--|
| Enter | Execute a WFI instruction under the following conditions: – SLEEPDEEP = 0 and – SLEEPONEXIT = 1 Refer to Cortex™-M0+ Control Register Note: In order to enter sleep mode, all external interrupt request bits (the pending register (EXTI_PR)) flags must be cleared, otherwise the entry into sleep mode will be skipped and the program will continue. |
| Exit | Interrupt: refer to interrupt vector table |
| Wake-up Delay | No |

Reset signal (RESET, WDT, BOD, LVR) can exit the sleep mode.

After the system wakes up, immediately restore the CPU clock, if the wake up is interrupted to execute the interrupt service program, then jump to the instruction after the instruction into sleep mode. The SRAM remains the same after awakening, while the value of the system register and peripheral register may change depending on the function module Settings (the clock and enable are not turned off).

If the reset signal wakes up, the system starts execution at address 0x0000 0000, and the SRAM and registers are processed according to the reset value.

9.4.2 Stop Mode

The stop mode allows the chip to enter a state of very low power consumption. The stop mode will stop all clocks for the CPU and peripherals (except for the LSI and LSE clock), and manually turn off the power supply of some analog modules according to the control requirements, and put the power supply module into low consumption state. Because the IWDT clock source is LSI, IWDT can continue to operate in stop mode, and LCD and LVR can also operate normally in stop mode (there are independent switches, which can be turned off if not needed). The states of all CPUs are saved before entering stop mode, SRAM and register contents are retained, and GPIO remains in level state as when it enters, but PWM outputs (PWM and PCA modules) are fixedly switched to the high-impedance state.

Note: For digital peripherals, such as UART, SPI, TIMER, etc., although the module enabling is not turned off during stop mode, the module is in stop state due to the clock cut off, and the module pin level remains at the level before stop mode.

For PWM module, not only is it stopped when the shutdown takes place, it will also be forced to switch to high-impedance output to protect the external drive circuit. In application, in order to avoid the state change before and after the shutdown from affecting the system operation, it is recommended that the user manually turn off the PWM module before entering stop mode, and then re-initialize the PWM after wake-up.



Note: When the PWM module is enabled normally, as long as the clock is turned off (through RCC_APB0ENR register), it will immediately be forced to switch to high-impedance state.

HSE, HSI, LSE, LSI and PLL circuits are all automatically turned off during stop mode. If the LCD circuit is turned on, the LSI or LSE clock remains output and is not turned off.

When the system enters the shutdown mode, the control bits of the RCC module, such as PLLON, HSEON, LSEON, PLLRDYIE, HSERDYIE, LSERDYIE, remain unchanged, and PLLRDY, HSERDY, LSERDY, PLLRDYIF, HSERDYIF, LSERDYIF and other flag bits restore to reset values. After exiting the shutdown mode, the hardware restores all clocks. The system clock is controlled by ESTCK@RCC_CR Register selects HSI or LSI and automatically turns on.

In stop mode, the following modules can operate normally:

- Independent watchdog (IWDG), once IWDG is started, it can no longer be stopped except for system reset ^(note);
- Wakeup Timer, which uses the TIM0~3 timer, requires to configure LSI, LSE, or external clock;
- LSI / LSE will not be turned off in stop mode, when LCD driver is turned on;

Note: By default, IWDG is turned off in stop mode to avoid IWDG from waking up the shutdown. The bit IWDTPE@IWDG_CR can be set, in which case the stop mode will be woken up by IWDG.

If the ADC circuit is not turned off in stop mode, these peripherals still consume current. The register enabled bits of these modules can be set to turn off these peripherals.

In stop mode, the internal LDO is fixed into the low consumption state.

The clock safety monitoring (CSM) module is invalid in stop mode, and the CSM can work normally in sleep mode.

Enter stop mode

About how to get into stop mode, refer to Table 9-6.

- If MCU are programming flash, the system does not go into stop mode until the flash programming is complete;
- If an access to AHB/APB is in progress, the system will not go into stop mode until the access to AHB/APB is completed.
- If a DMA transfer is in progress, the DMA transfer is suspended until it wakes up, but the system does not enter stop mode until current DMA operation is completed;
- If MCU executes interrupt service program, or there are interrupts in the queue, the system will enter the stop mode, except for EXTI and Wakeup Timer;

Notes: If there are EXTI, Wakeup Timer interrupts before entering stop mode, the entering stop mode process will be skipped unless the corresponding interrupt request flag is cleared.

Table 9-6 STOP mode

| STOP mode | description |
|---------------|---|
| Enter | Execute WFI or WFE instructions under the following conditions: -Set the SLEEPDEEP bit in the cortex-M0+ system control register Note: in order to enter the stop mode, all external interrupt request bits (pending register (EXTI_PR)) flags must be cleared, otherwise the entry into the stop mode will be skipped and the program will continue. |
| Exit | If the WFI is executed into stop mode: -Set any external interrupt to interrupt mode (the corresponding external interrupt vector must be enabled in NVIC).See NVIC's interrupt vector table interrupt vector table. If the WFE is executed into stop mode: Set any external interrupts to event mode. See "wake up event management" in EXTI. |
| Wake-up Delay | HSI RC wake up time + LDO wake up time from low power |



Exit stop mode

About how to exit stop mode, refer to Table 9-6.

When an interrupt or wake-up event causes an exit from shutdown mode, the ESTCK@RCC_CR register selects HSI or LSI to be automatically turned on and selected as system clock. Then, the software controls to turn on HSE, PLL (if used) and switch system clock to target clock source. Similar to when the system is powered on, it needs to wait for power supply and clock warm up.

Reset signal (RESET, IWDG, LVR) can exit the stop mode. After reset, the program starts to run from 0x0000 0000 address. If system exits the stop mode by interrupt awakening, system also need to execute the interrupt service routine, and then jumps to the instruction after the instruction into stop mode.

Interrupt Wake-up sources include external interrupt (EXTI) and Wakeup Timer interrupt (Wakeup Timer)

If the interrupt is disabled, you can also be awakened by an EXTI event and a wake timer overflow event.

Notes: EXTI and Wakeup Timer are both can generate interrupts and events. Because Wakeup Timer are non-maskable events, it will happen if they are turned on and the conditions are met, therefore, Wakeup Timer awakens the stop model actually is the result of the event (first event and next interrupt), even if the Wakeup Timer interrupt are disabled, they will also be able to wake up system form the stop mode. If the interrupt is enabled, the system will be waked up and enter the corresponding interrupt service routine.

9.5 Debug in stop mode

By default, if system enters the stop mode while debugging, system will lose the debugging connection because there is no clock in stop mode. Consumer can debug the software in stop mode by setting certain configuration bits in the DBGMCU_CR register.

Refer to the debug interface for "debugging support in stop mode" for more details



SH30F9/SA0 Series

9.6 Registers

SYSCFG Module Register list (Base Address:0x4004 0400)

| Address | Register | Description |
|-------------|----------|------------------------|
| 0x4004 0400 | PWRRCR | Power control register |
| 0x4004 0404 | PWRSR | Power Status Register |

9.6.1 Power control register (SYSCFG_PWRRCR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|--------|-----------|--------|--------|------------|----------|-----------|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | LVR EN | VLVR[1:0] | BOD EN | BOD IE | BODMD[1:0] | Reserved | VBOD[3:0] | | | | | |
| 0 | 0 | 0 | 0 | RW | RW | RW | RW | RW | - | RW | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 12 | Reserved | - |
| 11 | LVREN | LVR enable bit 0: Disable low voltage reset function 1: Enable low voltage reset function Note: LVR and SYSCFGRST will not reset the enable bit |
| 10 ~ 9 | VLVR[1:0] | LVR reset voltage gear setting 00: 4.1 V 01: 3.7 V 10: 2.8 V 11: 2.3 V Note: LVR and SYSCFGRST will not reset the enable bit |
| 8 | BODEN | BOD enable bit 0: Brownout detection is disabled 1: Brownout detection is enabled |
| 7 | BODIE | BOD interrupt enable bit 0: Disable BOD interrupt 1: Enable BOD interrupt |
| 6 ~ 5 | BODMD[1:0] | BOD mode selection control bit 00: The BODIF flag is set to 1 only when VDD voltage is from less than BOD threshold to greater than BOD threshold 01: The BODIF flag is set to 1 only when VDD voltage is from greater than BOD threshold to less than BOD threshold 10: When VDD voltage is from greater than threshold to less than threshold, or is from less than threshold to greater than threshold, the BODIF flag is set to 1 11: Reserved |
| 4 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|------------------|--|
| 3 ~ 0 | VBOD[3:0] | BOD threshold voltage setting bit 0000: 2.40 V 0001: 2.55 V 0010: 2.70 V 0011: 2.85 V 0100: 3.00 V 0101: 3.15 V 0110: 3.30 V 0111: 3.45 V 1000: 3.60 V 1001: 3.75 V 1010: 3.90 V 1011: 4.05 V 1100: 4.20 V 1101: 4.35 V 1110: 4.50 V 1111: 4.65 V |
|-------|------------------|--|

9.7 Power Status Register (SYSCFG_PWRSR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------|---------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | BOD F | BOD IF |
| - | | | | | | | | | | | | | | RO | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------------|--|
| 31 ~ 2 | Reserved | - |
| 1 | BODF | BOD flag bit 0: The current voltage is higher than BOD voltage set in VBOD[3:0] 1: The current voltage is lower than BOD voltage set in VBOD[3:0] |
| 0 | BODIF | BOD interrupt request flag 0: No interrupt pending, clear pending 1: Interrupt pending |



10. Reset and Clock Control (RCC)

10.1 Reset

The reset signals of SH30F9/SA0 series are divided into two types: system reset (SYSRESET) and power reset (PWRESET).

Note: The system Reset is Warm Reset and will not reset the Debug circuit. The power Reset is Cold Reset and will reset the Debug circuit.

10.1.1 System Rest

Except for the reset flag bit in RCC_CSR register of the clock controller, system reset will reset all registers to their reset state. A system reset is generated when any of the following events occur:

- Low level on the NRST pin (external reset)
- Independent watchdog count termination (IWDT reset)
- Window watchdog count termination (WWDT reset)
- Software reset (SW reset)

The source of the reset event can be identified by checking the reset status flag bit in RCC_CSR control status register.

10.1.1.1 Pin Reset

SH30F9/SA0 series has an independent reset pin, NRST, and a period of time of low level on this pin will cause a system reset.

10.1.1.2 Independent Watchdog Reset (IWDT)

The independent watchdog timer is a 12-bit decrement counter with an independent built-in RC oscillator as its clock source, which is off by default. It can be turned on by register IWDTON@IWDT_CR. Once turned on, it can only be closed by reset again.

After independent watchdog reset, IWDTRSTF@RCC_CSR flag is generated.

See the introduction of "Independent Watchdog" part for details.

10.1.1.3 Window Watchdog Reset (WWDT)

Window watchdog feeding area is a window area In order to detect software failures caused by external interference or unpredictable logic conditions that cause applications to deviate from the normal running sequence. It is suitable for applications that work in a precise timing window. A reset signal is generated when the window watchdog is refreshed too early or too late.

After window watchdog reset, WWDTRSTF@RCC_RSTSTR flag is generated.

See the introduction of "Window Watchdog" part for details.

10.1.1.4 Software Reset

Software reset is achieved by setting the SYSRESETREQ bit to '1' in application interrupt and reset control register of the Cortex™-M0+ core.

For more information, please see the ARM official manual" Cortex™-M0+ Technical Reference Manual ".

10.1.1.5 CPU LOCKUP

When CPU lockup occurs, point the PC pointer to 0xFFFFF0 and wait for WDT to reset. For lockup, please see the ARM official manual" Cortex™-M0+ Technical Reference Manual".



10.1.2 Power Reset

A power reset occurs when one of the following events occurs:

- Power-on reset (POR reset)
- Low voltage reset (LVR)

Power reset will reset all registers. The reset entry vector is fixed 0x0000_0004.

10.1.2.1 Power-on Reset

SH30F9/SA0 series has a complete power-on reset (POR) circuit inside, and the system works normally when the supply voltage reaches V_{POR} . When VDD is lower than the specified limit voltage V_{POR} , the system remains in the reset state.

For power-on reset, see the introduction of “Electrical Characteristics” part for details.

10.1.2.2 Low Voltage Reset

Low Voltage Reset (LVR) is a mandatory protection reset that can be turned on by the register LVREN@SYSCFG_PWRCR. When the power supply voltage is lower than V_{LVR} , the MCU will generate a reset.

See the introduction of “Low Voltage Reset (LVR)” part for details.



10.2 Clock

Four different clock sources can be used to drive the system clock (SYSCLK):

- HSI oscillator clock (24MHz)
- HSE oscillator clock (4MHz~12MHz Crystal/Ceramic)
- LSI oscillator clock (128kHz)
- LSE oscillator clock(32kHz Crystal)
- PLL clock(MAX:48MHz)

The PLL clock can select HSI and HSE as the clock source.

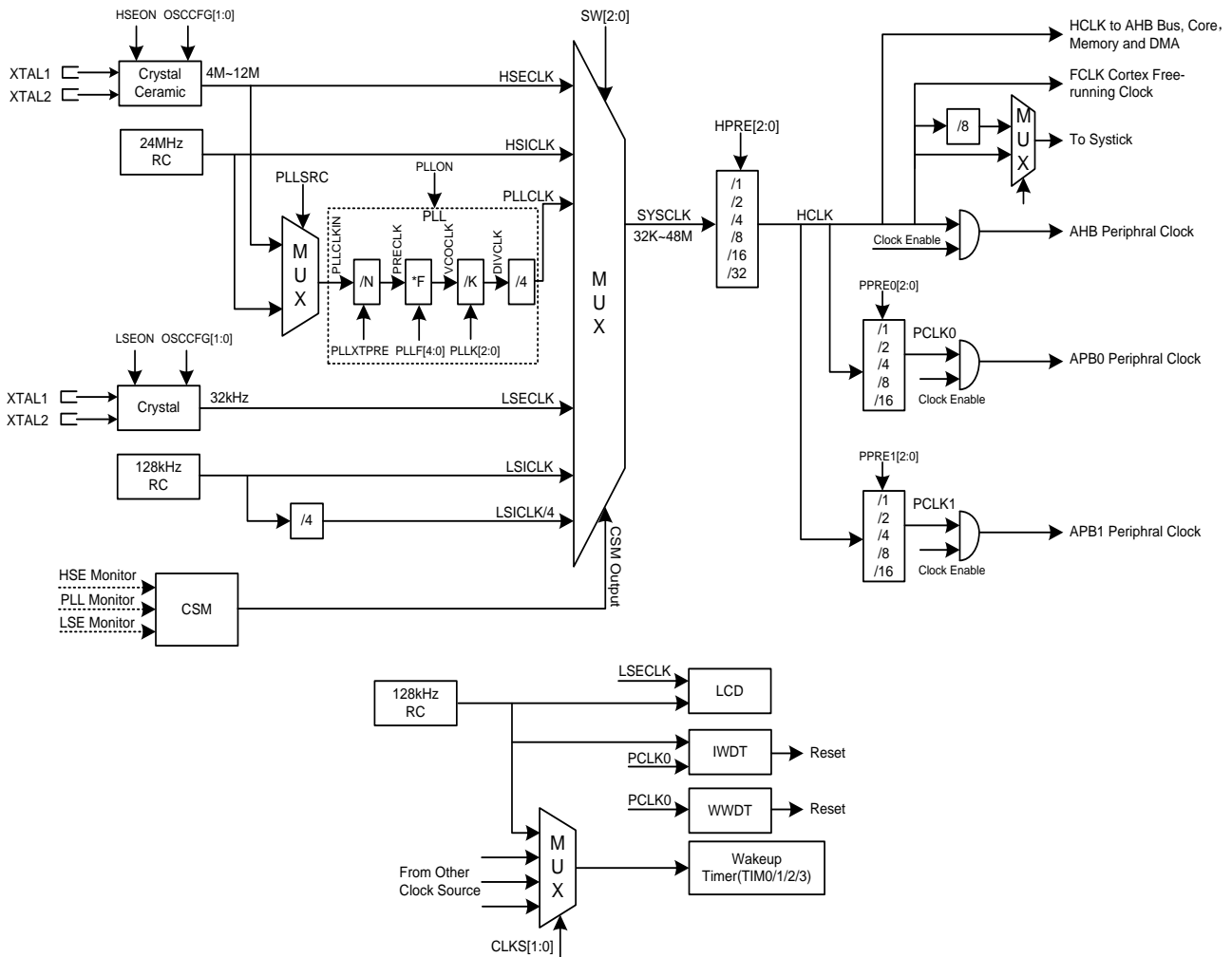


Figure 10-1 Clock Block Diagram of SH30F9/SA0 series

Note: The block diagram is used to illustrate the overall clock tree structure of the system. Please refer to this chapter for control details.



User can configure the frequencies of AHB, APB0, and APB1 through multiple prescalers.

- HCLK: AHB bus clock, with maximum frequency being 48MHz. Cortex-M0+ core, memory, DMA etc. are driven by HCLK. HCLK can be configured with frequencies and corresponding LACENCY [1:0], and PRFTEN can be configured as shown in Table 10-1. It is recommended that users configure HCLK frequencies to be used with the listed configurable frequencies, and it is not recommended to use frequencies not listed in the table. It is also recommended to choose the corresponding optional configuration items in the table for LACENCY [1:0] and PRFTEN configurations, and it is not recommended to choose configuration items that are not listed. For example, it is not recommended to allocate the HCLK frequency to 16MHz, 32MHz, etc. that are not listed in the table; When the HCLK frequency is set to 48MHz, it is not recommended to configure it as "00" and "11" that are not listed in the table, and PRFTEN does not recommend configuring it as "1" that is not listed in the table.

Table 10-1 HCLK configurable frequencies and corresponding LACENCY [1:0], PRFTEN optional configurations

| HCLK configurable frequencies | FLASH_ACR register | |
|-------------------------------|--|--|
| | LATENCY[1:0] corresponding optional configurations | PRFTEN corresponding optional configurations |
| ≤12MHz | 00, 01, 10, 11 | 0, 1 |
| 24MHz | 10, 11 | 0, 1 |
| 48MHz | 01, 10 | 0 |

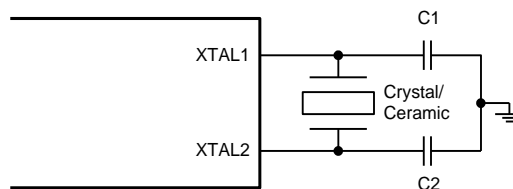
- PCLK0: APB0 bus clock, with maximum frequency being 24MHz. Peripherals on APB0 bus are driven by PCLK0. It is recommended that users do not set the frequency higher than 24MHz;
- PCLK1: APB1 bus clock, with maximum frequency being 24MHz. Peripherals on APB1 bus are driven by PCLK1. It is recommended that users do not set the frequency higher than 24MHz;

After the RCC is divided by 8 via the AHB clock (HCLK), RCC is used as the external clock of SysTick. The above clock or core clock can be selected as the SysTick clock source by setting SysTick control and status registers.

10.2.1 HSE Clock

High speed external clock signal (HSE) is generated by two clock sources:

- (1) External crystal/ceramic resonator: 4MHz - 12MHz



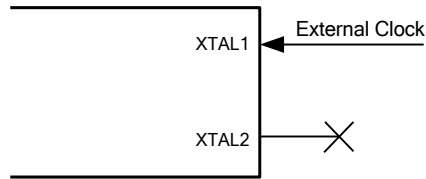
The 4MHz ~12MHz external oscillator (including crystal resonator and ceramic resonator) provides a more accurate master clock for the system. The HSERDY@RCC_CR bit in the clock control register is used to indicate if the high speed external oscillator is stable. At startup, the clock is not released until this bit is set to '1' by hardware. If an interrupt is allowed to be generated in the clock interrupt register RCC_CIENR, a corresponding interrupt will be generated.

HSE crystal can be turned on and off by setting the HSEON@RCC_CR bit in the clock control register.

In order to reduce the distortion of the clock output and shorten the startup stabilizing time, the crystal/ceramic resonator and the load capacitor must be as close as possible to the oscillator pins. The load capacitance value must be adjusted according to the selected oscillator.



(2) External input clock: 4MHz – 12MHz



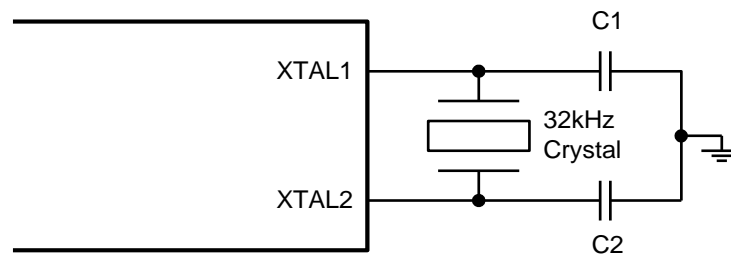
In this mode, external clock must be provided. Its frequency could be as high as 12MHz. In this mode, the user must first fill the external clock on XTAL1, then turn on the HSEON bit, while XTAL2 must be floating at this time. However, XTAL2 can be set to be used as a GPIO port by setting the OSCCFG@SYSCFG_SAFR bit to 10b.

Same as the external oscillator, the external clock also has a warm-up time and a Ready signal.

The external clock is also monitored by the CSM.

10.2.2 LSE Clock

Low speed external clock signal (LSE) is generated by external 32kHz crystal resonator:



32kHz external oscillator provides a more accurate low speed clock for the system. The LSERDY@RCC_CR bit in the clock control register is used to indicate if the low speed external oscillator is stable. At startup, the clock is not released until this bit is set to '1' by hardware. If an interrupt is allowed to be generated in the clock interrupt register RCC_CIENR, a corresponding interrupt will be generated.

LSE crystal can be turned on and off by setting the LSEON@RCC_CR bit in the clock control register.

In order to reduce the distortion of the clock output and shorten the startup stabilizing time, the crystal resonator and the load capacitor must be as close as possible to the oscillator pins. The load capacitance value must be adjusted according to the selected oscillator.

10.2.3 HSI Clock

The HSI clock signal is generated by internal 24MHz RC oscillator and can be used directly as the system clock or as PLL input. When acting as PLL input, it is recommended to turn on the PLL prescaler factor PLLXTPRE@RCC_CFGR, which also means pre-division is turned on.

HSI also has the following features:

- Startup time of HSI RC.
- After system reset, select HSI as the system clock.
- The start-up time of HSI RC oscillator is shorter than that of HSE crystal oscillator
- The HSI RC is calibrated to 0.3% (25°C) after production and does not exceed 1% over the full temperature range.



10.2.4 LSI Clock

The LSI clock signal is generated by an internal 128KHz RC oscillator

The LSI clock is generated by an internal 128KHz RC oscillator, providing a low-power clock source that can be used directly as system clock but not as PLL input.

LSI has the following features:

- If 128KHz RC is used as LCD clock source, it is still on in the stop mode
- RC calibration is not supported by LSI

10.2.5 WDT clock

WDT RC provides a low power clock source for the driver of the stop mode wake module, including IWDT and Wakeup Timer. WDT clock cannot be used for system clock. The WDT clock frequency is about 128KHz.

- The WDT clock is always on when system is powered on
- The WDT clock is also remain open in stop mode
- RC calibration is not supported by WDT clock
- IWDT uses 128KHz RC as clock source

10.2.6 PLL clock

The internal PLL can be used to multiply by frequency the internal RC output clock of the HSI or the external crystal oscillator output clock of the HSE.

The PLL settings must be completed before PLL is activated. These parameters include PLLSRC, PLLXTPRE, PLLF, PLLK, and they cannot be changed once PLL is activated.

If PLL interrupt is allowed in the clock interrupt register, an interrupt request can be generated when PLL is ready.

The PLL frequency multiplication factor consists of three parameters, N, F, and K. The calculation formula is:

$$\text{PRECLK} = \frac{\text{PLLCLKIN}}{N} \quad (1)$$

$$\text{VCOCLK} = \text{PRECLK} \times F \quad (2)$$

$$\text{PLLCLK} = \frac{\text{VCOCLK}}{K} \times \frac{1}{4} \quad (3)$$

N: Determined by the prescaler factor PLLXTPRE, taking 1 or 4;

F: Specified by the frequency doubling factor PLLF, ranging from 15 to 46;

K: Specified by the frequency division factor PLLK, ranging from 1 to 8;

For HSI clock source, it is recommended that the prescaler factor N be 2 or 4 to obtain an ideal clock input with 50% duty cycle.

For HSE clock source, it is recommended to take N as 1, input it directly.

For PLL module, the PLLCLKIN input clock range is 4MHz to 24MHz, the PRECLK prescaler clock is required to meet the range of 4MHz to 8MHz, and the VCOCLK multiplier output clock is required to meet the range of 150MHz to 300MHz. The final PLLCLK clock is required to meet the range of 4MHz to 48MHz. The configuration of N, F, and K parameters must meet the above clock range requirements.

Take the setting of 48MHz system main frequency as an example:

- (1) Select 24MHz HSI as the clock source, set PLLXTPRE=3 (N=4), PLLF=17 (F=32), PLLK=0 (K=1), then PRECLK=6MHz, VCOCLK=192MHz, output PLLCLK is $6 \times 32 / 4 / 1 = 48\text{MHz}$;
- (2) Select 8MHz HSE as the clock source, set PLLXTPRE=0 (N=1), PLLF=9 (F=24), PLLK=0 (K=1), then PRECLK=8MHz, VCOCLK=192MHz, output PLLCLK is $8 \times 24 / 4 / 1 = 48\text{MHz}$.



10.2.7 System Clock (SYSCLK) Selection

After system reset, the HSI oscillator is selected as the system clock. When the clock source is used directly or indirectly through PLL as the system clock, it cannot be stopped. Switching from one clock source to another is allowed only when the target clock source is ready (after a delay in the startup stabilization phase or PLL stabilization).

The status bit in the clock control register (RCC_CR) indicates which clock is ready and which clock is currently being used as the system clock.

10.2.8 Clock Safety Monitoring (CSM)

In order to enhance the reliability of the system and prevent serious consequences of system failure or even crash due to clock failure, SH30F9/SA0 series adds a clock safety monitoring (CSM) module. CSM monitors three sources:

- (1) The oscillation of the HSE, including the conditions of the crystal oscillator, the ceramic oscillator, or the external clock input. Once the vibration or oscillation is stopped or abnormal (the frequency is lower or higher than the normal value). The abnormality flag HSECSMF is set
- (2) The oscillation of the LSE. Once the 32kHz crystal oscillator is stopped (if the LSE is turned on, even if the LSE is not used as the system clock source), The abnormality flag LSECSMF is set
- (3) The loss of lock for PLL (in the case PLL is used), once loss of lock occurs, the exception flag PLLCSMF is set;

The user can query the corresponding interrupt flag at CISTR@RCC to determine what kind of exception it is. If the above exception affects the system clock, a CSM interrupt is requested, which is connected to NMI non-maskable interrupt (enabled by the IEN_CSM@SYSCFG.SAFR control bit). For specific system clock and exception relationships, and when to generate CSM interrupts, see "CSM Exception Function Definition and Processing Table" below.

CSM is turned off by default, but can be turned on by register CSMON@RCC_CR. When CSM is on, monitoring is only valid after LSE Ready, HSE Ready, or PLL Lock.

Table 10-1 CSM exception function definition

| System clock | HSE abnormalities | LSE abnormalities | PLL abnormalities | CSM flag | CSM handling (CSM interrupt is NMI) |
|--------------|-------------------|-------------------|-------------------|--|-------------------------------------|
| LSI/HSI | 0 | 0 | 1 | PLLCSMF=1 | Produce CSM interrupt |
| | 0 | 1 | 0 | LSECSMF=1 | |
| | 0 | 1 | 1 | PLLCSMF=1 LSECSMF=1 | |
| | 1 | 0 | 0 | HSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | |
| | 1 | 0 | 1 | HSECSMF=1 PLLCSMF=1 | |
| | 1 | 1 | 0 | HSECSMF=1 LSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | |
| | 1 | 1 | 1 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | |



SH30F9/SA0 Series

| | | | | | |
|---------|---|---|---|--|---|
| HSI+PLL | 0 | 0 | 1 | PLLCSMF=1 | (1) System clock switch to HSI clock; (2) Produce CSM interrupt; |
| | 0 | 1 | 0 | LSECSMF=1 | Produce CSM interrupt |
| | 0 | 1 | 1 | PLLCSMF=1 LSECSMF=1 | (1) System clock switch to HSI clock; (2) Produce CSM interrupt; |
| | 1 | 0 | 0 | HSECSMF=1 | Produce CSM interrupt |
| | 1 | 0 | 1 | HSECSMF=1 PLLCSMF=1 | (1) System clock switch to HSI clock; (2) Produce CSM interrupt; |
| | 1 | 1 | 0 | HSECSMF=1 LSECSMF=1 | Produce CSM interrupt |
| | 1 | 1 | 1 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | (1) System clock switch to HSI clock; (2) Produce CSM interrupt; |
| HSE | 0 | 0 | 1 | PLLCSMF=1 | Produce CSM interrupt |
| | 0 | 1 | 0 | LSECSMF=1 | |
| | 0 | 1 | 1 | PLLCSMF=1 LSECSMF=1 | |
| | 1 | 0 | 0 | HSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | (1) System clock switch to HSI clock; (2) Produce CSM interrupt; |
| | 1 | 0 | 1 | HSECSMF=1 PLLCSMF=1 | |
| | 1 | 1 | 0 | HSECSMF=1 LSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | |
| | 1 | 1 | 1 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | |
| HSE+PLL | 0 | 0 | 1 | PLLCSMF=1 | (1) System clock switch to HSI/2 clock; (2) Produce CSM interrupt; |
| | 0 | 1 | 0 | LSECSMF=1 | Produce CSM interrupt |
| | 0 | 1 | 1 | PLLCSMF=1 LSECSMF=1 | (1) System clock switch to HSI/2 clock; (2) Produce CSM interrupt; |
| | 1 | 0 | 0 | HSECSMF=1 PLLCSMF=1 | |
| | 1 | 0 | 1 | HSECSMF=1 PLLCSMF=1 | |
| | 1 | 1 | 0 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | |
| | 1 | 1 | 1 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | |



SH30F9/SA0 Series

| | | | | | |
|-----|---|---|---|--|---|
| LSE | 0 | 0 | 1 | PLLCSMF=1 | Produce CSM interrupt |
| | 0 | 1 | 0 | LSECSMF=1 | (1) System clock switch to LSI/4 clock (2) Produce CSM interrupt; |
| | 0 | 1 | 1 | PLLCSMF=1 LSECSMF=1 | |
| | 1 | 0 | 0 | HSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | Produce CSM interrupt |
| | 1 | 0 | 1 | HSECSMF=1 PLLCSMF=1 | Produce CSM interrupt |
| | 1 | 1 | 0 | HSECSMF=1 LSECSMF=1 When HSE is the PLL clock source: PLLCSMF=1 | (1) System clock switch to LSI/4 clock; (2) Produce CSM interrupt; |
| | 1 | 1 | 1 | HSECSMF=1 LSECSMF=1 PLLCSMF=1 | |

From the table above, even if HSE/LSE is not directly or indirectly used as a system clock, once the HSE/LSE stops, the CSM will set the flag, and the CSM interrupt is generated. Similarly, even if the PLL is not used as the system clock, once the PLL is out of lock, it will set the flag, which is set CSM interrupt is generated.

After the occurrence of CSM, in addition to HSECSMF, LSECSMF and PLLCSMF is placed, some control bits of the RCC will not be cleared, such as HSEON, LSEON, PLLON, SW, SWS are maintained at the original value, the system clock has switched to HSI or LSI/4 clock, but SW remains the original setting.

When recovering from a clock failure state, once the failed clock state is restored (whichever is set READY), the user only needs to clear CSM flag to restore the original system clock without switching SW (SW to remain the original setting). It is important to note that the READY bit is not automatically detected on the hardware when the clock is restored. If the target clock is still in a failed state when the clear CSM flag is still in the invalid state, the system clock switches to the target clock first, and then instantly switches back to the HSI clock, which is not expected, so it is not expected that the READY bit should be queried before the clock is restored, and the READY position indicates that the corresponding clock has been restored before the CSM flag can be restored.

If the HSE, LSE or PLL is turned off, the CSM to detect the failure of the corresponding module is automatically turned off.

Note: The system clock selects HSE or HSE + PLL. When the CSM module is turned on, the HSI shall not be turned off, otherwise the CSM will work abnormally.

Because CSM interrupts are non-shieldable interrupts (turned on by IEN_CSM@SYSCFG_SAFR), this must be handled in the software.

The following code indicates the process of clock failure and recovery: for example, the system clock is HSI + PLL, the PLL lock causes the clock to fail.



```
Void NMIException(void)
{
    if (RCC_GetITStatus(RCC_IT_CSM) != RESET) //A clock lose efficacy interrupt is generated
    {
        // The system clock has automatically switched to HSI
        ..... // users can add app-related protection code
        /* Below is ready to restore the clock by querying */
        /*****/
        if (RCC_GetINTStatus (RCC_INT_PLLRDY) != RESET)
        {
            RCC_ClearITPendingBit(RCC_IT_CSM); // Clear interrupt pending bit
            // Because SW remains at its original value, the original clock is immediately restored when the CSM is cleared
        }
        /*****/
    }
}
```

10.2.9 System Tick Timer (SysTick)

SysTick is a 24-bit down-count timer implemented in the Cortex- M0+ core. A fixed interval of time ticks can be provided, which can be used for monitoring or timing in the main flow. Because all Cortex- M0+ chips have SysTick, it helps cross-platform migration for operating systems or programs.

SysTick can use core clock or external reference clock, and its period can be set by the "Reload Value Register". For specific register definitions, please see the ARM official manual "Cortex- M0+ Technical Reference Manual".

SH30F9/SA0 series provides an external reference clock to the SysTick, fixed to HCLK/8, and its standard value is fixed at 60000(stored in the system timer calibration value register STCALIB), which is a 10ms time reference generated at a frequency of 48MHz for HCLK.



10.3 Registers

RCC Module Register list (Base Address:0x4004 0800)

| Address | Register | Description |
|-------------|----------|---|
| 0x4004 0800 | CR | Clock Control Register |
| 0x4004 0804 | CFGR | Clock Configuration Register |
| 0x4004 0808 | CIENR | Clock interrupt enable register |
| 0x4004 080C | CISTR | Clock Interrupt Status Register |
| 0x4004 0810 | CICLR | Clock Interrupt Flag Clear Register |
| 0x4004 0814 | AHBRSTR | AHB Peripheral Reset Register |
| 0x4004 0818 | APB1RSTR | APB1 Peripheral Reset Register |
| 0x4004 081C | APB0RSTR | APB0 Peripheral Reset Register |
| 0x4004 0820 | AHBENR | AHB Peripheral Reset Register |
| 0x4004 0824 | APB1ENR | APB1 Peripheral Reset Register |
| 0x4004 0828 | APB0ENR | APB0 Peripheral Reset Register |
| 0x4004 082C | RSTSTR | RESET Status Register |
| 0x4004 0830 | RSTCLR | RESET Status Clear Register |
| 0x4004 0834 | RCCLOCK | The RCC configuration lock register |
| 0x4004 0838 | HSERFB | Selection of feedback resistance of external high frequency crystal oscillator / ceramic oscillator driving circuit |

10.3.1 Clock Control Register (RCC_CR)

Offset Address: 0x0000

Reset value: 0x0000 2000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|--------|-------|---------|--------|---------|--------|---------|--------|--------|----------|-----|-----|---------|-------|--------|
| Reserved | | | | | | | | | | | | | | ESTCK | LSIRDY |
| | | | | | | | | | | | | | | RW | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LSION | HSIRDY | HSION | LSE RDY | LSE ON | HSE RDY | HSE ON | PLL RDY | PLL ON | CSM ON | SWS[2:0] | | | SW[2:0] | | |
| RW | RO | RW | RO | RW | RO | RW | RO | RW | RW | RO | | | RW | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 18 | Reserved | - |
| 17 | ESTCK | After exiting Shut Down mode, the system clock selects the control bit 0: HSI as the system clock 1: LSI as the system clock |
| 16 | LSIRDY | LSI clock ready flag Set to "1" by hardware after LSI oscillatory stability. 0: LSI oscillatory instability 1: LSI oscillatory stability |
| 15 | LSION | LSI enable Set to "1" or clear by software to open or close LSI When the LSI clock is used as system clock, the bit can't be cleared. 0: LSI is disabled. 1: LSI is enabled. |
| 14 | HSIRDY | HSI clock ready flag Set to "1" by hardware after HSI oscillatory stability. 0: HSI oscillatory instability 1: HSI oscillatory stability |



SH30F9/SA0 Series

| | | |
|-------|-----------------|--|
| 13 | HSION | HSI enable Set to "1" or clear by software to open or close HSI When the HSI clock is used as system clock, the bit can't be cleared. 0: HSI is disabled. 1: HSI is enabled |
| 12 | LSE RDY | External 32kHz oscillator ready flag Set to '1' by hardware to indicate the external 32kHz oscillator is stable. Clear this bit need to 6 external 32kHz clock period after clearing the LSEON bit. 0: External 32kHz clock instability 1: External 32kHz clock stability Note: This bit is rely on LSEON. It only indicate the external 32kHz oscillator ready state. |
| 11 | LSEON | External 32kHz oscillator enable Set or clear by software When LSE is directly or indirectly used as system clock, the bit can't be cleared. 0: LSE oscillator is turned off 1: LSE oscillator is turned on |
| 10 | HSE RDY | External high frequency oscillator ready flag Set to '1' by hardware to indicate the external high frequency oscillator is stable. Clear this bit need to 6 external 4-16MHz clock period after clearing the LSEON bit. 0: External 4-16MHz clock instability 1: External 4-16MHz clock stability Note: This bit is rely on LSEON. It only indicate the external 4-16MHz oscillator ready state. |
| 9 | HSEON | External high frequency oscillator enable Set or clear by software When HSE is directly or indirectly used as system clock, the bit can't be cleared. 0: HSE oscillator is turned off 1: HSE oscillator is turned on |
| 8 | PLL RDY | PLL clock ready flag After the PLL is locked, it is set to '1' by hardware. 0: PLL is not locked 1: PLL is locked |
| 7 | PLLON | PLL enable Set to '1' by software or cleared to enable or disable PLL This bit cannot be cleared when PLL clock is used as the system clock. 0: PLL is disabled 1: PLL is enabled |
| 6 | CSMON | CSM clock detect function enable 0: CSM clock detect function is disabled. 1: CSM clock detect function is enabled. |
| 5 ~ 3 | SWS[2:0] | SWS[2:0] System clock switch status Set to '1' or cleared by hardware to indicate the system clock source. 000: HSI as the system clock 001: PLL output as system clock 010: LSE as the system clock 011: HSE as the system clock 100: LSI as the system clock 101: LSI/4 as the system clock |



SH30F9/SA0 Series

| | | |
|-------|----------------|---|
| 2 ~ 0 | SW[2:0] | System clock switch Set to '1' or cleared by software to select the system clock source. 000: HSI as the system clock 001: PLL output as system clock 010: LSE as the system clock 011: HSE as the system clock 100: LSI as the system clock 101:LSI/4 as the system clock Others: HSI as the system clock |
|-------|----------------|---|

10.3.2 Clock Configuration Register (RCC_CFGR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-------------|--------------|--------------------------|------------------------|-------------------|------------|--------------|---------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | HSESEL[1:0] | Rese rved | CRY 32N OISE EN | CRY 32S PDU P | PLLXTPRE[1 :0] | PLL SRC | Rese rved | PLL4[0] | | | |
| | | | | | RW | - | RW | RW | RW | RW | - | RW | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---------|-----|-----|-----|-----------|-----|----|--------------|------------|----|----|------------|----|----|-----------|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PLL4[0] | | | | PLLK[2:0] | | | Rese rved | PPRE1[2:0] | | | PPRE0[2:0] | | | HPRE[2:0] | |
| RW | | | | RW | | | - | RW | | | RW | | | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 27 | Reserved | - |
| 26 ~ 25 | HSESEL[1:0] | HSE driving ability select bit(Normal is recommended) 00: Normal 01: Reserved 10: Reserved 11: Strong |
| 24 | Reserved | - |
| 23 | CRY32NOISEEN | LSE oscillator perturbation enable signal 0: Disable 1: Enable |
| 22 | CRY32SPDUP | 32.768kHz oscillator speed up mode control bit 0: 32.768kHz oscillator normal mode, cleared by software. 1: 32.768kHz oscillator speed up mode, set by hardware or software. This control bit is set by hardware automatically in all kinds of RESET such as Power on reset, watch dog reset etc. to speed up the 32.768kHz Oscillator oscillating, shorten the 32.768kHz oscillator start-oscillating time. And this bit also can be set or cleared by software if necessary. Such as set before entering Power-down mode and cleared when Power-down mode wakes up. It should be noted that turning off 32.768kHz oscillator speed up (clear this bit) could reduce the system power consumption. only when LSE be select is effective. |
| 21 ~ 20 | PLLXTPRE[1:0] | Pre-divider for PLL entry After being divided by software, it is used as a PLL input clock. This bit can only be written when the PLL is turned off. 00: No pre-division 01: 2 frequency division 10: 3 frequency division 11: 4 frequency division |



SH30F9/SA0 Series

| | | |
|---------|-------------------|---|
| 19 | PLLSRC | PLL entry clock source Set to '1' or cleared by software to select the PLL input clock source. This bit can only be written when the PLL is turned off. 0: HSI clock as PLL input clock 1: HSE clock as PLL input clock |
| 18 | Reserved | - |
| 17 ~ 13 | PLLF[4:0] | PLL multiplication factor F Determine the PLL multiplier parameter by software settings This can be written only when the PLL is turned off. 00000: F=15 00001: F=16 00010: F=17 11101: F=44 11110: F=45 11111: F=46 |
| 12 ~ 10 | PLLK[2:0] | PLL multiplication factor K Determine the PLL multiplier parameter by software settings This can be written only when the PLL is turned off. 000: K=1 001: K=2 010: K=3 011: K=4 100: K=5 101: K=6 110: K=7 111: K=8 |
| 9 | Reserved | - |
| 8 ~ 6 | PPRE1[2:0] | AHB1 Prescaler Set to '1' or cleared by software to control the pre-division coefficient of the low speed APB1 clock (PCLK1). 000: PCLK1 no division 001: PCLK1 2 frequency division 010: PCLK1 4 frequency division 011: PCLK1 8 frequency division 100: PCLK1 16 frequency division Others: PCLK1 16 frequency division Warning: The software must ensure that the APB1 clock frequency does not exceed 24MHz. |
| 5 ~ 3 | PPRE0[2:0] | AHB0 Prescaler Set to '1' or cleared by software to control the pre-division coefficient of the low speed APB0 clock (PCLK0). 000: PCLK0 no division 001: PCLK0 2 frequency division 010: PCLK0 4 frequency division 011: PCLK0 8 frequency division 100: PCLK0 16 frequency division Others: PCLK0 16 frequency division Warning: The software must ensure that the APB0 clock frequency does not exceed 24MHz. |



| | | |
|-------|------------------|---|
| 2 ~ 0 | HPRE[2:0] | AHB Prescaler Set to '1' or cleared by software to control the pre-division coefficient of the AHB clock 000: SYSCLK no division 001: SYSCLK 2 frequency division 010: SYSCLK 4 frequency division 011: SYSCLK 8 frequency division 100: SYSCLK 16 frequency division 101: SYSCLK 32 frequency division Others: SYSCLK 32 frequency division |
|-------|------------------|---|

10.3.3 Clock interrupt enable register (RCC_CIENR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------|----------|----------|----------|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | PLL RDYI | HSE RDYI | LSE RDYI | Reserved | |
| | | | | | | | | | | | E | E | E | | |
| | | | | | | | | | | | RW | RW | RW | - | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 5 | Reserved | - |
| 4 | PLL RDYI | PLL ready interrupt enable Set to '1' or cleared by software to enable or disable the PLL ready interrupt. 0: PLL ready interrupt is disabled; 1: PLL ready interrupt is enabled. |
| 3 | HSE RDYI | HSE ready interrupt enable Set to '1' or cleared by software to enable or disable external 4-16MHz oscillator. 0: HSE ready interrupt is disabled; 1: HSE ready interrupt is enabled. |
| 2 | LSE RDYI | LSE ready interrupt enable Set to '1' or cleared by software to enable or disable external 32KHz oscillator. 0: LSE ready interrupt is disabled; 1: LSE ready interrupt is enabled. |
| 1 ~ 0 | Reserved | - |

10.3.4 Clock Interrupt Status Register (RCC_CISTR)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|---------|---------|---------|----------|----------|----------|----------|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | PLL CSM | HSE CSM | LSE CSM | PLL RDYI | HSE RDYI | LSE RDYI | Reserved | |
| | | | | | | | | F | F | F | F | F | F | | |
| | | | | | | | | RO | RO | RO | RO | RO | RO | - | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|--------|----------|--|
| 31 ~ 8 | Reserved | - |
| 7 | PLLCSMF | PLL clock security monitor interrupt flag It is valid when the PLL loses lock and is set to '1' by hardware. Cleared by software by setting the CSMC bit to '1'. 0: CSM interrupt is not caused by PLL exception; 1: CSM interrupt is caused by a PLL exception; |
| 6 | HSECSMF | HSE clock security monitor interrupt flag It is valid when the HSE loses effect and is set to '1' by hardware. Cleared by software by setting the CSMC bit to '1'. 0: CSM interrupt is not caused by HSE invalidity ; 1: CSM interrupt is caused by a HSE invalidity ; |
| 5 | LSECSMF | LSE clock security monitor interrupt flag It is valid when the LSE loses effect and is set to '1' by hardware. Cleared by software by setting the CSMC bit to '1'. 0: CSM interrupt is not caused by LSE invalidity ; 1: CSM interrupt is caused by a LSE invalidity ; |
| 4 | PLLRDYIF | PLL ready interrupt flag It is set to '1' by hardware, and is cleared by setting PLLRDYC bit to '1'. 0: Clock ready interrupt is not generated by PLL lock; 1: Clock ready interrupt is caused by PLL lock. |
| 3 | HSERDYIF | HSE ready interrupt flag It is set to '1' by hardware, and is cleared by setting HSERDYC bit to '1'. 0: Clock ready interrupt is not generated by external 4-16MHz oscillator; 1: Clock ready interrupt is caused by external 4-16MHz oscillator. |
| 2 | LSERDYIF | LSE ready interrupt flag It is set to '1' by hardware, and is cleared by setting LSERDYC bit to '1'. 0: Clock ready interrupt is not generated by external 32KHz oscillator; 1: Clock ready interrupt is caused by external 32KHz oscillator. |
| 1 ~ 0 | Reserved | - |

10.3.5 Clock Interrupt Flag Clear Register (RCC_CICLR)

Offset Address: 0x0010

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|----------|----------|-----------------|-----------------|-----------------|----------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | CSM C | Reserved | PLL RDY C | HSE RDY C | LSE RDY C | Reserved | | |
| - | | | | | | | | WO | - | WO | WO | WO | - | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 8 | Reserved | - |
| 7 | CSMC | Clock security monitor interrupt clear The CSM interrupt flag bit CSMIF is cleared by setting this bit to '1' by software. 0: No effect; 1: Clear the CSMPLL and CSMHSEF interrupt flag bits. |
| 6 ~ 5 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|----------|--|
| 4 | PLLRDYC | PLL ready interrupt clear The PLL ready interrupt flag bit PLLRDYF is cleared by setting this bit to '1' by software. 0: No effect; 1: Clear the PLL ready interrupt flag bit PLLRDYF. |
| 3 | HSERDYC | HSE ready interrupt clear The HSE ready interrupt flag bit HSERDYF is cleared by setting this bit to '1' by software. 0: No effect; 1: Clear the HSE ready interrupt flag bit HSERDYF. |
| 2 | LSERDYC | LSE ready interrupt clear The LSE ready interrupt flag bit LSERDYF is cleared by setting this bit to '1' by software. 0: No effect; 1: Clear the LSE ready interrupt flag bit LSERDYF. |
| 1 ~ 0 | Reserved | - |

10.3.6 AHB Peripheral Reset Register (RCC_AHBRSTR)

Offset Address: 0x0014

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------|---------|-------------|----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | DMA RST | CRC RST | SYS CFG RST | Reserved |
| | | | | | | | | | | | | RW1s | RW1s | RW1s | - |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 4 | Reserved | - |
| 3 | DMARST | DMA reset Set to '1' by software, cleared by hardware 0: no effect 1: DMA reset |
| 2 | CRCRST | CRC module reset Set to '1' by software, cleared by hardware 0: No effect 1: CRC module reset |
| 1 | SYSCFGRST | SYSCFG reset Set to '1' by software, cleared by hardware 0: No effect 1: SYSCFG reset Note: when using external crystal oscillator (HSE / LSE), the register cannot be set to 1. |
| 0 | Reserved | - |



10.3.7 APB1 Peripheral Reset Register (RCC_APB1RSTR)

Offset Address: 0x0018

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----------|------------|------------|-------------|-------------|-------------|------------------|------------------|------------------|------------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | TKR ST | LCD RST | LED RST | TWI0 RST | SPI1 RST | SPI0 RST | UAR T3R ST | UAR T2R ST | UAR T1R ST | UAR T0R ST |
| - | | | | | | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 10 | Reserved | - |
| 9 | TKRST | Touchkey reset Set to '1' by software, cleared by hardware 0: No effect 1: Touchkey reset |
| 8 | LCDRST | LCD reset Set to '1' by software, cleared by hardware 0: No effect 1: LCD reset |
| 7 | LEDRST | LED Driver reset Set to '1' by software, cleared by hardware 0: No effect 1: LED reset |
| 6 | TWI0RST | TWI0 reset Set to '1' by software, cleared by hardware 0: No effect 1:TWI0 reset |
| 5 | SPI1RST | SPI1 reset Set to '1' by software, cleared by hardware 0: No effect 1:SPI1 reset |
| 4 | SPI0RST | SPI0 reset Set to '1' by software, cleared by hardware 0: No effect 1:SPI0 reset |
| 3 | UART3RST | UART3 reset Set to '1' by software, cleared by hardware 0: No effect 1:UART3 reset |
| 2 | UART2RST | UART2 reset Set to '1' by software, cleared by hardware 0: No effect 1:UART2 reset |
| 1 | UART1RST | UART1 reset Set to '1' by software, cleared by hardware 0: No effect 1:UART1 reset |
| 0 | UART0RST | UART0 reset Set to '1' by software, cleared by hardware 0: No effect 1:UART0 reset |



SH30F9/SA0 Series

10.3.8 APB0 Peripheral Reset Register (RCC_APB0RSTR)

Offset Address: 0x001C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | ADC RST | WWDTRST | EXTIRST | PCA3RST | PCA2RST | PCA1RST | PCA0RST | PWM3RST | PWM2RST | PWM1RST | PWM0RST | TIM3RST | TIM2RST | TIM1RST | TIM0RST |
| - | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s | RW1s |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 15 | Reserved | - |
| 14 | ADCRST | ADC reset Set to '1' by software, cleared by hardware 0: No effect 1:ADC reset |
| 13 | WWDTRST | WWDTRST reset Set to '1' by software, cleared by hardware 0: No effect 1:WWDTRST reset |
| 12 | EXTIRST | EXTI reset Set to '1' by software, cleared by hardware 0: No effect 1:EXTI reset |
| 11 | PCA3RST | PCA3 reset Set to '1' by software, cleared by hardware 0: No effect 1:PCA3 reset |
| 10 | PCA2RST | PCA2 reset Set to '1' by software, cleared by hardware 0: No effect 1:PCA2 reset |
| 9 | PCA1RST | PCA1 reset Set to '1' by software, cleared by hardware 0: No effect 1:PCA1 reset |
| 8 | PCA0RST | PCA0 reset Set to '1' by software, cleared by hardware 0: No effect 1:PCA0 reset |
| 7 | PWM3RST | PWM3 reset Set to '1' by software, cleared by hardware 0: No effect 1:PWM3 reset |
| 6 | PWM2RST | PWM2 reset Set to '1' by software, cleared by hardware 0: No effect 1:PWM2 reset |
| 5 | PWM1RST | PWM1 reset Set to '1' by software, cleared by hardware 0: No effect 1:PWM1 reset |



| | | |
|---|----------------|---|
| 4 | PWM0RST | PWM0 reset Set to '1' by software, cleared by hardware 0: No effect 1: PWM0 reset |
| 3 | TIM3RST | TIM3 reset Set to '1' by software, cleared by hardware 0: No effect 1: TIM3 reset |
| 2 | TIM2RST | TIM2 reset Set to '1' by software, cleared by hardware 0: No effect 1: TIM2 reset |
| 1 | TIM1RST | TIM1 reset Set to '1' by software, cleared by hardware 0: No effect 1: TIM1 reset |
| 0 | TIM0RST | TIM0 reset Set to '1' by software, cleared by hardware 0: No effect 1: TIM0 reset |

10.3.9 AHB Peripheral Reset Register (RCC_AHBENR)

Offset Address: 0x0020

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|---------------|-------------------|-----------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | DMA EN | CRC EN | SYS CFG EN | Reserved |
| - | | | | | | | | | | | | <i>RW</i> | <i>RW</i> | <i>RW</i> | - |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------------|---|
| 31 ~ 4 | Reserved | - |
| 3 | DMAEN | DMA clock enable Set to '1' or cleared by software 0: DMA clock is disabled 1: DMA clock is enabled |
| 2 | CRCEN | CRC clock enable Set to '1' or cleared by software 0: CRC module clock is disabled 1: CRC module clock is enabled |
| 1 | SYSCFGEN | SYSCFG clock enable Set to '1' or cleared by software 0: SYSCFG clock is disabled 1: SYSCFG clock is enabled |
| 0 | Reserved | - |



10.3.10 APB1 Peripheral Reset Register (RCC_APB1ENR)

Offset Address: 0x0024

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----------|-----------|-----------|------------|------------|------------|-----------------|-----------------|-----------------|-----------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | TKE N | LCD EN | LED EN | TWI0 EN | SPI1 EN | SPI0 EN | UAR T3E N | UAR T2E N | UAR T1E N | UAR T0E N |
| - | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 10 | Reserved | - |
| 9 | TKEN | Touch key clock enable Set to '1' or cleared by software 0: Touch key module clock is disabled 1: Touch key module clock is enabled |
| 8 | LCDEN | LCD clock enable Set to '1' or cleared by software 0: LCD module clock is disabled 1: LCD module clock is enabled |
| 7 | LEDEN | LED clock enable Set to '1' or cleared by software 0: LED module clock is disabled 1: LED module clock is enabled |
| 6 | TWI0EN | TWI0 clock enable Set to '1' or cleared by software 0: TWI0 module clock is disabled 1: TWI0 module clock is enabled |
| 5 | SPI1EN | SPI1 clock enable Set to '1' or cleared by software 0: SPI1 module clock is disabled 1: SPI1 module clock is enabled |
| 4 | SPI0EN | SPI0 clock enable Set to '1' or cleared by software 0: SPI0 module clock is disabled 1: SPI0 module clock is enabled |
| 3 | UART3EN | UART3 clock enable Set to '1' or cleared by software 0: UART3 module clock is disabled 1: UART3 module clock is enabled |
| 2 | UART2EN | UART2 clock enable Set to '1' or cleared by software 0: UART2 module clock is disabled 1: UART2 module clock is enabled |
| 1 | UART1EN | UART1 clock enable Set to '1' or cleared by software 0: UART1 module clock is disabled 1: UART1 module clock is enabled |
| 0 | UART0EN | UART0 clock enable Set to '1' or cleared by software 0: UART0 module clock is disabled 1: UART0 module clock is enabled |



10.3.11 APB0 Peripheral Reset Register (RCC_APB0ENR)

Offset Address: 0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|--------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | ADC EN | WWDTE N | EXTI EN | PCA3EN | PCA2EN | PCA1EN | PCA0EN | PWM3EN | PWM2EN | PWM1EN | PWM0EN | TIM3EN | TIM2EN | TIM1EN | TIM0EN |
| - | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 15 | Reserved | - |
| 14 | ADCEN | ADC clock enable Set to '1' or cleared by software 0: ADC module clock is disabled 1: ADC module clock is enabled |
| 13 | WWDTEN | WWDTE clock enable Set to '1' or cleared by software 0: WWDTE module clock is disabled 1: WWDTE module clock is enabled |
| 12 | EXTIEN | EXTI clock enable Set to '1' or cleared by software 0: EXTI module clock is disabled 1: EXTI module clock is enabled |
| 11 | PCA3EN | PCA3 Set to '1' or cleared by software 0: PCA3 module clock is disabled 1: PCA3 module clock is enabled |
| 10 | PCA2EN | PCA2 Set to '1' or cleared by software 0: PCA2 module clock is disabled 1: PCA2 module clock is enabled |
| 9 | PCA1EN | PCA1 Set to '1' or cleared by software 0: PCA1 module clock is disabled 1: PCA1 module clock is enabled |
| 8 | PCA0EN | PCA0 Set to '1' or cleared by software 0: PCA0 module clock is disabled 1: PCA0 module clock is enabled |
| 7 | PWM3EN | PWM3 Set to '1' or cleared by software 0: PWM3 module clock is disabled 1: PWM3 module clock is enabled |
| 6 | PWM2EN | PWM2 Set to '1' or cleared by software 0: PWM2 module clock is disabled 1: PWM2 module clock is enabled |
| 5 | PWM1EN | PWM1 Set to '1' or cleared by software 0: PWM1 module clock is disabled 1: PWM1 module clock is enabled |
| 4 | PWM0EN | PWM0 Set to '1' or cleared by software 0: PWM0 module clock is disabled 1: PWM0 module clock is enabled |
| 3 | TIM3EN | TIM3 Set to '1' or cleared by software 0: TIM3 module clock is disabled 1: TIM3 module clock is enabled |
| 2 | TIM2EN | TIM2 Set to '1' or cleared by software 0: TIM2 module clock is disabled 1: TIM2 module clock is enabled |



SH30F9/SA0 Series

| | | |
|---|---------------|--|
| 1 | TIM1EN | TIM1 Set to '1' or cleared by software 0: TIM1 module clock is disabled 1: TIM1 module clock is enabled |
| 0 | TIM0EN | TIM0 Set to '1' or cleared by software 0: TIM0 module clock is disabled 1: TIM0 module clock is enabled |

10.3.12 RESET Status Register (RCC_RSTSTR)

Offset Address: 0x002C

Reset value: 0x0000 0004

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------------------|---------------------------|--------------------|--------------------|--------------------|---------------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | WW DTR STF | IWD TRS TF | SWR STF | POR STF | LVR STF | PINR STF |
| - | | | | | | | | | | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------------|--|
| 31 ~ 6 | Reserved | - |
| 5 | WWDTRSTF | Window watchdog reset flag Set to '1' by hardware when window watchdog reset occurs. Cleared by software via writing on WWDTRSTFC bit. 0: No window watchdog reset occurs 1: A window watchdog reset occurs |
| 4 | IWDTRSTF | Independent watchdog reset flag Set to '1' by hardware when independent watchdog reset occurs. Cleared by software via writing on IWDTRSTFC bit. 0: No independent watchdog reset occurs 1: An independent watchdog reset occurs |
| 3 | SWRSTF | Software reset flag Set to '1' by hardware when software reset occurs. Cleared by software via writing on SWRSTFC bit. 0: No software reset occurs 1: A software reset occurs |
| 2 | PORSTF | POR reset flag Set to '1' by hardware when POR reset occurs. Cleared by software via writing on PORSTFC bit. 0: No POR reset occurs 1: A POR reset occurs |
| 1 | LVRSTF | LVR reset flag Set to '1' by hardware when LVR reset occurs. Cleared by software via writing on LVRSTFC bit. 0: No LVR reset occurs 1: An LVR reset occurs |
| 0 | PINRSTF | PIN reset flag Set to '1' by hardware when NRST pin reset occurs. Cleared by software via writing on PINRSTFC bit. 0: No NRST pin reset occurs 1: An NRST pin reset occurs |



SH30F9/SA0 Series

10.3.13 RESET Status Clear Register (RCC_RSTCLR)

Offset Address: 0x0030

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------------|-------------------|-----------------|-----------------|-----------------|------------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | WW DTR STF C | IWD TRS TFC | SWR STF C | POR STF C | LVR STF C | PINR STF C |
| - | | | | | | | | | | WO | WO | WO | WO | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 6 | Reserved | - |
| 5 | WWDTRSTFC | WWDTRSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |
| 4 | IWDTRSTFC | IWDTRSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |
| 3 | SWRSTFC | SWRSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |
| 2 | PORSTFC | PORSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |
| 1 | LVRSTFC | LVRSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |
| 0 | PINRSTFC | PINRSTFC reset flag clear Set to '1' by software to clear reset flag. 0: No effect 1: Clears reset flag |



10.3.14 The RCC configuration lock register (RCC_RCCLOCK)

Offset Address: 0x0034

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | LOCK[15:0] | RCC register lock bit 0x33CC: Unlock Other: Lock |

10.3.15 Selection of feedback resistance of external high frequency crystal oscillator / ceramic oscillator driving circuit (HSERFB)

Offset Address: 0x0038

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-------------|---|
| 31 ~ 2 | Reserved | - |
| 1 ~ 0 | RFBSEL[1:0] | HSE clock feedback resistance selection control (500k is recommended) 00: 150k 01: 200k 10: 300k 11: 500k |



11. Watchdog Timer (WDT)

11.1 Independent Watchdog Timer (IWDT)

The independent watchdog timer is a 12-bit decreasing counter with a built-in 128kHz RC oscillator as its clock source. IWDT is turned on by default after power on and cannot be turned off.^{Note}

Note: In DEBUG full-speed simulation mode, it can be turned off by `DBG_IWDT@SYSCFG_DBGCR`. When off, the IWDT counter is no longer to count until the next time the `DBG_IWDT@SYSCFG_DBGCR` is turned on, the counter is cleared and starts counting again. It is important to note that this control register is only valid in DEBUG simulation mode, and IWDT is always on and cannot be turned off in the offline mode.

IWDT can operate in stop mode, so it can be used as a wake-up timer. IWDT can be turned on or off by register during the stop mode (turned off in stop mode by default).

Before the timer overflows, the counter is restarted by writing `0xAAAA` to the dog feeding register `IWDT_CLR` to update the counter to avoid overflow.

The timer will reset the chip after overflow, and the `IWDRSTF@RCC_RSTSTR` flag is set.

After power-on is turned on, IWDT works for a longer overflow time, with approximately $4096 \times 32 / 128 = 1024\text{ms}$.

When the microcontroller enters debug mode, the IWDTG counter either continues to work normally or stops, depending on `DBG_IWDT` configuration bit.

11.2 Window Watchdog Timer (WWDT)

The window watchdog (WWDT) is usually used to monitor software failure by application deviating from normal operation sequence caused by external interference or unforeseen logic conditions.

The window watchdog is an 8-bit decreasing counter. The dog feeding area is a window area. Feeding the dog earlier than the window dog feeding area will generate an advanced abnormal event. Feeding the dog later than the window dog feeding area will generate a delayed abnormal event. In both cases, a reset will be caused. In addition, when the counter reaches the lower boundary of the window feeding dog area, it is the latest feeding time, and the WWDT interrupt can be applied. It can be used as the last remedy time of feeding dog in the window.

The lower boundary of the WWDT window in SH30F9/SA0 series is fixed to 0. A reset is generated when the counter reaches `0xFF`.

WWDT is turned on by the `WWDT_CR`. Once turned on, it can be turned off only when it is reset.

WWDT works on the APB0 bus and is driven by the APB0 clock.

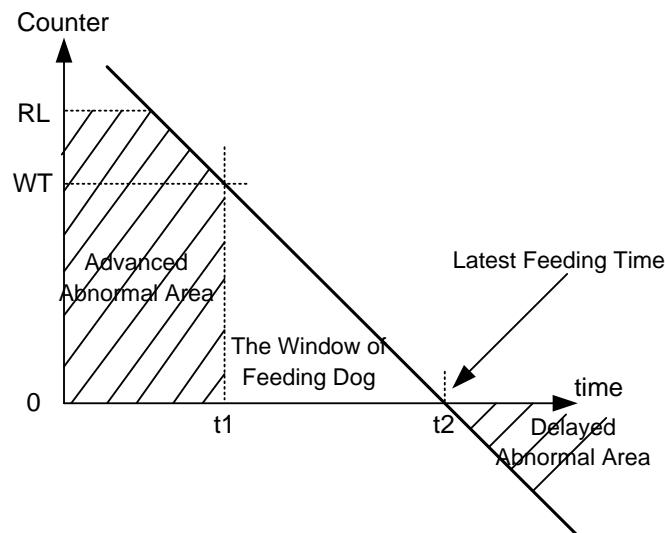


Figure 11-1 Window Watchdog Diagram



SH30F9/SA0 Series

When applying WWDT, first set the WWDT window reload value WWDTRLR (labeled RL in the figure) and the window value WWDTWTR (labeled WT in the figure), then turn on WWDT, and the counter starts running (Counter initializes overloaded values). In the normal feeding dog area, the dog is fed by writing 0x5555 to the feeding dog register WWDTCLR. And that can update the counter to avoid overflow.

If $RL \geq WT$ is set, it is the normal window monitoring function. If $RL < WT$ is set, there is no advanced abnormality monitoring function, leaving only delayed abnormality monitoring function.

Note: The window is the largest when $RL = WT = 0xFF$. And the function is similar to ordinary watchdog

When the microcontroller enters debug mode (Cortex-M0+ core halted), the WWDT counter can continue to operate or stop according to the state of the `DBG_WWDT@DBGCR` configuration bit. See the "Debug Interface" part for details.



11.3 Registers

IWDT Module Register list (Base Address:0x4000 3800)

| Address | Register | Description |
|-------------|----------|---------------------------------------|
| 0x4000 3800 | CR | Independent Watchdog Control Register |
| 0x4000 3804 | CLR | Independent Watchdog Feeding Register |

11.3.1 Independent Watchdog Control Register (IWDT_CR)

Offset Address: 0x0000

Reset value: 0x0000 3FFF

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| LOCK[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|------------|-------------|-----|-----|---------------|-----|----|----|----|----|----|----|----|----|----|----|
| IWD TPD | IWDTPR[2:0] | | | IWDTRLR[11:0] | | | | | | | | | | | |
| RW1 t | RW | | | RW | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|---------|---------------|--|
| 31 ~ 16 | LOCK[15:0] | Register lock bit 0x5AA5: Unlocks this register Other: Locks this register When configuring this register, the unlock bit must be input 0x5AA5 at the same time. After the operation is completed, it will be automatically locked. 0x0 will be fixedly read when reading this LOCK. |
| 15 | IWDTPD | Enable position in sleep/stop mode of IWDT 0: IWDT is disabled to work in sleep/stop mode 1: IWDT is enabled to work in sleep/stop mode |
| 14 ~ 12 | IWDTPR[2:0] | IWDT Pre-division setting 000: Pre-division factor =4 001: Pre-division factor =8 010: Pre-division factor =16 011: Pre-division factor =32 (Default) 100: Pre-division factor =64 101: Pre-division factor =128 110: Pre-division factor =256 111: Pre-division factor =512 Note: The IWDTPR [2:0] power on default value is 011b |
| 11 ~ 0 | IWDTRLR[11:0] | IWDT reload register Used to define the reload value of the watchdog counter. Note: The IWDTRLR[11:0] reset value is 0x0FFF. |



11.3.2 Independent Watchdog Feeding Register (IWDT_CLR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| IWDTCLR[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | IWDTCLR[15:0] | IWDT feeding register 0xAAAA: Feed the dog (update the watchdog counter with IWDTRLR) Other: No action Reading this register will fixedly return to 0x0 after the dog is fed. |

WWDT Module Register list (Base Address:0x4000 3C00)

| Address | Register | Description |
|-------------|----------|-----------------------|
| 0x4000 3C00 | CR | WWDT Control Register |
| 0x4000 3C04 | SR | WWDT Status Register |
| 0x4000 3C08 | CLR | WWDT Feeding Register |
| 0x4000 3C0C | WTR | WWDT Window Register |

11.3.3 WWDT Control Register (WWDT_CR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------|------------|----------|-----|-----|-------------|----|----|--------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| WW DTO N | WW DTIE | Reserved | | | WWDTPR[2:0] | | | WWDTRLR[7:0] | | | | | | | |
| RW1t | RW | - | | | RW | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|---------|---------------------|---|
| 31 ~ 16 | LOCK[15:0] | Register lock bit 0x5AA5: Unlocks this register Other: Lock this register When configuring this register, the unlock bit must be input 0x5AA5 at the same time. After the operation is completed, it will be automatically locked. 0x0000 will be fixedly read when reading this LOCK. |
| 15 | WWDTON | WWDT enable bit 0: Disable WWDT 1: Enable WWDT This bit is set by software but can only be cleared by hardware after reset. |
| 14 | WWDTIE | WWDT latest feeding interrupt enable 0: Disable WWDT interrupt 1: Enable WWDT interrupt This bit is set and cleared by software. |
| 13 ~ 11 | Reserved | - |
| 10 ~ 8 | WWDTPR[2:0] | WWDT pre-division setting 000: Pre-division factor =1X256 001: Pre-division factor =2X256 010: Pre-division factor =4X256 011: Pre-division factor =8X256 100: Pre-division factor =16X256 101: Pre-division factor =32X256 110: Pre-division factor =64X256 111: Pre-division factor =128X256 |
| 7 ~ 0 | WWDTRLR[7:0] | WWDT reload register Used to define the reload value of the watchdog counter. |

11.3.4 WWDT Status Register (WWDT_SR)

Offset Address: 0x0004

Reset value: 0x0000 00FF

| | | | | | | | | | | | | | | | |
|-------------|----------|-----|-----|-----|-----|-----|-----|------------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| WW | Reserved | | | | | | | TCNT[7:0] | | | | | | | |
| DTIF | | | | | | | | | | | | | | | |
| <i>RW</i> | - | | | | | | | <i>RO</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|---------|------------------|--|
| 31 ~ 16 | Reserved | - |
| 15 | WWDTIF | WWDT feeding interrupt flag bit 0: No WWDT interrupt is generated 1: WWDT interrupt is generated This bit is set by software or hardware, and cleared by software. |
| 14 ~ 8 | Reserved | - |
| 7 ~ 0 | TCNT[7:0] | WWDT count register Indicate the current watchdog count value, whose default value is the maximum value of 0xFF at startup. |



11.3.5 WWDT Feeding Register (WWDT_CLR)

Offset Address: 0x0008
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| WWDTCCLR[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | WWDTCCLR[15:0] | WWDT feeding register 0x5555: Feed the dog (update the watchdog counter with WWDTRLR) Other: No action Reading this register will fixedly return to 0x0 after the dog is fed. |

11.3.6 WWDT Window Register (WWDT_WTR)

Offset Address: 0x000C
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|--------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | WWDTWTR[7:0] | | | | | | | |
| - | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | LOCK[15:0] | Register lock bit 0x5AA5: Unlock this register Other: Lock this register When configuring this register, the unlock bit must be input 0x5AA5 at the same time. After the operation is completed, it will be automatically locked. 0x0000 will be fixedly read when reading this LOCK. |
| 15 ~ 8 | Reserved | - |
| 7 ~ 0 | WWDTWTR[7:0] | WWDT window value register Used to define the window value of the watchdog. This register has write protection, it can only be written when LOCK bit is 0x5AA5. No limit for reading. |



12. Code and Data Checking

12.1 CRC Calculation Unit

12.1.1 Introduction

The cyclic redundancy check (CRC) calculation unit gets calculation result of any 32-bit / 16-bit / 8-bit CRC according to the configurable generator polynomial. CRC technology is mainly used to verify the accuracy and integrity of data transmission or data storage. The standard of EN / IEC 60335-1 needs to provide a method for verifying the integrity of the flash memory. The CRC calculation unit can be used to calculate the software identification when the program is running and compare it with the reference identification generated during connection to check the integrity of the flash storage. In the communication process, the CRC value is added to the end of the data for transmission. The receiving unit sends the received data to the CRC unit one by one, and finally judges whether the final CRC value is 0 to verify the integrity of the transmitted data.

12.1.2 Main Features

- Support various CRC polynomial
 - (1) CRC-32: (0x04C11DB7)
Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
 - (2) CRC-16: (0x8005)
Polynomial: $x^{16} + x^{15} + x^2 + 1$
 - (3) CRC-CCITT: (0x1021)
Polynomial: $x^{16} + x^{12} + x^5 + 1$
 - (4) CRC-8: (0x7):
Polynomial: $x^8 + x^2 + x + 1$
- Provide CRC_INIT register to define first data in data register. When calculating CRC-16, the lower 16 bits of CRC_INIT are valid. When calculating CRC-8, the lower 8 bits of CRC_INIT are valid.
- Configurable input/output data conversion, includes byte reversal, bit reversal in byte, and 1's complement (bitwise NOT).
- Support 8/16/32-bit data input. If input data is 8-bit, then 8-bit data will be calculated. If input data is 16-bit, then 16-bit data will be calculated. If input data is 32-bit, then 32-bit data will be calculated. The bits of input data is independent of the CRC format. The atom unit for calculating data is 8-bit, so if multi-byte data inputted, low byte in order will be calculated first.
- CRC-8 computation done in 1 clock cycle, CRC-16 computation done in 2 clock cycles, CRC-32 computation done in 4 clock cycles.

12.1.3 CRC function description

As shown in Figure 12-1. When calculating CRC, firstly, you can sets format of the CRC_CR according to the actual situation. The parameters that can be set are (MODE, RBITR, RBITW, RBYTER, RBYTEW, COMPLR, COMPLW) @CRC_CR. Then set the generator polynomial and CRC initial value required by the format. After setting, you need to write 1 to RELOAD @CRC_CR to load the initial value into the CRC_DR unit. After that, you can write data to CRC_DR and read the calculation result from CRC_DR. That each write to CRC_DR will trigger a CRC operation, the user can get the current calculation result from reading CRC_DR register.

When writing, 8-bit, 16-bit and 32-bit data are allowed, but word alignment is required when writing/reading (4-byte alignment, which means it must be the address of CRC_DR), otherwise the result is unpredictable. The data in one write is not related to the bit number of digits of the specific CRC result, but it is related to the CRC result. When writing 8-bit data, only 8-bit data is calculated, when writing 16-bit data, only 16-bit data is calculated, and when writing 32-bit data, 32-bit data is calculated at a time. The basic calculation unit is one byte. When the written data is larger than one byte, the low byte first participates in the calculation (see the algorithm description for details). Therefore, when the input conversion is set to all 0s (which means no conversion is performed before written to data register), the CRC results obtained by writing 0x12345678 once and continuously writing 0x78, 0x56, 0x34, 0x12 are the same.



SH30F9/SA0 Series

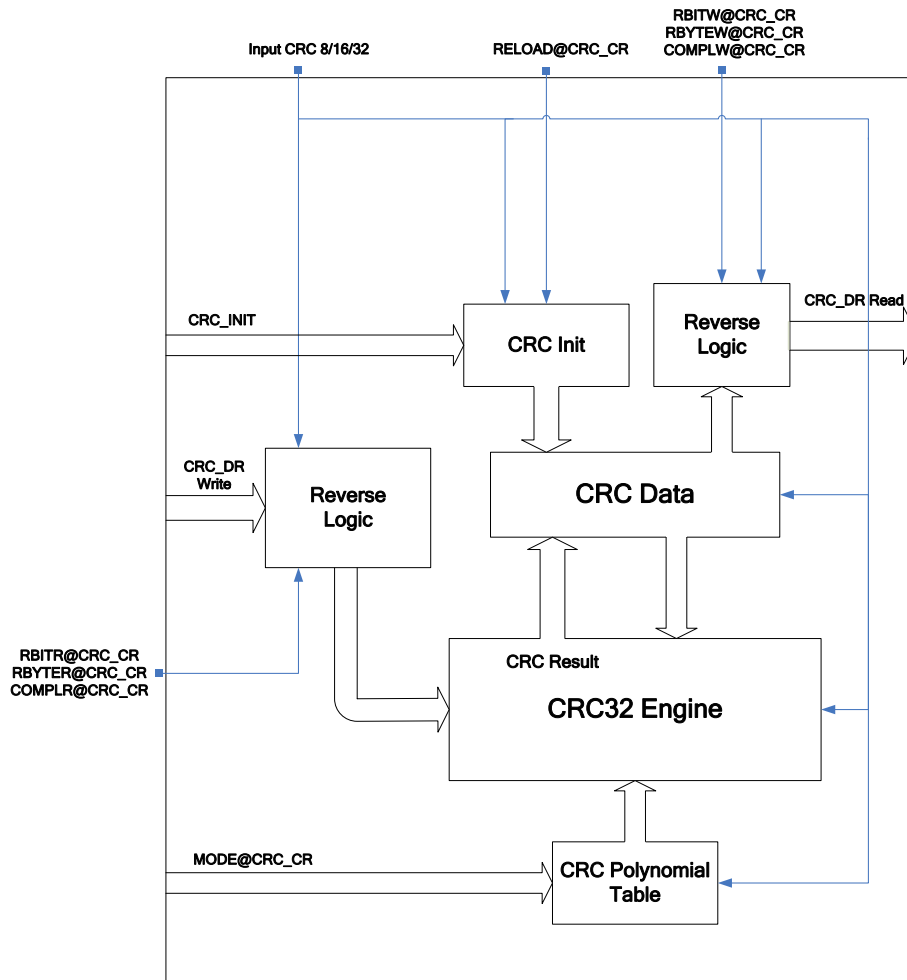


Figure 12-1 CRC logic module diagram



12.1.3.1 CRC Format Conversion Description

Since the data register operation can be 8-bit, 16-bit or 32-bit. Therefore, different processing will be carried out according to the difference of input data during calculating. Reading is only related to CRC Mode.

Table 12-1 Data Conversion Description

| | 8-bit Writing | | | Reading | | |
|------------------|--|---|--------------------------------|--|---|--------------------------------|
| | COMPLW | RBITW | RBYTEW | COMPLR | RBITR | RBYTER |
| CRC_8 MODE=2 | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore |
| CRC_16 MODE=1 | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 |
| CRC_32 MODE=0 | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 |
| | 16-bit Writing | | | Reading | | |
| | COMPLW | RBITW | RBYTEW | COMPLR | RBITR | RBYTER |
| CRC_8 MODE=2 | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore |
| CRC_16 MODE=1 | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 |
| CRC_32 MODE=0 | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 |
| | 32-bit Writing | | | Reading | | |
| | COMPLW | RBITW | RBYTEW | COMPLR | RBITR | RBYTER |
| CRC_8 MODE=2 | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 | Byte0 = ~Byte0 | Bit0<->Bit7 | Ignore |
| CRC_16 MODE=1 | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 | Byte0 = ~Byte0 Byte1 = ~Byte1 | Bit0<->Bit7 Bit8<->Bit15 | Byte1<->Byte0 |
| CRC_32 MODE=0 | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 | Byte0 = ~Byte0 Byte1 = ~Byte1 Byte2 = ~Byte2 Byte3 = ~Byte3 | Bit0<->Bit7 Bit8<->Bit15 Bit16<->Bit23 Bit24<->Bit31 | Byte0<->Byte3 Byte1<->Byte2 |

12.1.3.2 Configuration of Common Formats

(1) Configuration of CRC32 (0x04C11DB7)

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

```

CRC_INIT      : 0xFFFFFFFF
CRC_CR.MODE   : 0
CRC_CR.RBITW  : 0
CRC_CR.RBYTEW : 0
CRC_CR.COMPLW : 0
CRC_CR.RBITR  : 0
CRC_CR.RBYTER : 0
CRC_CR.COMPLR : 0

```



Input:

8-bit: Calculate 1 byte at a time

16-bit: Calculate 2 bytes at a time, low byte first

32-bit: Calculate 4 bytes at a time, low byte first

CRC result: all 32-bit of CRC_DR

(2) Configuration of CRC16 (0x8005)

$$x^{16} + x^{15} + x^2 + 1$$

CRC_INIT : 0xFFFF

CRC_CR.MODE : 1

CRC_CR.RBITW : 0

CRC_CR.RBYTEW : 0

CRC_CR.COMPLW : 0

CRC_CR.RBITR : 0

CRC_CR.RBYTER : 0

CRC_CR.COMPLR : 0

Input:

8-bit: Calculate 1 byte at a time

16-bit: Calculate 2 bytes at a time, low byte first

32-bit: Calculate 4 bytes at a time, low byte first

CRC result: low 16-bit of CRC_DR

(3) Configuration of CRC16-CITT (0x1021)

$$x^{16} + x^{12} + x^5 + 1$$

CRC_INIT : 0xFFFF

CRC_CR.MODE : 2

CRC_CR.RBITW : 0

CRC_CR.RBYTEW : 0

CRC_CR.COMPLW : 0

CRC_CR.RBITR : 0

CRC_CR.RBYTER : 0

CRC_CR.COMPLR : 0

Input:

8-bit: Calculate 1 byte at a time

16-bit: Calculate 2 bytes at a time, low byte first

32-bit: Calculate 4 bytes at a time, low byte first

CRC result: low 16-bit of CRC_DR

(4) Configuration of CRC8 (0x7)

$$x^8 + x^2 + x + 1$$

CRC_INIT : 0xFF

CRC_CR.MODE : 3

CRC_CR.RBITW : 0

CRC_CR.RBYTEW : 0

CRC_CR.COMPLW : 0

CRC_CR.RBITR : 0

CRC_CR.RBYTER : 0

CRC_CR.COMPLR : 0



SH30F9/SA0 Series

Input:

8-bit: Calculate 1 byte at a time

16-bit: Calculate 2 bytes at a time, low byte first

32-bit: Calculate 4 bytes at a time, low byte first

CRC result: low 8-bit of CRC_DR



12.1.4 CRC Register

CRC Module Register list (Base Address:0x4004 1000)

| Address | Register | Description |
|-------------|----------|----------------------------|
| 0x4004 1000 | DR | CRC data register |
| 0x4004 1004 | CR | CRC Control Register |
| 0x4004 1008 | INIT | CRC Initial Value Register |

12.1.4.1 CRC data register (CRC_DR)

Offset Address: 0x0000

Reset value: 0xFFFF FFFF

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| DR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 0 | DR[31:0] | 1. Used as an input register when writing new data to the CRC calculator; 2. Hold the previous CRC calculation result when it is read; 3. 8-bit, 16-bit, and 32-bit written operations are allowed. |

12.1.4.2 CRC Control Register (CRC_CR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----|----------|----|----|----|----|----|----|-----------|
| RBYTER | RBITR | COMPLR | RBYTEW | RBITRW | COMPLW | MODE[1:0] | | Reserved | | | | | | | RELOAD |
| <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | | - | | | | | | | <i>WO</i> |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 | RBYTER | Reverse the byte order of the data in DR and then read. 0: Remain unchanged 1: Reverse byte order (refer to Data Conversion Description table for detailed conversion) |
| 14 | RBITR | Reverse the byte inner bit order of the data in DR and then read. 0: Remain unchanged 1: Reverse byte inner bit order (refer to Data Conversion Description table for detailed conversion) |



| | | |
|-------|-----------|---|
| 13 | COMPLR | Do a 1's complement transformation to the data in DR and then read. (i.e. do bitwise inversion to the read data) 0: Remain unchanged 1: Do a 1's complement transformation |
| 12 | RBYTEW | Reverse the byte order of the written data 0: Remain unchanged 1: Reverse byte order (refer to Data Conversion Description table for detailed conversion) |
| 11 | RBITW | Reverse the byte inner bit order of the written data. 0: Remain unchanged 1: Reverse byte inner bit order (refer to Data Conversion Description table for detailed conversion) |
| 10 | COMPLW | Do a 1's complement transformation to the written data. 0: Remain unchanged 1: Do a 1's complement transformation |
| 9 ~ 8 | MODE[1:0] | CRC mode selection. 00: CRC-32 (0x04C11DB7) 01: CRC-16 (generator polynomial is 0x8005--modbus) 10: CRC-CITT (generator polynomial is 0x1021) 11: CRC-8 (generator polynomial is 0x7) |
| 7 ~ 1 | Reserved | - |
| 0 | RELOAD | Reload the initial value into the data register 1: Write 1 to reload. Reload INIT@CRC_INIT into the DR@CRC_DR register and other register values remain unchanged. 0: Write 0 to remain unchanged, this Bit cannot be read, and it is always 0. |

12.1.4.3 CRC Initial Value Register (CRC_INIT)

Offset Address: 0x0008

Reset value: 0xFFFF FFFF

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| INIT[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| INIT[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|--------|------------|--|
| 31 ~ 0 | INIT[31:0] | Store initial value of CRC. After writing 1 to RELOAD@CRC_CR, this value is automatically loaded into the DR@CRC_DR register. MODE=0: 32-bit valid MODE=1: Lower 16-bit valid MODE=2: Lower 16-bit valid MODE=3: Lower 8-bit valid |



12.2 SRAM Detection Module (RAMBIST)

12.2.1 SRAM Detection Module Introduction

To meet the requirements of the IEC60730 standard for RAM, SH30F9/SA0 series provides a hardware-implemented RAM detection module. This module supports March-C and March-X algorithms, you can choose one of them when you use it. The detection module is allowed to be interrupted during the program running. Compared with the traditional RAM detection software implementation, it greatly guarantees the efficiency of application execution.

12.2.2 SRAM detection module feature description

1. This module supports March-C algorithm or March-X algorithm to detect SRAM. March-C can detect SAF, TF, CFin, CFid, BF, SCF and other errors, and March-X can detect SAF, TF, CFin, AF errors. The detection speed of March-X algorithm is faster than that of March-C. When using it, you can choose the appropriate detection algorithm according to the specific application requirements.
2. This module needs to use the high address area of the RAM being tested as the data backup area of the RAM. During the detection process, this area is used to save the data that needs to be tested in RAM (SRAM), and then restore RAM after the detection is completed. Therefore, the user program is not allowed to access this backup area during the detection process (guaranteed by the software program, the hardware logic does not do special checks), otherwise unpredictable errors may occur. The size of the data backup area is determined by the BLKSZ @ RAMBIST_CFG register. For example, if BLKSZ is 1, the detection size is 32Bytes, so the size of the backup area is also 32Bytes. Then the address is 0x20003FE0~0x20003FFF.
3. The detection range includes SRAM areas. The starting address and area size of the inspection can be set. The start address must be aligned according to the size of the detection area. After the size of the detection area is determined, this module will occupy a space of the same size at the high-order address in the area to be tested as the data backup area. Therefore, it is necessary to avoid overlapping with the backup area when setting the detection area. This conflict is avoided by the software, and the hardware logic does not make any judgment. If a conflict occurs, it may cause unpredictable errors.
4. Self-checking function in data backup area. Data backup is not required with this function, the size of the detection is determined by the BLKSZ @ RAMBIST_CFG register.
5. Allows the user program to access areas other than the data backup area when this module is running. (Note: The access prohibition of the backup area needs to be avoided by software, and the hardware logic does not deal with it. If a conflict occurs, it may cause unpredictable errors in the program). When the user program accesses the RAM, this module will suspend the detection action and release access of the RAM to the user program. After the access of user program is completed, it will continue the previous detection process. If the user program accesses the detected area, this module will map the access address to the backup area, and the user program access the data required by the program from the backup area, without affecting the normal operation of the program. The access of the user program has priority over the detection of the module, and the application function of the application program is guaranteed as much as possible.
6. This module backs up and restores the contents of the detected area before and after the test (except for the self-test of the backup area). During the backup and recovery process, the user program is allowed to access areas other than the backup area. The access of the user program will suspend the ongoing backup or restore process. After completing the access, the user program will continue the previous process. If accessing the detected area, this module will judge whether the access address has been backed up or restored according to the current backup or restore position, and thus decide whether to access from the actual RAM address or from the data backup area. The access of user programs takes priority over the backup and restoration of modules, as far as possible to ensure the application efficiency of application programs.
7. During the backup and recovery process, the number of operations needs to be verified. If a verification error occurs during the backup process, you need to set the ERR @ RAMBIST_CSR bit and stop the operation of this module. If a check error occurs during the recovery process, firstly, you need to set the ERR @ RAMBIST_CSR bit to 1 and continue the subsequent recovery actions until the recovery is complete.
8. The ERR @ RAMBIST_CSR bit is set to 1 immediately after discovering an error during the detection process, and then the ongoing detection process is interrupted and the recovery process is started (if it perform a self-check in the backup area, without starting the recovery process). BSY @ RAMBIST_CSR is not cleared until the consummation of restoration.

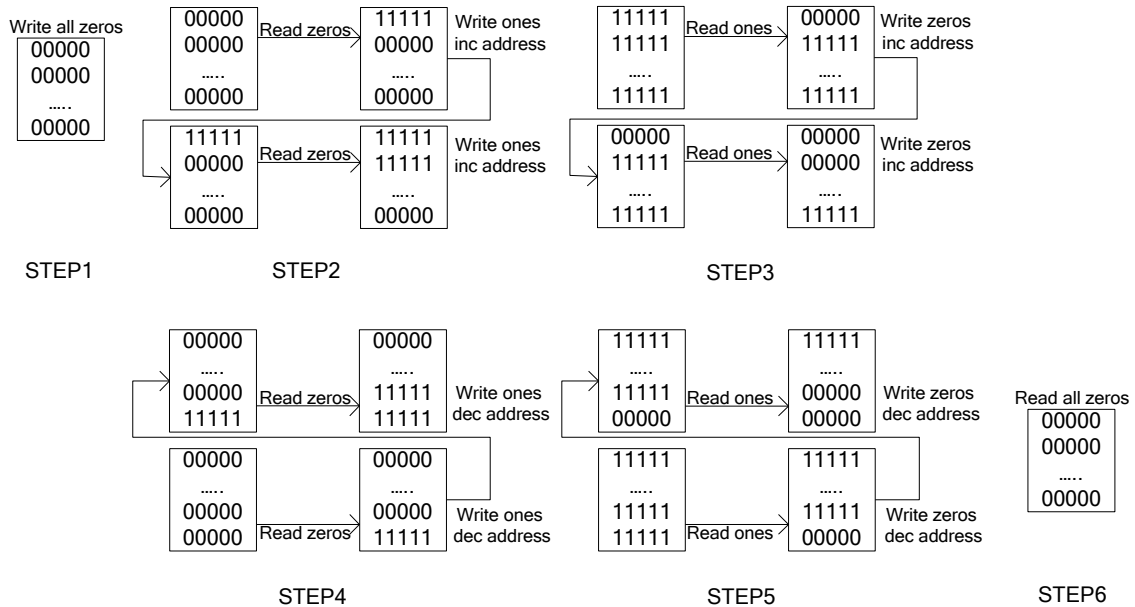


9. This module is started by RUN @ RAMBIST_CSR, and BSY @ RAMBIST_CSR is automatically set to 1 after startup. (The ERR @ RAMBIST_CSR bit needs to be cleared 0 by the software before the module is started. Clearing 0 is allowed to start at the same time as starting). During the detection process (including backup and recovery), if the error is found, the ERR @ RAMBIST_CSR bit is set to 1 immediately. After the entire module is completed or stopped, it will clear the BSY @ RAMBIST_CSR bit to 0.
10. During the operation of this module, the program is not allowed to modify the registers of this module, but the program is allowed to read these registers. This limitation is implemented by hardware logic.

12.2.2.1 March-C Algorithm Description

- Step 1: write 0 to all
- Step 2: check 0 by reading from begin to end, then write 1
- Step 3: check 1 by reading from begin to end, then write 0
- Step 4: check 0 by reading from end to begin, then write 1
- Step 5: check 1 by reading from end to begin, then write 0
- Step 6: check all 0 after reading.

12.2.2.2 March-C Algorithm logic diagram

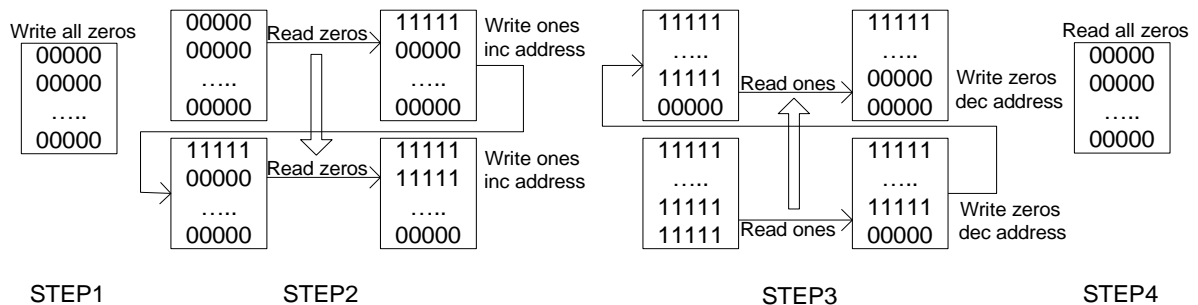


12.2.2.3 March-X Algorithm Description

- March-X is a subset of March-C algorithm.
- Step 1: write 0 to all
- Step 2: check 0 by reading from begin to end, then write 1
- Step 3: check 1 by reading from end to begin, then write 0
- Step 4: check all 0 after reading.



12.2.2.4 March-X Algorithm logic diagram



12.2.2.5 SRAM Detection Operating Steps

Software operating steps:

1. Backup area self-test
 - a) Set BLKSZ@RAMBIST_CFG to set the backup area size
 - b) Set the algorithm used by SEL@ RAMBIST_CFG selection (March-C or March-X)
 - c) Clear ERR@ RAMBIST_CSR. to clear the error flag.
 - d) Set MOD@RAMBIST_CSR to set the self-test mode.
 - e) Write 0x59A6 to RUN@ RAMBIST_CSR to start the module operation.
 - f) Check BYS@ RAMBIST_CSR to determine if the detection ends
 - g) Check ERR@ RAMBIST_CSR to determine if an error has occurred.

2. Normal RAM area detection
 - a) Set BLKSZ@RAMBIST_CFG to set the detection area size
 - b) Set the algorithm used by SEL@ RAMBIST_CFG selection (March-C or March-X)
 - c) Clear MOD@ RAMBIST_CSR to use the normal detection mode
 - d) Set ADDR@ RAMBIST_ADDR to set the detection block location
 - e) Clear ERR@ RAMBIST_CSR. to clear error flag
 - f) Write 0x59A6 to RUN@ RAMBIST_CSR to start the module operation.
 - g) Check ERR@RAMBIST_CSR to determine if an error has occurred
 - h) Check BYS@RAMBIST_CSR to determine if the detection ends
 - i) Repeat e to i until all blocks of RAM (except the backup area) are tested



Corresponding hardware operation process:

1. Write 0x59A6 to RUN @ RAMBIST_CSR to start the module.
2. The hardware sets the BSY @ RAMBIST_CSR bit to 1.
3. Back up the area to be checked. Copy the data in the checked area to the backup area (this step is not necessary if it is in self-check mode). Copy by word, after each copying a word, a reading verification will be performed (re-read the data from the target address and then compare with the original data), and after the verification is completed, copy the next one until the backup is completed. If the verification is error, set the error flag (ERR @ RAMBIST_CSR) to 1 and exit the entire process (no further steps), and then clear the BSY @ RAMBIST_CSR bit to 0. It can be interrupted by RAM access from the program during the copy. After completing the program access, the copying operation of the word before restarting (assuming that a word is interrupted when the copy has not been verified yet, then the word will be re-read, written and verified after the copy is restored, only after passing the verification, the word backup is considered successful). If the RAM accessed by the program happens to be the word being copied, the data that was accessed before the verification was completed is still accessed from the original area. If the verification is completed and correct, then access from the backup area. Accesses that are not within the detection range are accessed from the actual address.
4. After the backup is successful, perform March-C or March-X detection on the detected area. For specific detection actions, refer to the March-C and March-X algorithm descriptions above. If an error is found during the detection process, the error flag (ERR @ RAMBIST_CSR) is set first, and then the detection process is interrupted before entering the recovery process. During the detection process, it can be interrupted by RAM access from the program, and the previous detection action can be continued after the program access is completed. If the area accessed by the program is within the current detection range, it is accessed from the corresponding address in the backup area, and if it is not within the current detection range, it is accessed from the actual address of the RAM.
5. After the detection is completed (regardless of whether there is an error), the backup data is restored. When recovering, copy the data from the backup area to the detected area by word. If ERR @ RAMBIST_CSR is 1 (with error) during recovery, verification of the recovered data will not be performed. If ERR @ RAMBIST_CSR is 0 (no error) during recovery, it will be verified after each word is recovered (re-read the data from the target address and compare with the original data), if it is inconsistent, set ERR @ RAMBIST_CSR. Verification After the end, regardless of whether the previous verification is correct, the next word will be recovered (no recovery data will be verified after an error) until all are completed. It can be interrupted by RAM access from the program during recovery. After the program access is completed, the recovery action of the word before restarting (assuming that a word is interrupted when it is copied and not verified, then the word will be re-read, written and verified after recovery, only the correction Only after passing the test will the word be restored successfully). If the RAM accessed by the program happens to be the word being copied, then the data that was accessed before the verification was completed is also accessed from the backup area. If the verification is completed and correct, then access from the actual area. Accesses that are not within the detection range are accessed from the actual address.
6. After the recovery is completed, clear the BSY @ RAMBIST_CSR bit to end the test.
7. When the BSY @ RAMBIST_CSR bit is 1, software is not allowed to modify the register values in this module. This limitation is implemented by hardware.



12.2.3 RAMBIST Registers

RAMBIST Module Register list (Base Address:0x4004 1400)

| Address | Register | Description |
|-------------|----------|-----------------------------|
| 0x4004 1400 | ADDR | SRAM Address Register |
| 0x4004 1404 | CFG | SRAM Configuration Register |
| 0x4004 1408 | CSR | SRAM Control Register |

12.2.3.1 SRAM Address Register (RAMBIST_ADDR)

Offset Address: 0x0000

Reset value: 0x2000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| RAMADR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RAMADR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|--------------|--|
| 31 ~ 0 | RAMADR[31:0] | <p>Set the address to check the RAM area [A31:A0]. Error flag is set when data address is not in RAM range The address of the area to be detected must be aligned according to the size of BLKSZ, otherwise low address will be discarded according to the size of BLKSZ. Example: The BLKSZ of the detection area is 1 (32-byte alignment), so the legal RAMADDR is "0010 0000 0000 0000 00xx xxxx xxx0 0000". If "0010 0000 0000 0000 00xx xxxx xxx0 0101" is written, it is also start from "0010 0000 0000 0000 00xx xxxx xxx0 0000" during testing.</p> |

12.2.3.2 SRAM Configuration Register (RAMBIST_CFG)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SEL | Reserved | | | | | | | | | | | | BLKSZ[2:0] | | |
| <i>RW</i> | | | | | | | | | | | | | <i>RW</i> | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 | SEL | <p>Select detection algorithm 0: March-C 1: March-X</p> |
| 14 ~ 3 | Reserved | - |



| | | |
|-------|------------|--|
| 2 ~ 0 | BLKSZ[2:0] | Test area size 000: 16Bytes 001: 32Bytes 010: 64Bytes 011: 128Bytes 100: 256Bytes 101: 512Bytes 110: 1024Bytes 111: 2048Bytes |
|-------|------------|--|

12.2.3.3 SRAM Control Register (RAMBIST_CSR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | ERR | BSY | MOD |
| | | | | | | | | | | | | | RW | RO | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RUN[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 19 | Reserved | - |
| 18 | ERR | Module error flag, cleared by software and go on at the same time as the RUN bit. 0: No error is detected 1: There is detected error (including: data backup verification error, RAM detection failure, data recovery verification error, RAMADR invalidity) |
| 17 | BSY | Module running flag bit, automatically set and cleared by the module. 0: Indicate that no detection or detection has been completed 1: Indicate that detection is ongoing |
| 16 | MOD | Select detection mode 0: Detect the RAM area specified by the registers ADDR@RAMBIST and BLKSZ[2:0]@RAMBIST_CFG 1: Detect data backup area. The data backup area size is determined by the BLKSZ[2:0] @RAMBIST_CFG register, and the data backup area is located at the upper address of the SRAM. The ADDR@RAMBIST setting is ignored in this mode. |
| 15 ~ 0 | RUN[15:0] | Start detection, BSY Flag is automatically set to 1 after startup 0x59A6: Start detection, is always 0 if being read Other: Does not start detection |



13. General-Purpose I/O (GPIO)

13.1 Introduction

SH30F9/SA0 series provides up to 61 general-purpose I/O ports, divided into 4 groups of ports (A to D), each group contains up to 16 input/output pins, and has separate configuration registers and data registers.

Each I/O port has alternate functions, some alternate with the digital peripherals on chip, some alternate with analog peripherals on chip, and some alternate with external interrupt and debug interfaces.

All I/O support weak pull-up and push-pull output. The I/O port output drive capability is adjustable in multiple stages, and several I/O ports provide superior current sinking capability to meet the requirements of directly driving LED or driving low power device in certain applications.

13.2 Main Features

- All port groups can be mounted on the single cycle IO bus (SCIOB), supporting input and output of the single cycle IO
- Input/output direction control
- Input pin can be configured with weak pull-up
- Output pin can be configured as push-pull or open-drain, and the drive capability can be adjusted in multiple stages.
- All ports support per-bit set/reset operation
- All Ports support per-byte read/write operation
- All ports can be used as external interrupt input, with trigger edge settable (configure in EXTI part)
- I/O port can be configured as normal input/output function or digital peripheral function or analog peripheral function
- I/O port alternate function uses multiplexers, which can only be configured as one function at a time
- It is allowed to read pin input status, when configured as digital peripheral function
- I/O port supports alternate function remapping to other pin
- Support port configuration lock function

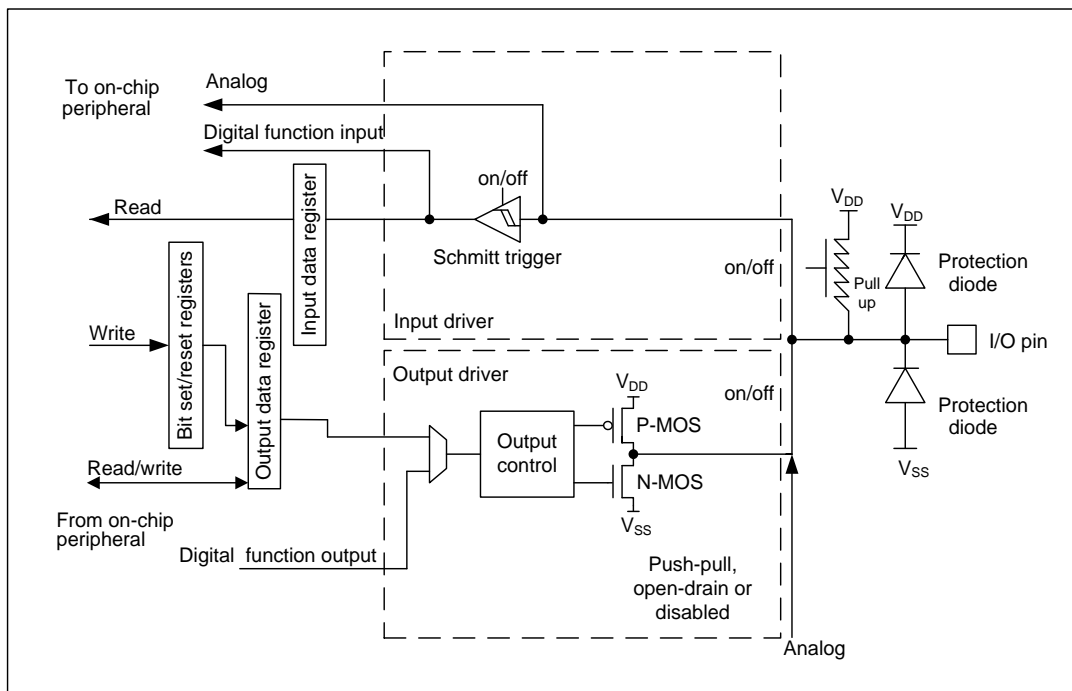


Figure 13-1 I/O Port Basic Structure Block Diagram



13.3 Function Description

Each I/O port configuration is completed by five basic configuration registers (MODER, OTYPE, ODRVR, PUPDR, TTLEN). Each register supports 32-bit word, 16 bit half word and 8-bit byte read-write mode, which is used to configure I/O input / output mode (direction), output push-pull/open-drain mode, output driver capability and pull-up attribute.

The basic working mode of the I/O port is shown in the following table:

Table 13-1 I/O Port Basic Working Mode List

| Port Function | AFRL/AFRH | MODER | OTYPE | ODRVR | PUPDR | I/O Attribute | | | |
|---------------------|-----------|--------------------|-------|-------|-------|------------------------------|--------------------------|---------------------------|----------------------|
| All off | AF0 | X | x | x | x | I/O all off ^{Note1} | | | |
| GPIO | AF1 | 0 | x | x | 0 | GPIO input | Floating | | |
| | | | | | 1 | | Pull-up | | |
| | | 1 | | | 0 | 0 | GPIO push-pull output | Push-pull | |
| | | | | | | 1 | | Push-pull + Pull-up | |
| | | | | | 1 | 0 | 0 | GPIO open-drain output | Open-drain |
| | | | | | | | 1 | | Open-drain + pull-up |
| Digital Peripherals | AF2~AF10 | x ^{Note2} | x | x | 0 | Digital input | Floating | | |
| | | | | | 1 | | Pull-up | | |
| | | | 0 | | 0 | 0 | Digital push-pull output | Push-pull | |
| | | | | | | 1 | | Push-pull + Pull-up | |
| | | | | | 1 | 0 | 0 | Digital open-drain output | Open-drain |
| | | | | | | | 1 | | Open-drain + pull-up |
| Analog Peripherals | AF2~AF10 | x | x | x | 0 | Analog input/output | Floating | | |
| | | | | | 1 | | Pull-up ^{Note3} | | |

x in the table means no care.

Note 1: The all off state is not floating (high resistance), it is true that the input and output channels are cut off.

Note 2: The input and output configuration of the digital peripheral is bound to the peripheral. After enabling a digital peripheral alternate function, the user only needs to configure the OTYPE, ODRVR and PUPDR registers without configuring the MODER register.

Note 3: The pull-up is not enabled by default. In special applications, the pull-up can be enabled to provide a specific input level.

In addition to the 4 basic port functions in the above table, the port also has a set of system port, which includes the debug interface SWD and the external oscillator interface XTAL, a total of 4 ports. System port can also be alternated as GPIO or digital peripheral interfaces. However, the above basic configuration registers are invalid when they are used as system port. These two interfaces are further described in SYSCFG.

13.3.1 Default Configuration of I/O Pins

During reset and after reset, the I/O port function is configured as AF0 "GPIO input and output off" function, except for the SWD interface.

After system reset, SWD interface is selected as the default function. In order to ensure proper entry into the debug mode, it is necessary to keep the SWD port as non-floating status. To achieve this, the corresponding SWD pins are set as follows after reset:

- SWDIO: Input pull-up
- SWCLK: Input pull-down

Once the SWD interface is released by the user code (see control bit SWJCFG@SYSCFG_SAFR), the software can use these I/O ports as general purpose I/O ports. Then the status of these I/O ports will be determined by the corresponding I/O configuration registers. By default, all I/O ports are configured as "GPIO input and output off".



Unlike the SWD interface, the external oscillator interface XTAL1/XTAL2 is "input and output off" function by default. The XTAL interface attribute needs to be turned on by the user code (see control bit OSCCFG@SYSCFG_SAFR). For more detailed definitions, see the "SYSCFG" chapter.

13.3.2 I/O Control Register

Each I/O port has five 32-bit registers for configuring the working modes: MODER, OTyPER, ODRVR, PUPDR, TTLLEN.

MODER: set input and output direction;

OTyPER: set the output type, which are push-pull output and open-drain output (when output 0, only N-MOS is turned on);

ODRVR: set the output drive capability, divided into 2 gears: normal, strong drive (only several pins have this capability);

PUPDR: set internal weak pull-up. This register can be set during input and output.

TTLLEN: set the input level logic to CMOS or TTL logic.

13.3.3 I/O Data Register

Each GPIO port has two 16-bit data registers: IDR, ODR.

IDR: input data register, read access to the input data register can get I/O status

ODR: output data register, the value written to the output data register (ODR) is outputted to the corresponding I/O pin. The output driver can be used in push-pull mode or open drain mode.

13.3.4 I/O Bit Set And Bit Reset

Bit set and reset register (BSRR) is a 32-bit register that allows each bit of the Output Register (ODR) to be independently set and cleared.

The lower 16 bits of the BSRR are the set operations, and the higher 16 bits are the clear operations, allowing simultaneous set and clear operations on different I/O ports in a single slot of AHB write operation. If set and clear the same I/O port at the same time, the set operation has higher priority.

The reading and writing of BSRR can be word, half-word or byte. The following example shows how to use word access to simultaneously switch two I/O ports in the same level or opposite level:

```
;Switch PA.0 and PA.1 simultaneously in the same level
```

```
GPIOA_BSRR = 0x00000003
```

```
GPIOA_BSRR = 0x00030000
```

```
;Switch PA.0 and PA.1 simultaneously in the opposite level
```

```
GPIOA_BSRR = 0x00020001
```

```
GPIOA_BSRR = 0x00010002
```

If half-word is used to achieve switching simultaneously in the same level, the method is as follows:

```
#define GPIOA_BitSet_ADDR = 0xXX ; point to the base address of GPIO_BSRR_A (used to set)
```

```
#define GPIOA_BitReset_ADDR = 0xXX + 2; point to the high half-word offset address of GPIO_BSRR_A (used to reset)
```

```
*(u16*)GPIOA_BitSet_ADDR = 0x03
```

```
*(u16*)GPIOA_BitReset_ADDR = 0x03
```

The bit operation of BSRR on ODR is one-shot, and the corresponding bit of ODR will not be locked. The user can still directly operate the ODR register.

13.3.5 I/O Locking Mechanism

The locking mechanism allows freezing the IO configuration. The lockable registers are: OTyPER, ODRVR, PUPDR, TTLLEN, AFRL and AFRH.

Each lock bit LCKy is used to lock an I/O port. Writing lock bit also needs to write a special control word (0x5AA5) to the corresponding unlock field (LOCK).



13.3.6 I/O Configured as External Interrupt/Wake-up Line

All ports can be selected to connect to the external interrupt input line via the external interrupt configuration register (EXTI_CFGL/EXTI_CFGH, in the EXTI module). In order to use an external interrupt line, the port needs to be configured in input mode.

For more information on external interrupts, see the "External Interrupt/Event Controller (EXTI)" chapter.

Note: It is recommended to configure as GPIO input mode, which is used as EXTI, but if it is configured as digital peripheral function, GPIO input channel is also turned on at this time, so it can also trigger external interrupt.

13.3.7 I/O Alternate Function

The registers AFRL and AFRH define alternate function of the port, allowing the same port to act as a shared input/output port for multiple multiplexing functions. Each port has a 16-input I/O multiplex selector for selecting the desired alternate function, which means only one peripheral connection is allowed per port at the same time.

The 16 channels of multiplex selector (AF0 to AF10) are divided into 3 categories:

- AF0: GPIO input and output off (default);
- AF1: GPIO basic functions;
- AF2~AF10: Digital/analog peripheral alternate function;

GPIO input and output off channel: AF0

This channel is a special state, forcing to shut down all input and output channels, but allowing pull up and down.

This is the default configuration after power on and reset. All I/O ports power-on are configured to GPIO input and output off function, except for the SWD interface.

GPIO basic channel: AF1

This is the general function GPIO, which can be configured as I/O basic channel AF1 by AFRL and AFRH registers.

The SWD interface pin can be used as the AF1 function after the SWJ-DP interface is closed by SWJCFG@SYSCFG_SAFR and configured by AFRL and AFRH registers.

Digital peripheral alternate channel: AF2 ~ AF10

The input and output attribute of digital peripheral alternate function is automatically set by hardware, but the output type (OTYPE), pull-up (PUPDR), output drive (ODRVR) and channel selection (AFRL/AFRH) parameters require user software configuration.

See the "Alternate Function Mapping Table" for complete mapping information.

Note: The bidirectional ports required by some peripherals can be automatically switched by hardware. Peripherals will ensure that they are not affected by the data fed back to the receiving channel when they are sent (if the receiving channel is turned on).

Analog peripheral alternate channel: AF2 ~ AF10

This channel is used for analog peripherals on chip. Input and output attribute, output types, and output driver configurations are meaningless, but pull-up have a special purpose in some applications.

The analog channel turns off the output buffer and the input schmitt trigger, which equals to disconnecting the digital input and output channels to avoid leakage.

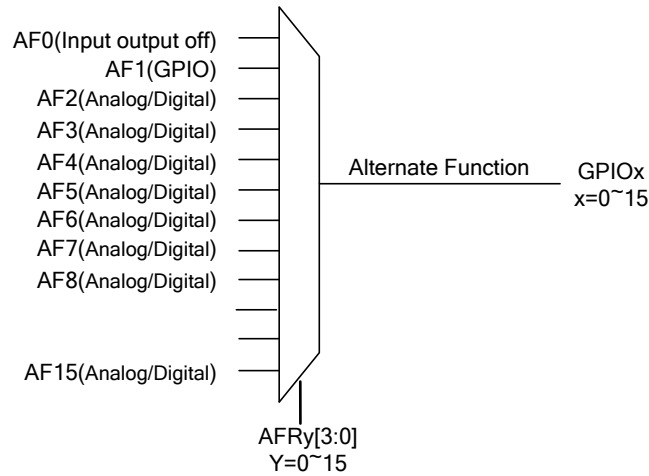


Figure 13-2 Multiplex selector block diagram for I/O alternate function

Based on the above-mentioned multiplexing basic functions, SH30F9/SA0 series also supports pin remapping, that is, the same peripheral can be mapped to different I / O ports. In order to adapt to the optimal configuration of resources in different packaging forms, it is also convenient for application wiring. The specific mapping can be compared with the definition in the "Alternate Function Mapping Table".

Note: The system allows the same peripheral to be configured to multiple different I/O ports at the same time. For example, if a peripheral is configured to PA0 and PA10 at the same time. When I/O is set as output, both ports output effectively; When I/O is set as input, the operation of both ports will affect this peripheral. This situation is guaranteed by the user software. In principle, the user software should avoid configuring the same function port to different ports.

13.3.8 GPIO Input Function

When I/O port is configured as a GPIO input:

- Output is invalid.
- Schmitt trigger input is activated.
- Pull-up resistors are connected or disconnected depending on pull-up configuration.
- I/O status can be obtained by read access to the input data register.

The figure below shows the GPIO input configuration of the I/O port bits.

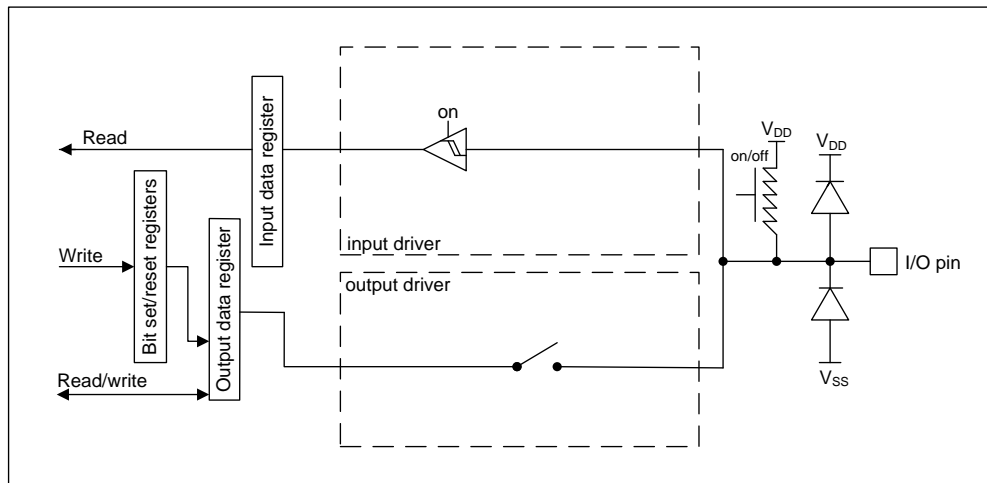


Figure 13-3 GPIO Input Configuration Block Diagram of I/O Port Bits



13.3.9 GPIO Output Function

When the I/O port is configured as a GPIO output:

- Output is valid.
- Open-drain mode: '0' on the output register activates N-MOS, while '1' on the output register places the port in a high-resistance state (P-MOS is inactive).
- Push-pull mode: '0' on the output register activates N-MOS, while '1' on the output register activates P-MOS.
 - Schmitt trigger input is activated.
 - Pull-up resistors are connected or disconnected depending on pull-up configuration.
 - I/O status can be obtained by read access to the input data register.
 - The last written value is obtained by read access to the output data register.

The figure below shows the GPIO output configuration for the I/O port bits.

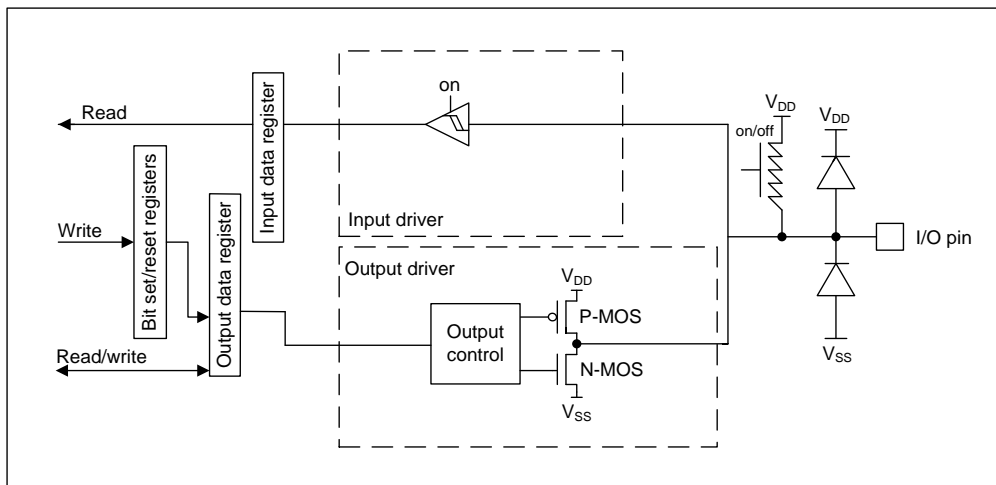


Figure 13-4 GPIO Output Configuration Block Diagram of I/O Port Bits

13.3.10 Digital Alternate Function

When the I/O port is configured for digital alternate:

- The output buffer is turned off as digital alternate input.
- The output buffer is turned on as digital alternate output (disconnect GPIO output data register).
- Digital alternate input or output attributes are determined by peripheral and are automatically configured when the function is activated.
- Schmitt trigger input is activated no matter it is input or output.
- Pull-up resistors are connected or disconnected depending on the pull-up configuration.
- I/O port status is available when reading the input data register.
- Except for the PWM/TXD port, the I/O port status can be obtained when reading input data registers from other ports.



The following figure shows the digital alternate function configuration of I/O Port Bits:

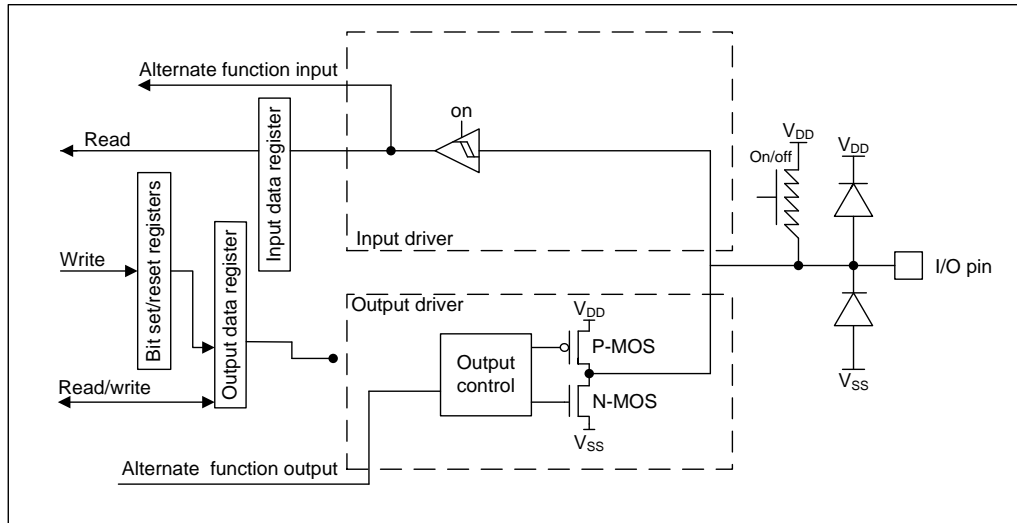


Figure 13-5 Digital Alternate Function Configuration Block Diagram of I/O Port Bits

If software configures an I/O pin as a digital alternate output function, but the peripheral is not activated, the output buffer is turned off and the output channel is in high-resistance state. Of course, the user can give a certain level by configuring pull-up. Note: GPIO input function, GPIO output function and digital alternate function are all turned on, so in fact when the I/O port is set as GPIO port (AF1), only I/O is additionally turned on Output channel.

13.3.11 Analog Alternate Function

When the I/O port is configured for analog alternate configuration:

- The output buffer is disabled.
- Automatically turn off the schmitt trigger input to achieve zero consumption on each analog I/O pin, and the schmitt trigger output value is forced to '1'.
- When reading the input data register, the value is '1'.
- Allow turn on pull-up.



The following figure shows the analog alternate configuration of the I/O port bits:

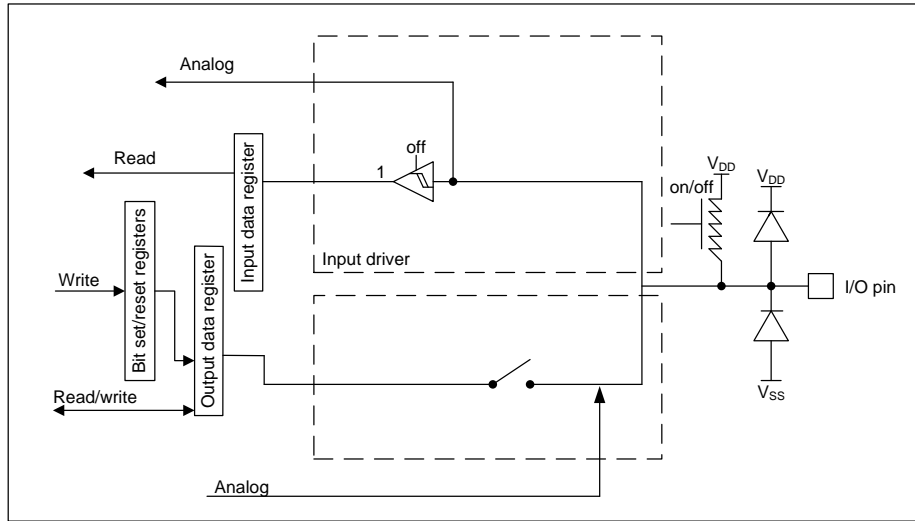


Figure 13-6 Analog Alternate Function Configuration Block Diagram of I/O Port Bits

13.3.12 Basic application of I/O pins

From the application point of view, I/O pins have only 2 functions: GPIO function and peripheral function (analog / digital). When using digital peripheral functions, the port bits need to be configured according to different peripheral types, including push-pull / open-drain, pull-up, and output drive capability.

Peripheral functions (analog / digital)

- Analog peripherals
Configured as an analog channel: 0010b(AF2)~1010b(AF10)
Turn off the pull-up resistor: PUPDRx = 0b (general application is not turned on by default)
- Digital peripherals
Configured as a digital channel: 0010b(AF2)~1010b(AF10)
Select push-pull / open drain, pull-up and output drive capabilities: OTyPERx, PUPDRx, ODRVRx

The general configuration rules for digital peripherals:

1. For the digital alternate input port, the port automatically switches to input, the user can configure the input float, pull up and the input pin is driven externally.
2. For the digital alternate output port, the port is automatically switched to output, and the user can configure push-pull or open-drain, driving capability, and pull up. Obviously, the pin and the data output register are disconnected and connected to the output signal of the on-chip peripheral.
3. For digital alternate bidirectional port, the port automatically switches between input and output, and the user can configure push-pull or open-drain, driving capability, and pull up.

A recommended peripheral configuration process:

- (1) According to the input and output properties of the alternate port, configure pull up (if necessary), push-pull or open drain (output), driving capacity (output);
- (2) Switch to the target peripheral channel;
- (3) Turn on the corresponding peripherals (some devices require the pull-up to be closed after turning on to avoid affecting the signal), the device works;
- (4) If necessary, configure the pull-up to give a certain level when the peripheral is turned off (the output channel will automatically be switched as high-resistance when the peripheral is turned off).

In actual applications, the order of (2) and (3) for some peripherals can also be exchanged, that is, the peripheral is enabled first, and then the target peripheral channel is switched.



GPIO function (input / output)

- Configuration direction
Input: MODERx = 0b
Output: MODERx = 1b
- For Input
Configure pull-up resistance: PUPDRx
Configure CMOS / TTL logic level: TTLEN_x
- For Output

Select push-pull / open-drain, pull-up and output drivers: OTyPERx, PUPDRx, ODRVRx

Take serial port module UART1 as an example: UART1 is a digital peripheral, the configuration is as follows:

| Alternate pin | Function definition | GPIO configuration |
|---------------|--|---|
| PA9/TXD1 | Asynchronous full-duplex mode, data transmission | Push-pull output, close pull-up, drive capacity default AFR9 = 0100b(AF4) |
| | Synchronous half-duplex mode, clock output | Push-pull output, close pull-up, drive capacity default AFR9 = 0100b(AF4) |
| PA10/RXD1 | Asynchronous full-duplex mode, data reception | Floating input or with pull-up input, AFR10 = 0100b(AF4) |
| | Synchronous half-duplex mode, two-way data | The push-pull output and the floating input are automatically switched. The pull-up is turned off, or only the pull-up is turned on. And the driving capacity is selected by default. AFR10 = 0100b(AF4) |



13.4 Registers

GPIO Module Register list (Base Address:0x4004 0000)

| Address | Register | Description |
|----------------------|----------|---|
| 0x4004 0000 + 0x80*x | MODER | Port x Mode Configuration register (PortA~D Correspond to x:0~3) |
| 0x4004 0004 + 0x80*x | IDR | Port x Input Data Register (PortA~D Correspond to x:0~3) |
| 0x4004 0008 + 0x80*x | ODR | Port x Output Data Register (PortA~D Correspond to x:0~3) |
| 0x4004 000C + 0x80*x | BSRR | Port x Reset/Set Register (PortA~D Correspond to x:0~3) |
| 0x4004 0010 + 0x80*x | LCKR | Port x Configuration Lock Register (PortA~D Correspond to x:0~3) |
| 0x4004 0014 + 0x80*x | IE | Port x input enable configuration register (PortA~D Correspond to x:0~3) |
| 0x4004 0018 + 0x80*x | OTYPER | Port x output type configuration register (PortA~D Correspond to x:0~3) |
| 0x4004 001C + 0x80*x | ODRVR | Port x output driver configuration register (PortA~D Correspond to x:0~3) |
| 0x4004 0020 + 0x80*x | PUPDR | Port x Pull-up Configuration register (PortA~D Correspond to x:0~3) |
| 0x4004 0024 + 0x80*x | TTLEN | Port x Partial Pin TTL Level Selection Register (PortA~D Correspond to x:0~3) |
| 0x4004 0028 + 0x80*x | AFRL | Port x Alternate Function Register Low Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (PortA~D Correspond to x:0~3) |
| 0x4004 002C + 0x80*x | AFRH | Port x Alternate Function Register High Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (PortA~D Correspond to x:0~3) |

13.4.1 Port x Mode Configuration register (GPIOx_MODER)

Offset Address: 0x0000 + 0x80*x

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----------|-----------|-----------|-----------|-----------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| MOD ER1 5 | MOD ER1 4 | MOD ER1 3 | MOD ER1 2 | MOD ER1 1 | MOD ER1 0 | MOD ER9 | MOD ER8 | MOD ER7 | MOD ER6 | MOD ER5 | MOD ER4 | MOD ER3 | MOD ER2 | MOD ER1 | MOD ER0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 | MODER15 | Port x mode configuration 0: input mode (reset state) 1: output mode |



| | | |
|----|----------------|---|
| 14 | MODER14 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 13 | MODER13 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 12 | MODER12 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 11 | MODER11 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 10 | MODER10 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 9 | MODER9 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 8 | MODER8 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 7 | MODER7 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 6 | MODER6 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 5 | MODER5 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 4 | MODER4 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 3 | MODER3 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 2 | MODER2 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 1 | MODER1 | Port x mode configuration 0: input mode (reset state) 1: output mode |
| 0 | MODER0 | Port x mode configuration 0: input mode (reset state) 1: output mode |

13.4.2 Port x Input Data Register (GPIOx_IDR)

Offset Address: 0x0004 + 0x80*x

Reset value: 0x0000 XXXX

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| IDRy(y=15~0) | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |



| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | IDRy(y=15~0) | Port x input data register This register is read-only and stores the level value which is read by the port. |

13.4.3 Port x Output Data Register (GPIOx_ODR)

Offset Address: 0x0008 + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ODRy(y=15~0) | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | ODRy(y=15~0) | Port x output data register This register can be read and written by software. Through the BSRR register after it, the ODR can be bit-set and reset independently to implement atomic bit operations. |

13.4.4 Port x Reset/Set Register (GPIOx_BSRR)

Offset Address: 0x000C + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| BRy(y=15~0) | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| BSy(y=15~0) | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|---|
| 31 ~ 16 | BRy(y=15~0) | Port x reset bit These bits are write only, reading these bits returns 0x0000. 0: No effect 1: Clear the corresponding IDRy bit Note: If the BRy and BSy of the same port are set at the same time, the BSy priority is higher. |
| 15 ~ 0 | BSy(y=15~0) | Port x set bit These bits are write only, reading these bits returns 0x0000. 0: No effect 1: Set the corresponding ODRy bit Note: If the same BRy and BSy of the same port are set at the same time, the BSy priority is higher. |



13.4.5 Port x Configuration Lock Register (GPIOx_LCKR)

Offset Address: 0x0010 + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCK15 | LCK14 | LCK13 | LCK12 | LCK11 | LCK10 | LCK9 | LCK8 | LCK7 | LCK6 | LCK5 | LCK4 | LCK3 | LCK2 | LCK1 | LCK0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 16 | LOCK[15:0] | LCK control bit operation unlock bit (Port x reset bit) These bits are written only, and reading these bits returns 0x0000. 0x5AA5: Unlock Other: Lock |
| 15 | LCK15 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 14 | LCK14 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 13 | LCK13 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 12 | LCK12 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 11 | LCK11 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 10 | LCK10 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 9 | LCK9 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |



| | | |
|---|-------------|---|
| 8 | LCK8 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 7 | LCK7 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 6 | LCK6 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 5 | LCK5 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 4 | LCK4 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 3 | LCK3 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 2 | LCK2 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 1 | LCK1 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |
| 0 | LCK0 | Port x lock bit 0: Port bit configuration is unlocked 1: Port bit configuration is locked These bits are readable and writable, and writing to these bits must satisfy the LOCK unlock condition (not required for reading) |

13.4.6 Port x input enable configuration register (GPIOx_IE)

Offset Address: 0x0014 + 0x80*x

Reset value: 0x0000 FFFF

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| IE15 | IE14 | IE13 | IE12 | IE11 | IE10 | IE9 | IE8 | IE7 | IE6 | IE5 | IE4 | IE3 | IE2 | IE1 | IE0 |
| <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 | IE15 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 14 | IE14 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 13 | IE13 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 12 | IE12 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 11 | IE11 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 10 | IE10 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 9 | IE9 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 8 | IE8 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 7 | IE7 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 6 | IE6 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 5 | IE5 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 4 | IE4 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 3 | IE3 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 2 | IE2 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 1 | IE1 | Port x input enable configuration 0: input disable 1: input enable (reset state) |
| 0 | IE0 | Port x input enable configuration 0: input disable 1: input enable (reset state) |



13.4.7 Port x output type configuration register (GPIOx_OTYPER)

Offset Address: 0x0018 + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|----------|----------|----------|--------------|----------|-----|-----|-----|-----|-----|-----|----------|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| OT1 5 | OT1 4 | OT1 3 | OT1 2 | Rese rved | OT1 0 | OT9 | OT8 | OT7 | OT6 | OT5 | OT4 | Reserved | | | |
| RW | RW | RW | RW | - | RW | RW | RW | RW | RW | RW | RW | - | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 | OT15 | PC15 output type configuration is configured by software Note: only PC15 is valid 0: output push-pull (reset state) 1: output open-drain |
| 14 | OT14 | PA14 output type configuration is configured by software Note: only PA14, PC14 is valid 0: output push-pull (reset state) 1: output open-drain |
| 13 | OT13 | PA13 output type configuration is configured by software Note: only PA13, PB13, PC13 is valid 0: output push-pull (reset state) 1: output open-drain |
| 12 | OT12 | PB12 output type configuration is configured by software Note: only PB12, PC12 is valid 0: output push-pull (reset state) 1: output open-drain |
| 11 | Reserved | - |
| 10 | OT10 | PC10 output type configuration is configured by software Note: only PC10 is valid 0: output push-pull (reset state) 1: output open-drain |
| 9 | OT9 | PB9 output type configuration is configured by software Note: only PB9, PC9 is valid 0: output push-pull (reset state) 1: output open-drain |
| 8 | OT8 | PB8 output type configuration is configured by software Note: only PB8 is valid 0: output push-pull (reset state) 1: output open-drain |
| 7 | OT7 | PC7 output type configuration is configured by software Note: only PC7, PD7 is valid 0: output push-pull (reset state) 1: output open-drain |
| 6 | OT6 | PC6 output type configuration is configured by software Note: only PC6, PD6 is valid 0: output push-pull (reset state) 1: output open-drain |
| 5 | OT5 | PA5 output type configuration is configured by software Note: only PA5, PB5, PC5 is valid 0: output push-pull (reset state) 1: output open-drain |



SH30F9/SA0 Series

| | | |
|-------|----------|--|
| 4 | OT4 | PA4 output type configuration is configured by software Note: only PA4, PB4 is valid 0: output push-pull (reset state) 1: output open-drain |
| 3 ~ 0 | Reserved | - |

13.4.8 Port x output driver configuration register (GPIOx_ODRVR)

Offset Address: 0x001C + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|----------|---------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | ODR VR_SINK 9 | Reserved | ODR VR_SINK 8 |
| | | | | | | | | | | | | | RW | - | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | ODR VR_SINK 7 | Reserved | ODR VR_SINK 6 | Reserved | ODR VR_SINK 5 | Reserved | ODR VR_SINK 4 | Reserved | ODR VR_SINK 3 | Reserved | ODR VR_SINK 2 | Reserved | | | |
| - | RW | - | RW | - | RW | - | RW | - | RW | - | RW | - | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|---|
| 31 ~ 19 | Reserved | - |
| 18 | ODRVR_SINK9 | Port x output sink current driver configuration is configured by software Note: only PB9 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 17 | Reserved | - |
| 16 | ODRVR_SINK8 | Port x output sink current driver configuration is configured by software Note: only PB8 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 15 | Reserved | - |
| 14 | ODRVR_SINK7 | Port x output sink current driver configuration is configured by software Note: only PB7 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 13 | Reserved | - |
| 12 | ODRVR_SINK6 | Port x output sink current driver configuration is configured by software Note: only PB6 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 11 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|--------------------|--|
| 10 | ODRVR_SINK5 | Port x output sink current driver configuration is configured by software Note: only PB5 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 9 | Reserved | - |
| 8 | ODRVR_SINK4 | Port x output sink current driver configuration is configured by software Note: only PB4 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 7 | Reserved | - |
| 6 | ODRVR_SINK3 | Port x output sink current driver configuration is configured by software Note: only PB3 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 5 | Reserved | - |
| 4 | ODRVR_SINK2 | Port x output sink current driver configuration is configured by software Note: only PB2 is valid 0: Normal state (reset value) 1: strong drive See "electrical characteristics" section of DataSheet for detailed driver capacity parameters |
| 3 ~ 0 | Reserved | - |

13.4.9 Port x Pull-up Configuration register (GPIOx_PUPDR)

Offset Address: 0x0020 + 0x80*x
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Rese rved | PHD R15 | Rese rved | PHD R14 | Rese rved | PHD R13 | Rese rved | PHD R12 | Rese rved | PHD R11 | Rese rved | PHD R10 | Rese rved | PHD R9 | Rese rved | PHD R8 |
| - | RW | - | RW | - | RW | - | RW | - | RW | - | RW | - | RW | - | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|--------------|------------|--------------|------------|--------------|------------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Rese rved | PHD R7 | Rese rved | PHD R6 | Rese rved | PHD R5 | Rese rved | PHD R4 | Rese rved | PHD R3 | Rese rved | PHD R2 | Rese rved | PHD R1 | Rese rved | PHD R0 |
| - | RW | - | RW | - | RW | - | RW | - | RW | - | RW | - | RW | - | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|-----|-----------------|--|
| 31 | Reserved | - |
| 30 | PHDR15 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 29 | Reserved | - |
| 28 | PHDR14 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 27 | Reserved | - |



SH30F9/SA0 Series

| | | |
|----|-----------------|--|
| 26 | PHDR13 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 25 | Reserved | - |
| 24 | PHDR12 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 23 | Reserved | - |
| 22 | PHDR11 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 21 | Reserved | - |
| 20 | PHDR10 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 19 | Reserved | - |
| 18 | PHDR9 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 17 | Reserved | - |
| 16 | PHDR8 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 15 | Reserved | - |
| 14 | PHDR7 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 13 | Reserved | - |
| 12 | PHDR6 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 11 | Reserved | - |
| 10 | PHDR5 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 9 | Reserved | - |
| 8 | PHDR4 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 7 | Reserved | - |
| 6 | PHDR3 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 5 | Reserved | - |
| 4 | PHDR2 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 3 | Reserved | - |
| 2 | PHDR1 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |
| 1 | Reserved | - |
| 0 | PHDR0 | Port x pull-up configuration is configured by software 0: No pull-up (reset state) 1: Pull-up |



SH30F9/SA0 Series

13.4.10 Port x Partial Pin TTL Level Selection Register (GPIOx_TTLEN)

Offset Address: 0x0024 + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TTL EN1 5 | TTL EN1 4 | TTL EN1 3 | TTL EN1 2 | TTL EN1 1 | TTL EN1 0 | TTL EN9 | TTL EN8 | TTL EN7 | TTL EN6 | TTL EN5 | TTL EN4 | TTL EN3 | TTL EN2 | TTL EN1 | TTL EN0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 | TTLEN15 | PA15 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 14 | TTLEN14 | PA14 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 13 | TTLEN13 | PA13 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 12 | TTLEN12 | PA12 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 11 | TTLEN11 | PA11 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 10 | TTLEN10 | PA10 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 9 | TTLEN9 | PA9 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 8 | TTLEN8 | PA8 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |



SH30F9/SA0 Series

| | | |
|---|---------------|--|
| 7 | TTLEN7 | PA7 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 6 | TTLEN6 | PA6 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 5 | TTLEN5 | PA5 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 4 | TTLEN4 | PA4 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 3 | TTLEN3 | PA3 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 2 | TTLEN2 | PA2 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 1 | TTLEN1 | PA1 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |
| 0 | TTLEN0 | PA0 pin TTL/CMOS input selection 0: Select CMOS input 1: Select TTL input See "electrical characteristics" for TTL parameters, and do not do any processing under stop mode. |

13.4.11 Port x Alternate Function Register Low Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (GPIOx_AFRL)

Offset Address: 0x0028 + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------------|-----|-----|-----|------------------|-----|-----|-----|------------------|-----|-----|-----|------------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| AFR7[3:0] | | | | AFR6[3:0] | | | | AFR5[3:0] | | | | AFR4[3:0] | | | |
| <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------------|-----|-----|-----|------------------|-----|----|----|------------------|----|----|----|------------------|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| AFR3[3:0] | | | | AFR2[3:0] | | | | AFR1[3:0] | | | | AFR0[3:0] | | | |
| <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|-----------|---|
| 31 ~ 28 | AFR7[3:0] | Port x alternate function selection is configured by software: AFR7 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 27 ~ 24 | AFR6[3:0] | Port x alternate function selection is configured by software: AFR6 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 23 ~ 20 | AFR5[3:0] | Port x alternate function selection is configured by software: AFR5 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |



SH30F9/SA0 Series

| | | |
|---------|------------------|---|
| 19 ~ 16 | AFR4[3:0] | Port x alternate function selection is configured by software: AFR4 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 15 ~ 12 | AFR3[3:0] | Port x alternate function selection is configured by software: AFR3 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 11 ~ 8 | AFR2[3:0] | Port x alternate function selection is configured by software: AFR2 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |



SH30F9/SA0 Series

| | | |
|-------|------------------|---|
| 7 ~ 4 | AFR1[3:0] | Port x alternate function selection is configured by software: AFR1 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 3 ~ 0 | AFR0[3:0] | Port x alternate function selection is configured by software: AFR0 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |

13.4.12 Port x Alternate Function Register High Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (GPIOx_AFRH)

Offset Address: 0x002C + 0x80*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-------------------|-----|-----|-----|-------------------|-----|-----|-----|-------------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| AFR15[3:0] | | | | AFR14[3:0] | | | | AFR13[3:0] | | | | AFR12[3:0] | | | |
| <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-------------------|-----|----|----|------------------|----|----|----|------------------|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| AFR11[3:0] | | | | AFR10[3:0] | | | | AFR9[3:0] | | | | AFR8[3:0] | | | |
| <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|------------|--|
| 31 ~ 28 | AFR15[3:0] | <p>Port x alternate function selection is configured by software:</p> <p>AFR15 function selection:</p> <ul style="list-style-type: none"> 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 27 ~ 24 | AFR14[3:0] | <p>Port x alternate function selection is configured by software:</p> <p>AFR14 function selection:</p> <ul style="list-style-type: none"> 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 23 ~ 20 | AFR13[3:0] | <p>Port x alternate function selection is configured by software:</p> <p>AFR13 function selection:</p> <ul style="list-style-type: none"> 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |



SH30F9/SA0 Series

| | | |
|---------|-------------------|--|
| 19 ~ 16 | AFR12[3:0] | Port x alternate function selection is configured by software: AFR12 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 15 ~ 12 | AFR11[3:0] | Port x alternate function selection is configured by software: AFR11 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 11 ~ 8 | AFR10[3:0] | Port x alternate function selection is configured by software: AFR10 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |



SH30F9/SA0 Series

| | | |
|-------|------------------|---|
| 7 ~ 4 | AFR9[3:0] | Port x alternate function selection is configured by software: AFR9 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |
| 3 ~ 0 | AFR8[3:0] | Port x alternate function selection is configured by software: AFR8 function selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 |



14. System Configuration Module (SYSCFG)

This module is used for system level auxiliary modules and system configuration. Including SRAM sector lock control definition, pin function reimage, debugging interface and other related configuration. The module is attached to the AHB bus and has independent enable and reset control.

14.1 System Configuration Correlation

14.1.1 SWD and XTAL pin alternate function definition

The debugger interface is a two-wire SW interface (SWDIO and SWCLK), which is used as a SWD interface by default, and can be configured as a GPIO port through SWJCFG @ SYSCFG_SAFR. Once the configuration is completed, it can only be reset to the debug interface state.

The external oscillator pins XTAL1 and XTAL2 are used as input and output are all off by default, and can be configured as an external oscillator interface or an external clock interface through OSCCFG @ SYSCFG_SAFR. Once the configuration is completed, they can only be reset to the default interface state.

When the SWD interface and XTAL interface are used as non-GPIO interfaces, the GPIO configuration parameters are invalid (you can write but have no effect), and the configuration including the input and output directions is also invalid.

Note: When the XTAL pin is used as an oscillator interface or an external clock interface, the hardware needs to be shielded from the pull-up, otherwise, due to software misconfiguration, it will cause serious consequences of oscillator stop.

14.1.2 Debug Interface Register

DBGCR is the debug interface register. The working status of each major peripheral module in debug mode and the system state when entering low consumption mode are defined. The debugging switch options are provided. For more details, see "Debug Interface" part.



14.2 Registers

SYSCFG Module Register list (Base Address:0x4004 0400)

| Address | Register | Description |
|-------------|----------|--|
| 0x4004 0408 | SAFR | System Configuration Register |
| 0x4004 040C | SRAMLOCK | SRAM Sector Lock Control Register |
| 0x4004 0410 | DBGCR | Debug Interface Control Register |
| 0x4004 0414 | GPIOBCR | GPIO module mount bus configuration register |

14.2.1 System Configuration Register (SYSCFG_SAFR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|-------------------|-----|-----|-----|-----|-----|-----|----------------|----------------|------------------|----------------|-----------------|-----|-----|--------------------|-----|
| LOCK[15:0] | | | | | | | | | | | | | | | |
| <i>WO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | IEN_CSM | IEN_BOD | IEN_EXTI0 | SWJ_CFG | Reserved | | | OSCCFG[1:0] | |
| - | | | | | | | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW1t</i> | - | | | <i>RW1t</i> | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------------|---|
| 31 ~ 16 | LOCK[15:0] | Unlock bit of this register These bits are write only, reading these bits returns 0x00. 0x5AA5: Unlocked Other: Locked Note: Writing to this register need to cooperate with unlock control. Reading this register does not require unlocking. |
| 15 ~ 9 | Reserved | - |
| 8 | IEN_CSM | CSM causes NMI interrupt enable bit |
| 7 | IEN_BOD | BOD causes NMI interrupt enable bit |
| 6 | IEN_EXTI0 | EXTI0 causes NMI interrupt enable bit |
| 5 | SWJCFG | Serial-Wire SWD Pin Multiplexing Function Definition These bits can only be written by software (reading these bits will return undefined values) for configuring SW and I/O ports which track alternate functions. The default state after system reset is to enable SW. To release the debug ports to general purpose I/O ports, the user software can set SWJCFG after reset. 0: The SWDIO/SWCLK pin is used as debugging port, not as multiplexed as general purpose I/O. 1: Close SW debugging function; SWCLK \ SWDIO port can be multiplexed as general purpose I/O. But SWDIO and SWCLK can be multiplexed as general purpose I/O in normal operation state, and they are still forced to be used as debug interfaces in debug mode state. Note: Once this control bit is set, it can only be modified by resetting. The user should carefully configure this bit, because some options will cause the debugging mode to enter only with the cooperation of the reset signal. |
| 4 ~ 2 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|--------------------|---|
| 1 ~ 0 | OSCCFG[1:0] | XTAL1/XTAL2 Pin Multiplexing Function Definition 00: XTAL1/XTAL2 is Input and output all off (default) 01: External oscillator interface (crystal and ceramic) 10: XTAL1 is externally clocked, XTAL2 is the input and output fully closed state 11: External low frequency oscillator interface(Crystal 32.768KHz) Note: Once this control bit is set, it can only be modified by resetting. |
|-------|--------------------|---|

14.2.2 SRAM Sector Lock Control Register (SYSCFG_SRAMLOCK)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| <i>WO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-------------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SRAMLCKy(y=15~0) | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------------------|--|
| 31 ~ 16 | LOCK[15:0] | Unlock bit of SRAM LCKy control bit operation These bits are written only, reading these bits returns 0x0000. 0x5AA5: Unlock Other: Locked |
| 15 ~ 0 | SRAMLCKy(y=15~0) | SRAM sector y lock bit(SRAM sector y lock bit) These bits are readable and writable, and writing to these bits must satisfy the unlock condition (not required for reading). 0: SRAM sector y is not locked 1: SRAM sector y is locked Note: SRAM sector start address is 0x2000 0000, every sector is 1K byte, total 16K byte. |

14.2.3 Debug Interface Control Register (SYSCFG_DBGCR)

Offset Address: 0x0010

Reset value: 0x0000 0020

| | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LOCK[15:0] | | | | | | | | | | | | | | | |
| <i>WO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-------------|-------------|------------------|-----------------|-------------|-----------------|------------------|------------------|-----------------|-----------------|-----------------|--------------|------------------|-------------------|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | DBG _TWI | DBG _SPI | DBG _UA RT | DBG _PW M | DBG _TIM | DBG _PC A | DBG _W WDT | DBG _IW DT | DBG _DM A | DBG _LE D | DBG _LC D | Rese rved | DBG _ST OP | DBG _SL EEP | |
| - | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | - | <i>RW</i> | <i>RW</i> | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|-------------------|--|
| 31 ~ 16 | LOCK[15:0] | Unlock bit of this register These bits are written only, reading these bits returns 0x0000. 0x5AA5:Unlocked Other: Locked Note: Writing to this register need to cooperate with unlock control. Reading this register does not require unlocking. |
| 15 ~ 14 | Reserved | - |
| 13 | DBG_TWI | TWI stopped when core is halted 0:TWI is stopped 1:TWI is still working properly |
| 12 | DBG_SPI | SPI stopped when core is halted 0:SPI is stopped 1:SPI is still working properly |
| 11 | DBG_UART | UART stopped when core is halted 0:UART is stopped 1:UART is still working properly |
| 10 | DBG_PWM | PWM stopped when core is halted 0:PWM is stopped 1:PWM is still working properly |
| 9 | DBG_TIM | TIM0-TIM3 counter stopped when core is halted 0:TIM0-TIM3 module is stopped 1:TIM0-TIM3 module is still working properly |
| 8 | DBG_PCA | PCA stopped when core is halted 0:PCA is stopped 1:PCA is still working properly |
| 7 | DBG_WWDT | Window watchdog stopped when core is halted 0:Window watchdog counter is stopped 1:Window watchdog counter is still working properly |
| 6 | DBG_IWDT | Watchdog stopped when core is halted 0:Watchdog counter is stopped 1:Watchdog counter is still working properly |
| 5 | DBG_DMA | DMA stopped when core is halted 0:DMA is stopped 1:DMA is still working properly |
| 4 | DBG_LED | LED stopped when core is halted 0:LED is stopped 1:LED is still working properly |
| 3 | DBG_LCD | LCD stopped when core is halted 0:LCD is stopped 1:LCD is still working properly |
| 2 | Reserved | - |
| 1 | DBG_STOP | Debug stop mode 0: (FCLK on, HCLK on) In stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting the stop mode, the software must reconfigure the clock system to start the PLL, crystal oscillator, etc. (the operation is the same as when this bit is set to 0) 1: (FCLK off, HCLK off) In stop mode, the clock controller prohibits all clocks (including HCLK and FCLK). When exiting from stop mode, the clock configuration is the same as the configuration after reset (the microcontroller is clocked by the 24MHz internal RC oscillator (HSI)). Therefore, the software must reconfigure the clock control system to start the PLL, crystal oscillator, etc. |
| 0 | DBG_SLEEP | Debug sleep mode 0:(FCLK on, HCLK on) In sleep mode, both FCLK and HCLK clocks are provided by the originally configured system clock. 1:(FCLK on, HCLK off) In sleep mode, FCLK is provided by the system clock that has been configured before, and HCLK is off. Since sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode. |

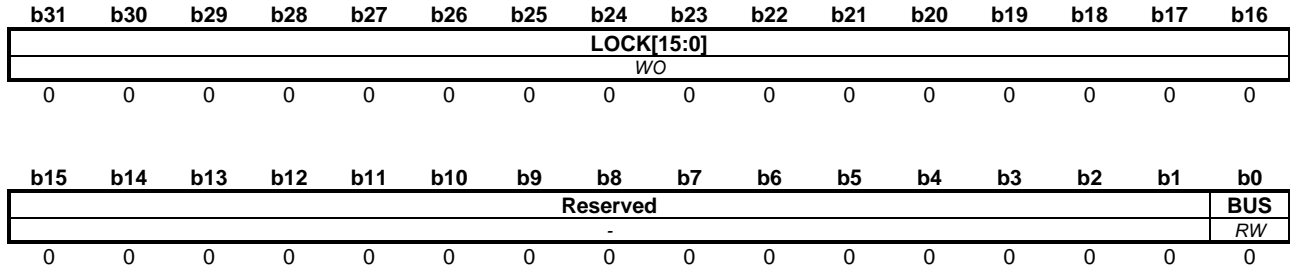


SH30F9/SA0 Series

14.2.4 GPIO module mount bus configuration register (SYSCFG_GPIOBCR)

Offset Address: 0x0014

Reset value: 0x0000 0000



| Bit | Symbol | Description |
|---------|-------------------|---|
| 31 ~ 16 | LOCK[15:0] | Unlock bit of this register These bits are written only, reading these bits returns 0x0000. 0x5AA5: Unlocked Other: Locked Note: Writing to this register need to cooperate with unlock control. Reading this register does not require unlocking. |
| 15 ~ 1 | Reserved | - |
| 0 | BUS | GPIO module mount bus selection bit 0: GPIO module is mounted on the single-cycle IO bus (SCIOB) (CPU can be accessed normally, DMA is not accessible) 1: GPIO module is mounted on the AHB bus (CPU and DMA can be accessed normally) |



15. Interrupts and events

15.1 Nestable Interrupt Vector Controller (NVIC)

15.1.1 Introduction

NVIC is an integrated part of Cortex-M0+ processor. It is closely linked to CPU, reducing interrupt latency and allowing new interrupts to be processed efficiently.

NVIC can also handle system exceptions, including: Reset, Non-maskable Interrupt (NMI), Hard Fault, Call System Services (SVCall), PendSV, and SysTick.

15.1.2 Main Features

- Nested vector controller is an integral part of ARM Cortex-M0+ core.
- Highly coupled interrupt controllers provide low interrupt latency.
- Processing system exceptions and peripheral interrupts.
- SH30F9/SA0 series support 32 maskable interrupts (excluding 16 Cortex-M0+ interrupts).
- 4 programmable interrupt priorities (using 2-bit interrupt priority)
- Low latency exceptions and interrupt handling
- Resettable vector table
- Interrupts maskable
- Software interrupt

Nested interrupt support

The NVIC provides nested interrupt support. All external interrupts and most of system exceptions can be programmed to different priority levels. The current priority is stored in a dedicated field of xPSR. When an interrupt occurs, the NVIC compares the priority of this interrupt to the current exception priority to see which one is higher. If an exception with higher priority is found, the processor interrupts the current interrupt service routine (or ordinary procedure) and serves the new one, which means interrupt preemption.

Vectored interrupt support

The Cortex-M0+ processor has vectored interrupt support. When an interrupt is accepted, the starting address of the interrupt service routine (ISR) is located from a vector table in memory. Priority levels of interrupts can be changed by software during runtime. Interrupts that are being serviced are blocked from further activation until the ISR is completed, so their priority can be changed without risk of accidental reentry.

Reduction of Interrupt Latency

The Cortex-M0+ introduces several new features to reduce interrupt latency. Includes automatic on-site protection and recovery, as well as other measures to reduce ISR delay when interrupting nesting.

Interrupts mask

You can either mask an interrupt/exception individually (set NVIC -> ISER register) or block all interrupts (set PRIMASK register). They can be used to ensure that time-critical tasks can be finished on time without being interrupted.

15.1.3 Interrupts Summary

The summary of all interrupts for SH30F9/SA0 series is shown in the following table:

The NMI source contains CSM, EXTIO, BOD, and every source has separate control bits (see the SYSCFG_SAFR register). EXTIO can select PA0, PB0, PC0, PD0 as inputs. In addition to the NMI entry, EXTIO also has a separate EXTIO entry, so user needs to choose from generating NMI or EXTIO (can be enabled at the same time). BOD also has two interrupt entries, while CSM has only NMI interrupt entry.



SH30F9/SA0 Series

5 Exception + 32 HWINT

| ID | Interrupt Source | Vector Address | Interrupt mask bit | Interrupt enable bit | Interrupt flag bit |
|-----|------------------|----------------|---|--|--|
| -15 | RESET | 0x0004 | | | |
| -14 | NMI | 0x0008 | SYSCFG_SAFR_IEN_CSM SYSCFG_SAFR_IEN_BOD SYSCFG_SAFR_IEN_EXTI0 | | RCC_CISTR_PLLCSMF RCC_CISTR_HSECSMF RCC_CISTR_LSECSMF SYSCFG_PWRSR_BODIF EXTI_PR_PR0 |
| -13 | HardFault | 0x000C | | | |
| -12 | | 0x0010 | | | |
| -11 | | 0x0014 | | | |
| -10 | | 0x0018 | | | |
| -9 | (reserved) | 0x001C | | | |
| -8 | (reserved) | 0x0020 | | | |
| -7 | (reserved) | 0x0024 | | | |
| -6 | (reserved) | 0x0028 | | | |
| -5 | SVC | 0x002C | PRIMASK | | SCB_SHCSR_SVCALLPENDE |
| -4 | | 0x0030 | | | |
| -3 | (reserved) | 0x0034 | | | |
| -2 | PendSV | 0x0038 | PRIMASK | | |
| -1 | SysTick | 0x003C | PRIMASK | SysTick_CTRL_TICKINT | |
| 0 | WWDT | 0x0040 | NVIC->ISER[0].0 | WWDT_CR_WWDTIE | WWDT_SR_WWDTIF |
| 1 | BOD | 0x0044 | NVIC->ISER[0].1 | SYSCFG_PWRCR_BODIE | SYSCFG_PWRSR_BODIF |
| 2 | RCC | 0x0048 | NVIC->ISER[0].2 | RCC_CIENR_PLLRDYIE RCC_CIENR_HSERDYIE RCC_CIENR_LSERDYIE | RCC_CISTR_PLLRDYIF RCC_CISTR_HSERDYIF RCC_CISTR_LSERDYIF |
| 3 | EXTI0 | 0x004C | NVIC->ISER[0].3 | EXTI_IMR_IMR0 | EXTI_PR_PR0 |
| 4 | EXTI1 | 0x0050 | NVIC->ISER[0].4 | EXTI_IMR_IMR1 | EXTI_PR_PR1 |
| 5 | EXTI2 | 0x0054 | NVIC->ISER[0].5 | EXTI_IMR_IMR2 | EXTI_PR_PR2 |
| 6 | EXTI3 | 0x0058 | NVIC->ISER[0].6 | EXTI_IMR_IMR3 | EXTI_PR_PR3 |
| 7 | DMA_CH0_1 | 0x005C | NVIC->ISER[0].7 | DMA_CCR0_TEIE DMA_CCR0_BEIE DMA_CCR0_HTIE DMA_CCR0_TCIE DMA_CCR1_TEIE DMA_CCR1_BEIE DMA_CCR1_HTIE DMA_CCR1_TCIE | DMA_IFSR_TEIF0 DMA_IFSR_BEIF0 DMA_IFSR_HTIF0 DMA_IFSR_TCIF0 DMA_IFSR_TEIF1 DMA_IFSR_BEIF1 DMA_IFSR_HTIF1 DMA_IFSR_TCIF1 |
| 8 | DMA_CH2_3 | 0x0060 | NVIC->ISER[0].8 | DMA_CCR2_TEIE DMA_CCR2_BEIE DMA_CCR2_HTIE DMA_CCR2_TCIE DMA_CCR3_TEIE DMA_CCR3_BEIE DMA_CCR3_HTIE DMA_CCR3_TCIE | DMA_IFSR_TEIF2 DMA_IFSR_BEIF2 DMA_IFSR_HTIF2 DMA_IFSR_TCIF2 DMA_IFSR_TEIF3 DMA_IFSR_BEIF3 DMA_IFSR_HTIF3 DMA_IFSR_TCIF3 |
| 9 | ADC | 0x0064 | NVIC->ISER[0].9 | ADC_ADCON1_ADIE ADC_ADCMPCON_ADGIE ADC_ADCMPCON_ADLIE | ADC_ADINTF_ADIF ADC_ADINTF_ADGIF ADC_ADINTF_ADLIF |
| 10 | PWM0 | 0x0068 | NVIC->ISER[0].10 | PWM0_CR_PWMIE PWM0_CR_EFLTIE | PWM0_PWMINTF_PWMIF PWM0_PWMINTF_FLTS |
| 11 | PWM1 | 0x006C | NVIC->ISER[0].11 | PWM1_CR_PWMIE PWM1_CR_EFLTIE | PWM1_PWMINTF_PWMIF PWM1_PWMINTF_FLTS |
| 12 | PWM2 | 0x0070 | NVIC->ISER[0].12 | PWM2_CR_PWMIE PWM2_CR_EFLTIE | PWM2_PWMINTF_PWMIF PWM2_PWMINTF_FLTS |



SH30F9/SA0 Series

| | | | | | |
|----|-----------|--------|------------------|--|--|
| 13 | PWM3 | 0x0074 | NVIC->ISER[0].13 | PWM3_CR_PWMIE PWM3_CR_EFLTIE | PWM3_PWMINTF_PWMIF PWM3_PWMINTF_FLTS |
| 14 | PCA0 | 0x0078 | NVIC->ISER[0].14 | PCA0_CFGR_PIE PCA0_CFGR_CIE PCA0_CCMR0_CCIE PCA0_CCMR1_CCIE | PCA0_SR_PIF PCA0_SR_CIF PCA0_SR_CC0IF PCA0_SR_CC1IF |
| 15 | PCA1 | 0x007C | NVIC->ISER[0].15 | PCA1_CFGR_PIE PCA1_CFGR_CIE PCA1_CCMR0_CCIE PCA1_CCMR1_CCIE | PCA1_SR_PIF PCA1_SR_CIF PCA1_SR_CC0IF PCA1_SR_CC1IF |
| 16 | PCA2 | 0x0080 | NVIC->ISER[0].16 | PCA2_CFGR_PIE PCA2_CFGR_CIE PCA2_CCMR0_CCIE PCA2_CCMR1_CCIE | PCA2_SR_PIF PCA2_SR_CIF PCA2_SR_CC0IF PCA2_SR_CC1IF |
| 17 | PCA3 | 0x0084 | NVIC->ISER[0].17 | PCA3_CFGR_PIE PCA3_CFGR_CIE PCA3_CCMR0_CCIE PCA3_CCMR1_CCIE | PCA3_SR_PIF PCA3_SR_CIF PCA3_SR_CC0IF PCA3_SR_CC1IF |
| 18 | EXTI9_4 | 0x0088 | NVIC->ISER[0].18 | EXTI_IMR_IMR4 EXTI_IMR_IMR5 EXTI_IMR_IMR6 EXTI_IMR_IMR7 EXTI_IMR_IMR8 EXTI_IMR_IMR9 | EXTI_PR_PR4 EXTI_PR_PR5 EXTI_PR_PR6 EXTI_PR_PR7 EXTI_PR_PR8 EXTI_PR_PR9 |
| 19 | UART0 | 0x008C | NVIC->ISER[0].19 | UART0_CR_TIE UART0_CR_RIE UART0_CR_TCIE UART0_CR_LBDIE | UART0_FR_TI UART0_FR_RI UART0_FR_TC UART0_FR_LBD |
| 20 | TWIO | 0x0090 | NVIC->ISER[0].20 | TWIO_CR_TWINTIE TWIO_CR_ETOT TWIO_CR_EFREE | TWIO_FR_TWINT TWIO_FR_TOUT TWIO_FR_TFREE |
| 21 | SPIO | 0x0094 | NVIC->ISER[0].21 | SPIO_CR_SPTIE SPIO_CR_SPRIE SPIO_CR_SSDIS | SPIO_FR_SPTI SPIO_FR_SPRI SPIO_FR_MODF |
| 22 | SPI1_LED | 0x0098 | NVIC->ISER[0].22 | SPI1_CR_SPTIE SPI1_CR_SPRIE SPI1_CR_SSDIS LED_COMCR_COMIE LED_AUCR_AUIE | SPI1_FR_SPTI SPI1_FR_SPRI SPI1_FR_MODF LED_FR_COMIF LED_FR_AUIF |
| 23 | UART1 | 0x009C | NVIC->ISER[0].23 | UART1_CR_TIE UART1_CR_RIE UART1_CR_TCIE UART1_CR_LBDIE | UART1_FR_TI UART1_FR_RI UART1_FR_TC UART1_FR_LBD |
| 24 | UART2 | 0x00A0 | NVIC->ISER[0].24 | UART2_CR_TIE UART2_CR_RIE UART2_CR_TCIE UART2_CR_LBDIE | UART2_FR_TI UART2_FR_RI UART2_FR_TC UART2_FR_LBD |
| 25 | UART3 | 0x00A4 | NVIC->ISER[0].25 | UART3_CR_TIE UART3_CR_RIE UART3_CR_TCIE UART3_CR_LBDIE | UART3_FR_TI UART3_FR_RI UART3_FR_TC UART3_FR_LBD |
| 26 | EXTI15_10 | 0x00A8 | NVIC->ISER[0].26 | EXTI_IMR_IMR10 EXTI_IMR_IMR11 EXTI_IMR_IMR12 EXTI_IMR_IMR13 EXTI_IMR_IMR14 EXTI_IMR_IMR15 | EXTI_PR_PR10 EXTI_PR_PR11 EXTI_PR_PR12 EXTI_PR_PR13 EXTI_PR_PR14 EXTI_PR_PR15 |



SH30F9/SA0 Series

| | | | | | |
|----|----------|--------|------------------|----------------|-------------------|
| 27 | TIM0 | 0x00AC | NVIC->ISER[0].27 | TIM0_CR_IE | TIM0_TIMINTF_TF |
| 28 | TIM1 | 0x00B0 | NVIC->ISER[0].28 | TIM1_CR_IE | TIM1_TIMINTF_TF |
| 29 | TIM2 | 0x00B4 | NVIC->ISER[0].29 | TIM2_CR_IE | TIM2_TIMINTF_TF |
| 30 | TIM3 | 0x00B8 | NVIC->ISER[0].30 | TIM3_CR_IE | TIM3_TIMINTF_TF |
| 31 | TouchKey | 0x00BC | NVIC->ISER[0].31 | TOUCHKEY_CR_IE | TOUCHKEY_FR_IFAVE |

For the above interrupt source (except for system reset interrupt), only when the triggering of interrupt is allowed by both corresponding interrupt mask bit and interrupt enable bit, the corresponding interrupt flag is set by hardware, then the interrupt will be triggered. The interrupt response process is determined by interrupt priority settings and nesting settings.

15.2 External Interrupt/Event Controller (EXTI)

The external interrupt/event controller consists of 16 levels and edge detectors, all of those 16 lines can be configured to the GPIO port. Each input line can be independently configured with an input type (level or edge) and the corresponding trigger event (high/low level, rising/falling edge or both edges). Each input line can also masked independently. A pending register maintains the interrupt requests of status line.

15.2.1 Main Features

- Independent trigger and mask on each interrupt/event line
- Dedicated status bit for each interrupt line
- Generation of up to 16 software interrupt requests
- DMA trigger on each interrupt line
- Adjustable sampling parameters for input detection, And EXTI0~ EXTI4 have the debounce ability in normal operation and stop mode

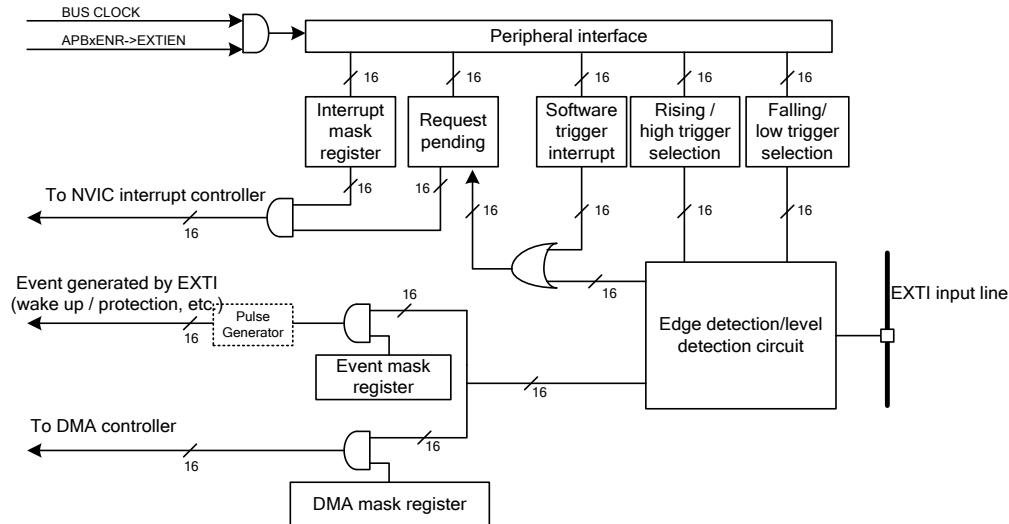


Figure 15-1 External Interrupt/Event Controller Block Diagram



The illustration above is as follows:

1. The interrupt and event are homologous before the logical or. The edge detection circuit detects the edge change event of the external input line, and then divides it into three channels. One channel enters the interrupt system, requests the CPU to respond to the interrupt. The second channel enters the event system, generates a hardware pulse through the pulse generator as the trigger signal of other modules in the system. The third channel enters the DMA controller, which can be touched to send DMA for operation (in fact, both the second and the third channels are events in nature, in this circuit, the two are independently for clarity);
2. The Edge detection circuit can detect rising edge, falling edge and rising-falling edge;
3. The level detection circuit can detect high level, low level and high-low level;
4. The software interrupt takes precedence over the external signal, that is, when the software interrupt position is "1", no matter what the external input is, the logical or always outputs the effective signal;

Note: On the basis of the interrupt system, the event occurrence path is reserved, which is mainly used as the trigger source of some special peripherals, such as DMA transmission, such as buffer register update, AD sampling trigger, event wake-up core (WFE), etc. These trigger operations directly affect the hardware and do not need to go through the CPU interrupt processing path.

Note: The request pending register and interrupt mask register of EXTI are independent of NVIC.

EXTI0 can cause NMI interrupt, which is enabled by IEN_EXTI0@SYSCFG_SAFR control bit.

15.2.2 Wake-up Event Management

SH30F9/SA0 series can handle external or internal events to wake up the core (WFE). Wake-up events can be generated by the following configuration:

- Enable an interrupt in the peripheral control register, but not in NVIC, and enabling the SEVONPEND bit in the Cortex-M0+ system control register. When the MCU resumes from the WFE, the peripheral interrupt pending bit (in the EXTI pending register) and the peripheral NVIC IRQ channel pending bit (in the NVIC interrupt pending clear register) have to be cleared.
- Configure an external or internal EXTI line to be in event mode. When the CPU resumes from WFE, it is not necessary to clear the peripheral interrupt pending bit or the NVIC IRQ channel pending bit as the pending bit corresponding to the event line is not set. To use the external I/O port as a wake-up event, see the "GPIO Configuration for External Interrupt/Wake-up Line" description in the GPIO part.

15.2.3 Function Description

To generate an interrupt, the interrupt line must be configured and enabled. Two trigger registers are set according to the required edge/level detection, and enabling interrupt request by writing '1' to the corresponding bit of the interrupt enable register. When the expected edge/level occurs on the external interrupt line, an interrupt request is generated and the corresponding pending bit is also set. This request is reset by writing '1' to the clear bit of the pending register.

To generate an event, the event line must be configured and enabled. Two trigger registers are set according to the required edge/level detection, and enabling event request by writing '1' to the corresponding bit of the event mask register. When the expected edge/level occurs on the event line, an event request pulse is generated and the pending bit corresponding to the interrupt path is also set. By writing '1' to the software interrupt register, interrupt requests can also be generated by software.

SH30F9/SA0 series supports level trigger. By setting the trigger mode to level trigger and setting the high level or low level trigger, EXTI can work in the level trigger state, which is different from the edge trigger. If the trigger level does not change, it will continue triggering.

The EXTI input signal supports sampling filtering. It uses bus clock as the clock source during normal operation, and supports sampling clock prescaler (up to 128 division), supports multiple sampling to achieve filtering effect. Use LSI as clock source in shutdown mode ^{Note}. See the "EXTI Sampling Control Register" for details.

Note: LSI is a filter clock source, because of its low frequency, it blocks clock division in hardware, but the multiple sampling function is reserved.



EXTI supports as a DMA trigger source. When an EXTI external event occurs, it can choose to send a DMA request to trigger a DMA transfer. The DMA request is released after the DMA controller responds. DMA enable signal of EXTI is described in the "DMA request mask controller".

EXTI0 can cause an NMI interrupt, which is enabled by the IEN_EXTI0@SYSCFG_SAFR control bit.

Hardware interrupt selection

Use the following procedure to configure 16 input lines as the interrupt sources:

- Configure the mask bits of 16 interrupt lines (EXTI_IMR)
- Configure the trigger selection bits (EXTI_RTISR and EXTI_FTISR) of the selected interrupt lines
- Configure the enable and mask bits of the NVIC interrupt channel mapped to the external interrupt controller (EXTI) so that the requests in the 16 interrupt lines can be correctly responded to.

Hardware event selection

Use the following procedure to configure 16 input lines as the event sources:

- Configure the mask bits of 16 event lines (EXTI_EMR)
- Configure the trigger selection bits (EXTI_RTISR and EXTI_FTISR) of the event line

Software triggered interrupt selection

The 16 lines can be configured as software trigger interrupt lines. The following is the process of generating a software interrupt:

- Configure mask bits of 16 interrupt lines (EXTI_IMR)
- Set the required bit of the software trigger interrupt register (EXTI_SWIER)



15.2.4 External Interrupt/Event Line Mapping

All GPIOs are connected to the 16 external interrupt/event lines in the following manner:

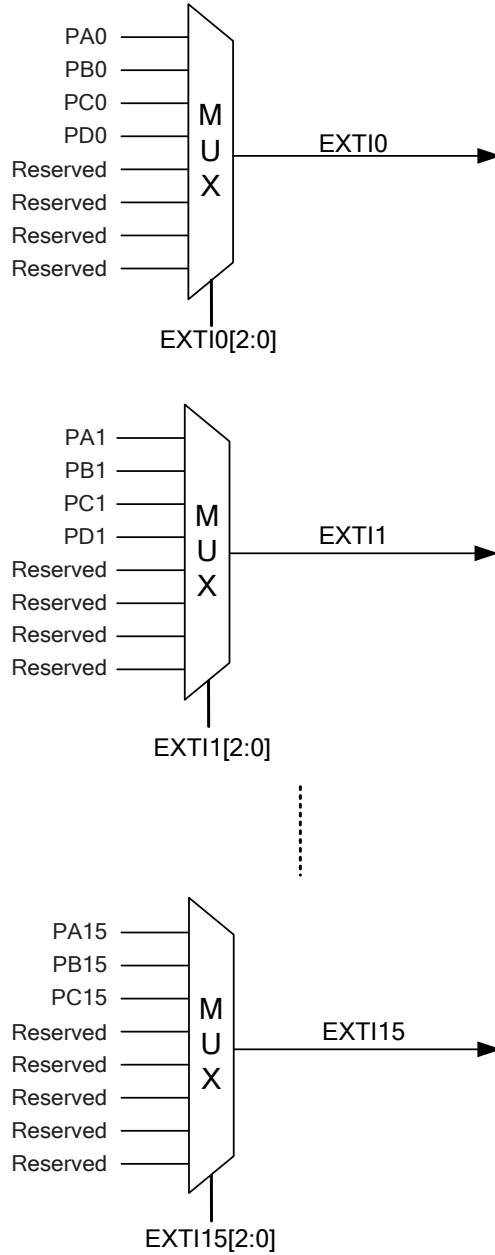


Figure 15-2 External Interrupt General I/O Mapping



15.3 Registers

EXTI Module Register list (Base Address:0x4000 3400)

| Address | Register | Description |
|-------------|----------|--|
| 0x4000 3400 | IMR | Interrupt Mask Register |
| 0x4000 3404 | EMR | Event Enable Register |
| 0x4000 3408 | TMSR | Trigger Mode Selection Register |
| 0x4000 340C | RTSR | Rising Edge/High Level Trigger Mode Selection Register |
| 0x4000 3410 | FTSR | Falling Edge/Low Level Trigger Mode Selection Register |
| 0x4000 3414 | SWIER | Software Trigger Interrupt Register |
| 0x4000 3418 | PR | Pending Register |
| 0x4000 341C | CFGL | External Interrupt Configuration Low Register |
| 0x4000 3420 | CFGH | External Interrupt Configuration High Register |
| 0x4000 3424 | SAMP | EXTI Sample Control Low Register |
| 0x4000 3428 | DMR | DMA Enable Register |

15.3.1 Interrupt Mask Register (EXTI_IMR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| IMRy(y=15~0) | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | IMRy(y=15~0) | Interrupt mask on line y 0: Mask interrupt requests from line y 1: Open interrupt request from line y |

15.3.2 Event Enable Register (EXTI_EMR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| EMRy(y=15~0) | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | EMRy(y=15~0) | Event mask on line y 0: Mask event requests from line y 1: Open event request from line y |



15.3.3 Trigger Mode Selection Register (EXTI_TMSR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TMRy(y=15~0) | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | TMRy(y=15~0) | Trigger mode selection on line y 0: Edge trigger 1: Level trigger Note: The edge trigger is one-shot mode. After the trigger, the software clear pending flag is used to clear the flag. The level trigger is level mode. After triggering, the flag can only be cleared by external level change and cannot be cleared by software. |

15.3.4 Rising Edge/High Level Trigger Mode Selection Register (EXTI_RTSTR)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RTRy(y=15~0) | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | RTRy(y=15~0) | Rising edge/high level trigger event configuration bit of line y 0: Rising edge/high level trigger on input line y (interrupt and event) is disabled 1: Rising edge/high level trigger on input line y (interrupt and event) is enabled |



15.3.5 Falling Edge/Low Level Trigger Mode Selection Register (EXTI_FTSTR)

Offset Address: 0x0010
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|--------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| FTRy(y=15~0) RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | FTRy(y=15~0) | Falling edge/low level trigger event configuration bit of line y 0: Falling edge/low level trigger on input line y (interrupt and event) is disabled 1: Falling edge/low level trigger on input line y (interrupt and event) is enabled |

15.3.6 Software Trigger Interrupt Register (EXTI_SWIER)

Offset Address: 0x0014
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SWIERy(y=15~0) WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | SWIERy(y=15~0) | Software interrupt on line y When the bit is '0', writing '1' will set the corresponding pending bit in EXTI_PR. If EXTI_IMR or EXTI_EMR allows interrupt to be generated, an interrupt will be generated at this time. 0: No action 1: Set the corresponding pending bit of EXTI_PR, causing software trigger This signal is automatically cleared by hardware. If the corresponding EXTI_PR is already set, this operation is invalid. |



15.3.7 Pending Register (EXTI_PR)

Offset Address: 0x0018

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| PRCy(y=15~0) | | | | | | | | | | | | | | | |
| WO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PRy(y=15~0) | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | PRCy(y=15~0) | Pending clear bit 0: No action 1: Clear |
| 15 ~ 0 | PRy(y=15~0) | Pending bit 0: No trigger request has occurred 1: Selected trigger request has occurred This bit is set to '1' when a selected trigger interrupt/event occurs on the external interrupt line. Note: For level trigger, if the external level does not change, pending bit will be generated again immediately after clearing the pending bit. |

15.3.8 External Interrupt Configuration Low Register (EXTI_CFGL)

Offset Address: 0x001C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|------------|-----|-----|----------|------------|-----|-----|----------|------------|-----|-----|----------|------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | EXTI7[2:0] | | | Reserved | EXTI6[2:0] | | | Reserved | EXTI5[2:0] | | | Reserved | EXTI4[2:0] | | |
| - | RW | | | - | RW | | | - | RW | | | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | EXTI3[2:0] | | | Reserved | EXTI2[2:0] | | | Reserved | EXTI1[2:0] | | | Reserved | EXTI0[2:0] | | |
| - | RW | | | - | RW | | | - | RW | | | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|--|
| 31 | Reserved | - |
| 30 ~ 28 | EXTI7[2:0] | EXTI 7 configuration These bits are written by software to select the source input for the EXTI7 external interrupt. 000: PA[7] pin 001: PB[7] pin 010: PC[7] pin 011: PD[7] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 27 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|-------------------|--|
| 26 ~ 24 | EXTI6[2:0] | EXTI 6 configuration These bits are written by software to select the source input for the EXTI6 external interrupt. 000: PA[6] pin 001: PB[6] pin 010: PC[6] pin 011: PD[6] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 23 | Reserved | - |
| 22 ~ 20 | EXTI5[2:0] | EXTI 5 configuration These bits are written by software to select the source input for the EXTI5 external interrupt. 000: PA[5] pin 001: PB[5] pin 010: PC[5] pin 011: PD[5] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 19 | Reserved | - |
| 18 ~ 16 | EXTI4[2:0] | EXTI 4 configuration These bits are written by software to select the source input for the EXTI4 external interrupt. 000: PA[4] pin 001: PB[4] pin 010: PC[4] pin 011: PD[4] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 15 | Reserved | - |
| 14 ~ 12 | EXTI3[2:0] | EXTI 3 configuration These bits are written by software to select the source input for the EXTI3 external interrupt. 000: PA[3] pin 001: PB[3] pin 010: PC[3] pin 011: PD[3] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 11 | Reserved | - |
| 10 ~ 8 | EXTI2[2:0] | EXTI 2 configuration These bits are written by software to select the source input for the EXTI2 external interrupt. 000: PA[2] pin 001: PB[2] pin 010: PC[2] pin 011: PD[2] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 7 | Reserved | - |



| | | |
|-------|-------------------|--|
| 6 ~ 4 | EXTI1[2:0] | EXTI 1 configuration These bits are written by software to select the source input for the EXTI1 external interrupt. 000: PA[1] pin 001: PB[1] pin 010: PC[1] pin 011: PD[1] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 3 | Reserved | - |
| 2 ~ 0 | EXTI0[2:0] | EXTI 0 configuration These bits are written by software to select the source input for the EXTI0 external interrupt. 000: PA[0] pin 001: PB[0] pin 010: PC[0] pin 011: PD[0] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |

15.3.9 External Interrupt Configuration High Register (EXTI_CFGH)

Offset Address: 0x0020

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-------------|-----|-----|----------|-------------|-----|-----|----------|-------------|-----|-----|----------|-------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | EXTI15[2:0] | | | Reserved | EXTI14[2:0] | | | Reserved | EXTI13[2:0] | | | Reserved | EXTI12[2:0] | | |
| - | RW | | | - | RW | | | - | RW | | | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-------------|-----|-----|----------|-------------|----|----|----------|------------|----|----|----------|------------|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | EXTI11[2:0] | | | Reserved | EXTI10[2:0] | | | Reserved | EXTI9[2:0] | | | Reserved | EXTI8[2:0] | | |
| - | RW | | | - | RW | | | - | RW | | | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------------|--|
| 31 | Reserved | - |
| 30 ~ 28 | EXTI15[2:0] | EXTI 15 configuration These bits are written by software to select the source input for the EXTI15 external interrupt. 000: PA[15] pin 001: PB[15] pin 010: PC[15] pin 011: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 27 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|--|
| 26 ~ 24 | EXTI14[2:0] | EXTI 14 configuration These bits are written by software to select the source input for the EXTI14 external interrupt. 000: PA[14] pin 001: PB[14] pin 010: PC[14] pin 011: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 23 | Reserved | - |
| 22 ~ 20 | EXTI13[2:0] | EXTI 13 configuration These bits are written by software to select the source input for the EXTI13 external interrupt. 000: PA[13] pin 001: PB[13] pin 010: PC[13] pin 011: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 19 | Reserved | - |
| 18 ~ 16 | EXTI12[2:0] | EXTI 12 configuration These bits are written by software to select the source input for the EXTI12 external interrupt. 000: PA[12] pin 001: PB[12] pin 010: PC[12] pin 011: PD[12] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 15 | Reserved | - |
| 14 ~ 12 | EXTI11[2:0] | EXTI 11 configuration These bits are written by software to select the source input for the EXTI11 external interrupt. 000: PA[11] pin 001: PB[11] pin 010: PC[11] pin 011: PD[11] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 11 | Reserved | - |
| 10 ~ 8 | EXTI10[2:0] | EXTI 10 configuration These bits are written by software to select the source input for the EXTI10 external interrupt. 000: PA[10] pin 001: PB[10] pin 010: PC[10] pin 011: PD[10] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 7 | Reserved | - |



| | | |
|-------|-------------------|--|
| 6 ~ 4 | EXTI9[2:0] | EXTI 9 configuration These bits are written by software to select the source input for the EXTI9 external interrupt. 000: PA[9] pin 001: PB[9] pin 010: PC[9] pin 011: PD[9] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |
| 3 | Reserved | - |
| 2 ~ 0 | EXTI8[2:0] | EXTI 8 configuration These bits are written by software to select the source input for the EXTI8 external interrupt. 000: PA[8] pin 001: PB[8] pin 010: PC[8] pin 011: PD[8] pin 100: Reserved 101: Reserved 110: Reserved 111: Reserved |

15.3.10 EXTI Sample Control Low Register (EXTI_SAMP)

Offset Address: 0x0024

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|-----------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | PS4[1:0] | SN4[1:0] | | |
| | | | | | | | | | | | | <i>RW</i> | <i>RW</i> | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PS3[1:0] | SN3[1:0] | PS2[1:0] | SN2[1:0] | PS1[1:0] | SN1[1:0] | PS0[1:0] | SN0[1:0] | | | | | | | | |
| <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------------|--|
| 31 ~ 20 | Reserved | - |
| 19 ~ 18 | PS4[1:0] | EXTI4 sampling clock prescaler selection bit 00: 1/1 01: 1/4 10: 1/16 11: 1/128 |
| 17 ~ 16 | SN4[1:0] | EXTI4 continuous sample number selection bit 00: 1 01: 2 10: 3 11: 4 |
| 15 ~ 14 | PS3[1:0] | EXTI3 sampling clock prescaler selection bit 00: 1/1 01: 1/4 10: 1/16 11: 1/128 |



| | | |
|---------|-----------------|--|
| 13 ~ 12 | SN3[1:0] | EXTI3 continuous sample number selection bit 00: 1 01: 2 10: 3 11: 4 |
| 11 ~ 10 | PS2[1:0] | EXTI2 sampling clock prescaler selection bit 00: 1/1 01: 1/4 10: 1/16 11: 1/128 |
| 9 ~ 8 | SN2[1:0] | EXTI2 continuous sample number selection bit 00: 1 01: 2 10: 3 11: 4 |
| 7 ~ 6 | PS1[1:0] | EXTI1 sampling clock prescaler selection bit 00: 1/1 01: 1/4 10: 1/16 11: 1/128 |
| 5 ~ 4 | SN1[1:0] | EXTI1 continuous sample number selection bit 00: 1 01: 2 10: 3 11: 4 |
| 3 ~ 2 | PS0[1:0] | EXTI0 sampling clock prescaler selection bit 00: 1/1 01: 1/4 10: 1/16 11: 1/128 |
| 1 ~ 0 | SN0[1:0] | EXTI0 continuous sample number selection bit 00: 1 01: 2 10: 3 11: 4 |

15.3.11 DMA Enable Register (EXTI_DMR)

Offset Address: 0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DMRy(y=15~0) RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | DMRy(y=15~0) | DMA Mask on line y 0: Mask DMA requests from line y 1: Open DMA requests from line y |



16. DMA Controller

16.1 Introduction

Direct memory access (DMA) is used to provide high-speed data transfer between peripherals and memory or between memory and memory. Without CPU intervention, data can be moved quickly through DMA, which saves CPU resources for other operations. The DMA controller has 4 channels, and each channel is dedicated to managing memory access requests from one or more peripherals. There is also an arbiter to coordinate the priority of the CPU and DMA requests.

Note: For a two-way data transmission application, two DMA channels are required to complete sending and receiving respectively.

16.2 Main Features

- 4 independent configurable channels, with each channel being able to support one-way transmitting
- Each channel is directly connected to a dedicated hardware DMA request, and each channel also supports software triggering
- Supports One-Shot and To-End trigger modes, supports software trigger mode
- The basic transmitting unit is burst, and the burst length can be set. When the burst length is 1, it is the same as single transmitting.
- The priority within multiple requests can be set by software programming (which has four levels: very high, high, medium and low), and the access order of the same priority will be processed by rotating.
- The transmitting mode, transmitting bit width, cycle mode, burst length, and pointer increment of data source and data destination can all be independently configured.
- Each channel has 4 event flags (half-transmitting, transmitting completion, burst transmitting end, transmitting error), and any one of these 4 events can generate an interrupt request.
- Both DMA source and destination support memory and peripherals, which means they support memory-to-memory, peripheral-to-memory, memory-to-peripheral, and peripheral-to-peripheral transfers.
- SRAM, APB0, APB1 and AHB peripherals can all be used as the source and destination of access. But flash can only be used as the source, not destination.
- Programmable number of data transmitting is up to 8192 bursts, supporting automatic reloading.
- DMA can be forcibly released for a period of time within each burst to ensure CPU executing efficiency.

Note: when flash is used as DMA destination, there is no restriction on DMA configuration, but DMA write operation is shielded on flash interface, which will not produce effect. DMA error and flash write error will not be generated during operation.

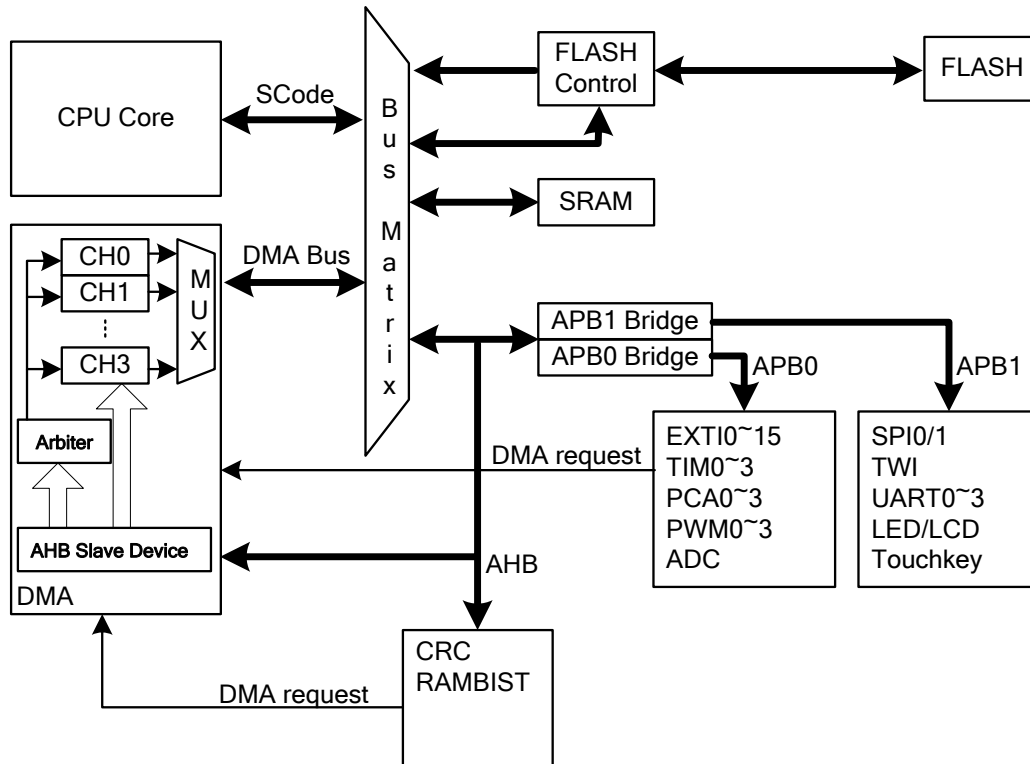


Figure 16-1 DMA Controller logic diagram

Explanation:

(1) DMA supports reading table data from Flash and transferring to SRAM or on-chip peripherals, but it does not support writing data to Flash. The DMA controller does not prohibit this configuration, but it has no effect when writing to Flash

(2)The DMA module is both an AHB master and an AHB slave. The former indicates that DMA and CPU have similar permissions to control the data transmission on the bus matrix, and the latter indicates that DMA as a device module requires parameter configuration through the CPU.

16.3 Function Description

DMA controller and CPU core share a part of system bus. When CPU and DMA access the same destination (RAM or peripheral) at the same time, the DMA request suspends the CPU access to the system bus for several cycles, and the bus arbiter performs cyclic scheduling to ensure that CPU can obtain at least half of the system bus resource. In burst transfer mode, CPU will be paused by DMA during burst transmitting.

16.3.1 DMA Processing

After an event occurs, the peripheral sends a request signal to the DMA controller. The DMA controller responses the request according to the priority of the DMA channel. When the DMA controller begins to process the requesting peripheral, the DMA controller immediately sends back an acknowledge. When an acknowledgment is received from the DMA controller, the peripheral immediately releases its request. Once the peripheral releases the request, the DMA controller also finishes the acknowledge signal at the same time. If there are more requests, the DMA controller can start the next cycle. In summary, each DMA transfer consists of 3 operations:

- Fetch data from the memory address indicated by the source address register. The first source address is the peripheral base address or memory location specified by the DMA_SARx register.
- Store data to memory address indicated by destination address register. The first destination address is the peripheral base address or memory location specified by the DMA_DARx register.
- Perform an Incremental operation of the DMA_CPkTx register, it contains the number of bursts which already be completed.



16.3.2 Arbitrator

The arbitrator has 2 basic functions:

- Responsible for coordinating CPU and DMA controller's occupation of the shared bus;
- Responsible for coordinating the priority processing of multiple DMA channels;

16.3.2.1 Bus occupation mode

DMA occupies the bus mode is called "bus stealing mode", in this case, DMA needs to pause and release the bus for at least one bus cycle for each transfer, and return the bus control to the CPU. The number of cycles to release the bus depends on the BURSTIDLE parameter, with a minimum of 1 and a maximum of 32.

The priority of DMA is higher than that of CPU. When the DMA request arrives, if the CPU is in the unlocked state (the locked state means that the current CPU execution cannot be interrupted), after executing the current instruction, it should respond to the DMA request and release the bus control.

Since the basic DMA transmission unit in SH30F9/SA0 series is a burst, after a burst is started, DMA will occupy bus resources until the burst transfer is completed. The length of the burst is set by BURSTLEN, with a minimum of 1 data and a maximum of 16 data. (Each DMA data may be 1 byte / 2 bytes / 4 bytes).

If the burst length is set to the maximum, it will have a certain impact on the execution efficiency of the CPU. Therefore, large bursts are generally used for insensitive to CPU efficiency, but there are occasions where small pieces of data need to be continuously sent and received quickly.

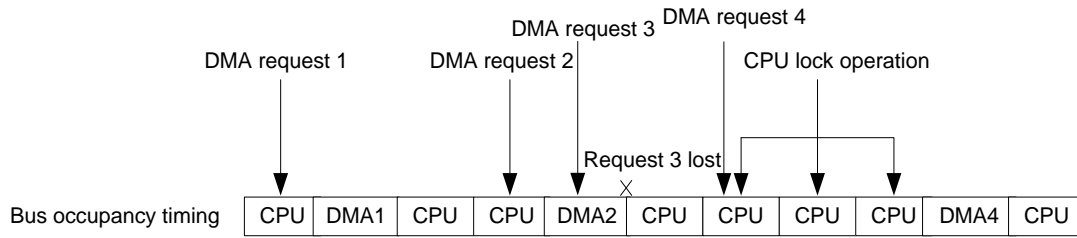


Figure 16-2 Schematic diagram of DMA bus stealing mode (burst length is 1)

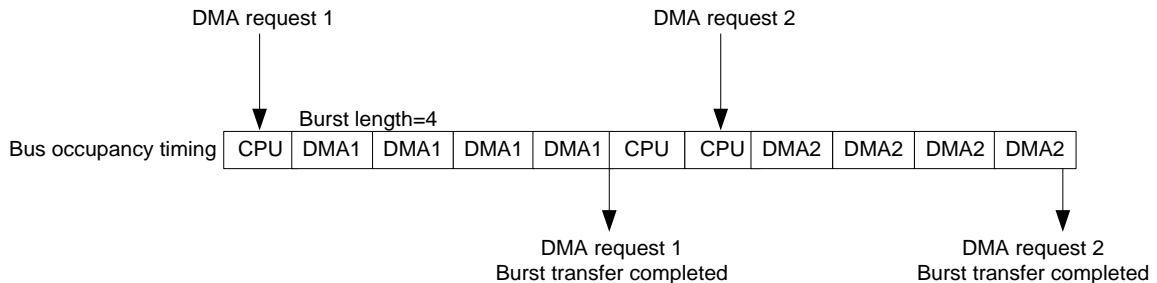


Figure 16-3 DMA Schematic diagram of bus stealing (burst length is 4)

16.3.2.2 Multi-channel priority processing

The access to the device is initiated according to the priority requested by the channel. The priority of each channel can be set in the DMA_CCRx register and has 4 levels:

- Highest priority
- High priority
- Medium priority
- Low priority

High priority always takes precedence over low priority access.



If multiple channels are set with the same priority and requests are made at the same time, the access order is determined according to the channel number, that is, channel 1 is accessed before channel 2, and so on, but if there are already channels that have been accessed, the access sequence will be rotated. That is, after the access to a channel within the same priority is completed, the access sequence is ranked last, and so on, to ensure access opportunities to other channels within the same priority.

For example, channels 1, 2, and 3 are high priority, and 0 is low priority.

- (1) 1, 2 triggers occur simultaneously, 1 >> 2;
- (2) After that, 2 and 3 occur at the same time, because 2 has been transmitted once, so 3 >> 2;
- (3) After 1, 2, 3 occur at the same time, then 1 >> 3 >> 2;
- (4) After 0, 1, 2, 3 occur at the same time, then 1 >> 3 >> 2 >> 0, if 1 occurred before the start of channel 0, then 1 >> 3 >> 2 >> 1 >> 0.

16.3.3 DMA Channel

Each channel can perform DMA transfers between source and destination memory. There's a total of 4 DMA channels, and each channel can be connected to multiple peripherals, but only one peripheral can be selected at a time, the peripheral is selected via the STRMSEL[2:0] field of the DMA_CCRx register.

16.3.3.1 DMA Transfer Quantity(Number of Packet) And Automatic Reloading

It is the number of bursts of DMA transfer which is defined by the DMA transfer quantity register (DMA_NPKTx), indicating the number of (NPKT+1) burst transfers. The minimum setting 0 indicates 1 burst, and the maximum setting 8191 indicates 8192 bursts. DMA_CPKTx is a DMA transfer count register. After each burst transferring, the counter will increase by one until DMA_CPKTx reaches DMA_NPKTx. According to the automatic reload function setting, the DMA will stop transmitting or continue transmitting after reloading.

When RELOADy@DMA_CSR is 0, the auto-reload is turned off. CPKTx will stop transmitting after the count reaches NPKTx, and the DMA enable bit will be automatically cleared. At this time, the counter DMA_CPKTx remains unchanged. If the DMA transmitting completion interrupt is turned on, the DMA interrupt will be requested. DMA_CPKTx is automatically cleared when the next DMA enable bit is turned on.

When RELOADy@DMA_CSR is 1, the auto-reload is turned on, CPKTx is automatically cleared after the count reaches NPKTx, the DMA enable bit remains on state, the DMA transfer source and destination address are restored to the initial value(pointer reset), and the DMA transfer is restarted (uninterrupted). If the DMA transmitting completion interrupt is turned on, the DMA interrupt will also be requested. If the user turns off the RELOAD control bit, the DMA will wait until the current round of reload transmitting is completed before stopping transferring.

Special: When setting To-end trigger mode, the transmission will not start automatically after the auto-reload is turned on, it needs to be retriggered by hardware or software.

Special: When DMA_NPKTx=0, DMA enable is turned on, and the enable will be turned off after once transmitting, and DMA_CPKTx is 0, indicating 1 burst transmitting.

Note: DMA_CPKTx is initialized to 0. This initialization value is meaningless when DMA enable is turned off. After the DMA turns on the transmitting, CPKT+1 is used to indicate the number of burst which actually transferred.

Each burst contains 1 to 16 DMA data, as shown in the BURSTLEN @DMA_CCRx bit setting.

Depending on the SIZE@DMA_CCRx setting, each DMA data may be 1, 2, or 4 bytes.

The DMA transfer quantity register(DMA_NPKTx) can only be set when DMA is turned off. This register becomes read-only when the DMA channel is turned on.

Note: The count units of DMA_NPKTx and DMA_CPKTx are both bursts.

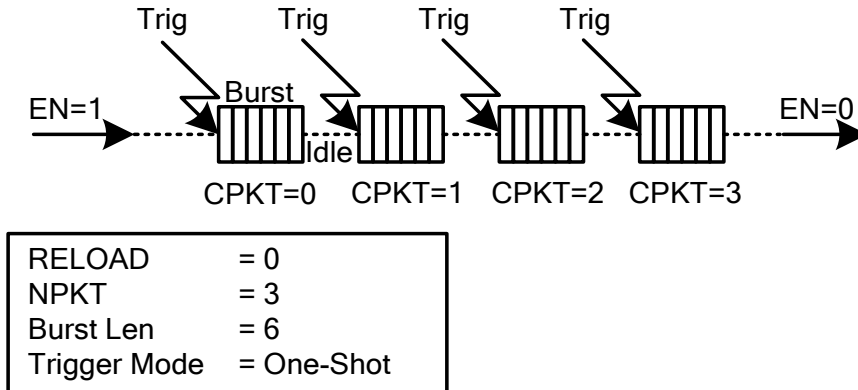


Figure 16-4 Schematic diagram of DMA transmission based on Burst

16.3.3.2 Programmable Data Bit Width

The transmitting data bit width of source end and destination end can be programmed by the SIZE field in the DMA_CCRx register. Depending on the SIZE@DMA_CCRx setting, each DMA data may be 1, 2, or 4 bytes.

16.3.3.3 Pointer Modification Method

By setting SPTYP and DPTYP flag bits in the DMA_CCRx register, the internal current address pointers (which cannot be read) of source end and destination end can selectively complete automatic modify after each transfer. DMA supports the pointer INC, DEC, FIX, and WRAP modes, of which the WRAP mode is used for the cycle mode. The pointer modification method is all set by software.

When set to INC mode, the next address to be transmitted will be the previous address plus the increment value, the increment value depends on the selected data width, and can be 1, 2 or 4.

DMA_SARx/DMA_DARx stores the initial pointer of the source/destination address and remains unchanged during DMA transferring. The software cannot get the current accessing address because of internal address attribute.

Note: If the user wants to know the current internal pointer position, it can only be calculated indirectly through (initial pointer + burst transmission count x burst length x SIZE parameter).

16.3.3.4 Cycle Mode

Cycle mode is used to process circular buffer and continuous data transfers (such as the continuous conversion mode of ADC).

The cycle period is fixed to the burst length. After the cycle transfer is completed, its address pointer will automatically return to the initial value, and need not to change the pointer by software.

When setting SPTYP/DPTYP=WRAP method, the corresponding cycle mode is enabled.

Various settings for the DMA cycle mode are given in the table below.

Table 16-1 DMA Cycle Mode Setting Table

| Cycle Mode | Setting | Function Description |
|-------------------|---|---|
| Non Cyclic mode | SPTYP=00/01/10,DPTYP=00/01/10 | Non cyclic mode. Requires software for pointer operations. |
| Single cycle mode | Source end cycle: SPTYP=11,DPTYP=00/01/10 Destination end cycle: SPTYP=00/01/10,DPTYP=11 | Single cycle mode. For example, receiving multi-channel ADC sampling data, the loop length is set according to the number of channels in the ADC data register end, but the data buffer end pointer also needs to be processed by software. |
| Double cycle mode | SPTYP=11,DPTYP=11 | Double cycle mode. Simplify CPU pointer operation at both ends. |

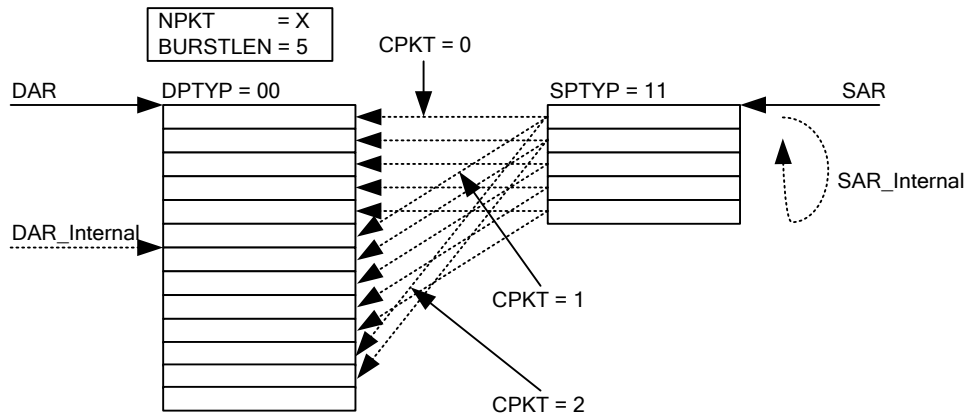


Figure 16-5 Single cycle mode

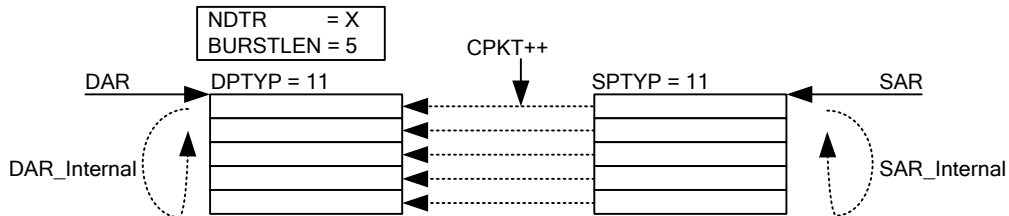


Figure 16-6 Double cycle mode

16.3.3.5 Software Trigger Method

The operation of the DMA channel can be performed without a peripheral request, which is software trigger method.

The software trigger method is generally used for data transfer between memory and memory, but is not limited thereto. When the SWTRGy bit in the DMA_CSR register is set, a software trigger is initiated with the same effect as the pulse trigger.

When doing memory-to-memory data movement, need be aware that Flash can only be read as a source and cannot be written.

16.3.3.6 Single transmitting and burst transmitting

Single transmitting is a special case of burst transmitting, and it will be a single transmitting when the burst length is 1.

Single transfer is the basic transfer method of DMA. The DMA arbitration strategy can ensure that CPU can obtain at least half slot of running time, thus achieving an ideal balance between ensuring CPU running time and liberating the load of intervention from CPU to peripheral, which helps to improve system operation efficiency.

Burst transmitting (which means the burst length greater than 1) is suitable for fast movement of small block data area, as described in the "Arbiter" part.

16.3.3.7 One-shot Trigger and To-end Trigger Method

Only one DMA burst transfer is done at a time with the One-shot trigger method, and with the To-end trigger method, it is only stopped after the whole transfers appointed by the DMA_NPKTx register are completed.

There is also an IDLE time slot for CPU to run between the bursts before and after during transmitting with the To-end trigger.



16.3.3.8 BUSY Signal

The DMA channel will set the busy signal during transmitting. The user can determine which channel is currently transmitting by querying the busy signal of each channel. The BUSY signal only indicates one burst transmitting during One-shot trigger, and will indicate whole transmitting during To-end trigger, means not disappear between burst and burst.

Note: The busy signal only indicates that burst is being transmitted, and busy returns to idle during burst idle.

16.3.3.9 Burst Idle Setting

After DMA completes the transmitting of a burst, it releases the bus to CPU. If there are multiple DMA channels turned on, DMA controller need wait until the current channel's Burst Idle slot is finished, then can execute the next DMA channel, which means the CPU execution time will not be preempted by the queueing DMA. This design ensures the execution efficiency of CPU.

16.3.3.10 DMA Channel Configuration Progress

The following is the process of configuring DMA channel x (x represents the channel number, using non-cyclic single transmitting single trigger mode description):

1. Set the first address of the source data area in the DMA_SARx register.
2. Set the first address of the destination data area in the DMA_DARx register.
3. Set the amount of data to be transfered in the DMA_NPKTx register.
4. Set the DMA channel mapping table STRMSEL[2:0] in the DMA_CCRx register. Only one peripheral can be selected for each DMA channel.
5. Set the burst length, trigger method, pointer modification method, data bit width, channel priority, and interrupt enable in the DMA_CCRx register.
6. Set the ENABLE bit in the DMA_CCRx register and start the channel.

Once the DMA channel is enabled, it can respond to DMA requests (note) from the memory or peripherals connected to that channel.

The half transfer flag (HTIF) is set to 1 when half of the data has been transferred, and an interrupt request is generated when the half transfer interrupt enable bit (HTIE) is set. After the end of the data transfer, the transfer completion flag (TCIF) is set to 1. When the transfer is complete interrupt enable bit (TCIE) is set, an interrupt request will be generated.

Note: The DMA request enable for related peripherals can be turned on or off independently by setting the control bits in the corresponding peripheral registers (in the corresponding peripheral part). If the corresponding DMA channel is configured and turned on, but the DMA request of the corresponding peripheral is not turned on, the hardware will not work, and the DMA transfer will not occur.

16.3.4 Source and Destination Data Transfer Format

When SSIZE and DSIZE are not the same, the DMA module performs data transfer according to the format of the following table.

Table 16-2 DMA Source and Destination Data Transfer Format Table (When SPTYP=DPTYP=00)

| Source End Width | Destination Width | Transfer Quantity | Source: Address/Data | The transfer operation | Destination: Address/Data |
|------------------|-------------------|-------------------|--|--|--|
| 8 | 8 | 4 | 0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3 | 1: Read B0[7:0] in 0x0, Write B0[7:0] on the 0x0 2: Read B1[7:0] in 0x1, Write B1[7:0] on the 0x1 3: Read B2[7:0] in 0x2, Write B2[7:0] on the 0x2 4: Read B3[7:0] in 0x3, Write B3[7:0] on the 0x3 | 0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3 |
| 8 | 16 | 4 | 0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3 | 1: Read B0[7:0] in 0x0, Write 00B0[15:0] on the 0x0 2: Read B1[7:0] in 0x1, Write 00B1[15:0] on the 0x2 3: Read B2[7:0] in 0x2, Write 00B2[15:0] on the 0x4 4: Read B3[7:0] in 0x3, Write 00B3[15:0] on the 0x6 | 0x0 / 00B0 0x2 / 00B1 0x4 / 00B2 0x6 / 00B3 |



SH30F9/SA0 Series

| | | | | | |
|----|----|---|--|--|--|
| 8 | 32 | 4 | 0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3 | 1: Read B0[7:0] in 0x0, Write 000000B0[31:0] on the 0x0 2: Read B1[7:0] in 0x1, Write 000000B1[31:0] on the 0x4 3: Read B2[7:0] in 0x2, Write 000000B2[31:0] on the 0x8 4: Read B3[7:0] in 0x3, Write 000000B3[31:0] on the 0xC | 0x0 / 000000B0 0x4 / 000000B1 0x8 / 000000B2 0xC / 000000B3 |
| 16 | 8 | 4 | 0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6 | 1: Read B1B0[15:0] in 0x0, Write B0[7:0] on the 0x0 2: Read B3B2[15:0] in 0x2, Write B2[7:0] on the 0x1 3: Read B5B4[15:0] in 0x4, Write B4[7:0] on the 0x2 4: Read B7B6[15:0] in 0x6, Write B6[7:0] on the 0x3 | 0x0 / B0 0x1 / B2 0x2 / B4 0x3 / B6 |
| 16 | 16 | 4 | 0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6 | 1: Read B1B0[15:0] in 0x0, Write B1B0[15:0] on the 0x0 2: Read B3B2[15:0] in 0x2, Write B3B2[15:0] on the 0x2 3: Read B5B4[15:0] in 0x4, Write B5B4[15:0] on the 0x4 4: Read B7B6[15:0] in 0x6, Write B7B6[15:0] on the 0x6 | 0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6 |
| 16 | 32 | 4 | 0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6 | 1: Read B1B0[15:0] in 0x0, Write 0000B1B0[31:0] on the 0x0 2: Read B3B2[15:0] in 0x2, Write 0000B3B1[31:0] on the 0x4 3: Read B5B4[15:0] in 0x4, Write 0000B5B2[31:0] on the 0x8 4: Read B7B6[15:0] in 0x6, Write 0000B7B3[31:0] on the 0xC | 0x0 / 0000B1B0 0x4 / 0000B3B2 0x8 / 0000B5B4 0xC / 0000B7B6 |
| 32 | 8 | 4 | 0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC | 1: Read B3B2B1B0[31:0] in 0x0, Write B0[7:0] on the 0x0 2: Read B7B6B5B4[31:0] in 0x4, Write B4[7:0] on the 0x1 3: Read BBBAB9B8[31:0] in 0x8, Write B8[7:0] on the 0x2 4: Read BFBEBDBC[31:0] in 0xC, Write BC[7:0] on the 0x3 | 0x0 / B0 0x1 / B4 0x2 / B8 0x3 / BC |
| 32 | 16 | 4 | 0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC | 1: Read B3B2B1B0[31:0] in 0x0, Write B1B0[15:0] on the 0x0 2: Read B7B6B5B4[31:0] in 0x4, Write B5B4[15:0] on the 0x2 3: Read BBBAB9B8[31:0] in 0x8, Write B9B8[15:0] on the 0x4 4: Read BFBEBDBC[31:0] in 0xC, Write BDBC[15:0] on the 0x6 | 0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC |
| 32 | 32 | 4 | 0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC | 1: Read B3B2B1B0[31:0] in 0x0, Write B3B2B1B0[31:0] on the 0x0 2: Read B7B6B5B4[31:0] in 0x4, Write B7B6B5B4[15:0] on the 0x4 3: Read BBBAB9B8[31:0] in 0x8, Write BBBAB9B8[15:0] on the 0x8 4: Read BFBEBDBC[31:0] in 0xC, Write BFBEBDBC[15:0] on the 0xC | 0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC |



16.3.4.1 Operate an AHB device that does not support byte or halfword writing

When the DMA module starts a byte or halfword write operation of an AHB device, the data will be repeated in the unused part of the HWDATA [31: 0] bus. Therefore, if the DMA writes bytes or halfwords to an AHB device that does not support byte or halfword write operations (that is, HSIZE is not suitable for the module), no error will occur, and the DMA will write to 32 bits according to the following two examples HWDATA data:

- When HSIZE = halfword, write halfword '0xABCD', DMA will set HWDATA bus to '0xABCDABCD'.
- When HSIZE = byte, write byte '0xAB', DMA will set HWDATA bus to '0xABABABAB'.

Assuming that the AHB / APB bridge is an AHB 32-bit slave device, it does not process the HSIZE parameter, it will transfer any bytes or halfwords on the AHB to the APB in 32 bits as follows:

- An AHB operation on the write byte data '0xB0' at address 0x0 (or 0x1, 0x2, or 0x3) will be converted to an APB operation on the write data '0xB0B0B0B0' at address 0x0.
- An AHB operation on the write halfword data '0xB1B0' at address 0x0 (or 0x2) will be converted to an APB operation on the write data '0xB1B0B1B0' at address 0x0.

16.3.5 Error Management

Reading and writing a reserved address area will result in a DMA transfer error, including

- (1) Write to the SRAM area (when SRAM is set to Read Only)
- (2) Read and write the Cortex-M0+ private peripheral area
- (3) Other HardFault^{Note} that occur during DMA read and write

When DMA transfer error occurs during a DMA read/write operation, the hardware automatically clears the EN bit corresponding to the channel which occurs the DMA error, and the operation on that channel will be stopped (note: need to wait for the current Burst transfer to complete). At this time, the transmitting error interrupt flag bit (TEIF_y) corresponding to the channel in the DMA_IFSR register will be set. If the transmitting error interrupt enable bit is set in the DMA_CCR_x register, an interrupt will be generated.

Note: When the DMA reads or writes the reserved address area or illegally reads or writes, a HardFault is generated. This exception will be sent to the DMA controller to generate a DMA error.

16.3.6 DMA Interrupt

Each DMA channel has four kinds of transmitting events, and interrupt can be generated when a transmitting event occurs. The four kinds are burst transmitting end (BE), half transmitting (HT), transmitting completion (TC) and transmitting error (TE). Setting the relevant control bits in the DMA configuration register CCR_x enables the associated interrupt event. An interrupt request will be generated if any of the four events BE, HT, TC, or TE occurs.

In the interrupt vector table, the DMA module occupies two entries, which are DMA_CH0~1, and DMA_CH2~3.

Table 16-3 DMA Interrupt Request

| Interrupt Event | Event Flag Bit | Enable Control Bit |
|-------------------------|----------------|--------------------|
| Burst transmitting end | BEIF | BEIE |
| Half transmitting | HTIF | HTIE |
| transmitting Completion | TCIF | TCIE |
| transmitting Error | TEIF | TEIE |



16.3.7 DMA Request Image

Requests from peripherals (including PCAx(x=0~3), PWMx(x=0~3), ADC, UARTx(x=0~3), SPIx(x=0, 1), EXTIx(x=0~15), TIMx(x=0~3), LED), will enter the arbiter through the input source selector (STRMSEL[2:0]), which means that only one request can be valid at the same time for each DMA channel. See the DMA request image in the figure below.

Table 16-4 List of DMA Requests for Each Channel

| DMA input source STRMSEL[2:0] | Peripheral Type | DMA Channel 0 | DMA Channel 1 | DMA Channel 2 | DMA Channel 3 |
|----------------------------------|--------------------|------------------|------------------|------------------|------------------|
| stream0 | UART | UART0_TX | UART0_RX | UART1_TX | UART1_RX |
| stream1 | SPI | SPI0_TX | SPI0_RX | SPI1_TX | SPI1_RX |
| stream2 | EXTI | EXTI0/1/8/9 | EXTI2/3/10/11 | EXTI4/5/12/13 | EXTI6/7/14/15 |
| stream3 | TIM | TIM0 | TIM1 | TIM2 | TIM3 |
| stream4 | UART | UART2_RX | UART2_TX | UART3_RX | UART3_TX |
| stream5 | PCA | PCA0 | PCA1 | PCA2 | PCA3 |
| stream6 | ADC/LED/TK | LED/TK | LED/TK | ADC/TK | ADC/TK |
| stream7 | PWM | PWM0 | PWM1 | PWM2 | PWM3 |

Note: DMA supports the sending and receiving of GPIO, but the trigger signal needs to be generated by other peripherals such as EXTI and TIM. At this time, the GPIO module needs to be mounted on the AHB bus through Bus@SYSCFG_GPIOBKR of the configuration register.

16.3.8 Precautions

Before resetting the DMA trigger source module, temporarily close the corresponding DMA channel and reopen it after the reset is completed. For example, when an external interrupt triggers DMA, the external interrupt module is reset (i.e. set EXTIRST@RCC_APB0RSTR) previously, it was necessary to close the DMA channel corresponding to the external interrupt and wait for the external interrupt module to reset before opening it.

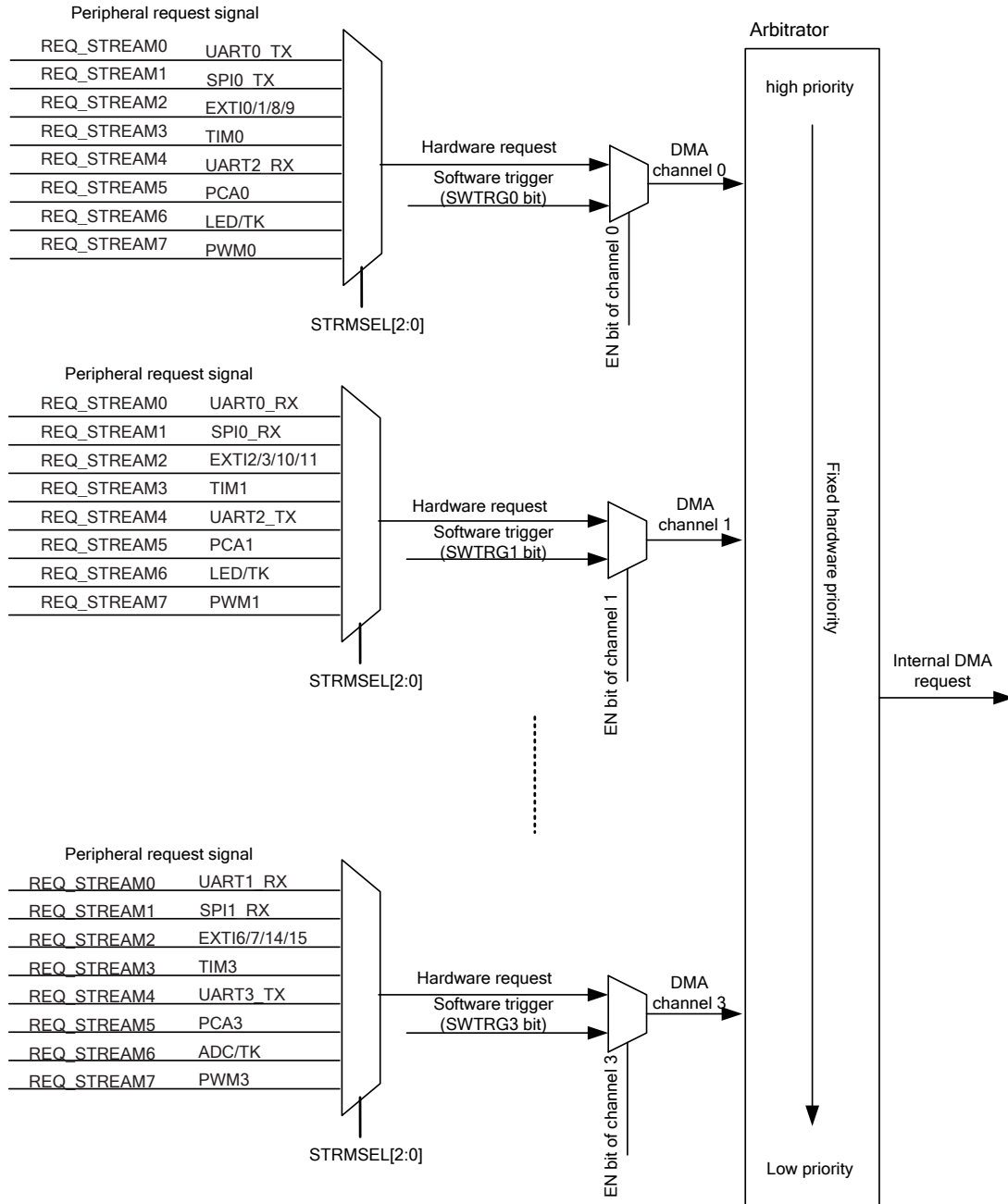


Figure 16-7 DMA request image



16.4 Registers

DMA Module Register list (Base Address:0x4004 1800)

| Address | Register | Description |
|-------------|----------|--|
| 0x4004 1800 | IFSR | DMA Interrupt Status Register |
| 0x4004 1804 | IFCR | DMA Interrupt Flag Clear Register |
| 0x4004 1808 | CSR | DMA Control Status Register |
| 0x4004 1810 | CCR0 | DMA Channel 0 Configuration Register |
| 0x4004 1814 | NPKT0 | DMA Channel 0 Transmitting Quantity Register |
| 0x4004 1818 | CPKT0 | DMA Channel 0 Transmitting Count Register |
| 0x4004 181C | SAR0 | DMA Channel 0 Source Address Register |
| 0x4004 1820 | DAR0 | DMA Channel 0 Target Address Register |
| 0x4004 1830 | CCR1 | DMA Channel 1 Configuration Register |
| 0x4004 1834 | NPKT1 | DMA Channel 1 Transmitting Quantity Register |
| 0x4004 1838 | CPKT1 | DMA Channel 1 Transmitting Count Register |
| 0x4004 183C | SAR1 | DMA Channel 1 Source Address Register |
| 0x4004 1840 | DAR1 | DMA Channel 1 Target Address Register |
| 0x4004 1850 | CCR2 | DMA Channel 2 Configuration Register |
| 0x4004 1854 | NPKT2 | DMA Channel 2 Transmitting Quantity Register |
| 0x4004 1858 | CPKT2 | DMA Channel 2 Transmitting Count Register |
| 0x4004 185C | SAR2 | DMA Channel 2 Source Address Register |
| 0x4004 1860 | DAR2 | DMA Channel 2 Target Address Register |
| 0x4004 1870 | CCR3 | DMA Channel 3 Configuration Register |
| 0x4004 1874 | NPKT3 | DMA Channel 3 Transmitting Quantity Register |
| 0x4004 1878 | CPKT3 | DMA Channel 3 Transmitting Count Register |
| 0x4004 187C | SAR3 | DMA Channel 3 Source Address Register |
| 0x4004 1880 | DAR3 | DMA Channel 3 Target Address Register |

16.4.1 DMA Interrupt Status Register (DMA_IFSR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|--------------|-----|-----|-----|----------|-----|-----|-----|--------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | TEIFy(y=3~0) | | | | Reserved | | | | HTIFy(y=3~0) | | | |
| - | | | | RO | | | | - | | | | RO | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | TCIFy(y=3~0) | | | | Reserved | | | | BEIFy(y=3~0) | | | |
| - | | | | RO | | | | - | | | | RO | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 28 | Reserved | - |
| 27 ~ 24 | TEIFy(y=3~0) | The transmitting error flag of channel y, set by hardware, corresponding flag can be cleared by writing '1' to the corresponding bit in the DMA_IFCR register. 0: There's no transmitting error (TE) event in channel y 1: There's transmitting error (TE) event generated in channel y |
| 23 ~ 20 | Reserved | - |
| 19 ~ 16 | HTIFy(y=3~0) | The half transmitting flag of channel y, set by hardware, corresponding flag can be cleared by writing '1' to the corresponding bit in the DMA_IFCR register. 0: There's no half transmitting (HT) event in channel y 1: There's half transmitting (HT) event generated in channel y |
| 15 ~ 12 | Reserved | - |
| 11 ~ 8 | TCIFy(y=3~0) | The transmitting completion flag of channel y, set by hardware, corresponding flag can be cleared by writing '1' to the corresponding bit in the DMA_IFCR register. 0: There's no transmitting completion (TC) event in channel y 1: There's transmitting completion (TC) event generated in channel y |
| 7 ~ 4 | Reserved | - |
| 3 ~ 0 | BEIFy(y=3~0) | The block transmitting end interrupt flag of channel y, set by hardware, corresponding flag can be cleared by writing '1' to the corresponding bit in the DMA_IFCR register. 0: There's no block transmitting end (BE) event in channel y 1: There's block transmitting end (BE) event generated in channel y |

16.4.2 DMA Interrupt Flag Clear Register (DMA_IFCR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|---------------|-----|-----|-----|----------|-----|-----|-----|---------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | CTEIFy(y=3~0) | | | | Reserved | | | | CHTIFy(y=3~0) | | | |
| - | | | | WO | | | | - | | | | WO | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | CTCIFy(y=3~0) | | | | Reserved | | | | CBEIFy(y=3~0) | | | |
| - | | | | WO | | | | - | | | | WO | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 28 | Reserved | - |
| 27 ~ 24 | CTEIFy(y=3~0) | Clear transmitting error flag of channel y, cleared by software 0: No effect 1: Clears the corresponding TEIF flag in the DMA_ISR register |
| 23 ~ 20 | Reserved | - |
| 19 ~ 16 | CHTIFy(y=3~0) | Clear half transmitting flag of channel y, cleared by software 0: No effect 1: Clears the corresponding HTIF flag in the DMA_ISR register |
| 15 ~ 12 | Reserved | - |
| 11 ~ 8 | CTCIFy(y=3~0) | Clear transmitting completion flag of channel y, cleared by software 0: No effect 1: Clears the corresponding TCIF flag in the DMA_ISR register |
| 7 ~ 4 | Reserved | - |
| 3 ~ 0 | CBEIFy(y=3~0) | Clear block transmitting completion interrupt flag of channel y, cleared by software 0: No effect 1: Clears the corresponding BEIF flag in the DMA_ISR register |



SH30F9/SA0 Series

16.4.3 DMA Control Status Register (DMA_CSR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|----------------|-----|-----|-----|----------|-----|-----|-----|----------------|-----|-----|-----|
| Reserved | | | | RELOADy(y=3~0) | | | | Reserved | | | | BURSTIDLE[3:0] | | | |
| - | | | | RW | | | | - | | | | RW | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----------|-----|-----|-----|---------------|-----|----|----|----------|----|----|----|---------------|----|----|----|
| Reserved | | | | DBUSYy(y=3~0) | | | | Reserved | | | | SWTRGy(y=3~0) | | | |
| - | | | | RO | | | | - | | | | RW1s | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------|---|
| 31 ~ 28 | Reserved | - |
| 27 ~ 24 | RELOADy(y=3~0) | Reload control bit of DMA channel y, this bit is set and cleared by software 0: Does not automatically reload 1: Automatically reloads |
| 23 ~ 20 | Reserved | - |
| 19 ~ 16 | BURSTIDLE[3:0] | Release cycle setting bit after DMA burst transfer, this bit is set and cleared by software 0000: Releases 1 cycle 0001: Releases 2 cycles 0010: Releases 3 cycles 0011: Releases 4 cycles 0100: Releases 5 cycles 0101: Releases 6 cycles 0110: Releases 7 cycles 0111: Releases 8 cycles 1000: Releases 9 cycles 1001: Releases 10 cycles 1010: Releases 12 cycles 1011: Releases 16 cycles 1100: Releases 20 cycles 1101: Releases 24 cycles 1110: Releases 28 cycles 1111: Releases 32 cycles |
| 15 ~ 12 | Reserved | - |
| 11 ~ 8 | DBUSYy(y=3~0) | The status of DMA channel y, set and cleared by hardware, including software and hardware trigger status 0: Idle 1: Busy |
| 7 ~ 4 | Reserved | - |
| 3 ~ 0 | SWTRGy(y=3~0) | Software trigger of DMA channel y, this bit is set by software and cleared by hardware 0: There's no software trigger, can read 0, writing 0 is meaningless 1: Sets software trigger, automatically cleared after DMA response |



SH30F9/SA0 Series

16.4.4 DMA Channel n Configuration Register (DMA_CCRn)(n=0..3)

Offset Address: 0x0010
 :0x0030
 :0x0050
 :0x0070

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----------------|--------------|-----|-----|---------------|-----|-----|-----|
| Reserved | | | | | | | | TRG MOD E | STRMSEL[2:0] | | | BURSTLEN[3:0] | | | |
| - | | | | | | | | RW | RW | | | RW | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------|-----|------------|-----|------------|-----|-----------|----|-----------|----|--------------|------|------|------|------|----|
| PL[1:0] | | SSIZE[1:0] | | DSIZE[1:0] | | SPTY[1:0] | | DPTY[1:0] | | Rese rved | TEIE | BEIE | HTIE | TCIE | EN |
| RW | | RW | | RW | | RW | | RW | | - | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | Reserved | - |
| 23 | TRGMODE | DMA trigger mode selection bit, this bit is set and cleared by software 0: One-shot trigger method 1: To-end trigger method |
| 22 ~ 20 | STRMSEL[2:0] | DMA channel input source select control bits, these bits are set and cleared by software 000: Selects stream0 input 001: Selects stream1 input 111: Selects stream7 input |
| 19 ~ 16 | BURSTLEN[3:0] | DMA burst length setting bit, this bit is set and cleared by software 0000: Each burst contains 1 data (i.e. single transmitting) 0001: Each burst contains 2 data ... 1111: Each burst contains 16 data Note: Depending on the SIZE setting, 1 data transfer may contain 1 byte, 2 bytes or 4 bytes. |
| 15 ~ 14 | PL[1:0] | Channel priority, these bits are set and cleared by software 00: Low 01: Medium 10: High 11: Highest |
| 13 ~ 12 | SSIZE[1:0] | Source end data width, these bits are set and cleared by software 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved |
| 11 ~ 10 | DSIZE[1:0] | Destination end data width, these bits are set and cleared by software 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved |



| | | |
|-------|-------------------|---|
| 9 ~ 8 | SPTYP[1:0] | Source end pointer modification method, this bit is set and cleared by software 00: Increment 01: Decrement 10: Fixed 11: Cycle (incremental cycle) |
| 7 ~ 6 | DPTYP[1:0] | Destination end pointer modification method, this bit is set and cleared by software 00: Increment 01: Decrement 10: Fixed 11: Cycle (incremental cycle) |
| 5 | Reserved | - |
| 4 | TEIE | Allows transmitting error interrupt, this bit is set and cleared by software 0: TE interrupt is disabled 1: TE interrupt is enabled |
| 3 | BEIE | Allows burst transmitting end interrupt, this bit is set and cleared by software 0: BE interrupt is disabled 1: BE interrupt is enabled |
| 2 | HTIE | Allows half transmitting interrupt, this bit is set and cleared by software 0: HT interrupt is disabled 1: HT interrupt is enabled |
| 1 | TCIE | Allows transmitting completion interrupt, this bit is set and cleared by software 0: TC interrupt is disabled 1: TC interrupt is enabled |
| 0 | EN | Channel enable control bit, set and cleared by software 0: channel is closed 1: channel is open |

16.4.5 DMA Channel n Transmitting Quantity Register (DMA_NPKTn)(n=0..3)

Offset Address: 0x0014

: 0x0034

: 0x0054

: 0x0074

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|------------|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | NPKT[12:0] | | | | | | | | | | | | |
| - | | | RW | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|--|
| 31 ~ 13 | Reserved | - |
| 12 ~ 0 | NPKT[12:0] | Channel n data transmitting quantity (burst number) The quantity of data transmitting is NPKT, and the setting range is 0 to 8191, indicating that the quantity of transmitting bursts is from 1 to 8192. This register can only be written when the channel is not working (EN = 0 for DMA_CCRx). This register becomes read only after the channel is turned on. |



16.4.6 DMA Channel n Transmitting Count Register (DMA_CPKTn)(n=0..3)

Offset Address: 0x0018
 :0x0038
 :0x0058
 :0x0078

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|------------|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | CPKT[12:0] | | | | | | | | | | | | |
| - | | | RO | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 13 | Reserved | - |
| 12 ~ 0 | CPKT[12:0] | Channel n data transmitting count (burst number) This register is used to indicate the number of DMA transfers, read only. After the DMA channel is turned on, the counter is automatically cleared and starts counting according to the DMA transfer. When the number of NPkTs is counted, the DMA transfer is stopped, the DMA channel enable bit is cleared, and the count value is kept unchanged. This register is set from 0 to 8191, indicating that the number of transmitting bursts is from 1 to 8192. |

16.4.7 DMA Channel n Source Address Register (DMA_SARn)(n=0..3)

Offset Address: 0x001C
 :0x003C
 :0x005C
 :0x007C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| SAR[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SAR[31:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

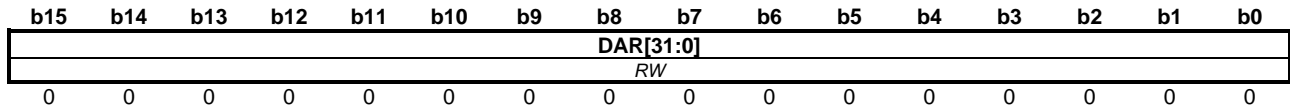
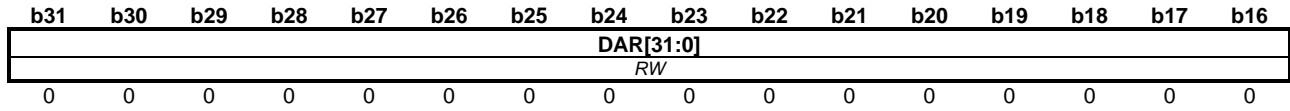
| Bit | Symbol | Description |
|--------|-----------|---|
| 31 ~ 0 | SAR[31:0] | DMA source address register Base address of the source data register When SSIZE = '01' (16 bits), the SAR[0] bit is not used. The operation is automatically aligned with the half-word address. When SSIZE = '10' (32 bits), the SAR[1:0] bits are not used. The operation is automatically aligned with the word address. This register cannot be written when the channel is turned on (EN=1 of DMA_CCRx) |



16.4.8 DMA Channel n Target Address Register (DMA_DARn)(n=0..3)

Offset Address: 0x0020
 :0x0040
 :0x0060
 :0x0080

Reset value: 0x0000 0000



| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | DAR[31:0] | <p>DMA destination address register</p> <p>Base address of the data destination register</p> <p>When DSIZE = '01' (16 bits), the DAR[0] bit is not used. The operation is automatically aligned with the half-word address.</p> <p>When DSIZE = '10' (32 bits), the DAR[1:0] bits are not used. The operation is automatically aligned with the word address.</p> <p>This register cannot be written when the channel is turned on (EN=1 of DMA_CCRx)</p> |



17. LED Driver (constant current)

17.1 Overview

The SH30F9/SA0 series contains a LED driver, with 8 Common signal pins and 16 segment driver pins

- (1) Support 8*8 Row and column Matrix Drive
- (2) LED Serial lattice drive Support 7*8、6*7、5*6、4*5 and so on
- (3) Maximum support lattice: 8*8+7*8
- (4) Support constant current drive (8mA-32mA adjustable)
- (5) Supports two modes of operation

Model 1: Single scan

Select the COM (single COM) to be scanned by the software and start scanning. When the COM is scanned, it means that the scan is over. During COM scan, each bit in the LED_SEGMOD register controls a SEG. When the bit is 0, the state of the SEG port is controlled by IO. When the bit is 1, the SEG port outputs constant current.

At the end of this scan, The LED driver corresponds to the interrupt flag at COMIF position 1, the LED stops scanning.

The output width of SEG can be controlled in two ways.

Mode 1: The output cycle width of SEG is the same as that of COM (when the MODE bit in the LED_COMCR register is 0)

Mode 2: The output cycle width of SEG is controlled by the SEG width register (when the MODE bit 1 in the LED_COMCR register)

Model 2: Continuous scanning

When the LED works in continuous scanning mode, the LED_SEGMOD register is used to select the scanned SEG. When $SxMS@LED_SEGMOD = 1$, the bit in the LED_SORX register controls the corresponding SEGX output. When the bit is 0, the state of the SEG port is controlled by IO. When the bit is 1, the SEG port outputs constant current. When $SEGX=0$, the corresponding port of the SEGX is IO port and is not controlled by the corresponding SEGX in the LED_SORX register. At the end of the LED frame scan, the interrupt flag bit AU_IF corresponding to the LED driver marks position 1. The output cycle width of the continuous scan mode SEG is the same as that of COM.

LED drivers support constant current drivers

The LED driver supports constant current drive and can be adjusted from 8mA to 32mA. There are 8 constant-current sources in the LED, and one constant-current source is shared for every two SEGs. Seg0 and Seg8 share constant current source 0, Seg1 and Seg9 share constant current source 1, Seg2 and Seg10 share constant current source 2, Seg3 and Seg11 share constant current source 3, Seg4 and Seg12 share constant current source 4, Seg5 and Seg13 share constant current source 5, Seg6 and Seg14 share constant current source 6, Seg7 and Seg15 share constant current source 7. When the constant current source is closed, the corresponding SEG port function of the constant current source is IO.

Effective output of LED driver COM/SEG

COM's effective output level is low, and SEG's effective output is constant current.

In Sleep mode, the LED driver remains working, and in Stop mode, the LED driver is off.

When the constant current source is closed, the port function corresponding to the constant current source is IO

For example:

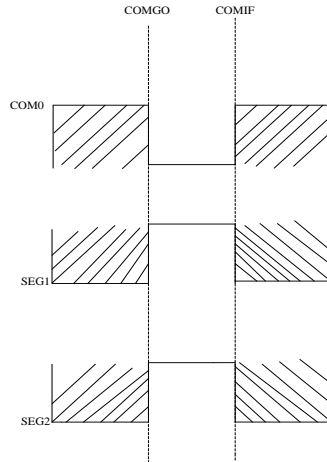
Single scan:

- 1.The COMSEL[2:0] bit in the LED_COMCR register is used to select the COM currently scanned. Configure COMSEL[2:0]=001, that is, the COM currently scanned is COM1;
- 2.Configure the DISCOM[7:0] register, set the width of each COM cycle, TB is the LED single COM scan width, TBUS is the bus clock width, $TB = TBUS \times 256 \times (DISCOM[7:0]+1)$
- 3.Configure the output of SEG. SEG output is controlled by the LED_SEGMOD register. LED_SEGMOD corresponding bit is 1, SEG output constant current, otherwise SEG output state is controlled by IO.
4. SEG output width configuration, when the Mode bit in the LED_COMCR register is 0, the width of SEG and COM is the same; When the MODE bit in the LED_COMCR register is 1, the width of the SEG is controlled by the SEG width register
5. COMGO set 1 to start scanning. After scanning, COMIF set 1 to stop scanning.

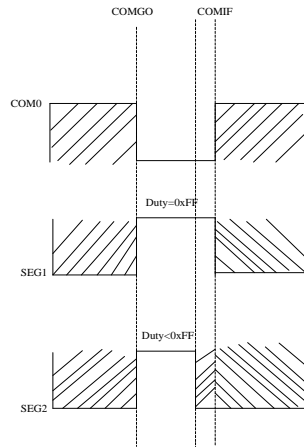


SH30F9/SA0 Series

Example 1: (SEG is the same width as COM)

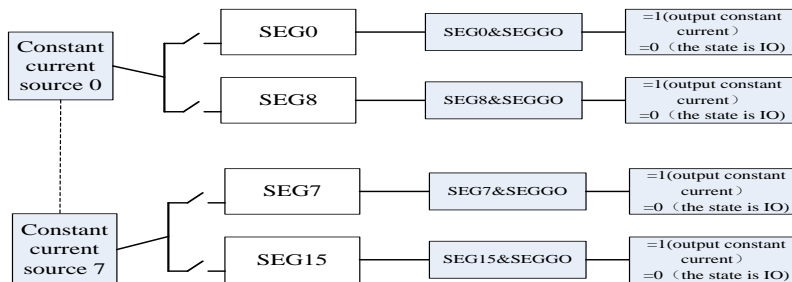


Example 2: (The width of the SEG is controlled by the SEG width register)



- Description:
1. Low represents the output low level
 2. High represents the output constant current
 3. Shaded parts represent IO states

Single scan constant current control diagram:



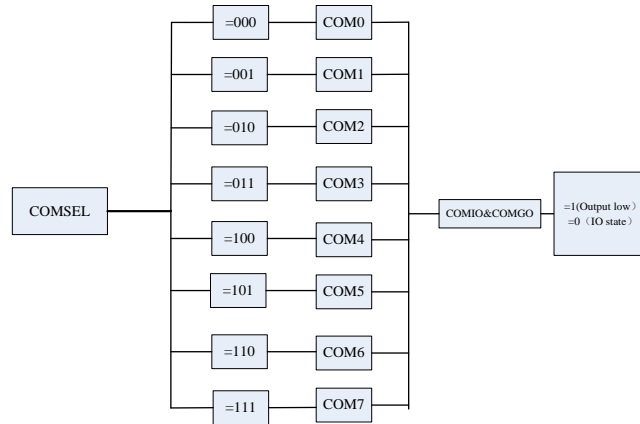


SH30F9/SA0 Series

Note:

1. SEGGO is an internal signal, Start with the COMGO signal, the end is controlled by SEG timing width.
2. VDD of SH30F9/SA0 series has a maximum through current limit (See the section on electrical parameters), enable more than 8 SEGs. Please pay attention to evaluate the current flowing through the VDD.

Schematic diagram of COM control for single scan:

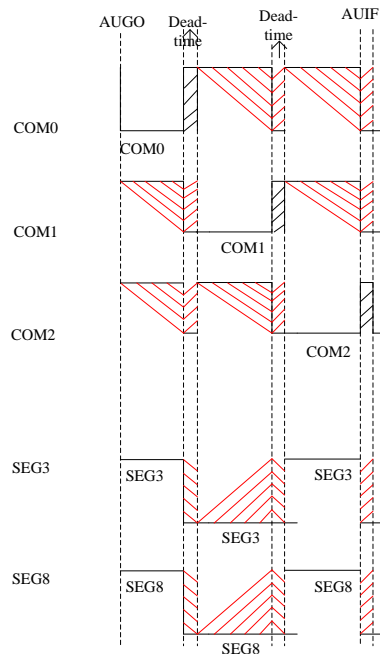


Note:

COM is controlled by three signals, COMIO, COMSEL and COMGO. Take COM1 as an example, if and only when COMIO=1, COMSEL=001, COMGO=1, COM1 output is low, otherwise COM1 is IO, and the state is controlled by IO port.

Continuous scanning: AUEN=1 in the LED_AUCR register is used to select continuous scanning, COM[7:0] bit is used to select COM to be scanned, LED_SEGMOD register is used to select SEG to be scanned, and AUGO=1 in the LED_AUCR register is used to start continuous scanning. The output of SEG is controlled by the register LED_SORX. When the value in LED_SORX is 1, the output is constant current; when the value is 0, the state is IO. SEG output cycle width is the same as COM; When a frame is scanned, the interrupt flag bit AUIF@LED_AUCR is set to 1.

Control timing (continuous scan mode):





SH30F9/SA0 Series

Description:

1. Low represents the output low level
2. High represents the output constant current source
3. The dark shaded part represents the floating state
4. The red shaded area represents IO status

Note: serial dot matrix continuous scanning is not supported

17.2 SEG width control

| Register | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| LED_SDR0 | SEG0DUTY | DY0.7 | DY0.6 | DY0.5 | DY0.4 | DY0.3 | DY0.2 | DY0.1 | DY0.0 |
| | SEG1DUTY | DY1.7 | DY1.6 | DY1.5 | DY1.4 | DY1.3 | DY1.2 | DY1.1 | DY1.0 |
| | SEG2DUTY | DY2.7 | DY2.6 | DY2.5 | DY2.4 | DY2.3 | DY2.2 | DY2.1 | DY2.0 |
| | SEG3DUTY | DY3.7 | DY3.6 | DY3.5 | DY3.4 | DY3.3 | DY3.2 | DY3.1 | DY3.0 |
| LED_SDR1 | SEG4DUTY | DY4.7 | DY4.6 | DY4.5 | DY4.4 | DY4.3 | DY4.2 | DY4.1 | DY4.0 |
| | SEG5DUTY | DY5.7 | DY5.6 | DY5.5 | DY5.4 | DY5.3 | DY5.2 | DY5.1 | DY5.0 |
| | SEG6DUTY | DY6.7 | DY6.6 | DY6.5 | DY6.4 | DY6.3 | DY6.2 | DY6.1 | DY6.0 |
| | SEG7DUTY | DY7.7 | DY7.6 | DY7.5 | DY7.4 | DY7.3 | DY7.2 | DY7.1 | DY7.0 |
| LED_SDR2 | SEG8DUTY | DY8.7 | DY8.6 | DY8.5 | DY8.4 | DY8.3 | DY8.2 | DY8.1 | DY8.0 |
| | SEG9DUTY | DY9.7 | DY9.6 | DY9.5 | DY9.4 | DY9.3 | DY9.2 | DY9.1 | DY9.0 |
| | SEG10DUTY | DY10.7 | DY10.6 | DY10.5 | DY10.4 | DY10.3 | DY10.2 | DY10.1 | DY10.0 |
| | SEG11DUTY | DY11.7 | DY11.6 | DY11.5 | DY11.4 | DY11.3 | DY11.2 | DY11.1 | DY11.0 |
| LED_SDR3 | SEG12DUTY | DY12.7 | DY12.6 | DY12.5 | DY12.4 | DY12.3 | DY12.2 | DY12.1 | DY12.0 |
| | SEG13DUTY | DY13.7 | DY13.6 | DY13.5 | DY13.4 | DY13.3 | DY13.2 | DY13.1 | DY13.0 |
| | SEG14DUTY | DY14.7 | DY14.6 | DY14.5 | DY14.4 | DY14.3 | DY14.2 | DY14.1 | DY14.0 |
| | SEG15DUTY | DY15.7 | DY15.6 | DY15.5 | DY15.4 | DY15.3 | DY15.2 | DY15.1 | DY15.0 |

Description:

1. The register is valid only if AUEN@LED_AUCR = 0 and the MODE bit of LED_COM_CR is 1.
2. SEG width = bus clock width X (DISCOM+1) X (SegxDuty +1).
3. After the end of a SEG width, the state of SEG port is switched to IO.

17.3 Continuous scan SEG output control (if and only if AUEN@LED_AUCR = 1)

| Register | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------|-------|-------|-------|-------|-------|-------|------|------|
| LED_SOR0 | COM0SEG | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | - |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| | COM1SEG | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | - | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| LED_SOR1 | COM2SEG | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | - | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| | COM3SEG | SEG7 | SEG6 | SEG5 | SEG4 | - | SEG2 | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |



SH30F9/SA0 Series

| | | | | | | | | | |
|----------|---------|-------|-------|-------|-------|-------|-------|------|------|
| LED_SOR2 | COM4SEG | SEG7 | SEG6 | SEG5 | - | SEG3 | SEG2 | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| | COM5SEG | SEG7 | SEG6 | - | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| LED_SOR3 | COM6SEG | SEG7 | - | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |
| | COM7SEG | - | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| | | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 |

17.4 Matters attention

17.4.1 LED output status in stop mode

When the AUGO@LED_AUR bit is 1, before entering the stop mode, the AUGO@LED_AUR bit will be cleared by the hardware, and the output state will remain (to turn off the output, the software needs to configure it to the off state before entering the stop mode). When the MCU wakes up from the stop mode, it needs to write AUGO@LED_AUR bit to 1 again through software, and the LED pin can output waveform normally.



17.5 Registers

LED Module Register list (Base Address:0x4002 1C00)

| Address | Register | Description |
|-------------|----------|--|
| 0x4002 1C00 | COMCR | LED COM control register |
| 0x4002 1C04 | FR | LED interrupt flag and clear register |
| 0x4002 1C08 | CCCR | LED constant current switch control register |
| 0x4002 1C0C | CCDR | LED data register |
| 0x4002 1C10 | AUCR | LED continuous scan control register |
| 0x4002 1C14 | SEGMOD | LED mode control register |
| 0x4002 1C18 | SDR0 | LED SEG width control register |
| 0x4002 1C1C | SDR1 | LED SEG width control register |
| 0x4002 1C20 | SDR2 | LED SEG width control register |
| 0x4002 1C24 | SDR3 | LED SEG width control register |
| 0x4002 1C28 | SOR0 | Continuous scan of the SEG output control register |
| 0x4002 1C2C | SOR1 | Continuous scan of the SEG output control register |
| 0x4002 1C30 | SOR2 | Continuous scan of the SEG output control register |
| 0x4002 1C34 | SOR3 | Continuous scan of the SEG output control register |

17.5.1 LED COM control register (LED_COMCR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|--------|----------|--------|-------|--------|-------------|-----|-----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | LED CDM A |
| | | | | | | | | | | | | | | | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DISCOM[7:0] | | | | | | | | COM GO | Reserved | COM IE | MOD E | COM IO | COMSEL[2:0] | | |
| RW | | | | | | | | RW | - | RW | RW | RW | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|---|
| 31 ~ 17 | Reserved | - |
| 16 | LEDCDMA | LED_COM interrupts DMA enable bits 0: Disable DMA requests 1: Enable DMA requests |
| 15 ~ 8 | DISCOM[7:0] | COM width selection bit (after a COM width, the COM port state is switched to IO) COM width = bus clock width X 256 X (DISCOM[7:0]+1) |
| 7 | COMGO | COM starts to output control bits (if and only if AUEN = 0) 0: Disable COM/SEG output 1: COM/SEG start output (after a COM width ends, cleared by hardware) |
| 6 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|--------------------|---|
| 5 | COMIE | LED_COM interrupt enable bit 0: Disable LED_COM interrupts 1: Enable LED_COM interrupts |
| 4 | MODE | SEG output width selection 0: The SEG output width is the same as the COM width 1: The SEG output width is controlled by the SEG width register (SDR0/SDR1/SDR2/SDR3) |
| 3 | COMIO | COM/IO Share (if and only if AUEN = 0) 0: COM as IO 1: COM as the currently selected COMX |
| 2 ~ 0 | COMSEL[2:0] | Current scan COM selection (valid if and only if AUEN = 0) 000: Select the current scan COM as COM0 001: Select the current scan COM as COM1 010: Select the current scan COM as COM2 011: Select the current scan COM as COM3 100: Select the current scan COM as COM4 101: Select the current scan COM as COM5 110: Select the current scan COM as COM6 111: Select the current scan COM as COM7 |

17.5.2 LED interrupt flag and clear register (LED_FR)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | AUIF C | COM IFC |
| | | | | | | | | | | | | | | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|------|-----------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | AUIF | COM IF |
| | | | | | | | | | | | | | | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 18 | Reserved | - |
| 17 | AUIFC | Continuously scanning of a frame end interrupt flag clears bits 0: Invalid 1: Clear |
| 16 | COMIFC | COM output end interrupt flag clear bit 0: Invalid 1: Clear |
| 15 ~ 2 | Reserved | - |
| 1 | AUIF | Continuously scan a frame end interrupt flag 0: A frame scan is not finished 1: A frame scan ends |
| 0 | COMIF | COM outputs the end interrupt flag bit 0: COM output is not finished 1: End of COM output |



SH30F9/SA0 Series

17.5.3 LED constant current switch control register (LED_CCCR)

Note: When the constant current source is closed, the corresponding SEG port of the constant current source is IO

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | CCS S7 | CCS S6 | CCS S5 | CCS S4 | CCS S3 | CCS S2 | CCS S1 | CCS S0 |
| - | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|--|
| 31 ~ 8 | Reserved | - |
| 7 | CCSS7 | Constant current source 7 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 6 | CCSS6 | Constant current source 6 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 5 | CCSS5 | Constant current source 5 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 4 | CCSS4 | Constant current source 4 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 3 | CCSS3 | Constant current source 3 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 2 | CCSS2 | Constant current source 2 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 1 | CCSS1 | Constant current source 1 switch 0:Turn off the constant current source 1:Turn on the constant current source |
| 0 | CCSS0 | Constant current source 0 switch 0:Turn off the constant current source 1:Turn on the constant current source |

17.5.4 LED data register (LED_CCDR)

Offset Address: 0x000C

Reset value: 0x6666 6666

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-------------|-----|-----|-----|-------------|-----|-----|-----|-------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CCS7DS[3:0] | | | | CCS6DS[3:0] | | | | CCS5DS[3:0] | | | | CCS4DS[3:0] | | | |
| RW | | | | RW | | | | RW | | | | RW | | | |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CCS3DS[3:0] | | | | CCS2DS[3:0] | | | | CCS1DS[3:0] | | | | CCS0DS[3:0] | | | |
| RW | | | | RW | | | | RW | | | | RW | | | |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|--------------------|---|
| 31 ~ 28 | CCS7DS[3:0] | Constant current source 7 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 27 ~ 24 | CCS6DS[3:0] | Constant current source 6 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20Ma(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 23 ~ 20 | CCS5DS[3:0] | Constant current source 5 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 19 ~ 16 | CCS4DS[3:0] | Constant current source 4 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 15 ~ 12 | CCS3DS[3:0] | Constant current source 3 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 11 ~ 8 | CCS2DS[3:0] | Constant current source 2 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 7 ~ 4 | CCS1DS[3:0] | Constant current source 1 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |
| 3 ~ 0 | CCS0DS[3:0] | Constant current source 0 output current size selection bit 0000: Constant current for8mA 0001: Constant current for10mA 0010: Constant current for12mA 0011: Constant current for14mA 0100: Constant current for16mA 0101: Constant current for18mA 0110: Constant current for20mA(default) 0111: Constant current for22mA 1000: Constant current for24mA 1001: Constant current for26mA 1010: Constant current for28mA 1011: Constant current for30mA 1100: Constant current for32mA other: Constant current for32mA |



SH30F9/SA0 Series

17.5.5 LED continuous scan control register (LED_AUCR)

Offset Address: 0x0010

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|----------|----------|------|----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | AUD MA | AUE N | AUG O | AUIE | AUM D |
| | | | | | | | | | | | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|---------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| COM[7:0] | | | | | | | | DZ[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 21 | Reserved | - |
| 20 | AUDMA | Continuous scanning frames interrupt DMA allowed bits 0: Disallow DMA requests 1: DMA requests are allowed |
| 19 | AUEN | Continuous scanning of control bits 0: Single scan (SEG output controlled by register SEGMOD) 1: Continuous scan (SEG output controlled by registers SOR0/SOR1/SOR2/SOR3) |
| 18 | AUGO | The continuous scan begins to output control bits 0: Disallow COM/SEG output 1: COM/SEG starts output |
| 17 | AUIE | Scan the end interrupt enabler bit continuously for one frame 0: Disallow frame interrupts 1: Frame interrupt allowed |
| 16 | AUMD | Continuous scan of interrupt operation mode control bits 0: When AUIF = 1, Scanning continues (AUGO is not zeroed, software zeroed) 1: When AUIF = 1, the scan stops. You need to set AUGO to 1 to start the next frame scan (AUGO hardware reset). |
| 15 ~ 8 | COM[7:0] | Continuous mode COM port selection bit 0: As I/O 1: COM as LED during LED continuous scanning(COM0-COM7) Note: Apply to continuous scan mode, if and only if AUEN=1. |
| 7 ~ 0 | DZ[7:0] | Continuous mode LED dead-zone width selector bit Dead width = Bus clock width X (DZ[7:0] + 1) Note: 1. It is recommended to set the LED dead time more than 10 bus clocks (DZ[7:0] > 0AH); 2. Apply to continuous scanning mode, if and only if AUEN=1. |

17.5.6 LED mode control register (LED_SEGMOD)

Offset Address: 0x0014

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| S15 MS | S14 MS | S13 MS | S12 MS | S11 MS | S10 MS | S9M S | S8M S | S7M S | S6M S | S5M S | S4M S | S3M S | S2M S | S1M S | S0M S |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 | S15MS | <p>Seg15 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 14 | S14MS | <p>Seg14 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 13 | S13MS | <p>Seg13 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 12 | S12MS | <p>Seg12 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 11 | S11MS | <p>Seg11 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 10 | S10MS | <p>Seg10 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 9 | S9MS | <p>Seg9 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 8 | S8MS | <p>Seg8 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |
| 7 | S7MS | <p>Seg7 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan.</p> |



SH30F9/SA0 Series

| | | |
|---|-------------|--|
| 6 | S6MS | Seg6 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 5 | S5MS | Seg5 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 4 | S4MS | Seg4 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 3 | S3MS | Seg3 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 2 | S2MS | Seg2 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 1 | S1MS | Seg1 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |
| 0 | S0MS | Seg0 port mode selection bit 0: As I/O 1: As SEG Note: 1. In a single scan, the register is used to control the output of constant current; 2. This register is used for IO Share during continuous scan. |



17.5.7 LED SEG width control register (LED_SDR0)

Offset Address: 0x0018

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| SEG3DUTY[7:0] | | | | | | | | SEG2DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|----|----|---------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SEG1DUTY[7:0] | | | | | | | | SEG0DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|--|
| 31 ~ 24 | SEG3DUTY[7:0] | Seg3 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 23 ~ 16 | SEG2DUTY[7:0] | Seg2 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 15 ~ 8 | SEG1DUTY[7:0] | Seg1 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 7 ~ 0 | SEG0DUTY[7:0] | Seg0 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |

17.5.8 LED SEG width control register (LED_SDR1)

Offset Address: 0x001C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| SEG7DUTY[7:0] | | | | | | | | SEG6DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|----|----|---------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SEG5DUTY[7:0] | | | | | | | | SEG4DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|---------------|--|
| 31 ~ 24 | SEG7DUTY[7:0] | Seg7 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 23 ~ 16 | SEG6DUTY[7:0] | Seg6 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 15 ~ 8 | SEG5DUTY[7:0] | Seg5 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 7 ~ 0 | SEG4DUTY[7:0] | Seg4 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |

17.5.9 LED SEG width control register (LED_SDR2)

Offset Address: 0x0020

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|----------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| SEG11DUTY[7:0] | | | | | | | | SEG10DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|----|----|---------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SEG9DUTY[7:0] | | | | | | | | SEG8DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------|---|
| 31 ~ 24 | SEG11DUTY[7:0] | Seg11 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 23 ~ 16 | SEG10DUTY[7:0] | Seg10 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 15 ~ 8 | SEG9DUTY[7:0] | Seg9 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 7 ~ 0 | SEG8DUTY[7:0] | Seg8 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |



17.5.10 LED SEG width control register (LED_SDR3)

Offset Address: 0x0024

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|----------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| SEG15DUTY[7:0] | | | | | | | | SEG14DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SEG13DUTY[7:0] | | | | | | | | SEG12DUTY[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------|---|
| 31 ~ 24 | SEG15DUTY[7:0] | Seg15 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 23 ~ 16 | SEG14DUTY[7:0] | Seg14 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 15 ~ 8 | SEG13DUTY[7:0] | Seg13 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |
| 7 ~ 0 | SEG12DUTY[7:0] | Seg12 width selection bit (after a SEG width ends, the SEG port state switches to IO) Segx width = bus clock width X (DISCOM[7:0]+1) X (SEGxDUTY[7:0]+1) The register is valid if and only if AUEN = 0 and the MODE bit in the COMCR register is 1 |

17.5.11 Continuous scan of the SEG output control register (LED_SOR0)

Offset Address: 0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| COM1SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| COM0SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 16 | COM1SEG[15:0] | SEG output control bits when scanning COM1 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |
| 15 ~ 0 | COM0SEG[15:0] | SEG output control bits when scanning COM0 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |



17.5.12 Continuous scan of the SEG output control register (LED_SOR1)

Offset Address: 0x002C
 Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| COM3SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| COM2SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------------|---|
| 31 ~ 16 | COM3SEG[15:0] | SEG output control bits when scanning COM3 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |
| 15 ~ 0 | COM2SEG[15:0] | SEG output control bits when scanning COM2 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |

17.5.13 Continuous scan of the SEG output control register (LED_SOR2)

Offset Address: 0x0030
 Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| COM5SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| COM4SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------------|---|
| 31 ~ 16 | COM5SEG[15:0] | SEG output control bits when scanning COM5 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |
| 15 ~ 0 | COM4SEG[15:0] | SEG output control bits when scanning COM4 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |



SH30F9/SA0 Series

17.5.14 Continuous scan of the SEG output control register (LED_SOR3)

Offset Address: 0x0034

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| COM7SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| COM6SEG[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------------|---|
| 31 ~ 16 | COM7SEG[15:0] | SEG output control bits when scanning COM7 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |
| 15 ~ 0 | COM6SEG[15:0] | SEG output control bits when scanning COM6 (if and only if AUEN = 1) 0: As I/O 1: As SEG (Constant Current Output) |



18. LCD driver (Resistance type)

18.1 Introduction

- Support 4*40, 5*39, 6*38, 8*36
- Support 1/3bias and 1/4bias
- Support 1/4, 1/5, 1/6, 1/8 duty
- Support fast charge mode to reduce power consumption
- The COM port that is not enabled can be shared as SEG port
- LCD is turned off during power on reset, pin reset, low voltage reset or watchdog reset. When LCD is turned off, Common and Segment output low level.
- Support internal resistance series voltage division to realize bias voltage
- Support operation in stop mode

The traditional resistance LCD display mode has the following characteristics:

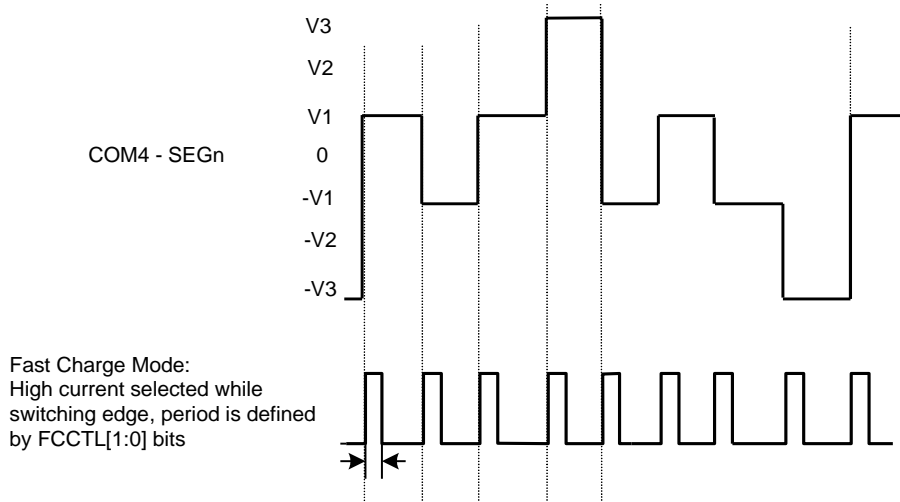
- The driving voltage of LCD, VLCD, is directly supplied by VDD. VDD is controlled by VOL[2:0] of LCD_CR register for 8-level contrast adjustment. Then, the corresponding V1, V2, and V3 are obtained by internal resistor divider
- The LCD frame rate calculation is as follows:
When selecting 32.768KHz Crystal for LCD clock, the fixed frame rate is 64Hz, DCK [1:0] @ LCD_CR register is invalid;
Choose different duty cycles with slightly different frame rates;
When selecting LSI for LCD clock, set LCD_ The DCK [1:0] bit of the CR register selects 1/4, 1/3, 1/2, 1/1 division ratio, and the corresponding LCD frame rates are 256/4Hz, 256/3Hz, 256/2Hz, 256/1Hz. Selecting different duty cycles results in slight differences in frame rates;
- Fast charge mode is supported to reduce power consumption.

The bit MOD[1:0]@LCD_CR controls LCD bias resistance (R_{LCD}) which can be selected as 30K or 2.4M. Choosing 30K bias resistor can get a better display effect, but the current is relatively larger, which is not suitable for low-power applications. When The bit MOD[1:0]@LCD_CR is set to 00, the 2.4M bias resistor is selected. Although low power consumption can be achieved, the LCD display effect will become worse.

Therefore, MCU provides a display mode with low power consumption and display effect: fast charge mode. Set MOD[1:0]=10 to select this display mode. Select 30K bias resistor at the display data refresh time to provide a larger drive current. Select 2.4M bias resistor during data retention to provide a smaller drive current.

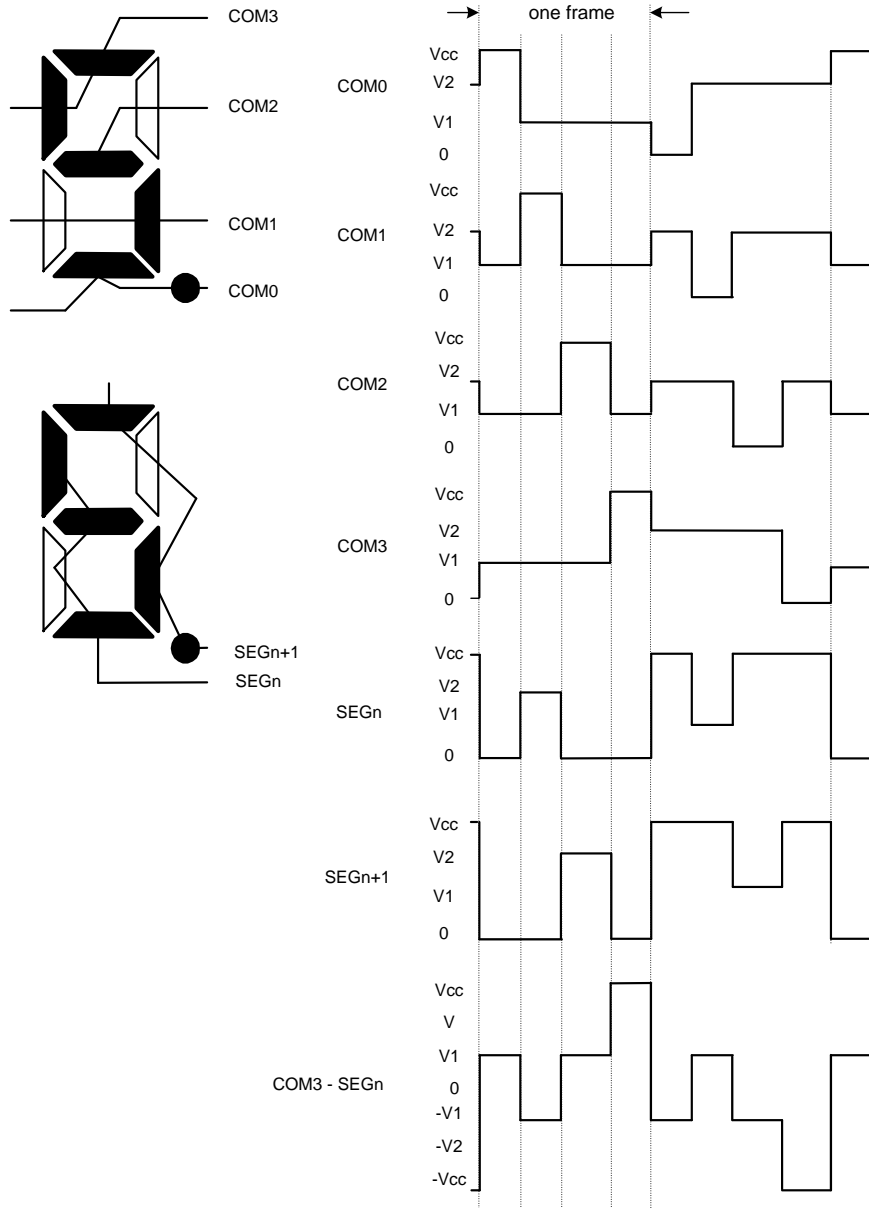
The charge time of fast charge mode is 1/8, 1/16, 1/32 or 1/64 of LCD com cycle, which is selected by the FCCTL[1:0] bit@LCD_CR

The waveform of fast charge mode





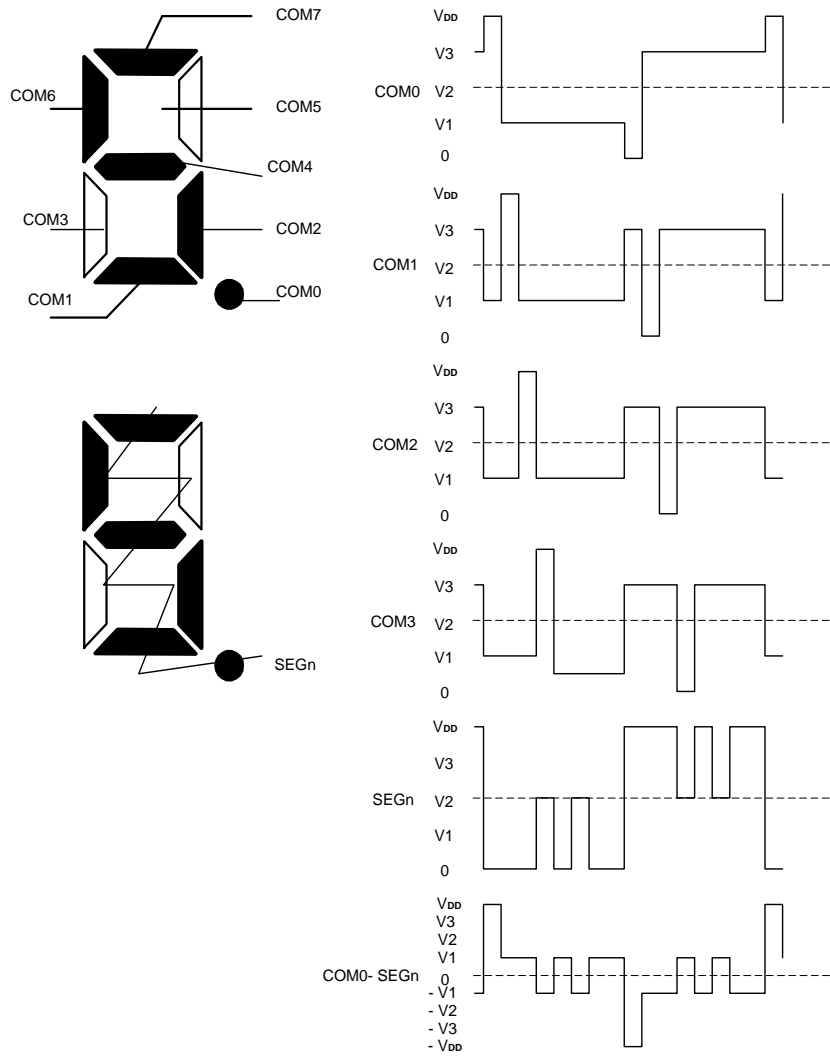
SH30F9/SA0 Series



LCD waveform (1/4 duty, 1/3 bias)



SH30F9/SA0 Series



LCD waveform (1/8 duty, 1/4 bias)



SH30F9/SA0 Series

18.2 LCD RAM CONFIGURATION

| REGISTERS | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | COM7 | COM6 | COM5 | COM4 | COM3 | COM2 | COM1 | COM0 |
| LCD_BUFS0 | LCD_BUF0 | SEG0 | SEG0 | SEG0 | SEG0 | SEG0 | SEG0 | SEG0 | SEG0 |
| | LCD_BUF1 | SEG1 | SEG1 | SEG1 | SEG1 | SEG1 | SEG1 | SEG1 | SEG1 |
| | LCD_BUF2 | SEG2 | SEG2 | SEG2 | SEG2 | SEG2 | SEG2 | SEG2 | SEG2 |
| | LCD_BUF3 | SEG3 | SEG3 | SEG3 | SEG3 | SEG3 | SEG3 | SEG3 | SEG3 |
| LCD_BUFS1 | LCD_BUF4 | SEG4 | SEG4 | SEG4 | SEG4 | SEG4 | SEG4 | SEG4 | SEG4 |
| | LCD_BUF5 | SEG5 | SEG5 | SEG5 | SEG5 | SEG5 | SEG5 | SEG5 | SEG5 |
| | LCD_BUF6 | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 |
| | LCD_BUF7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 |
| LCD_BUFS2 | LCD_BUF8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 |
| | LCD_BUF9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 |
| | LCD_BUF10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 |
| | LCD_BUF11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 |
| LCD_BUFS3 | LCD_BUF12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 |
| | LCD_BUF13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 |
| | LCD_BUF14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 |
| | LCD_BUF15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 |
| LCD_BUFS4 | LCD_BUF16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 |
| | LCD_BUF17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 |
| | LCD_BUF18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 |
| | LCD_BUF19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 |
| LCD_BUFS5 | LCD_BUF20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 |
| | LCD_BUF21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 |
| | LCD_BUF22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 |
| | LCD_BUF23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 |
| LCD_BUFS6 | LCD_BUF24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 |
| | LCD_BUF25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 |
| | LCD_BUF26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 |
| | LCD_BUF27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 |
| LCD_BUFS7 | LCD_BUF28 | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 |
| | LCD_BUF29 | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 |
| | LCD_BUF30 | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 |
| | LCD_BUF31 | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 |
| LCD_BUFS8 | LCD_BUF32 | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 |
| | LCD_BUF33 | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 |
| | LCD_BUF34 | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 |
| | LCD_BUF35 | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 |
| LCD_BUFS9 | LCD_BUF36 | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 |
| | LCD_BUF37 | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 |
| | LCD_BUF38 | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 |
| | LCD_BUF39 | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 |



18.3 Registers

LCD Module Register list (Base Address:0x4002 2000)

| Address | Register | Description |
|-------------|----------|----------------------|
| 0x4002 2000 | CR | LCD control register |
| 0x4002 2004 | BUFS0 | LCD data register |
| 0x4002 2008 | BUFS1 | LCD data register |
| 0x4002 200C | BUFS2 | LCD data register |
| 0x4002 2010 | BUFS3 | LCD data register |
| 0x4002 2014 | BUFS4 | LCD data register |
| 0x4002 2018 | BUFS5 | LCD data register |
| 0x4002 201C | BUFS6 | LCD data register |
| 0x4002 2020 | BUFS7 | LCD data register |
| 0x4002 2024 | BUFS8 | LCD data register |
| 0x4002 2028 | BUFS9 | LCD data register |

18.3.1 LCD control register (LCD_CR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|------------|----------|----------|--------|-------|----------|----------|-----|-----------|-----|-------|-----|-----|----------|---------|-----|
| Reserved | | | | | | | | | | | | | DCK[1:0] | CNT MOD | |
| | | | | | | | | | | | | | RW | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| FCCTL[1:0] | MOD[1:0] | Reserved | MOD SW | ELC C | Reserved | VOL[2:0] | | DUTY[2:0] | | LCL K | EN | | | | |
| RW | RW | - | RW | RW | - | RW | | RW | | RW | RW | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 19 | Reserved | - |
| 18 ~ 17 | DCK[1:0] | LCD clock selection control bit 00: Divided by 4 01: Divided by 3 10: Divided by 2 11: Divided by 1 Note: These two are only valid if the LCD clock selects LSI |
| 16 | CNTMOD | LCD scan counter mode control bit 0: When MODSW is 1, LCD scan counter keep counting. 1: When MODSW is 1, the LCD scan counter will stop counting and keep the current COM value. When LCD is opened again, the scan will continue from the current COM port |
| 15 ~ 14 | FCCTL[1:0] | Charging time control bit 00: 1/8 LCD com period 01: 1/16 LCD com period 10: 1/32 LCD com period 11: 1/64 LCD com period |
| 13 ~ 12 | MOD[1:0] | Driver mode selection bit 00: Traditional mode, bias resistor sum is 2.4M 01: Traditional mode, bias resistor sum is 30k 10: Fast charge mode, bias resistor sum switch between 30k and 2.4M 11: No used |



SH30F9/SA0 Series

| | | |
|-------|------------------|---|
| 11 | Reserved | - |
| 10 | MODSW | LCD port output total control bit 0: Enable LCD port output 1: Disable LCD port output |
| 9 | ELCC | LCD contrast enable control bits 0: Disable LCD contrast control 1: Enable LCD contrast control |
| 8 | Reserved | - |
| 7 ~ 5 | VOL[2:0] | LCD contrast control bit 000: VLCD=0.60VDD 001: VLCD=0.66VDD 010: VLCD=0.72VDD 011: VLCD=0.78VDD 100: VLCD=0.84VDD 101: VLCD=0.90VDD 110: VLCD=0.96VDD 111: VLCD=1.00VDD |
| 4 ~ 2 | DUTY[2:0] | LCD duty selected bits 000:000: 1/4 Duty, 1/3 Bias(4 COM X 40 SEG) COM: COM0-3 SEG: SEG0-35 COM4-7 shared as SEG36-39 001:1/8Duty, 1/4Bias(8 COM X 36 SEG) COM: COM0-7 SEG: SEG0-35 010:1/4Duty, 1/3Bias(4 COM X 40 SEG) COM: COM4-7 shared as COM0-3 SEG: SEG0-35, COM0-3 shared as SEG36-39 011:1/5Duty, 1/3Bias(5 COM X 39 SEG) COM: COM0-4 SEG: SEG0-35, COM5-7 shared as SEG36-38 100:1/6Duty, 1/3Bias(6 COM X 38 SEG) COM: COM0-5 SEG: SEG0-35, COM6-7 shared as SEG36-SEG37 101:1/6Duty, 1/4Bias(6 COM X 38 SEG) COM: COM0-5 SEG: SEG0-35, COM6-7 shared as SEG36-SEG37 Others:1/4Duty, 1/3Bias(4 COM X 40 SEG) COM: COM0-3 SEG: SEG0-35 COM4-7 shared as SEG36-39 |
| 1 | LCLK | LCD clock selection bit 0: LCD clock source is LSI 1: LCD clock source is 32.768Khz Crystal |
| 0 | EN | LCD turn on/off control bit 0: Disable LCD driver 1: Enable LCD driver |



18.3.2 LCD data register (LCD_BUFS0)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|--------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF3[7:0] | | | | | | | | LCDBUF2[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF1[7:0] | | | | | | | | LCDBUF0[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 24 | LCDBUF3[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF2[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF1[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF0[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |

18.3.3 LCD data register (LCD_BUFS1)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|--------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF7[7:0] | | | | | | | | LCDBUF6[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF5[7:0] | | | | | | | | LCDBUF4[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|---|
| 31 ~ 24 | LCDBUF7[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF6[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF5[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF4[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |



18.3.4 LCD data register (LCD_BUFS2)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF11[7:0] | | | | | | | | LCDBUF10[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF9[7:0] | | | | | | | | LCDBUF8[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF11[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF10[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF9[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF8[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |

18.3.5 LCD data register (LCD_BUFS3)

Offset Address: 0x0010

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF15[7:0] | | | | | | | | LCDBUF14[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF13[7:0] | | | | | | | | LCDBUF12[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF15[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF14[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF13[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF12[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |



18.3.6 LCD data register (LCD_BUFS4)

Offset Address: 0x0014

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF19[7:0] | | | | | | | | LCDBUF18[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF17[7:0] | | | | | | | | LCDBUF16[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF19[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF18[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF17[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF16[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |

18.3.7 LCD data register (LCD_BUFS5)

Offset Address: 0x0018

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF23[7:0] | | | | | | | | LCDBUF22[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF21[7:0] | | | | | | | | LCDBUF20[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF23[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF22[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF21[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF20[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |



18.3.8 LCD data register (LCD_BUFS6)

Offset Address: 0x001C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF27[7:0] | | | | | | | | LCDBUF26[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF25[7:0] | | | | | | | | LCDBUF24[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF27[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF26[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF25[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF24[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |

18.3.9 LCD data register (LCD_BUFS7)

Offset Address: 0x0020

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF31[7:0] | | | | | | | | LCDBUF30[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF29[7:0] | | | | | | | | LCDBUF28[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF31[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF30[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF29[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF28[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |



18.3.10 LCD data register (LCD_BUF58)

Offset Address: 0x0024

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF35[7:0] | | | | | | | | LCDBUF34[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF33[7:0] | | | | | | | | LCDBUF32[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF35[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF34[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF33[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF32[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |

18.3.11 LCD data register (LCD_BUF59)

Offset Address: 0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| LCDBUF39[7:0] | | | | | | | | LCDBUF38[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LCDBUF37[7:0] | | | | | | | | LCDBUF36[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 24 | LCDBUF39[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 23 ~ 16 | LCDBUF38[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 15 ~ 8 | LCDBUF37[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |
| 7 ~ 0 | LCDBUF36[7:0] | SEG display data 0: The corresponding cell is not displayed 1: The corresponding cell is displayed |



19. Basic Timer (TIM0~TIM3)

19.1 Introduction

SH30F9/SA0 series built in 3 basic 16 bit timers and 1 basic 32 bit timer. Each clock can run with bus clock at most, and 4 timers can run in stop mode (clock source is built-in 128kHz RC or LSE).

19.2 Main Features

- 3x16-bit timers (TIM0~TIM2)
- 1x32-bit timer (TIM3)
- Each timer counts incrementally
- Each timer can choose independent clock source
- Each timer can select external clock source, which can be used as counter
- Able to run synchronously

19.3 Function Description

○: CAN —: CANNOT

ICLK is the clock source.

| item | TIM0 | TIM1 | TIM2 | TIM3 |
|---------------------------------|---|---|---|---|
| Clock source (ICLK) | (1) Bus clock (2) T0 (3) 128kHz RC (LSI) (4) 32kHz Cry (LSE) | (1) Bus clock (2) T1 (3) 128kHz RC (LSI) (4) 32kHz Cry (LSE) | (1) Bus clock (2) T2 (3) 128kHz RC (LSI) (4) 32kHz Cry (LSE) | (1) Bus clock (2) T3 (3) 128kHz RC (LSI) (4) 32kHz Cry (LSE) |
| Clock source division | ICLK/(PSQ[15:0]+1) | ICLK/(PSQ[15:0]+1) | ICLK/(PSQ[15:0]+1) | ICLK/(PSQ[15:0]+1) |
| Periodic setting register | TIM0.TPR | TIM1.TPR | TIM2.TPR | TIM3.TPR |
| Input pin | T0 | T1 | T2 | T3 |
| Output pin | T0 | T1 | T2 | T3 |
| DMA start-up | All interrupt sources | | | |
| A/D conversion start-up trigger | — | — | • TCNT overflow (TIM2.TF) | — |
| Interrupt source | • TCNT overflow (TIM0.TF) | • TCNT overflow (TIM1.TF) | • TCNT overflow (TIM2.TF) | • TCNT overflow (TIM3.TF) |

19.3.1 Operation of Counter/Timer

SH30F9/SA0 series has four basic timers. Among them, TIM0~TIM2 have a 16 bit data register TIMx_TCNT (x=0~2) and TIM3 has a 32 bit data register TIMx_TCNT (x=3). TIM0~TIM3 have a 2bit clock source selector CLKS[1:0]@TIMx_CR and a clock source frequency divider TIMx_PSQ[15:0](x=0~3). TIMx_TCNT (x=0~3) interrupt is enabled by setting IE@TIMx_CR to 1.

When STR@TIMx_CR is set, the timer starts to increase the count. And when TIMx_TPR (x=0~3) is reached, TIMx overflows and the system sets the timer overflow flag TFx, TIMx_TCNT (x = 0-3) is cleared. If TIMx interrupt is allowed, an interrupt will be generated.

CLKS[1:0]@TIMx_CR bit selects the clock source of the timer: bus clock, rising edge of TIMx input pin (Tx), low frequency internal 128kHz RC and 32kHz Cry can be selected.

When STR@TIMx is set to 1, the timer is not reset, which means that if the STR@TIMx_CR bit is set to 1, the timer register will start counting from TIMx_TCNT. Therefore, the initial value of the timer counter should be set before the timer is allowed.

In timer mode, the TC@TIMx_CR bit can be configured to automatically flip the Tx pin when timer x overflows. With this configuration, the Tx pin is automatically set to output.

Note 1:

When the initial value written to TIMx_TCNT(x=0~2) is greater than TIMx_TPR, TIMx_TCNT will continue to count up until it reaches 0xFFFF, and then it returns to 0x0000.



When the initial value written to $TIMx_TCNT(x=3)$ is greater than $TIMx_TPR$, $TIMx_TCNT$ will continue to count up until it reaches $0xFFFFFFFF$, and then it returns to $0x00000000$.

Note 2: Timer overflow occurs when the $TIMx_TCNT$ counts from the value in the cycle register($TIMx_TPR$) to 0.

Note 3: The prescaler register can be modified while the timer is stopped.

Note 4: The fastest frequency of Tx input is $1/2 * \text{bus clock}$. Select bus clock as filter clock.

Note 5: The initial state of Tx output level is low level.

The following figure shows the timer block diagram, and $TIMx_INT / TIMx_EVENT$ can wake up low power consumption.

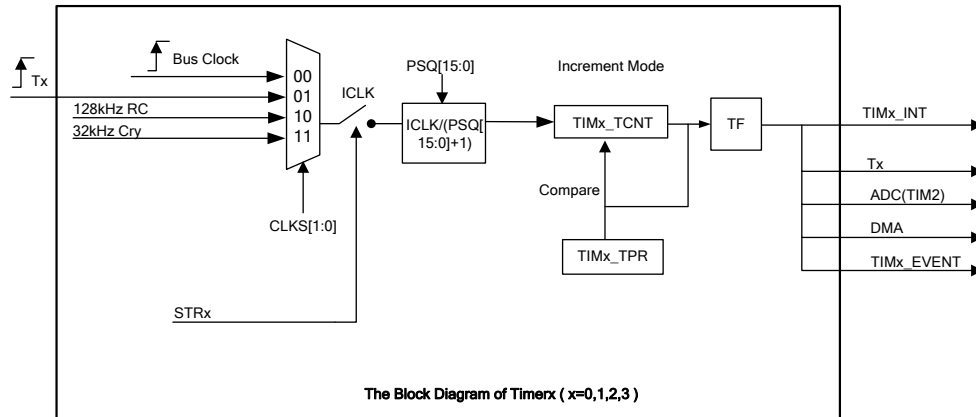


Figure 19-1 Timer Block Diagram

Note1: Timer0/1/2/3 can operate in low consumption mode and can wake up low consumption.

Note2: In stop mode, the Tx port can also work, and Tx port can use 128K as a filter or close the filtering function.

Note3: Before Modifying $TIMx_TCNT$, make sure $STR @ TIMx_CR$ is 0.

If $CLKS[1:0] @ TIMx_CR$ is 00, $TIMx$ can not work in stop mode. If $CLKS[1:0] @ TIMx_CR$ is 01 and the Tx port inputs external clock, $TIMx$ can operate in normal mode or stop mode. If $CLKS[1:0] @ TIMx_CR$ is 10, $TIMx$ selects LSI 128K RC for clock source. When $CLKS[1:0] @ TIMx_CR$ is 11, $TIMx$ selects LSE 32K Crystal for clock source.

19.3.2 External input clock source filtering

When $CLKS[1:0] @ TIMx_CR$ is set to 01, the clock of the counter is provided by the rising edge of the external Tx pin. Different filter parameters can be selected by setting the $ECF @ TIMx_CR$. when $ECF = 0$, there is no filter effect. When $ECF = 1, 2, 3$, the filter time is 8 / 16 / 32 bus clock cycles respectively. When $ECF = 0$ in stop mode, it has no filter effect, and the rest filter time is one LSI clock cycle.

Filter algorithm description: using digital integration, if the sampling result is high level, the counter is added with 1. The counter result exceeds the set constant. The filter outputs 1. And the counter is set as the filter constant at the same time; If the sampling result is low level, the counter is decreased by 1. The counter result is less than the filter constant. Then the filter outputs 0. And the counter is decreased to 0.



19.4 Registers

TIMER0~2 Module Register list (Base Address:0x4000 0400)

| Address | Register | Description |
|---------------------|----------|--|
| 0x4000 0400+0x400*x | CR | TIMx Control Register(TIM0~2 corresponds to x: 0~2) |
| 0x4000 0404+0x400*x | TCNT | TIMx Count Register(TIM0~2 corresponds to x: 0~2) |
| 0x4000 0408+0x400*x | TPR | TIMx period register(TIM0~2 corresponds to x: 0~2) |
| 0x4000 040C+0x400*x | PSQ | TIMx Pre-division Register(TIM0~2 corresponds to x: 0~2) |
| 0x4000 0410+0x400*x | TIMINTF | TIMx Interrupt Flag and Clear Register(TIM0~2 corresponds to x: 0~2) |

19.4.1 TIMx Control Register (TIMx_CR)

Offset Address: 0x0000+0x400*x

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|----------|----------|-----|------|--------|-----|----------|-----------|-----|----------|-----|-----|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | ECF[1:0] | Reserved | TC | ETEN | TRIGEN | IE | Reserved | CLKS[1:0] | OPM | Reserved | STR | | | | |
| - | RW | - | RW | RW | RW | RW | - | RW | RW | - | RW | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 14 | Reserved | - |
| 13 ~ 12 | ECF[1:0] | The filtering time of clock source Tx input 00: No filtering function 01: 8*PCLK 10: 16*PCLK 11: 32*PCLK |
| 11 | Reserved | - |
| 10 | TC | Compare output function enable bit 0: Compare output function of timer x is disabled 1: Compare output function of timer x is enabled Note: When the timer allows compare output (TC = 1), the clock source cannot select the Tx (x = 0, 1, 2) port. |
| 9 | ETEN | TIMx overflow event enable bit 0: Timer overflow from waking up CM0+ core is disabled 1: Timer overflow to wake up CM0+ core is enabled |
| 8 | TRIGEN | TIMx overflow trigger external module enable bit 0: Timer x overflow is disabled to trigger other modules 1: Timer x overflow is enabled to trigger other modules Note 1: Other modules include DMA, ADC Note 2: All modules can trigger DMA, but only TIM2 can trigger ADC |
| 7 | IE | TIMx overflow interrupt enable bit 0: TIMx overflow interrupt is disabled 1: TIMx overflow interrupt is enabled |
| 6 | Reserved | - |
| 5 ~ 4 | CLKS[1:0] | TIMx clock source (ICLK) selection bit 00: Internal APB0 clock (PCLK0) 01: Tx (x=0, 1, 2) port input, automatic pull-up 10: 128kHz RC (LSICLK) 11: 32kHz Crystal (LSECLK) |



| | | |
|-------|-----------------|--|
| 3 | OPM | Single pulse mode 0: When timer x overflows, the timer does not stop 1: When timer x overflows, the timer stops Note: In stop mode, single pulse mode is invalid, OPM must be set to 0 |
| 2 ~ 1 | Reserved | - |
| 0 | STR | TIMx start-up bit Note 1: Can be triggered by software or by external module Note 2: When STR writes "0" or "1", TCNT register will not be cleared |

19.4.2 TIMx Count Register (TIMx_TCNT)

Offset Address: 0x0004+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TCNT[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | TCNT[15:0] | Timer count value Note: When CLKS[1:0]=01, the TCNT register does not support write operation. |

19.4.3 TIMx period register (TIMx_TPR)

Offset Address: 0x0008+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TPR[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|---------------------------------|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | TPR[15:0] | Timer automatic reload register |



19.4.4 TIMx Pre-division Register (TIMx_PSQ)

Offset Address: 0x000C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PSQ[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PSQ[15:0] | Timer pre-division value Note1: The timer clock is (ICLK/(PSQ[15:0]+1))(ICLK/(PSQ[15:0]+1)) Note2: when writing TCNT value, the internal CNT value of PSQ will also be cleared. |

19.4.5 TIMx Interrupt Flag and Clear Register (TIMx_TIMINTF)

Offset Address: 0x0010+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | TFC |
| - | | | | | | | | | | | | | | | <i>WO</i> |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | | TF |
| - | | | | | | | | | | | | | | | <i>RO</i> |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 17 | Reserved | - |
| 16 | TFC | TIMx clear bit of overflow flag 0: Invalid 1: Clear |
| 15 ~ 1 | Reserved | - |
| 0 | TF | TIMx overflow flag bit 0: Timer x has no overflow and can be cleared by software. 1: Timer x overflows and is set to 1 by hardware |



TIMER3 Module Register list (Base Address:0x4000 1000)

| Address | Register | Description |
|-------------|----------|--|
| 0x4000 1000 | CR | TIM3 Control Register |
| 0x4000 1004 | TCNT | TIM3 Count Register |
| 0x4000 1008 | TPR | TIM3 period register |
| 0x4000 100C | PSQ | TIM3 Pre-division Register |
| 0x4000 1010 | TIMINTF | TIM3 Interrupt Flag and Clear Register |

19.4.6 TIM3 Control Register (TIMER32_CR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|----------|-----|----------|-----|------|--------|-----|----------|-----------|-----|-----|----------|-----|-----|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | ECF[1:0] | | Reserved | TC | ETEN | TRIGEN | IE | Reserved | CLKS[1:0] | | OPM | Reserved | STR | | |
| - | RW | | - | RW | RW | RW | RW | - | RW | | RW | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 14 | Reserved | - |
| 13 ~ 12 | ECF[1:0] | The filtering time of clock source T3 input 00: No filtering function 01: 8*PCLK 10: 16*PCLK 11: 32*PCLK |
| 11 | Reserved | - |
| 10 | TC | Compare output function enable bit 0: Compare output function of timer3 is disabled 1: Compare output function of timer3 is enabled Note: When the timer allows compare output (TC = 1), the clock source cannot select the T3 port. |
| 9 | ETEN | TIM3 overflow event enable bit 0: Timer3 overflow from waking up CM0+ core is disabled 1: Timer3 overflow to wake up CM0+ core is enabled |
| 8 | TRIGEN | TIM3 overflow trigger external module enable bit 0: Timer3 overflow is disabled to trigger other modules 1: Timer3 overflow is enabled to trigger other modules Note 1: Other modules include DMA, ADC Note 2: All modules can trigger DMA, but only TIM2 can trigger ADC |
| 7 | IE | TIM3 overflow interrupt enable bit 0: Timer3 overflow interrupt is disabled 1: Timer3 overflow interrupt is enabled |
| 6 | Reserved | - |
| 5 ~ 4 | CLKS[1:0] | TIM3 clock source (ICLK) selection bit 00: Internal APB0 clock (PCLK0) 01: T3 port input, automatic pull-up 10: 128kHz RC (LSICKL) 11: 32kHz Crystal (LSECLK) |
| 3 | OPM | Single pulse mode 0: When timer3 overflows, the timer does not stop 1: When timer3 overflows, the timer stops Note: In stop mode, single pulse mode is invalid, OPM must be set to 0 |
| 2 ~ 1 | Reserved | - |



| | | |
|---|------------|---|
| 0 | STR | TIM3 start-up bit Note1: Can be triggered by software or by external module Note2: When STR writes "0" or "1", TCNT register will not be cleared |
|---|------------|---|

19.4.7 TIM3 Count Register (TIMER32_TCNT)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| TCNT[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| | | | | | | | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TCNT[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| Bit | Symbol | Description |
|--------|------------|--|
| 31 ~ 0 | TCNT[31:0] | Timer count value Note: When CLKS[1:0]=01, the TCNT register does not support write operation. |

19.4.8 TIM3 period register (TIMER32_TPR)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| TPR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TPR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | TPR[31:0] | Timer automatic reload register |

19.4.9 TIM3 Pre-division Register (TIMER32_PSQ)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PSQ[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| Bit | Symbol | Description |
|---------|-----------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PSQ[15:0] | Timer pre-division value Note1: The timer clock is (ICLK/(PSQ[15:0]+1))(ICLK/(PSQ[15:0]+1)) Note2: when writing TCNT value, the internal CNT value of PSQ will also be cleared. |



19.4.10 TIM3 Interrupt Flag and Clear Register (TIMER32_TIMINTF)

Offset Address: 0x0010

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | TFC |
| - | | | | | | | | | | | | | | | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | | TF |
| - | | | | | | | | | | | | | | | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 17 | Reserved | - |
| 16 | TFC | TIM3 clear bit of overflow flag 0: Invalid 1: Clear |
| 15 ~ 1 | Reserved | - |
| 0 | TF | TIM3 overflow flag bit 0: Timer3 has no overflow and can be cleared by software. 1: Timer3 overflows and is set to 1 by hardware |



20. 16 bit Pulse Width Modulation (PWM0/1/2/3)

20.1 Feature

- 4x2 complementary outputs with dead zone control
- Provide overflow interrupt per pwm_x (x = 0,1,2,3) cycle
- Two output polarities can be selected independently
- Provide error detection function to turn off PWM output in an emergency
- Provide protection register to protect important registers from interference and errors

The SH30F9/SA0 series integrates four 16-bit PWM modules. The PWM module can generate PWM waveforms with adjustable cycle and duty cycle respectively.

The PWM timer also provides four interrupt sources for PWM_x (x = 0,1,2,3), which generate interrupts during each PWM cycle. This allows the user to change the cycle or duty cycle of the next cycle in each PWM cycle.

20.2 PWM Enable Control

When `PWMEN@PWMx->CR` is cleared, the PWM output High resistance state

20.3 PWM protection register

PWM protection register is used to control the change of PWM control register, PWM period register, PWM duty cycle register and PWM dead time control register. Only when the data in the protection register is 0x5AA5, the contents of these registers can be modified, otherwise they cannot be modified.

This register can enhance the anti-interference ability of SH30F9/SA0 series.

20.4 16 bit PWM timer

SH30F9/SA0 series consists of four 16 bit PWM modules. PWM module can produce PWM waveform whose period and duty cycle are adjustable respectively. `TCK[2:0]@PWMx_CR` register is used to control the clock of PWM module, `PP[15:0]@PWMx->PWMPR` register is used to control the period of PWM output waveform, `PD[15:0]@PWMx->PWMDR` register is used to control the duty cycle of PWM module output waveform.

These three registers can be modified during the PWM output permission period, but the changes will not take effect until the next PWM cycle.

Be careful:

- (1) When PWM_x output is off (FLT occurs), PWM_xA and PWM_xB (x=0,1,2,3) output fixed low level (PWMSA=0, PWMSB=1) or high level (PWMSA=1, PWMSB=0).
- (2) Once the high/low level of FLT_x pin is detected, the internal state will be maintained and the PWM_x output will be turned off.
- (3) When the FLT_x input signal is valid, the FLTS bit cannot be cleared. The FLTS status bit can only be cleared when the FLT_x input signal disappears.

PWM_x output period = `PP[15:0]` X PWM_x clock.

When `PP[15:0] = 0000H`, if PWMSA=0, no matter what the PWM_x duty cycle is, PWM_xA (x=0,1,2,3) outputs low level.

When `PP[15:0] = 0000H`, if PWMSA=1, no matter what the PWM_x duty cycle is, PWM_xA (x=0,1,2,3) outputs high level.

PWM_x output duty cycle = `PD[15:0]` X PWM_x clock.

When `PD[15:0] = 0000H`, if PWMSA=0, no matter what the PWM_x period is, PWM_xA (x=0,1,2,3) outputs low level.

When `PD[15:0] = 0000H`, if PWMSA=1, no matter what the PWM_x period is, PWM_xA (x=0,1,2,3) outputs high level.

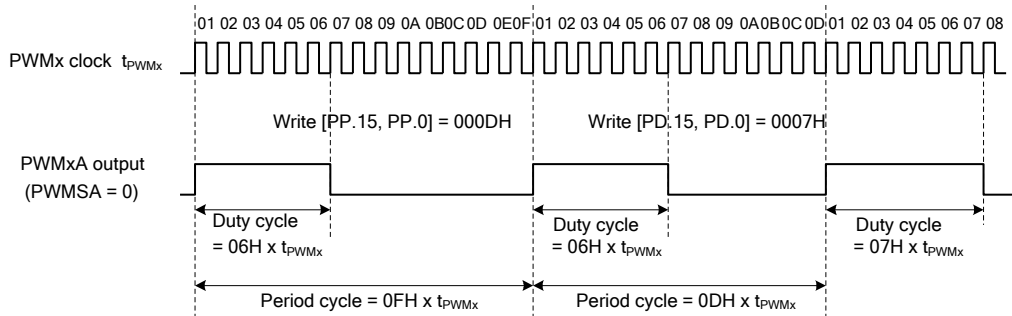
When `PP[15:0] ≠ 0000H` and `PP[15:0] ≤ PD[15:0]`, if PWMSA=0, PWM_xA (x=0,1,2,3) outputs high level.

When `PP[15:0] ≠ 0000H` and `PP[15:0] ≤ PD[15:0]`, if PWMSA=1, PWM_xA (x=0,1,2,3) outputs low level.



Programming attentions:

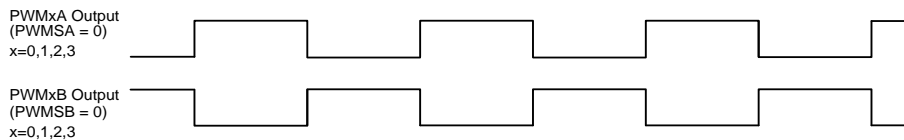
- (1) Set PWMx-> PWMLOCK (x = 0,1,2,3, the same below) register content to 0x5AA5; select PWMx module clock source.
- (2) Set the PWM period/duty cycle by writing the appropriate value to the PWM period control register (PWMx->PWMPR) or PWM duty cycle register (PWMx->PWMDR).
- (3) Select the PWMxA and PWMxB output modes (active high level or active low level) by setting the PWMSA and PWMSB bits of the PWMx control register (PWMx->CR).
- (4) If the PWMx period or duty cycle needs to be changed, the operation process is as described in step 2 or step 3. The value of the modified overload counter is valid at the beginning of the next cycle.
- (5) To avoid interference, set the data in PWMx->PWMLOCK register not equal to 0x5AA5.



Example of PWMx output period or duty cycle change

20.5 PWMxB complementary output (x = 0,1,2,3)

As shown in the figure below, generally, when no dead time is inserted, the period and duty cycle of PWMxB output waveform are the same as PWMxA. By setting PWMSB, PWMxB (x = 0,1,2,3) can be set to the complement output waveform of PWMxA. When the EPWMxB bit is set to 1 in PWMx control register, the effective level of PWMxB is set by PWMSB.

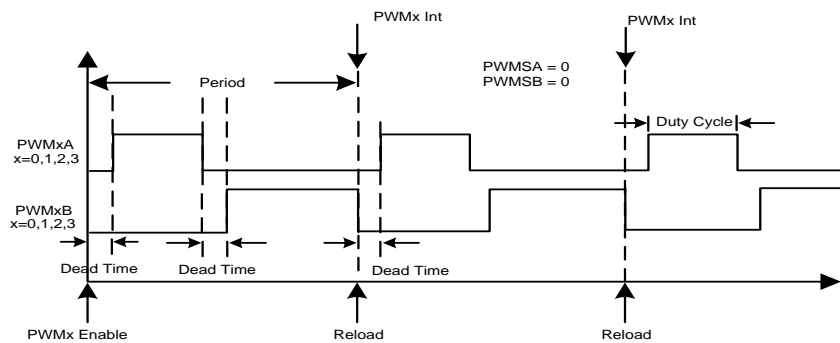


PWMxA and PWMxB pin output waveform

20.6 Dead Time

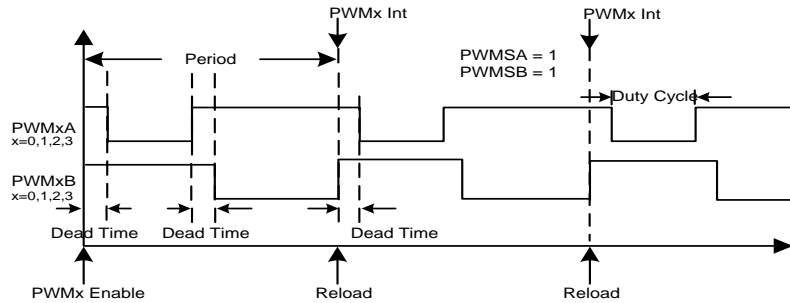
SH30F9/SA0 series PWMx provides dead time control function.

When PWMSA = 0 and PWMSB = 0, the generation of dead time is shown in the figure below.





When PWMSA = 1 and PWMSB = 1, the generation of dead time is shown in the figure below.



By writing the PWMx dead time control register, the dead time is generated between PWMxA (x = 0,1,2,3) and PWMxB (x = 0,1,2,3).

PWMxA has the same cycle as PWMxB.

Note:

- (1) The dead time must be set before the PWMx (x = 0,1,2,3) output allowing, otherwise, the dead time will not be changed. So in order to modify the dead time, first unlock the PWMxLOCK register (PWMx->PWMLOCK = 5AA5H) and disable PWMx output, then change the dead time and allow PWMx output. Finally, in order to ensure that the PWMx correlation register is not affected by interference, the content of the PWMLOCK register is modified not equal to 5AA5H.
- (2) In order to generate a dead time, make sure $(\text{PWMxA period} - \text{PWMxA duty cycle}) > 2 * (\text{dead time of PWMxB})$. Otherwise, PWMxB outputs high level when PWMSB = 1 and low level when PWMSB = 0.
- (3) DT[15:0]@PWMx->PWMD TR register is used to control dead time. Its time base is APB0 clock, while the time base of period and duty cycle is controlled by TCK[2:0]@PWMx->CR, with a minimum of 1 APB0 clocks.



20.7 Registers

PWM0~3 Module Register list (Base Address:0x4000 1400)

| Address | Register | Description |
|---------------------|----------|---|
| 0x4000 1400+0x400*x | CR | PWM control register(PWM0~3 corresponds to x: 0~3) |
| 0x4000 1404+0x400*x | PWMLOCK | PWM protection register(PWM0~3 corresponds to x: 0~3) |
| 0x4000 1408+0x400*x | PWMPR | PWM cycle control register(PWM0~3 corresponds to x: 0~3) |
| 0x4000 140C+0x400*x | PWMDR | PWM duty cycle control register(PWM0~3 corresponds to x: 0~3) |
| 0x4000 1410+0x400*x | PWMDTR | PWM dead time control register(PWM0~3 corresponds to x: 0~3) |
| 0x4000 1414+0x400*x | PWMINTF | PWM interrupt flag and clear register(PWM0~3 corresponds to x: 0~3) |

20.7.1 PWMx control register (PWMx_CR)

Offset Address: 0x0000+0x400*x

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|------------|----------------|-----------|------------|-----------|-----------|----------|----------|----------|-----|-----------|-----|
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | FLT DMA | PW MDM A | PW MIE | EFL TIE | PW MSA | PW MSB | EFL T | FLT C | TCK[2:0] | | PW MEN | |
| 0 | 0 | 0 | 0 | RW | RW | RW | RW | RW | RW | RW | RW | RW | | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 12 | Reserved | - |
| 11 | FLTDMA | FLT DMA enable bit 0: Disable DMA request 1: Enable DMA request |
| 10 | PWMDMA | PWM overflow DMA enable bit 0: Disable DMA request 1: Enable DMA request |
| 9 | PWMIE | PWM overflows interrupt enable bits 0: Disable PWM overflow interrupts 1: Enable PWM to overflow interrupts |
| 8 | EFLTIE | PWM pin level interrupt enable bit 0: Disable PWM pin error interrupts 1: Enable PWM pin fault to interrupt |
| 7 | PWMSA | PWMXA output mode 0: PWMXA outputs high level during duty cycle and low level after duty cycle overflow 1: PWMXA outputs low level during duty cycle and high level after duty cycle overflow |
| 6 | PWMSB | PWMXB output mode 0: PWMXB outputs low level during duty cycle and high level after duty cycle overflow 1: PWMXB outputs high level during duty cycle and low level after duty cycle overflow |



| | | |
|-------|-----------------|---|
| 5 | EFLT | PWM fault detection input pin enable bit 0: Disable 1: Enable |
| 4 | FLTC | FLT pin level configuration bit 0: When FLT is low level, PWM output is off 1: When FLT is high, PWM output is off |
| 3 ~ 1 | TCK[2:0] | 16-bit PWM clock source selection bit 000: Bus clock / 2 001: Bus clock / 4 010: Bus clock / 8 011: Bus clock / 16 100: Bus clock / 1 101: Bus clock / 32 110: Bus clock / 64 111: Bus clock / 256 |
| 0 | PWMEN | PWM module enable bit 0: Disable 1: Enable |

20.7.2 PWMx protection register (PWMx_PWMLOCK)

Offset Address: 0x0004+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PWMLO[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PWMLO[15:0] | PWM lock bits Only when PWMLO = 0x5AA5, other PWMx registers can be modified |

20.7.3 PWMx cycle control register (PWMx_PWMPR)

Offset Address: 0x0008+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PP[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PP[15:0] | 16 bit PWM cycle register PWM output cycle = PP[15:0] × PWM clock source cycle |



20.7.4 PWMx duty cycle control register (PWMx_PWMDR)

Offset Address: 0x000C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PD[15:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PD[15:0] | 16 bit PWM duty cycle register PWM output aspect ratio = PD[15:0] × PWM clock source cycle |

20.7.5 PWMx dead time control register (PWMx_PWMDTR)

Offset Address: 0x0010+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DT[15:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | DT[15:0] | 16 bit PWM dead time control The dead time is DT[15:0] × APB0 clock cycle |

20.7.6 PWMx interrupt flag and clear register (PWMx_PWMINTF)

Offset Address: 0x0014+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------------|-----------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | PW MIF C | FLT SC |
| - | | | | | | | | | | | | | | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | PW MIF S | FLT S |
| - | | | | | | | | | | | | | | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 18 | Reserved | - |
| 17 | PWMIFC | PWM interrupt flag clear bit 0: Invalid 1: clear |
| 16 | FLTSC | FLT status clear bit 0: Invalid 1: clear |
| 15 ~ 2 | Reserved | - |
| 1 | PWMIF | PWM interrupt flag bit 0: PWM cycle counter does not overflow, need to clear 0 by software 1: PWM cycle counter overflow, which is set by hardware |
| 0 | FLTS | FLT status bit 0: PWM is in normal state, which needs to be cleared by software 1: PWM is abnormal, PWM output is off, which is set by hardware |



21. Programmable counter array (PCA, PCA0~PCA3)

21.1 Introduction

The programmable counter array PCA provides enhanced timer functions, which can provide four enhanced functions: input capture, comparison matching (output), square wave output with adjustable frequency, and PWM modulation output. SH30F9/SA0 series built-in 4 PCA modules (PCA0/1/2/3), each PCA module provides 2 input and output channels.

21.2 Feature

- 16-bit timer
- 2 independent input and output channels
- Support input capture, compare match (output)
- Support square wave output with adjustable frequency
- 16/8 bit PWM, 4 output modes
- PCA0/1 and PCA2/3 can be cascaded to form a 32-bit PCA module

21.3 Function description

The programmable counter array PCA consists of a basic counting unit, two 16-bit capture/compare modules. Each capture/compare module has an independent pin (PCAxA/PCAxB), which can be used as Capture signal input or waveform output.

Note: The small subscript x is the PCA serial number, such as PCAx (x=0, 1,2,3). PCAx will be used uniformly below, and the value of x will not be explained.

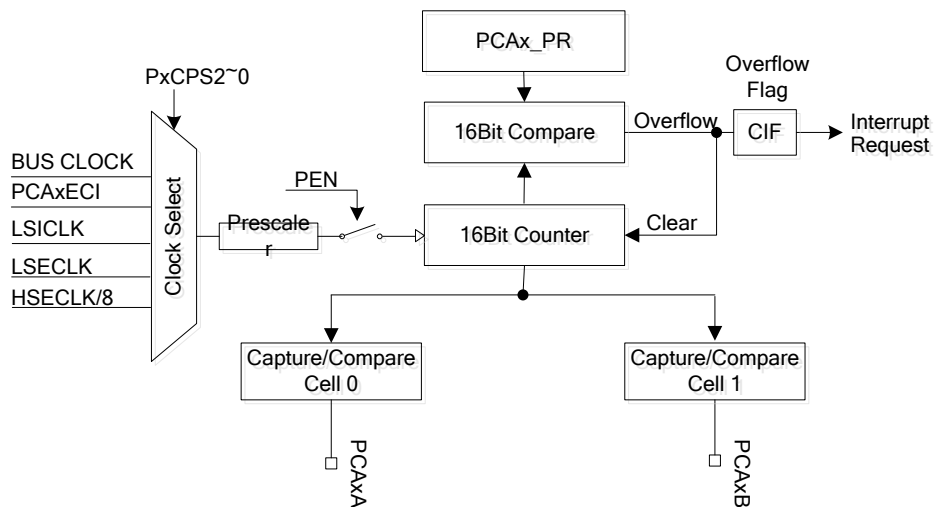


Figure 21-1 PCA principle block diagram

21.3.1 Counting unit

PCAx is composed of a 16-bit period register PCAx_PR, a 16-bit counting register PCAx_CNT and a 16-bit prescaler register PCAx_PSC to form a basic counting/timing unit.

The 16-bit counter has two counting modes, namely up counting and center aligned counting. When the SDEN bit in the register PCAx_CFGR is 0, the counter works in up-counting mode. When SDEN is 1, the counter works in center-aligned counting mode.

In the up-counting mode, the counter counts up from 0 to the value of the period register PCAx_PR, and then restarts counting from 0 and generates a counter overflow event. At the same time, the CIF bit in the status register PCAx_SR will be set by hardware. If the CIE bit in the PCAx_CFGR register is set to 1 by software, an interrupt will be generated.



SH30F9/SA0 Series

In the center-aligned counting mode, the counter starts counting up from 0 to the value of the period register PCAx_PR, and then counts down to 1 to complete a cycle. The next cycle starts counting from 0 again and continues to loop. When the counter value is the PR value of the period register, a period match event will be generated, and the PIF bit in the status register PCAx_SR will be set by hardware. If the PIE in the CFGR register is 1, an interrupt will be triggered. When the counter value reaches 0, a counter overflow event will be generated, and the CIF bit in the status register PCAx_SR will be set by hardware. If the CIE bit in the CFGR register is set to 1 by software, an interrupt will be triggered.

Note: From the shape of the generated waveform, the upward counting mode can also be called the sawtooth wave mode, and the center-aligned counting mode can also be called the triangle wave mode.

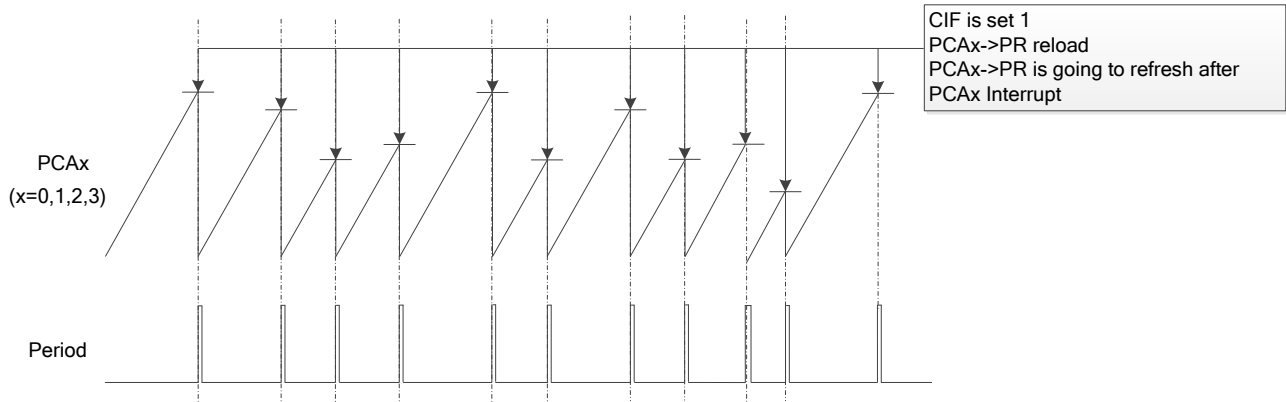


Figure 21-2 Waveform diagram of PCAx counter in up counting (sawtooth wave) mode

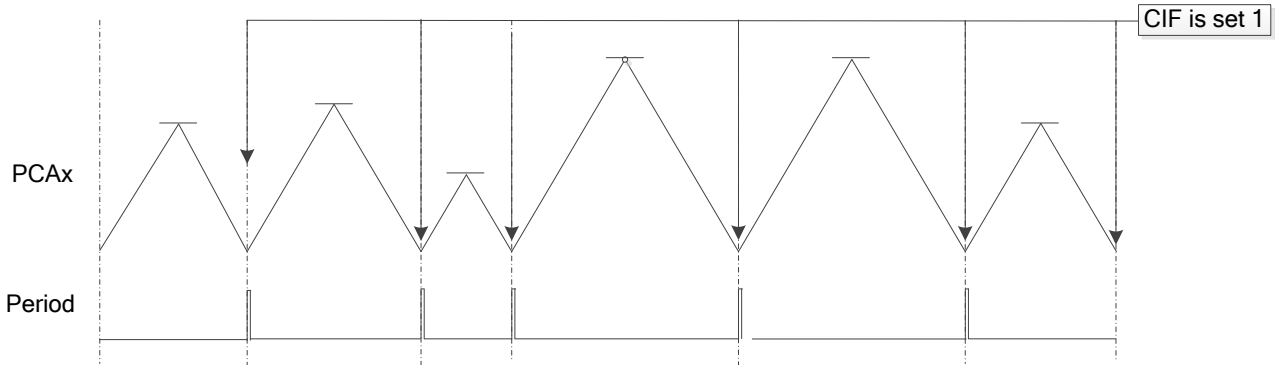


Figure 21-3 Waveform diagram of PCAx counter in center-aligned counting (triangular wave) mode

21.3.2 Counter clock source

The clock source of the counter can be selected by setting CPS[2:0] in register PCAx_CFGR, which are PCLK clock, PCAx_ECI pin clock input, low-frequency RC 128KHz clock source, LSECLK and HSECLK/8 as shown in the following table.

| CPS[2] | CPS[1] | CPS[0] | |
|--------|--------|--------|---|
| 0 | 0 | 0 | PCLK clock |
| 0 | 0 | 1 | External pin input PCAxECl clock falling edge |
| 0 | 1 | 0 | LSI(RC 128KHz) |
| 0 | 1 | 1 | LSECLK (Cry32K) |
| 1 | 0 | 0 | HSECLK/8 |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | Reserved |



When CPS[2:0]=001, the external ECI pin input is selected as the clock source, and the filter parameters of the input clock source can be selected by setting the ECF[1:0] bit segment in the CFGR register. The filter parameters are no filter, 8PCLK, 16PCLK, 32PCLK.

Note: The bus clock cycle must not be less than 4 times the counting clock cycle (except when the counting clock is the bus clock), otherwise the PCAx counter will not count correctly.

21.4 Working mode

PCAx has 2 capture/compare modules, each capture/compare module corresponds to 1 channel and can be configured independently. You can select the module to work in one of the following four working modes by configuring SM[1:0] in the respective module register PCAx_CCMR: input capture, compare match output, frequency output, and PWM output mode. The FS[1:0] in the PCAx_CCMR register can be further configured for each working mode. Only when SM[1:0]=10, the setting of FS[1:0] is invalid.

The working mode selection is shown in the following table:

Table 21-1 PCAx mode selection table

| Mode | SDEN | SM[1] | SM[0] | FS[1] | FS[0] | Function Description |
|--------|------|-------|-------|-------|-------|--|
| Mode0 | 0 | 0 | 0 | 0 | X | Positive edge trigger capture (sawtooth wave) |
| | | | | 1 | 0 | Negative edge trigger capture (sawtooth wave) |
| | | | | 1 | 1 | Any edge trigger capture (sawtooth wave) |
| Mode1 | 0 | 0 | 1 | 0 | X | Continuous comparison match (output) (sawtooth wave) |
| | | | | 1 | X | Single comparison match (output) (sawtooth wave) |
| Mode2 | 0 | 1 | 0 | X | X | Frequency output (sawtooth wave) |
| Mode3 | 0 | 1 | 1 | 0 | 0 | 8-bit PWM1 mode (sawtooth wave) |
| | | | | 0 | 1 | 16-bit PWM2 mode (sawtooth wave) |
| | | | | 1 | 0 | 16-bit PWM3 mode (triangular wave) |
| | 1 | | | 1 | 1 | 16-bit PWM4 mode (triangular wave) |
| Others | | | - | | | The PCAx counter counts correctly, but the compare/capture module does not work. |

Note: X represents any value;

When PCAx is set to one of two waveforms (sawtooth or triangle), the other waveform mode of the compare capture module is invalid even if it is configured. (Note: If it is configured as sawtooth wave mode, the configuration of triangle wave related registers is invalid)

21.4.1 Input Capture Mode

When SM[1:0] in register PCAx_CCMR=00, the capture/compare module works in input capture mode, and FS[1:0] in register PCAx_CCMR is used to select the type of level change that triggers the capture. When FS[1:0]=0X, the rising edge capture; when FS[1:0]=10, the falling edge capture; when FS[1:0]=11, the rising/falling edge can trigger capture. When the capture occurs, the value of the counter PCAx_CNT will be loaded into the capture/compare register PCAx_CCR of the corresponding module. When a capture occurs, the capture/compare flag CCIF in PCAx_SR is set. If the CCIE bit of the corresponding module is 1, an interrupt request will be generated. The CCIF flag bit cannot be automatically cleared by hardware, it must be cleared by writing 1 to the corresponding CCIFC bit in the PCAx_SR register by software.

In the input capture mode, you can read the TCP bit in the register PCAx_CCMR to determine whether the current capture is a rising edge capture or a falling edge capture. When TCP=0, it means that the falling edge capture is currently taking place; when TCP=1, it means that the rising edge capture is currently taking place.



When a capture occurs, you can choose whether to clear the counter value by setting the CC bit in the register PCAx_CCMR. If CC is 0, the counter value is not cleared when a capture occurs. If CC is 1, the hardware saves the current counter value to the PCAx_CCR register when a capture occurs and then clears the counter value to 0. When the input signal is actually captured, if CC is 1, the hardware clears the counter value, then the value saved in the CCR register is the actual pulse width of the input signal.

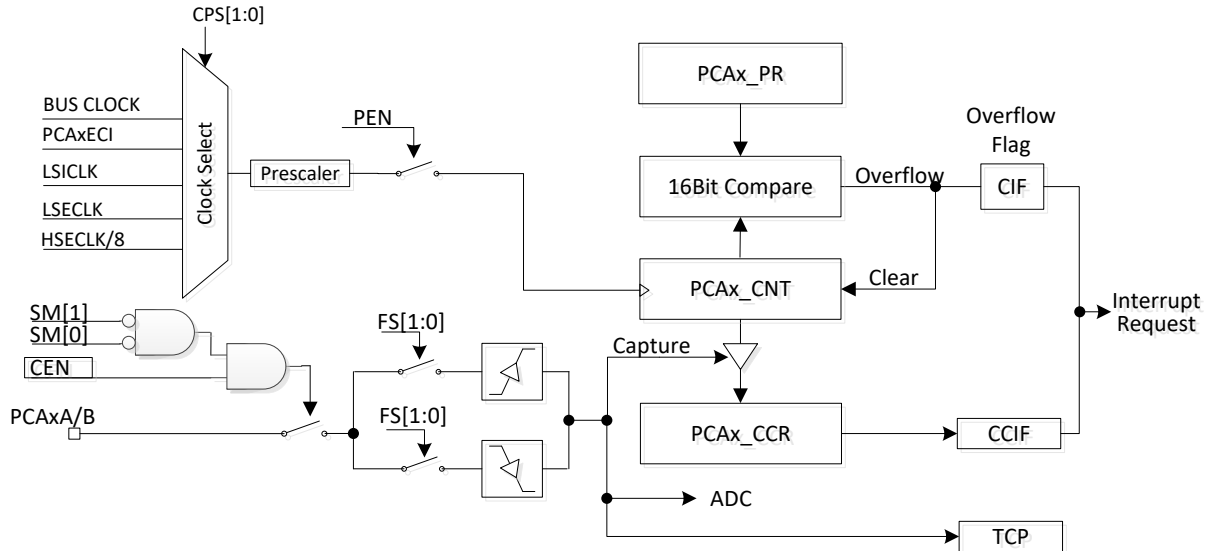


Figure 21-4 Block diagram of PCA capture method

21.4.2 Compare match output

By setting SM[1:0]=01 in the CCMR register, the corresponding module can be made to work in the compare match mode. Set FS[1:0]=0X to achieve continuous matching, and the counter will count repeatedly from 0 to PR value. When a match occurs, the level of the module output pin PCAxA/B will be inverted, and the capture/compare flag CCIF in the register PCAx_SR will be set to 1, if the CCIE of the corresponding module is 1, an interrupt will be triggered. When FS[1:0]=1X, this mode is a single match, that is, the counter counts from 0 to the PR value and then overflows and counts back to 0 and keeps counting repeatedly, but the level of the output pin PCAxA/B is also The flip occurs only during the first match, and then the level after the flip does not change. The flag CCIF of the corresponding module will be set to 1, if CCIE is 1, an interrupt will be triggered. Whether it is continuous matching or single matching, the initial level of the output waveform can be changed through the TCP bit in PCAx_CCMR. When TCP is 0, the initial level of the output waveform is high, and when the TCP bit is 1, the output waveform is initial The level is low.

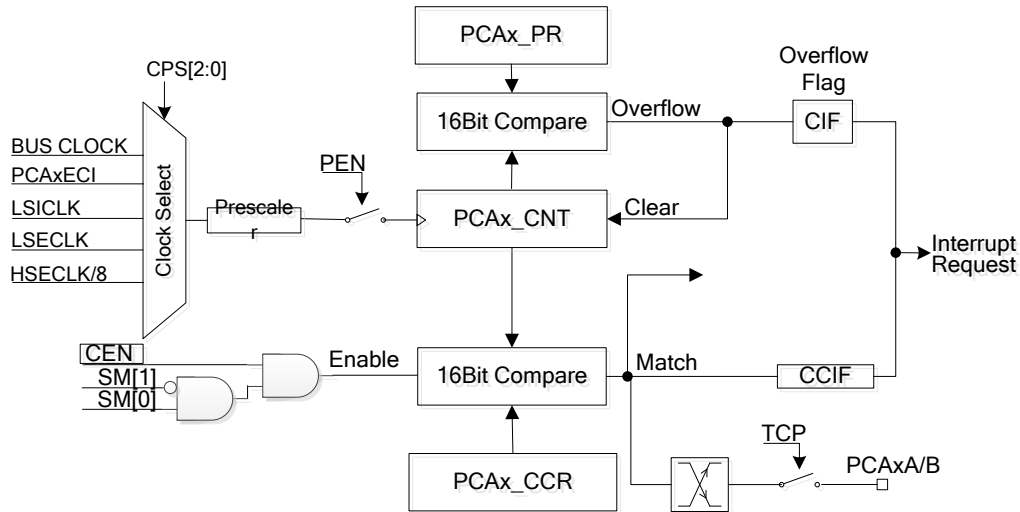


Figure 21-5 Block diagram of comparison matching method

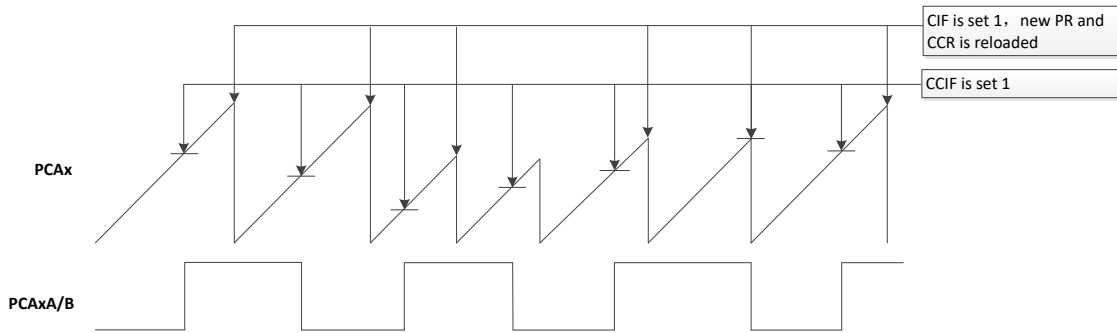


Figure 21-6 Waveform diagram of comparison matching output mode

In addition, you can force a comparison match to be generated by setting the FCO_n bit of the PCAx_FORCE register. However, such a match is not a true match. It will neither set the compare match flag nor affect the register value, but only cause a level flip on the PCAxA/B output pin. When a forced match takes effect, the FCO_n bit will be automatically cleared by hardware.

21.4.3 Frequency output mode

By setting SM[1:0]=10 in the register CCMR, the corresponding module can be operated in the frequency output mode, and the waveform is output through the pin PCAxA/B. In this mode, the period register PR, the counter CNT and the capture/output register CCR are divided into two 8-bit registers by 16 bits, which are distinguished by high-order register (PRH) and low-order register (PRL).

In this mode, the lower 8 bits of the period register PRL is fixed to 0xFF and cannot be modified. The 16 digits composed of PRH and PRL are the period value. If PRH is set to 0x05, the period is 0x05FF. When the counter reaches this value When the PCAx overflow flag CIF is set to 1, if the counter overflow interrupt enable bit CIE is 1, it will trigger an interrupt.

When the lower 8 bits of CCRL of the capture/compare register are compared with the lower 8 bits of CNTL of the counter, if the two match, the level of the PCAxA/B pin will be reversed and the value of the upper 8-bit register CCRH will be added to CCRL as an offset value and it will take effect immediately. When CNTL continues to count to CCRL+CCRH, a match occurs again, and the output pin level will flip. As long as a match occurs, the value of CCRL will be added to the original value and the value of CCRH will be updated immediately.

PCAx works in the frequency output mode, only the low 8 bits will match, so when the value of CCRL continues to accumulate more than 0xFF, it will overflow, and the low 8 bits after overflow will continue to be compared with the counter CNTL.



The initial level of the output pin PCAxA/B can be selected according to the TCP bit. When TCP is 0, the initial level is high, and when TCP is 1, the initial level is low.

The frequency of the square wave generated in this mode is $F = F_{PCA_x} / (2 \times CCRH)$, where F_{PCA_x} is the clock divided by the PCAx prescaler. If CCRH is 0, it is equivalent to 256.

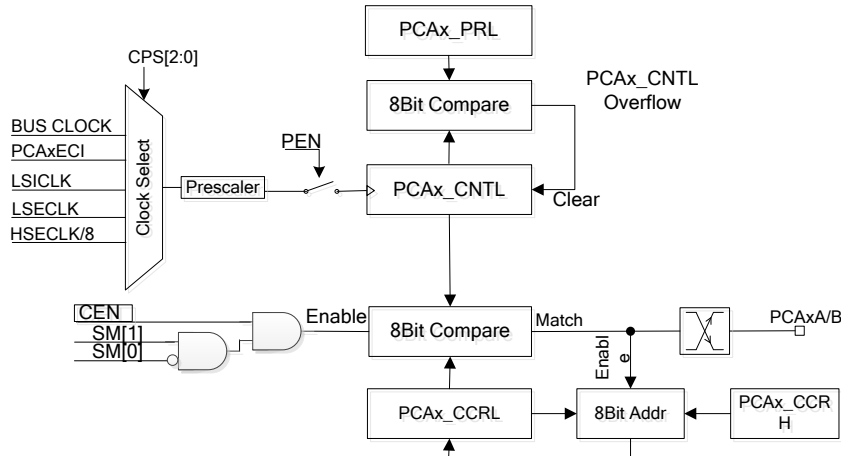


Figure 21-7 Block diagram of frequency output mode

21.4.4 PWM output

Each module of PCAx can independently generate pulse width modulation (PWM) output. Configure SM[1:0]=11 to make the compare/capture module work in PWM mode. In this mode, the compare/capture module can output the following 4 PWM waveforms by configuring the FS[1:0] bits.

| FS[1] | FS[0] | Function Description |
|-------|-------|-------------------------------|
| 0 | 0 | 8 bit PWM1 (sawtooth wave) |
| 0 | 1 | 16 bit PWM2 (sawtooth wave) |
| 1 | 0 | 16 bit PWM3 (triangular wave) |
| 1 | 1 | 16 bit PWM4 (triangular wave) |

■ 8-bit PWM1 mode

When the compare/capture module is working in the 8-bit PWM function, the lower 8-bit PCAx_CNTL of the counter PCAx_CNT counts up from 0x0 to PCAx_PRL (the counter is in sawtooth wave mode), when PCAx_CNTL overflows (from 0xFF to 0x0, this mode PCAx_PRL is fixed to 0xFF), the value saved in PCAx_CCRH is automatically loaded into PCAx_CCRL, this process does not require software intervention. The maximum value of the count can be configured through PCAx_PRL, that is, if PCAx_PRL is 0x35, the counter needs to count to 0x35FF before it overflows. At this time, the CIF bit is set. If CIE is 1, an interrupt will be triggered.

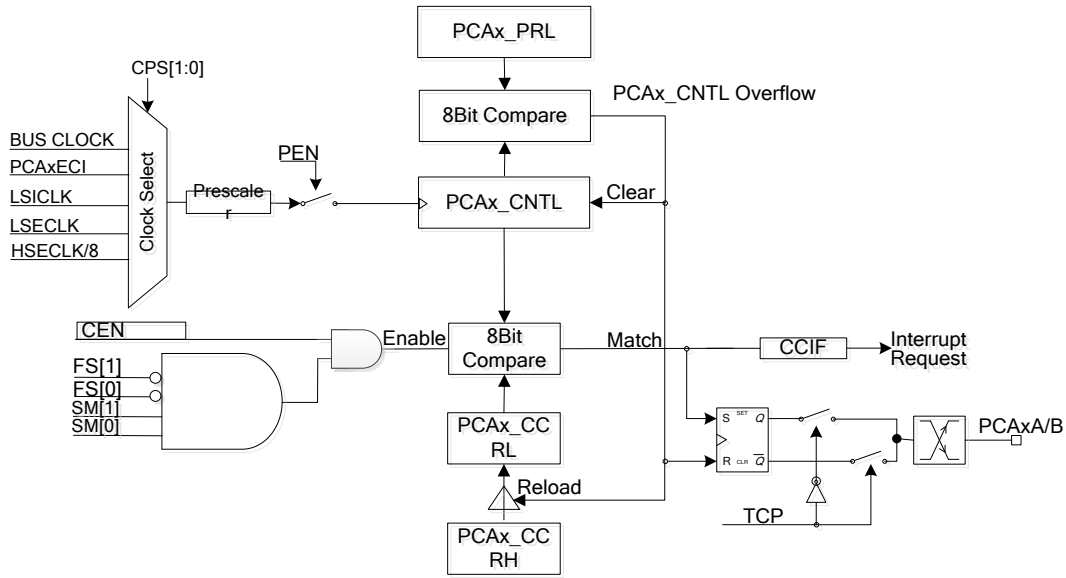


Figure 21-8 8-bit pulse width modulator (PWM) principle block diagram

The module's capture/compare register `PCAx_CCRH` is used to change the duty cycle of the PWM output signal. When `TCP=0`, when the `PCAx` counter low byte `PCAx_CNTL` and `PCAx_CCRL` have the same value, the output on the `PCAxA/B` pin changes from high to low; when the counter `PCAx_CNTL` overflows, the output on the `PCAxA/B` pin changes from low to high; when `TCP=1`, the `PCAxA/B` pin outputs a waveform with the opposite polarity.

The duty cycle in this mode is $\text{Duty} = \frac{256 - (\text{CCRH} + 1)}{256}$.

As shown in the figure below, cycle 1 is the default value of the `PCAxA/B` pin when it is powered on; when the cycle 1 is `CCRL = 00H`, the `PCAxA/B` pin output level; When cycles 2 to 4 are `CCRL = 01H, 80H, FEH`, the waveform of the corresponding duty cycle of the `PCAxA/B` pin; cycle 5 is that the `CCRL = FFH` `PCAxA/B` pin is pulled high.

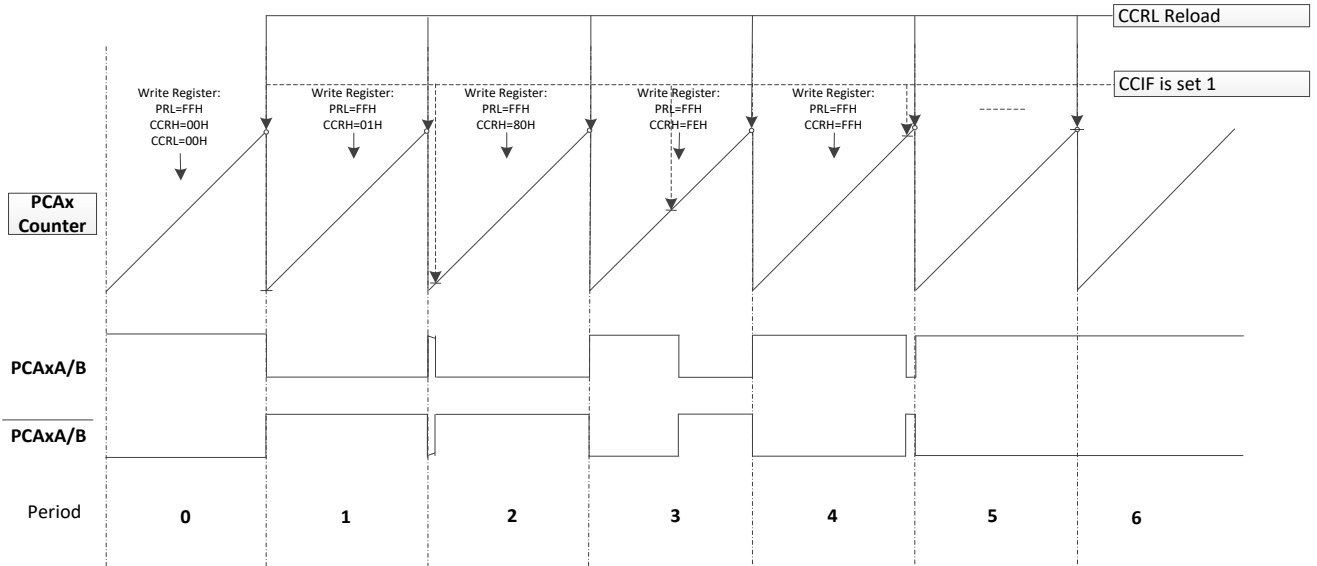


Figure 21-9 8 bit PWM modulation waveform



16-bit PWM2 mode

In this mode, when TCP = 0, when the PCAx counter matches the module's matching register CCR value, the output on the PCAx/A/B pin changes from high to low; when the counter overflows, The output on the PCAx/A/B pin changes from low to high. When TCP=1, the PCAx/A/B pin outputs a waveform with the opposite polarity. The realization principle block diagram is shown in Figure 21-10 below.

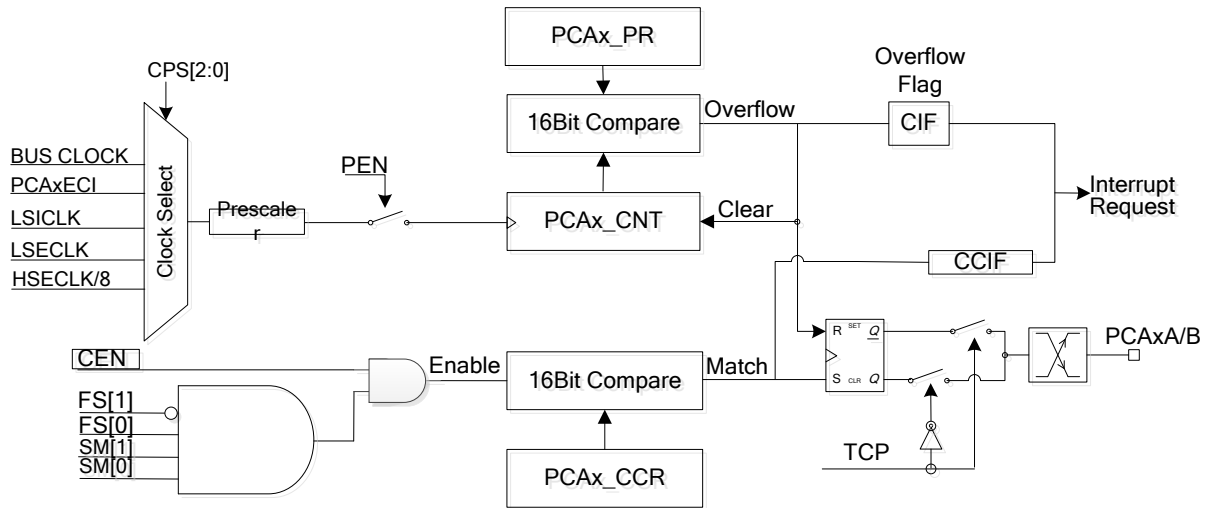


Figure 21-10 16-bit PWM2 mode block diagram (sawtooth mode)

The PWM duty cycle in this mode $Duty = (PCAx \rightarrow PR - PCAx \rightarrow CCR) / (PCAx \rightarrow PR + 1)$

The output waveform is shown in the figure below, cycle 1 is the default value of the PCAx/A/B pin when power is on; cycle 1 is the output level of the PCAx/A/B pin when CCR = 0000H; cycles 2~4 are CCR respectively = 0001H, 8000H, FFFEh, PCAx/A/B pin corresponding duty cycle waveform; cycle 5 is when CCR = FFFFH, pin PCAx/A/B is pulled high; cycle 6~8 is when PCAx_PR and CCR values are changed, PCAx/A/B pin output waveform.

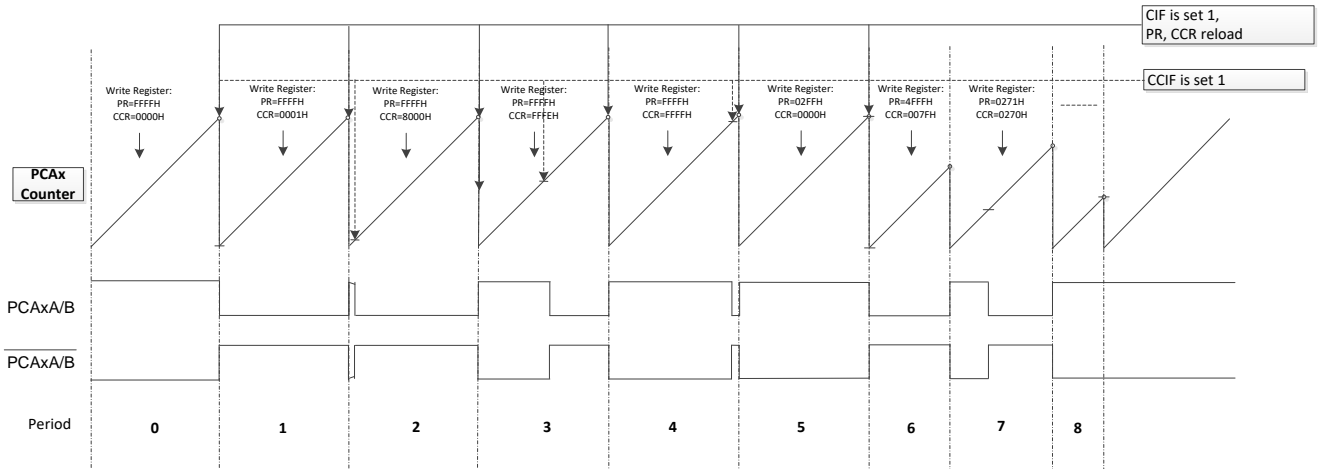


Figure 21-11 16 bit PWM2 waveform



16-bit PWM3 mode

In this mode, the TCP bit in the PCAx_CCMR register can be used to select the level of the Duty zone. When TCP=0, when the timer counts from 0 to PCAx_PR, if the counter value matches PCAx_CCR, the output on the PCAx/A/B pin will change from high to low; and when the timer goes from the period value to If the counter value matches PCAx_CCR when counting at 0x0, the output on the PCAx/A/B pin will change from low to high. When TCP=1, PCAx/A/B pins output waveforms with opposite polarity. The realization principle block diagram is shown in Figure 21-12 below. When a match occurs, the match flag bit CCIF is set to 1, if the interrupt enable bit CCIE is 1, a PCAx interrupt will be generated. When PCAx counts down from the period value back to 0x0 overflow, the PCAx interrupt flag CIF will also be set to 1, if the interrupt enable bit CIE is 1, a PCAx interrupt will be generated.

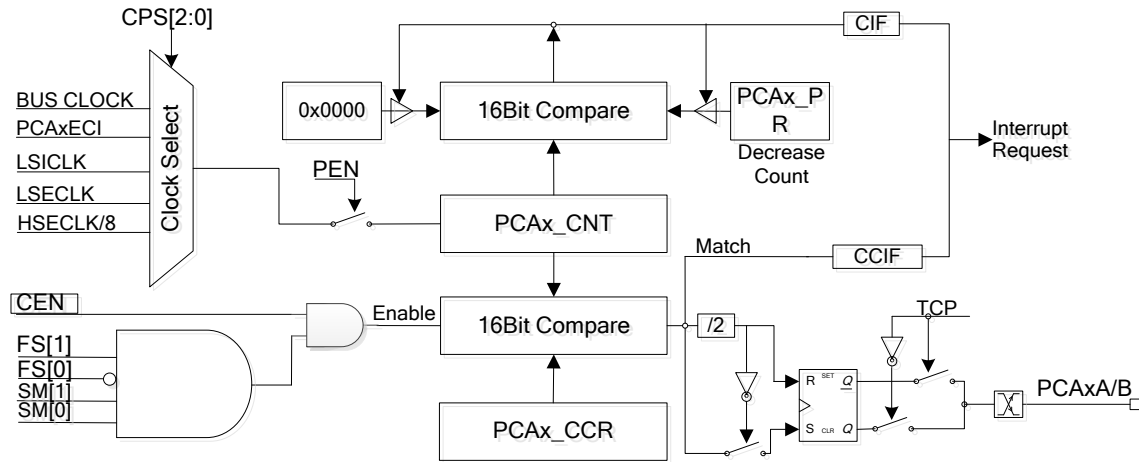


Figure 21-12 16 bit PWM3 principle block diagram

In this mode, modifying the values of PR and CCR will only take effect when the counter reaches the PR value. Therefore, if the PR value and the CCR value are modified when the counter counts from 0 to PR, the CCR value can be used in the half cycle when the counter returns from the PR value to 0x0, and the period value should be used when the counter returns from PR to 0x0 After that, the modified PR value in a new counting cycle is used. Therefore, no matter the PR value is modified at any time, the value of the counter will not change suddenly.

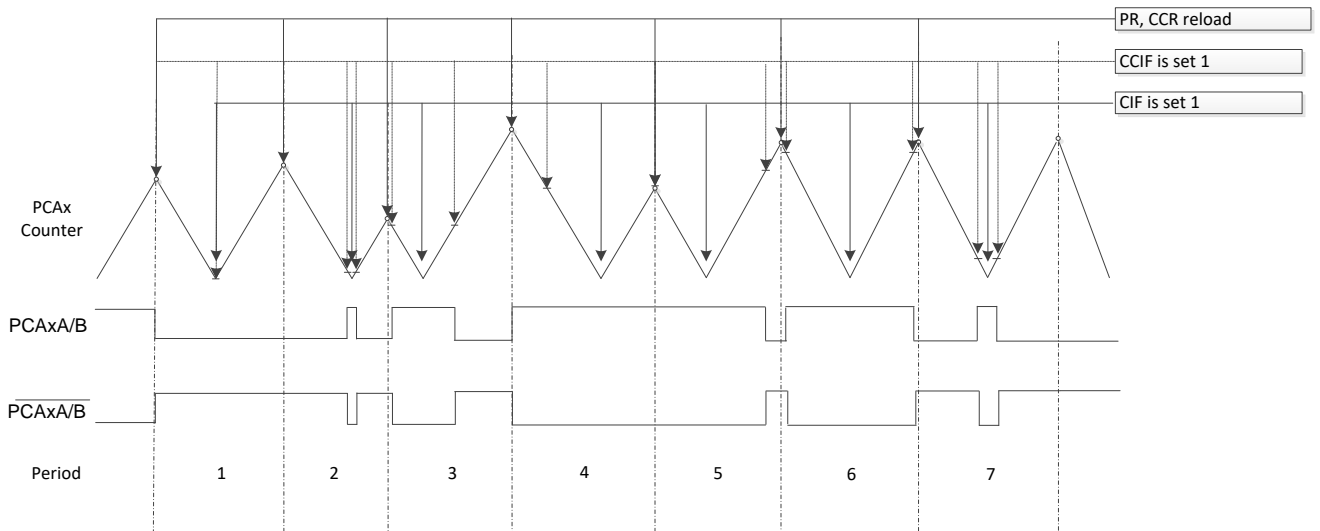


Figure 21-13 16 bit PWM3 waveform



SH30F9/SA0 Series

The resolution of PWM3 mode can be defined by PCAx_PR, the minimum resolution is 2 bits (PCAx_PR is set to 0x0003), and the maximum resolution is 16 bits. The number of PWM resolution bits $R = \log_2(\text{PCAx_PR} + 1)$, and the PWM frequency $F = F_{\text{PCAx}} / (2 * \text{PCAx_PR})$.

16-bit PWM4 mode

When TCP=0, when the timer counts to the period value PR, if the counter value PCAx_CNT matches PCAx_CCR, the output on the PCAx/A/B pin changes from high to low; and when the timer counts to 0x0 If the counter value PCAx_CNT matches PCAx_CCR, the output on the PCAx/A/B pin changes from low to high. When TCP=1, PCAx/A/B pins output waveforms with opposite polarity. When a match occurs, the match flag PCAx/A/B is set to 1, if the interrupt is enabled, a PCAx interrupt will be generated. When the counter counts down from PR to 0x0 overflow, the PCAx interrupt flag CIF will also be set to 1, if the interrupt is enabled, a PCAx interrupt response will be generated.

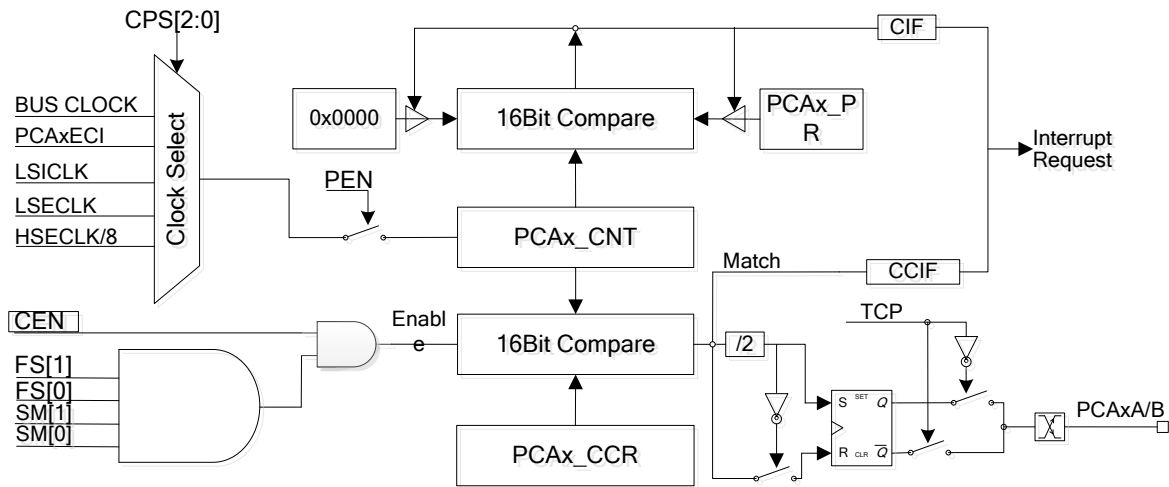


Figure 21-14 16 bit PWM4 principle block diagram

The difference between PWM4 and PWM3 modes is that the update time of PCAx_PR and PCAx_CCR are different, so PWM3 can output asymmetrical waveforms, while PWM4 always outputs symmetrical waveforms. This is because PWM4 updates the CCR and PR register values at 0x0000, while PWM3 updates the CCR and PR register values at the PR point.

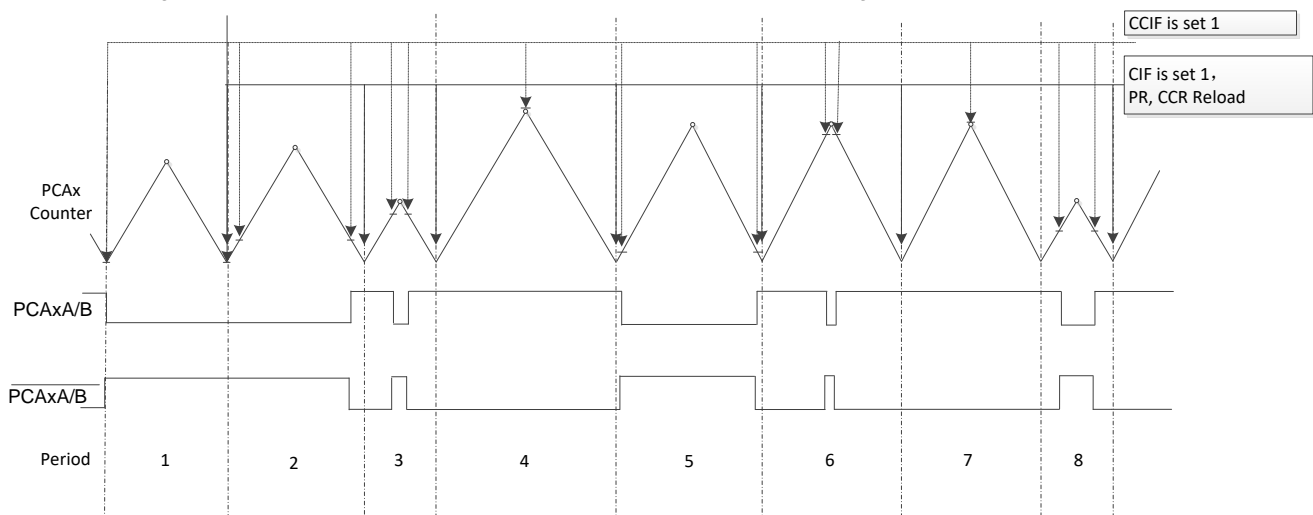


Figure 21-15 16-bit PWM4 waveform



SH30F9/SA0 Series

The PWM resolution of PWM4 mode is defined by PCAx_PR. The minimum resolution is 2 bits (PCAx_PR is set to 0x0003), and the maximum resolution is 16 bits (PCAx_PR is set to 0xFFFF).

$$\text{Output PWM frequency } F = F_{\text{PCAx}} / (2 * \text{PCAx_PR})$$

$$\text{Output duty cycle } \text{Duty} = (\text{PCAx_PR} - \text{PCAx_CCR}) / \text{PCAx_PR}$$

21.5 Cascade operation

The cascade operation is the function of connecting two 16-bit PCAs into a 32-bit PCA, thereby expanding the setting range of the PR value of the PCA and the comparison value CCR of the capture module.

The PCA pairwise cascade function is enabled by setting the CASCEN bit in PCAx_CFGR to '1'. When PCA is cascaded in pairs, the PCA related configuration is set by the PCA0/2 register, and the PCA1/3 related control register setting is invalid. The 32-bit PCA after cascade also has Mode0~Mode3 working modes.

The two combinations of cascade are as follows:

| Cascade combination | High 16 bits | Low 16 bits |
|---------------------|--------------|-------------|
| PCA0 and PCA1 | PCA1 | PCA0 |
| PCA2 and PCA3 | PCA3 | PCA2 |

Examples of PCA cascade operation:

Set PCA0 and PCA1 to be cascaded, the clock source of PCA1 timer is the overflow of PCA0, and the clock source of PCA0 is set to the falling edge of PCA0ECI, and the count is as follows:

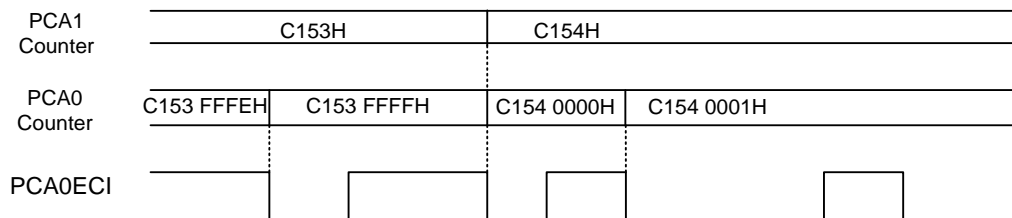


Figure 21-16 Example diagram of cascade operation

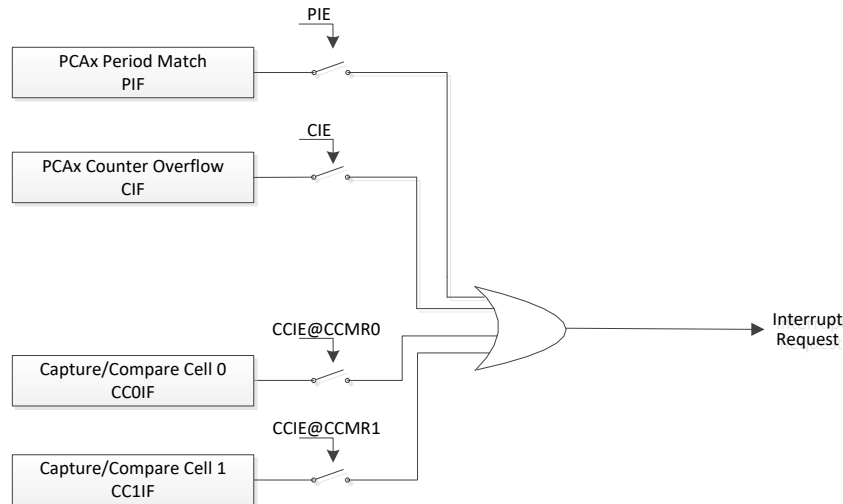
Note:

1. After cascading, the PR value of PCA1 and CCR0/CCR1 are automatically mapped to the upper 16 bits of the corresponding register of PCA0, so reading only needs to read the 32-bit register of PCA0, and the rest of the PCA is cascaded in the same way.
2. The 8-bit mode (frequency output mode and 8-bit pwm1 mode) of PCA does not support cascading.

21.6 Interrupt

PCAx has 1 independent interrupt vector addresses, among which counter overflow and 2 input capture/output comparison share an interrupt vector.

Although the counter and the 2-channel input capture/output comparison module share an interrupt vector, the counter and each input capture/output comparison module have independent interrupt request sources. When the counter overflows, the overflow flag is set to 1, as long as the overflow interrupt enable bit CIE is 1, and the interrupt can be triggered. The interrupt request process of the 2-channel input capture/output comparison module is the same as that of the counter, and each interrupt enable bit needs to be turned on.



21.7 Immediate update of registers

The output compare register CCR and period register can be selected to take effect immediately or take effect in the next cycle by setting the corresponding bits in CFGR. When setting CC0UE in CFGR to 1, the value written in CCR0 will take effect immediately. If CC0UE is 0, the value written in CCR0 will take effect in the next cycle.

The registers that can be set include CCR0/CCR1/ PR, and the corresponding bits are CC0UE/CC1UE/ PUE in CFGR.

21.8 Trigger DMA

DMA can be triggered when an input capture/output comparison or period overflow event occurs. When CC0DMA in CFGR is set to 1, when a capture or comparison event occurs in channel 0 of PCA, DMA transfer can be triggered.

Others that can trigger DMA transfer are the capture or compare events of channel 1, and the overflow event of the counter. The CC0DMA/CC1DMA and CUDMA bits in CFGR need to be enabled respectively.

21.9 Register Lock

PCAx has one lock registers LCKR. Only when LCKR=0x33CC, the PCAx register is allowed to be written. The protected registers include CR, CFGR, PR, PSC, CNT, CCMR0, CCMR1, CCR0, CCR1, FORCE.

21.10 matters needing attention

21.10.1 PCA output status in debug mode

There is no difference between the output waveform of PCA module in debug mode and normal mode, but if the CPU is in the stop state (such as breakpoint or single-step debugging, etc.) in the debug state, the PCA output will automatically switch to the high resistance state to avoid damaging the external power devices.

21.10.2 PCA output status in stop mode

When the PEN@ PCA_CR bit is 1, PEN@ PCA_CR bit will be cleared by hardware before entering stop mode and the PCA pin will switch the output to the high resistance state. When MCU wakes up from stop mode, it needs to write PEN@ PCA_CR bit to 1 again through software, and the PCA pin can output waveform normally.



21.11 Registers

PCA0~3 Module Register list (Base Address:0x4000 2400)

| Address | Register | Description |
|---------------------|----------|--|
| 0x4000 2400+0x400*x | CR | PCAx enable register(PCA0~3 corresponds to x: 0~3, the same below) |
| 0x4000 2404+0x400*x | CFGR | PCAx mode register |
| 0x4000 2408+0x400*x | SR | PCAx flag register |
| 0x4000 240C+0x400*x | FORCE | PCAx forced output control register |
| 0x4000 2410+0x400*x | CNT | PCAx counter register |
| 0x4000 2414+0x400*x | PSC | Prescaler |
| 0x4000 2418+0x400*x | PR | PCAx counting period value |
| 0x4000 241C+0x400*x | CCMR0 | PCAx capture/compare module 0 control register |
| 0x4000 2420+0x400*x | CCMR1 | PCAx capture/compare module 1 control register |
| 0x4000 2424+0x400*x | CCR0 | PCAx compare/capture module 0 register |
| 0x4000 2428+0x400*x | CCR1 | PCAx compare/capture module 1 register |
| 0x4000 242C+0x400*x | LCKR | PCAx lock register |

21.11.1 PCAx enable register (PCAx_CR)

Offset Address: 0x0000+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | | PEN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | RW |

| Bit | Symbol | Description |
|--------|----------|--|
| 31 ~ 1 | Reserved | - |
| 0 | PEN | PCAx counter/timer operation control bit 0: Disable PCAx counter/timer 1: Enable PCAx counter/timer |

21.11.2 PCAx mode register (PCAx_CFGR)

Offset Address: 0x0004+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----------|-----------|--------------|------------|------------|----------|-----|----------|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | PUD MA | CUD MA | Rese rved | CC1 DMA | CC0 DMA | Reserved | | ECF[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | RW | RW | - | RW | RW | | | RW | |

| | | | | | | | | | | | | | | | |
|-----|--------------|-----------|-----------|----------|-----|----|-----|-----|----------|----------|----|------------|----------|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PUE | Rese rved | CC1 UE | CC0 UE | Reserved | | | PIE | CIE | SDE N | Reserved | | CAS CEN | CPS[2:0] | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RW | - | RW | RW | | | | RW | RW | RW | | | RW | RW | | |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 25 | Reserved | - |
| 24 | PUDMA | PCA counter/timer triggers DMA when a periodic match event occurs 0: Disable triggering of DMA when counter/timer periodic match event 1: Enable triggering of DMA when counter/timer periodic match event |
| 23 | CUDMA | PCA counter/timer triggers DMA when an overflow event occurs 0: Disable triggering of DMA when counter/timer overflow event 1: Enable triggering of DMA when counter/timer overflow event |
| 22 | Reserved | - |
| 21 | CC1DMA | PCA capture/compare channel 1 triggers DMA when an event occurs 0: Disable triggering of DMA when an event occurs in capture/compare channel 1 1: Enable triggering of DMA when an event occurs in capture/compare channel 1 |
| 20 | CC0DMA | PCA capture/compare channel 0 triggers DMA when an event occurs 0: Disable triggering of DMA when an event occurs in capture/compare channel 0 1: Enable triggering of DMA when an event occurs in capture/compare channel 0 |
| 19 ~ 18 | Reserved | - |
| 17 ~ 16 | ECF[1:0] | Input signal filtering time 00: No filtering function 01: filter time is 8*bus clock 10: filter time is 16*bus clock 11: filter time is 32*bus clock |
| 15 | PUE | PCAx period value update enable bit immediately 0: The value written to the PR register takes effect in the next cycle 1: The value written to the PR register takes effect immediately |
| 14 | Reserved | - |
| 13 | CC1UE | PCAx compare module 1 immediately updates the enable bit 0: The value written to CCR1 will take effect in the next cycle 1: The value written to CCR1 takes effect immediately |
| 12 | CC0UE | PCAx compare module 0 immediately update the enable bit 0: The value written to CCR0 will take effect in the next cycle 1: The value written to CCR0 takes effect immediately |
| 11 ~ 9 | Reserved | - |
| 8 | PIE | PCAx counter/timer period interrupt enable bit 0: Disable counter cycle interrupt 1: When PIF is set, allow PCAx counter/timer cycle interrupt request This bit is the mask bit for the PCAx counter/timer period (PIF) interrupt |
| 7 | CIE | PCAx counter/timer overflow interrupt enable bit 0: Disable counter overflow interrupt 1: When CIF is set, the interrupt request of PCAx counter/timer overflow is allowed This bit is the mask bit for the PCAx counter/timer overflow (CIF) interrupt |
| 6 | SDEN | PCAx counting mode selection bit 0: PCAx works in up-counting (sawtooth wave) mode, at this time the triangle mode of the other comparison capture modules of this PCAx are disabled 1: PCAx works in the center-aligned counting (triangular wave) mode, at this time the sawtooth wave mode of other comparison capture modules of the PCAx are disabled When PCAx is set to one of the two counting modes, the other counting mode of the compare capture module is invalid even if it is configured. |
| 5 ~ 4 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|-----------------|--|
| 3 | CASCEN | PCAx cascade (Cascade) enable bit (x=0,2, only valid in the upper PCA0/2 of the cascade) 0: Prohibit PCA0/1 (or PCA2/3) cascade function 1: Enable PCA0/1 (or PCA2/3) cascade function Note: This bit can only be modified when the two PCAs in the cascade stop running (CR=0). |
| 2 ~ 0 | CPS[2:0] | PCAx counter/timer clock selection 000: bus clock 001: PCAxECl falling edge 010: RC128K 011: LSECLK 100: HSECLK/8 101: reserved other: bus clock Note: The bus clock cycle must not be less than 4 times the counting clock cycle (except when the counting clock is the bus clock), otherwise the PCAx counter will not count correctly. |

21.11.3 PCAx flag register (PCAx_SR)

Offset Address: 0x0008+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|------|----------|------|----------|--------|--------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | CIFC | Reserved | PIFC | Reserved | CC1IFC | CC0IFC | | |
| - | | | | | | | | WO | - | WO | - | WO | WO | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|-----|----------|-----|----------|-------|-------|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | CIF | Reserved | PIF | Reserved | CC1IF | CC0IF | | |
| - | | | | | | | | RO | - | RO | - | RO | RO | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 24 | Reserved | - |
| 23 | CIFC | PCAx counter/timer overflow flag clear bit 0: invalid 1: Clear |
| 22 ~ 21 | Reserved | - |
| 20 | PIFC | PCAx counter/timer period interrupt flag clear bit 0: invalid 1: Clear |
| 19 ~ 18 | Reserved | - |
| 17 | CC1IFC | CCF1: PCAx capture/compare module 1 flag bit clear bit 0: invalid 1: Clear |
| 16 | CC0IFC | CCF0: PCAx capture/compare module 0 flag bit clear bit 0: invalid 1: Clear |
| 15 ~ 8 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|-----------------|---|
| 7 | CIF | PCAx counter/timer overflow flag 0: The counter has not overflowed 1: Counter overflow Set by hardware when PCAx counter/timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CIF) interrupt is enabled, this bit will cause the CPU to switch to the PCAx interrupt service routine. This bit cannot be automatically cleared to 0 by hardware, and must be cleared to 0 with a special clear bit. |
| 6 ~ 5 | Reserved | - |
| 4 | PIF | PCAx counter/timer period interrupt flag 0: The counter has not reached the period value 1: The counter reaches the period value Only when SDEN=1, when the counter value reaches the period value, it will be set by hardware. When SDEN=0, this bit is invalid and is always 0. When PIE is 1, this bit '1' will cause the CPU to switch to the PCAx interrupt service routine. This bit cannot be automatically cleared to 0 by hardware, and must be cleared to 0 with a special clear bit. |
| 3 ~ 2 | Reserved | - |
| 1 | CC1IF | CC1IF: PCAx capture/compare module 1 flag 0: No comparison match or capture occurred 1: A comparison match or capture occurred This bit is set by hardware when a match or capture occurs. When the CC1IF interrupt is enabled, this bit '1' will cause the CPU to switch to the PCAx interrupt service routine. This bit cannot be automatically cleared to 0 by hardware, and must be cleared to 0 with a special clear bit. |
| 0 | CC0IF | CC0IF: PCAx capture/compare module 0 flag 0: No comparison match or capture occurred 1: A comparison match or capture occurred This bit is set by hardware when a match or capture occurs. When the CC0IF interrupt is enabled, this bit '1' will cause the CPU to switch to the PCAx interrupt service routine. This bit cannot be automatically cleared to 0 by hardware, and must be cleared to 0 with a special clear bit. |

21.11.4 PCAx forced output control register (PCAx_FORCE)

Offset Address: 0x000C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | | FCO | FCO |
| | | | | | | | | | | | | | | 1 | 0 |
| | | | | | | | | | | | | | | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------------|--|
| 31 ~ 2 | Reserved | - |
| 1 | FCO1 | Module 1 forced match control bit (this bit is valid only when SM[1:0]=01) 0: Disable forced matching 1: Enable forced match, hardware will automatically clear after match |



SH30F9/SA0 Series

| | | |
|---|------|--|
| 0 | FC00 | Module 0 forced match control bit (this bit is valid only when SM[1:0]=01) 0: Disable forced matching 1: Enable forced match, hardware will automatically clear after match |
|---|------|--|

21.11.5 PCAx counter register (PCAx_CNT)

Offset Address: 0x0010+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CNTCAS[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|-----------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CNTH[7:0] | | | | | | | | CNTL[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|--------------|--|
| 31 ~ 16 | CNTCAS[15:0] | High 16 bits of 32-bit PCAx count value: only this bit of PCA0/2 is valid Note: only cascading is effective, for example, the read and write values of PCA1/3.[CNTH:CNTH] and PCA0/2.CNTCAS are the same. |
| 15 ~ 8 | CNTH[7:0] | PCAx counter value high 8 bits |
| 7 ~ 0 | CNTL[7:0] | PCAx counter value lower 8 bits |

21.11.6 Prescaler (PCAx_PSC)

Offset Address: 0x0014+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PSC[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | PSC[15:0] | Prescaler The frequency division factor is PSC[15:0]+1 |



21.11.7 PCAx counting period value (PCAx_PR)

Offset Address: 0x0018+0x400*x

Reset value: 0x0000 FFFF

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| PRCAS[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| PRH[7:0] | | | | | | | | PRL[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|---------|-------------|---|
| 31 ~ 16 | PRCAS[15:0] | High 16 bits of 32-bit PCAx counting period value: only this bit of PCA0/2 is valid Note: only cascading is effective, for example, the read and write values of PCA1/3.[PRH:PRL] and PCA0/2.PRCAS are the same. |
| 15 ~ 8 | PRH[7:0] | PCAx count maximum high byte (MSB) |
| 7 ~ 0 | PRL[7:0] | PCAx maximum count low byte (LSB) |

21.11.8 PCAx capture/compare module 0 control register (PCAx_CCMR0)

Offset Address: 0x001C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----------|-----------|-----------|-----------|-----------|----------|-----------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | CC | SM[1:0] | FS[1:0] | CEN | TCP | Reserved | CCIE | | |
| - | | | | | | | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | - | <i>RW</i> | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 9 | Reserved | - |
| 8 | CC | Clear counter enable bit when capture occurs 0: Do not clear the counter value when a capture event occurs 1: Clear the counter value when a capture event occurs |
| 7 ~ 6 | SM[1:0] | PCAx mode selection bit 00: Capture mode 01: compare match 10: Frequency output 11: PWM output |



SH30F9/SA0 Series

| | | |
|-------|-----------------|---|
| 5 ~ 4 | FS[1:0] | <p>When SM[1:0] = 00: Capture mode selection bit 0X: In capture mode, trigger on positive edge 10: In capture mode, trigger on negative edge 11: In capture mode, trigger on any edge</p> <p>When SM[1:0] = 01: Software timing mode selection bit 0X: continuous match 1X: Single match</p> <p>When SM[1:0] = 11: PWM mode selection. 00: select 8-bit PWM1 mode 01: Select 16-bit PWM2 mode 10: Select 16-bit PWM3 mode 11: Select 16-bit PWM4 mode</p> <p>This bit is invalid when SM[1:0] = 10 (frequency output mode).</p> |
| 3 | CEN | <p>Compare/Capture Module Function Enable Bit 0: disable compare/capture module 0 1: Enable compare/capture module 0</p> |
| 2 | TCP | <p>When SM[1:0] = 00, this bit is the edge indicator bit of the capture signal 0: PCAXA channel is falling edge capture 1: PCAXA channel is rising edge capture</p> <p>When SM[1:0] = 01 or 10, this bit is the waveform output control bit 0: The initial level is high 1: The initial level is low</p> <p>When SM[1:0] = 11, this bit is PWM output inversion enable bit 0: PWM normal output waveform (Duty is active at low level) 1: PWM reverse output waveform (Duty is valid at high level)</p> |
| 1 | Reserved | - |
| 0 | CCIE | <p>Capture/Compare Flag Interrupt Enable Bit 0: Disable compare capture/output compare interrupt 1: Enable compare capture/output compare interrupt, when the CCIF bit is set to 1, enable capture/compare flag interrupt request</p> |

21.11.9 PCAX capture/compare module 1 control register (PCAX_CCMR1)

Offset Address: 0x0020+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|---------|---------|-----|-----|----------|------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | CC | SM[1:0] | FS[1:0] | CEN | TCP | Reserved | CCIE | | |
| - | | | | | | | RW | RW | RW | RW | RW | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------------|---|
| 31 ~ 9 | Reserved | - |
| 8 | CC | <p>Clear counter enable bit when capture occurs 0: Do not clear the counter value when a capture event occurs 1: Clear the counter value when a capture event occurs</p> |
| 7 ~ 6 | SM[1:0] | <p>PCAX mode selection bit 00: Capture mode 01: Comparison match 10: Frequency output 11: PWM output</p> |



| | | |
|-------|-----------------|--|
| 5 ~ 4 | FS[1:0] | <p>When SM[1:0] = 00: Capture mode selection bit 0X: In capture mode, trigger on positive edge 10: In capture mode, trigger on negative edge 11: In capture mode, trigger on any edge</p> <p>When SM[1:0] = 01: Software timing mode selection bit 0X: Continuous software timing mode 1X: Single software timing mode</p> <p>When SM[1:0] = 11: PWM mode selection. 00: select 16-bit PWM1 mode 01: Select 32-bit PWM2 mode 10: Select 32-bit PWM3 mode 11: Select 32-bit PWM4 mode</p> <p>This bit is invalid when SM[1:0]=10 (frequency output mode).</p> |
| 3 | CEN | <p>Compare/Capture Module Function Enable Bit 0: prohibit compare/capture module 1 1: Enable compare/capture module 1</p> |
| 2 | TCP | <p>When SM[1:0] = 00, this bit is the edge indicator bit of the capture signal 0: PCAxB channel is falling edge capture 1: PCAxB channel is rising edge capture</p> <p>When SM[1:0] = 01 or 10, this bit is the waveform output control bit 0: The initial level is high 1: The initial level is low</p> <p>When SM[1:0] = 11, this bit is PWM output inversion enable bit 0: PWM normal output waveform (Duty is active at low level) 1: PWM reverse output waveform (Duty is valid at high level)</p> |
| 1 | Reserved | - |
| 0 | CCIE | <p>Capture/Compare Flag Interrupt Enable Bit 0: Disable compare capture/output compare interrupt 1: Enable compare capture/output compare interrupt, when the CCIF bit is set to 1, enable capture/compare flag interrupt request</p> |

21.11.10 PCAx compare/capture module 0 register (PCAx_CCR0)

Offset Address: 0x0024+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CCR0CAS[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|----|----|-------------------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CCR0H[7:0] | | | | | | | | CCR0L[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------------------|--|
| 31 ~ 16 | CCR0CAS[15:0] | High 16 bits of 32-bit PCAx capture value: only this bit of PCA0/2 is valid Note: only cascading is effective, for example, the read and write values of PCA1/3.[CCR0H:CCR0L] and PCA0/2.CCR0CAS are the same. |
| 15 ~ 8 | CCR0H[7:0] | PCAx compare/capture module 0 register high byte |
| 7 ~ 0 | CCR0L[7:0] | PCAx compare/capture module 0 register low byte |



21.11.11 PCAx compare/capture module 1 register (PCAx_CCR1)

Offset Address: 0x0028+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| CCR1CAS[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CCR1H[7:0] | | | | | | | | CCR1L[7:0] | | | | | | | |
| <i>RW</i> | | | | | | | | <i>RW</i> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|---------------|---|
| 31 ~ 16 | CCR1CAS[15:0] | High 16 bits of 32-bit PCAx capture value: only this bit of PCA0/2 is valid Note: only cascading is effective, for example, the read and write values of PCA1/3.[CCR1H:CCR1L] and PCA0/2.CCR1CAS are the same. |
| 15 ~ 8 | CCR1H[7:0] | PCAx compare/capture module 1 register high byte (MSB) |
| 7 ~ 0 | CCR1L[7:0] | PCAx compare/capture module 1 register low byte (LSB) |

21.11.12 PCAx lock register (PCAx_LCKR)

Offset Address: 0x002C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| KEY[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | KEY[15:0] | Register modification control register 0x33CC: Allow software to modify the protected registers of the module Note: The protected registers are CR, CFGR, FORCE, PR, PSC, CNT, CCMR0, CCMR1, CCR0, CCR1 |



22. Universal Asynchronous Receiver/Transmitter (UART)

22.1 Introduction

Universal asynchronous receiver/transmitter, also known as UART, is a kind of asynchronous transceiver.

22.2 Main Feature

- Internal baud rate generator
- 1/2 stop bit can be set
- UART has four working modes
- Enhanced functions include frame error detection and automatic address recognition
- Parity check
- Baud rate detection
- Support LIN bus synchronization and discontinuous hardware generation and detection
- Support inverse logic function
- Support DMA communication

22.3 Function Description

22.3.1 UART Work Mode

UART has four working modes. The user must first initialize UARTx_CR, select mode, baud rate and stop bit before communicating.

In all four modes, any write operation with UARTx_TDR as the target register will initiate a transfer after TEN is enabled. In mode 0, the reception is initiated by conditions RI = 0 and REN = 1. This generates a clock signal on the TXD pin and then shifts in 8-bit data on the RXD pin. In other modes, the reception is initiated by the starting bit of input (if RI = 0 and REN = 1). External transmitter communication begins with the transmission of the starting bit.

Table 22-1 UART Working Mode List

| SM1 | SM0 | Mode | Type | Baud rate | Frame length | Starting bit | Stop bit | 9 th bit |
|-----|-----|------|--------------------------|---|----------------|--------------|----------|---------------------|
| 0 | 0 | 0 | Synchronous half duplex | Bus clock/(48 or 12) | 8bit | No | No | No |
| 0 | 1 | 1 | Asynchronous full duplex | Overflow rate of built-in baud rate generator /16 | 10 bit (or 11) | 1 | 1(or 2) | No |
| 1 | 0 | 2 | Asynchronous full duplex | Bus clock/(256 or 128) | 11 bit (or 12) | 1 | 1(or 2) | 0,1 |
| 1 | 1 | 3 | Asynchronous full duplex | Overflow rate of built-in baud rate generator /16 | 11 bit (or 12) | 1 | 1(or 2) | 0,1 |

Mode 0: Synchronous, half duplex communication

Mode 0 supports synchronous communication with external devices. The serial data is received and transmitted on the RXD pin, and the shift clock is sent by the TXD pin. SH30F9/SA0 series provides shift clock on the TXD pin, so this mode is a half-duplex mode of serial communication. In this mode, 8 bits are transmitted and received per frame, and the LSB is received or transmitted first.

The baud rate is programmable to either 1/12 or 1/48 of the bus clock that is set by SMOD_M0@UARTx_CR. When the SMOD_M0bit is set to 0, the serial port operates at 1/48 of the bus clock. When the SMOD_M0 bit is set to 1, the serial port operates with 1/12 of the bus clock.

The block diagram of the function block is shown below. Data is shifted in and out of the serial port through the RXD pin, and the shift clock is output by the TXD pin.

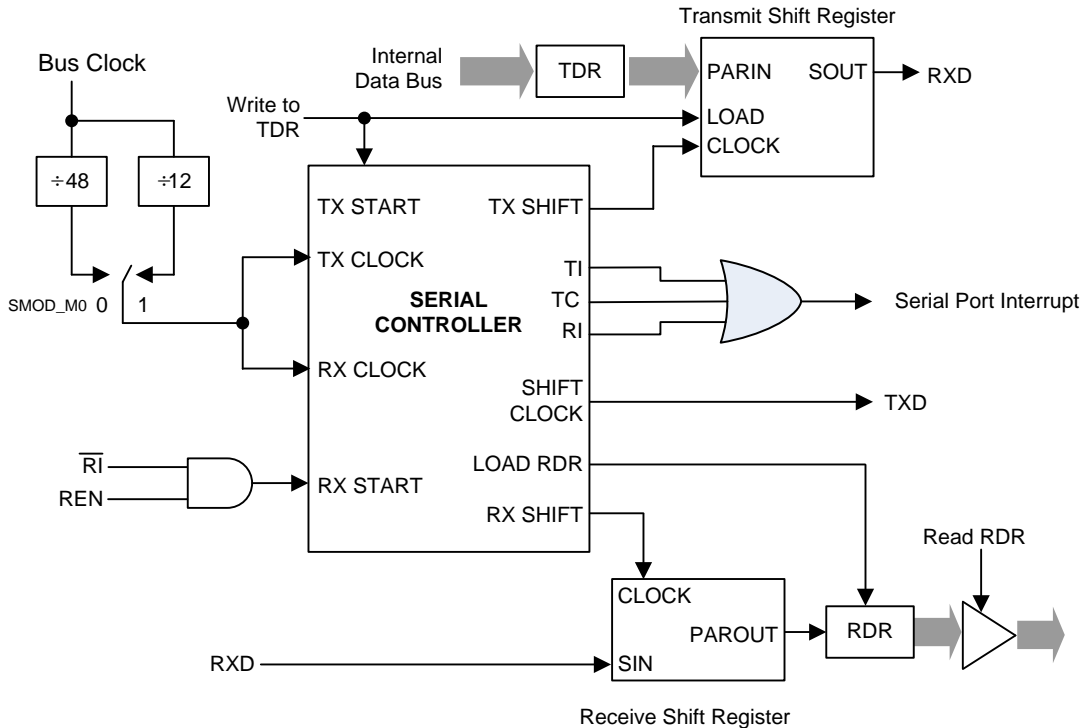


Figure 22-1 Mode 0 functional block diagram

After enabling `TEN@UARTx_CR`, the `UARTx_TDR` register is empty and the write operation can be performed. The TX control block of the next bus clock of the write data starts to transmit. Data conversion occurs on the falling edge of the shift clock, and the contents of the shift register are successively shifted from left to right, and the empty position is 0. When all 8 bits in the shift register are sent, the TX control module stops the sending operation, and then moves the data from the `UARTx_TDR` register to the SHIFT register on the rising edge of the next bus clock. At this time, `UARTx_TDR` is empty again, `TI @UARTx_FR` hardware Set (TI is the transmit interrupt flag, which reflects the status of the transmit register in real time, and set indicates that the transmit register is empty).

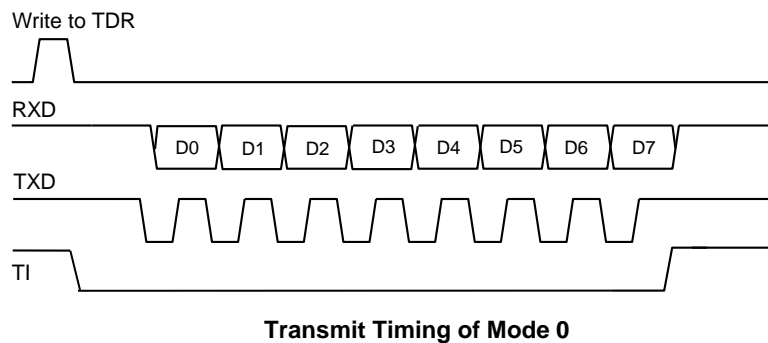
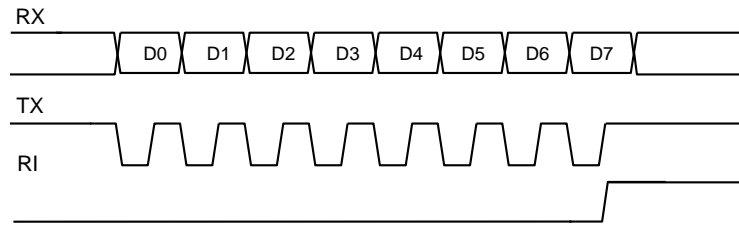


Figure 22-2 Transmit timing of Mode 0 diagram

`REN@UARTx_CR` is set and `RI@UARTx_FR` is cleared to initialize reception. The next bus clock starts the reception, latches the data on the rising edge of the shift clock, and shifts the contents of the receiving conversion register to the left one by one. When all 8 bits of data are moved to the shift register, the RX control block stops receiving, and RI is set on the rising edge of the next bus clock until it is read by software `RDR[7:0]@UARTx_RDR`, and RI is cleared accordingly. Zero will allow the next reception. (RI is the receiving interrupt flag, which reflects the status of the sending register in real time. Setting this bit indicates that the sending register is full.)



Receive Timing of Mode 0

Figure 22-3 Receive timing of Mode 0 diagram

Mode 1: 8-bit UART, variable baud rate, asynchronous full duplex

Mode 1 provides 10/11-bit full-duplex asynchronous communication, 10 bits are composed of a start bit (logic 0), 8 data bits (the low bit is the first bit) and 1 bit/2 stop bits (logic 1). When receiving, these 8 data bits are stored in RDR[7:0]@UARTx_RDR. The baud rate in mode 1 is fixed at 1/16 of the overflow rate of the built-in baud rate generator. The function block box Figure is shown in the following figure:

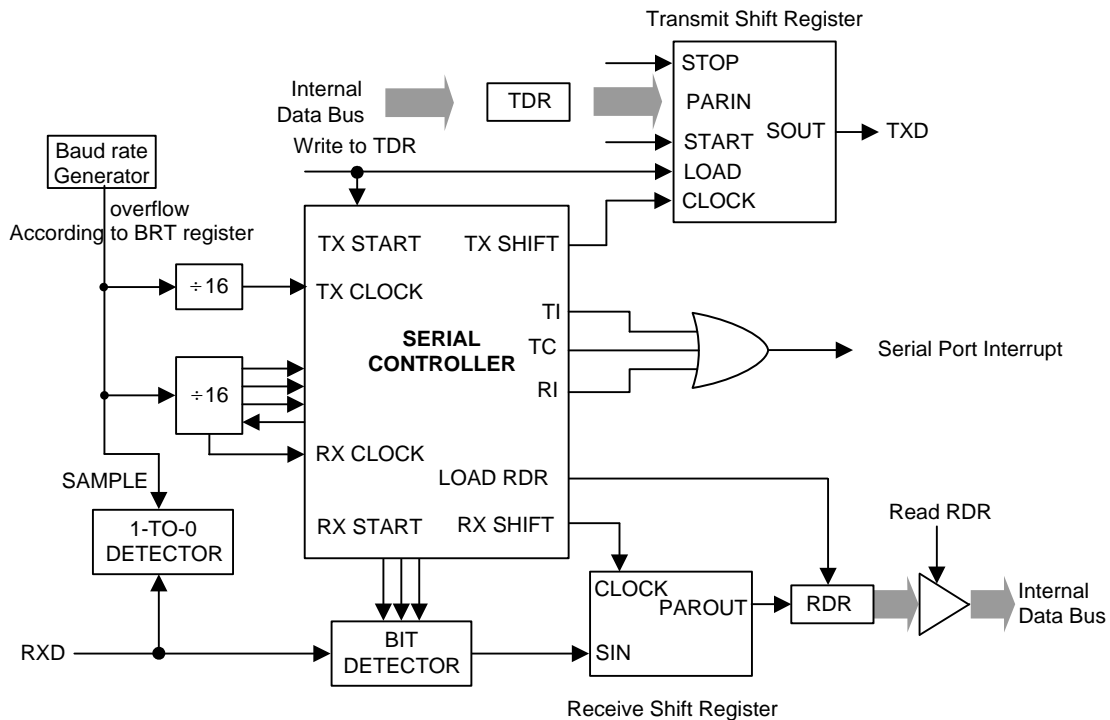
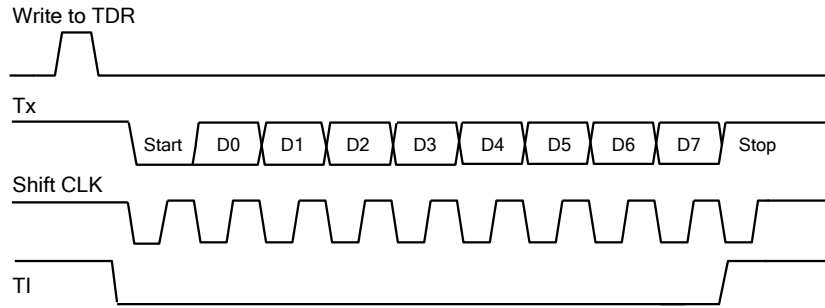


Figure 22-4 Mode 1 functional block diagram

After enabling TEN@UARTx_CR, the UARTx_TDR register is empty and the write operation can be performed. In fact, the TXD pin is sent from the bus clock after the next transition in the divide-by-16 counter, so the bit time is the divide-by-16 counter is synchronized. The start bit is first shifted out on the TXD pin, followed by 8 data bits. After all 8 bits of data in the transmit shift register are sent, the stop bit is shifted out on the TXD pin, and the TI flag is set when the stop bit transmission begins.



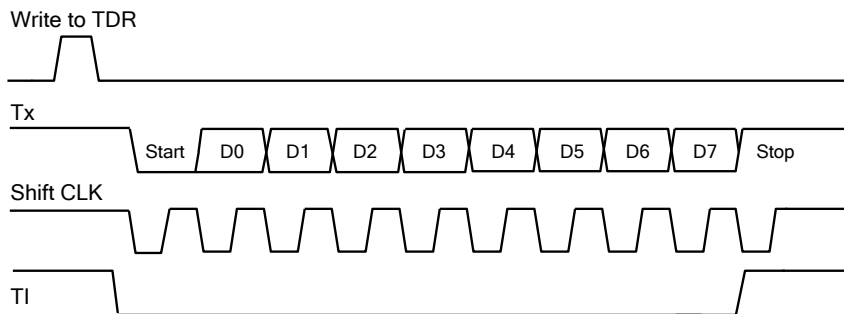
Transmit Timing of Mode 1

Figure 22-5 Transmit timing of Mode 1 diagram (STOP=1BIT)

Reception is allowed only when REN is set. When the RXD pin detects the falling edge, the serial port begins to receive serial data. For this reason, the CPU continuously samples RXD at a sampling rate of 16 times the baud rate. When the falling edge is detected, the baud rate 16 multiplier counter is reset immediately, and the filter is performed at this frequency (3 clocks debounce), and then the data is sampled at the middle position of the filtered waveform. If the first bit received is not 0, it means that this bit is not the start bit of a frame of data. This bit is ignored and the receiving circuit is reset, waiting for another falling edge on the RXD pin. If the start bit is valid, move into the shift register, and then move other bits into the shift register. After 8 data bits and 1 stop bit are shifted in, the contents of the shift register are loaded into RDR[7:0]@UARTx_RDR and RB8@UARTx_RDR respectively, RI is set, but the following conditions must be met:

1. RI = 0
2. FER@UARTx_CR=0 or received stop bit = 1

If these conditions are met, then 8 data bits are loaded into RDR[7:0]@UARTx_RDR, and RI is set. Otherwise, the received frame will be lost and continue to detect the next start bit. At this time, the receiver will re-detect whether there is another falling edge at the RXD end. After receiving 2 bytes of data continuously, the user should remove the data in RDR[7:0]@UARTx_RDR, otherwise the data received in SHIFT will be overwritten by new data, and the RXOV error flag will be generated at the same time.



Transmit Timing of Mode 1

Figure 22-6 Receiving timing of Mode1 diagram (STOP=1BIT)

Mode 2: 9-bit UART, fixed baud rate, asynchronous full duplex

This method uses 11 bits/12 bits in asynchronous full-duplex communication. A frame consists of a start bit (logic 0), 8 data bits (the low bit is the first bit), a programmable 9th data bit and 1 bit/2 stop bits (logic 1). Mode 2 supports multi-machine communication and hardware address recognition (see the chapter on multi-machine communication for details). During data transmission, the 9th data bit (TB8@UARTx_TDR) can be written as 0 or 1. The 9th bit can be used as the data/address flag bit and the parity bit in multi-machine communication. When data is received, the 9th data is shifted into RB8@UARTx_RDR and the stop bit is not saved. Use the SMOD_M2@UARTx_CR bit to select the baud rate of 1/256 or 1/128 of the bus clock. The function block box Figure is as follows:

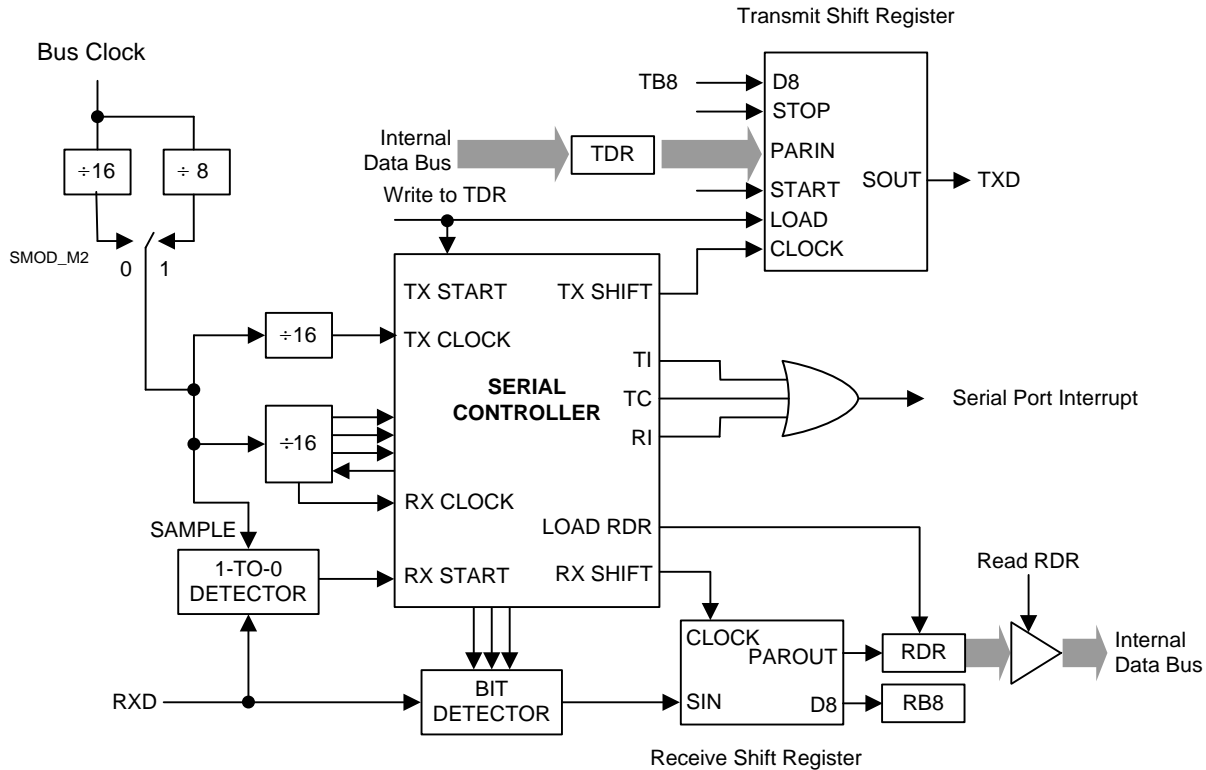
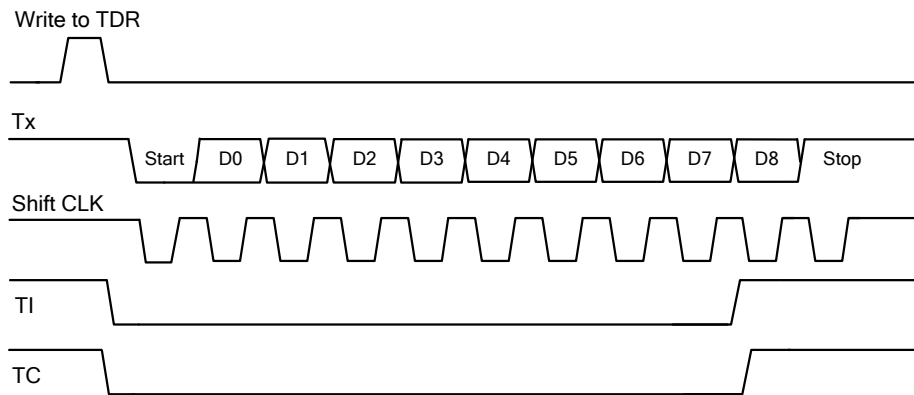


Figure 22-7 Mode 2 functional block diagram

After $TEN@UARTx_CR$ is enabled, the $UARTx_TDR$ register is empty and the write operation can be performed. At the same time, $TB8@UARTx_TDR$ is also loaded into the 9th bit of the transmit shift register. In fact, TXD transmission starts from the bus clock after the next transition in the divide-by-16 counter, so the bit time is synchronized with the divide-by-16 counter. The start bit is first shifted out on the TXD pin, followed by 9 bits of data. After all 9 bits of data in the transmit conversion register are sent, the stop bit is shifted out on the TXD pin, and the TI flag is set when the stop bit transmission ends.



Transmit Timing of Mode 2

Figure 22-8 Transmit timing of Mode 1 diagram (STOP=1BIT)



Reception is allowed only when REN is set. When the RXD pin detects the falling edge, the serial port begins to receive serial data. For this reason, the CPU continuously samples RXD at a sampling rate of 16 times the baud rate. When the falling edge is detected, the baud rate 16 multiplier counter is reset immediately, and the filter is performed at this frequency (3 clocks debounce), and then the data is sampled at the middle position of the filtered waveform. If the first bit received is not 0, it means that this bit is not the start bit of a frame of data. This bit is ignored and the receiving circuit is reset, waiting for another falling edge on the RXD pin. If the start bit is valid, move into the shift register, and then move other bits into the shift register. After 9 data bits and 1 stop bit are shifted in, the contents of the shift register are loaded into RDR[7:0]@UARTx_RDR and RB8@UARTx_RDR respectively, RI is set, but the following conditions must be met:

1. RI = 0
2. FER@UARTx_CR= 0 or received stop bit = 1

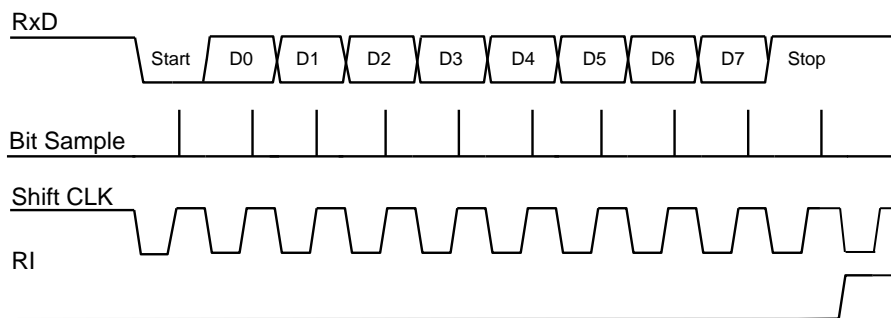
If these conditions are met, the 9th bit is shifted into RB8@UARTx_RDR, 8 bits of data are shifted into RDR[7:0]@UARTx_RDR, and RI is set. Otherwise, the received data frame will be lost. In the middle of the stop bit, the receiver will re-detect another falling edge on the RXD pin. After receiving 2 bytes of data continuously, the user should remove the data in RDR[7:0]@UARTx_RDR. Otherwise, the data received in the SHIFT will be overwritten by the new data, and the RXOV error flag will be generated at the same time.

Shift Register:

The full status of the shift register is set from 0 to 1: After the valid Start signal is detected, the full status of the shift register is set to 1 when 8 or 9bit+1stop data is received.

The full status of the shift register is set from 0 to 1: After the valid Start signal is detected, the full status of the shift register is set to 1 when 8 or 9bit+1stop data is received.

When the full state of the shift register is 1, after reading the RDR data once, the full state of the shift register is automatically cleared to 0.



Receive Timing of Mode 2

Figure 22-9 Receiving timing of Mode 2 diagram (STOP=1BIT)



Mode 3: 9-bit UART, variable baud rate, asynchronous full duplex

Mode 3 uses the transmission protocol of mode 2 and the baud rate generation of mode 1.

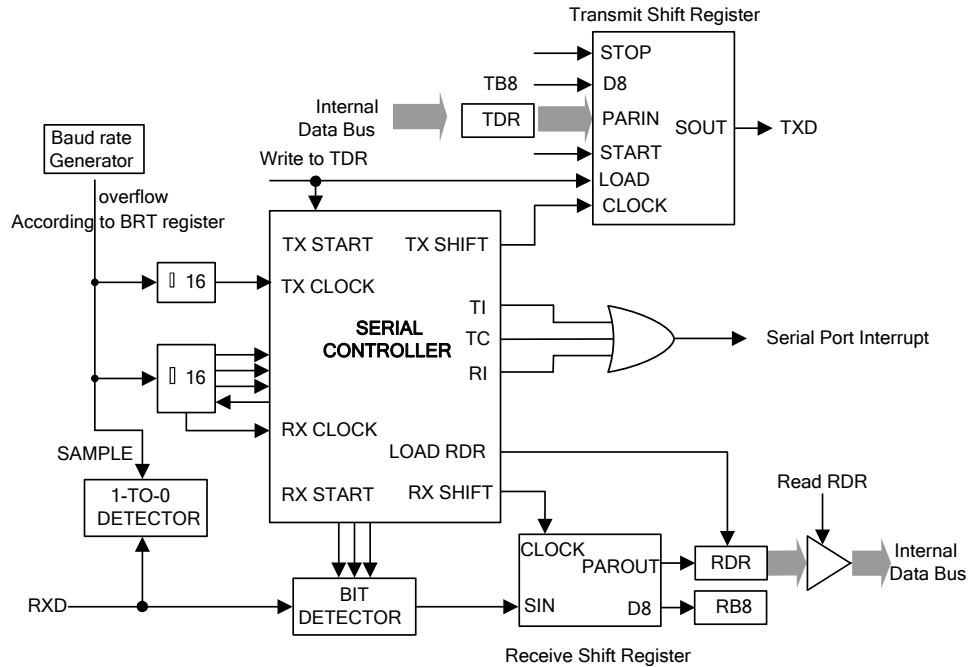


Figure 22-10 Mode 3 functional block diagram

Description about TI, TC flags

TI@UARTx_FR is the transmit interrupt flag bit. If TDR[7:0]@UARTx_TDR is empty, TI will be set. After writing data to TDR[7:0]@UARTx_TDR, TI will be cleared automatically. But it should be noted that when sending data continuously, the first data is directly written to the transmit shift register, and the second data is written to TDR[7:0]@UARTx_TDR. So after the second data is written into TDR[7:0]@UARTx_TDR, the TI flag bit will be cleared. (As shown in Figure 22-11)

TC@UARTx_FR is the flag bit for the end of continuous frame transmission. This bit is cleared by writing to TDR[7:0]@UARTx_TDR register. Writing 1 to TCC@UARTx_FR can also clear the TC bit. When the last Byte data of a frame of data sent continuously is finished (after the stop bit is sent), and it is detected that TI@UARTx_FR is high, the TC mark position starts. (As shown in Figure 22-11)

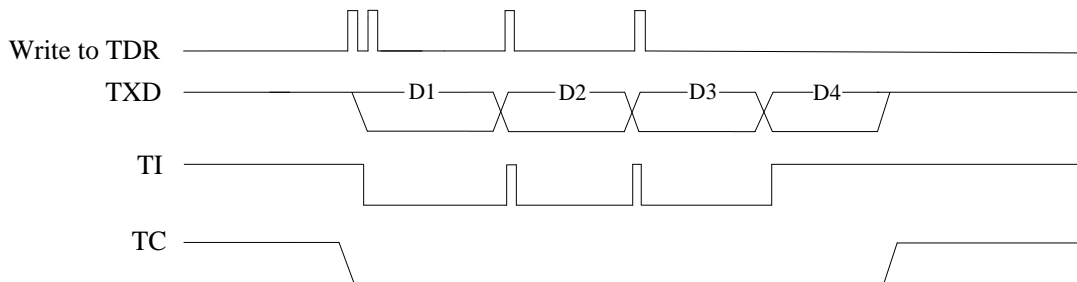


Figure 22-11 TC working sequence diagram



22.3.2 Fine-tunable Baud Rate

UART has an internal baud rate generator, which is actually a 15-bit incremental counter.

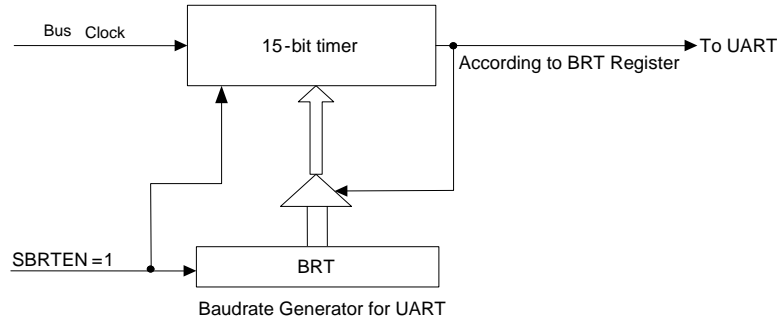


Figure 22-12 Baud rate generator functional block diagram

From the figure, it can be obtained that the overflow rate of the baud rate generator is:

$$SBRT \text{ overflowrate} = \frac{Bus \text{ Clock}}{BRT}$$

Therefore, the baud rate calculation formula for UART in each mode is as follows.

In mode 0, the baud rate can be programmed as 1/48 or 1/12 of the bus clock, which is determined by the SMOD_M0 bit. When SMOD_M0 is 0, the serial port runs at 1/48 of the bus clock. When SMOD_M0 is 1, the serial port runs at 1/12 of the bus clock.

In Mode 1 and Mode 3, the baud rate can be fine-tuned, and the accuracy is a bus clock. The formula is as follows:

$$BaudRate = \frac{Bus \text{ Clock}}{16 \times (SBRT + 1) + BFINE}$$

For example: bus clock = 16MHz, the baud rate of 9600Hz needs to be obtained, the calculation method of SBRT and BFINE values is as follows:

$$16000000/16/9600 = 104.17$$

$$SBRT = 104 - 1 = 103$$

$$\text{Calculated by BaudRate: } 9600 = 16000000 / (16 \times 104 + BFINE)$$

$$\text{Get: } BFINE = 3$$

The actual baud rate calculated by this fine-tuning method is 9598 with an error of 0.02%.

In mode 2, the baud rate is fixed to 1/256 or 1/128 of the bus clock, which is determined by the SMOD_M2 bit @UARTx_CR. When the SMOD_M2 bit is 0, the UART runs at 1/256th of the bus clock. When the SMOD_M2 bit is 1, the UART runs at 1/128th of the bus clock.

$$BaudRate = 2^{SMOD_M2} \times \left(\frac{Bus \text{ Clock}}{256} \right)$$

Note: No matter which method is used for communication, the UART baud rate generator must be enabled, that is, SBRTEN=1.



22.3.3 Multiprocessor Communication

Automatic (hardware) address recognition

In mode 2 and mode 3, when the ninth bit function selection bit MULTIE[1:0]=01, the multi-machine communication mode is entered, and the UART operating status is as follows: the stop bit is received, and the ninth bit of RB8 is 1 (address word Section mark), and the received data byte matches the UART slave address, the UART will start to receive subsequent data bytes.

The 9th bit is 1 to indicate that the byte is an address rather than data. When the master wants to send a group of data to one of several slaves, it must first send the target slave address. All slaves are waiting to receive the address byte. The characteristic of automatic address recognition is that only the slave with a matching address can generate an interrupt flag for subsequent received data, and the hardware completes the address comparison.

Only the slave with a matching address can continue to receive data bytes and generate a receive interrupt flag. The slave with a mismatched address will not be affected, and will continue to wait to receive the address byte that matches it.

When using the automatic address recognition function, the master can choose to communicate with one or more slaves by calling the given slave address. The host can address all the slaves by using the broadcast address. There is a special function register UARTx_ADDR, which contains the slave address (SADDR) and address mask (SAMR). The slave address is an 8-bit byte. SAMR is used to define the validity of each bit of SADDR. If a bit in SAMR is 0, the corresponding bit in SADDR is ignored. If a bit in SAMR is set, then SADDR is The corresponding bit will be used to generate the agreed address. This allows the user to flexibly address multiple slaves without changing the slave address in the SADDR register.

Table 22-2 Example table of automatic address recognition

| | Slave 1 | Slave 2 |
|-----------------------------------|----------|----------|
| SADDR | 10100100 | 10100111 |
| SAMR(bits being 0 are ignored) | 11111010 | 11111001 |
| Agreed address | 10100x0x | 10100xx1 |
| Broadcast address (SADDR or SAMR) | 1111111x | 11111111 |

The lowest bit of the agreed address of slave 1 and slave 2 are different. Slave 1 ignores the lowest bit, while the lowest bit of Slave 2 is 1. Therefore, when only communicating with slave 1, the master must send the address with the lowest bit of 0 (10100000). Similarly, the first bit of slave 1 is 0, and the first bit of slave 2 is ignored. Therefore, when only communicating with slave 2, the master must send the address (10100011) whose first digit is 1. If the master needs to communicate with two slaves at the same time, the 0th bit is 1, the 1st bit is 0, and the second bit is ignored by both slaves. Two different addresses are used to select two slaves (1010 0001 and 1010 0101).

The master can communicate with all slaves at the same time through the broadcast address. This address is equal to the bit-OR of SADDR and SAMR, and a 0 in the result indicates that the bit is ignored. In most cases, the broadcast address is 0xFFh, which can be answered by all slaves.

After the system is reset, the two registers SADDR and SAMR are initialized to 0. These two results set the agreed address and broadcast address as XXXXXXXX (all bits are ignored). This effectively removes the feature of multi-slave communication and prohibits the automatic addressing mode. Such a UART will respond to any address. The user can realize the multi-machine communication of software address recognition according to the method mentioned above.

22.3.4 Frame Error Detection

The UART has a frame error detection function. After the error flag bit is set, it can only be cleared by software by writing 1 to the corresponding clear bit, even if there is no error in the subsequent received frame, it will not be automatically cleared. The setting of this error flag will not trigger an interrupt, and will not affect the continuous transmission of serial port data. It is only used for software query.

Sending conflict

If there is data in TDR[7:0]@UARTx_TDR and the user software continues to write data to the TDR[7:0]@UARTx_TDR register, the transmission conflict bit TXCOL@UARTx_FR is set. If a conflict occurs, the new data will be ignored and cannot be written into the transmit buffer. The clearing of this bit is realized by writing 1 to TXCOLC@UARTx_FR.



Receive overflow

There are two levels of RDR receiving data register and shift register on the UART receiving end. When UARTx receives a byte, the data is stored in RDR[7:0]@UARTx_RDR, and RI is set to 1, and subsequent data can continue to enter the receive shift Register, if the data in the shift register is full, and new data enters, RXOV is set to 1. After RXOV is set to 1, the subsequent received data will continue to enter the receive shift register, that is, the data in the receive shift register will be overwritten, but the data in the RDR will not be overwritten. When the data in RDR[7:0]@UARTx_RDR is read, it is necessary to wait for the shift register to receive a complete frame of data (including start bit, data bit, stop bit) before changing the data in the shift register Move into RDR[7:0]@UARTx_RDR to avoid misalignment. Therefore, once RXOV@UARTx_FR is set to 1, it means that data has been discarded, and it is uncertain how many data are lost. In this case, the user is generally required to retransmit the data packet.

Note:

①The data in the RDR will not be overwritten, including the following meanings. As long as the RI flag is set, the data in the shift register cannot enter RDR[7:0]@UARTx_RDR, and once the RDR[7:0]@ is read UARTx_RDR to clear the RI flag will re-allow the data in the shift register to enter RDR[7:0]@UARTx_RDR.

②RXOV is invalid under Mode0

Parity error

If an incorrect parity bit is detected, then the parity bit PE@UARTx_FR is set. The clearing of this bit is realized by writing 1 to PEC@UARTx_FR.

Frame error

If an invalid (low) stop bit is detected, the frame error bit FE@UARTx_FR is set. This bit is cleared by writing 1 to FEC@UARTx_FR.

Note: Before sending TXD, the pin must be set to output high level.

22.3.5 Parity

Odd parity: The number of '1' in the parity bit and the transmitted data (binary representation) is an odd number.

Even parity: The number of '1' in the parity bit and the transmitted data (binary representation) is an even number.

When selecting mode 0 and mode 1, there is no parity check function. When selecting mode 2 and mode 3, if the ninth function selection bit MULTIE[1:0]=1x, the ninth bit TB8@UARTx_TDR will be used as the parity bit (the priority is higher than the multi-machine communication mode and user-defined Ninth bit mode), this bit of data is generated by the shift register and automatically assigned to the ninth bit. The check mode selection is set by PS@UARTx_CR.

The parity bit is automatically detected by the hardware. After each byte is transmitted, if a parity error occurs, the PE@UARTx_FR bit will be automatically checked. This bit is cleared by writing 1 to PEC@UARTx_FR by software.

Note: Regardless of whether the verification is correct or not, the data will be stored in RDR[7:0]@UARTx_RDR, and the parity bit will be stored in RB8@UARTx_RDR.

22.3.6 LIN (Local Area Internet) bus mode

The LIN bus mode of SH30F9/SA0 series supports SYNCH BREAK hardware generation and detection.

LIN mode is controlled by LINEN@UART_CR. In LIN mode, use the TXD and RXD pins of the UART as LIN transmit and receive function pins, and the UART mode should select MODE1.

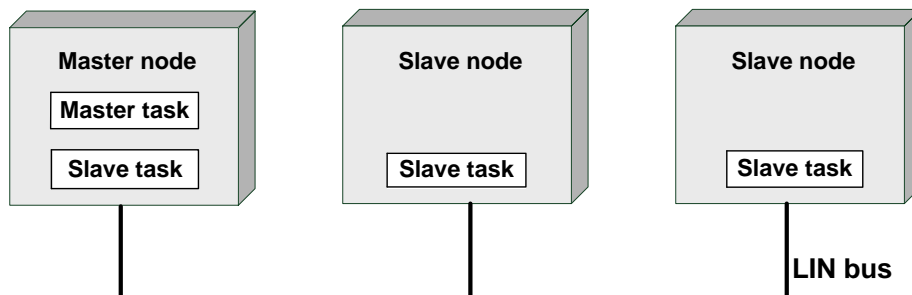


Figure 22-13 LIN bus mode



LIN sends and transmits bytes similar to ordinary UART. According to the LIN bus protocol, first send the synchronization interval (SBK@UART_CR) as the start of signal transmission on the LIN bus, then send 0X55 as the synchronization field (SYNCH FIELD) data adapted to the baud rate, and finally send the identifier field (IDENT FIELD). Control the data communication on the bus. The data format of the subsequent communication between the master and the slave is 8-bit data (LSB) without parity bit. There is a checksum field (CHECKSUM FIELD) at the end of the data frame. There are two types of checksum fields: the checksum calculated only based on the data byte is called the traditional checksum, which can be used for LIN1.3 slave communication; the checksum calculated based on the data byte and identifier is called the traditional checksum. The checksum is called an enhanced checksum, and it can communicate with the LIN2.0 slave. The host judges whether the goal is achieved according to the checksum, and then decides whether to initiate LIN bus communication again.

For LIN reception, there is a synchronous interval detection circuit on the RXD interface of the UART. When a low level (BREAK CHARACTER) with a length greater than 11 or 10 bits (set by LBDL@UART_CR) is detected, then at least 1 bit is detected The high level (DELIMITER) means that the synchronous discontinuity detection is successful, and an LBD interrupt flag is generated. If the response interrupt enable bit LBDIE is enabled, the interrupt service routine will be entered. When the LBD is cleared by writing 1 to LBDC, it is necessary to clear the receive shift register and receive buffer, and close LINEN in order to receive the serial data that is subsequently shifted in. If you need to re-detect the synchronization interruption signal, you need to re-enable LINEN.

22.3.7 Use DMA communication

Users can choose to operate UART transmission and/or reception by using DMA. The DMA mode is determined by the DMAT and DMAR mode selection bits in the UARTx_CR register to determine whether to use the DMA bus for communication between sending and receiving. After selecting the DMA mode, it is recommended to turn off the corresponding interrupt enable bit of the UART, because the DMA request in the DMA mode can handle the data stream transmission by itself without the participation of the CPU.

UART transmission

In the DMA mode, when the sending data buffer UARTx_TDR is empty after the data is sent, the DMA request sent by the serial port will take effect to request the DMA to transfer the subsequent data to the sending data buffer UARTx_TDR. The DMA request sent by the serial port is cleared by the DMA controller.

UART receive

In DMA mode, when the data in the receive data buffer UARTx_RDR is fully updated, the DMA request received by the serial port will take effect to request the DMA to remove the data in the receive data buffer UARTx_RDR. The serial port receives the DMA request to be cleared by the DMA controller.

22.3.8 Interrupt control

UART interrupts (TI, RI) are generated when UARTx_TDR is empty and UARTx_RDR is full. Note: When TEN is not enabled and TIE is enabled, an interrupt will be triggered to move the data to the transmit buffer, but the data will be transmitted after TEN is enabled.

When the DMA channel is used, the UART read and write request signal is directly transmitted to the DMA controller;

When a frame containing data is sent and TI=1, the TC bit is set to 1 by the hardware.

The LBD interrupt occurs when the LIN bus function is enabled and the break signal is detected on RXD.

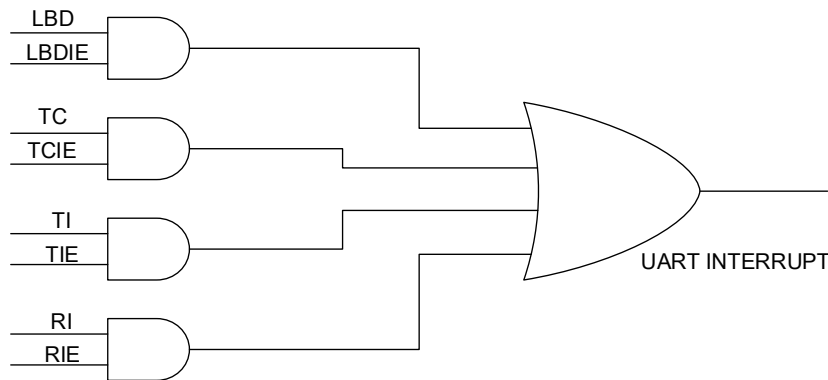


Figure 22-14 UART interrupt control diagram



22.4 UART registers

UART Module Register list (Base Address:0x4002 0000)

| Address | Register | Description |
|---------------------|----------|--|
| 0x4002 0000+0x400*x | FR | UART Interrupt Flag Register(UART0~3 corresponds to x: 0~3) |
| 0x4002 0008+0x400*x | TDR | UART Transmit Data Register(UART0~3 corresponds to x: 0~3) |
| 0x4002 000C+0x400*x | RDR | UART receive data register(UART0~3 corresponds to x: 0~3) |
| 0x4002 0010+0x400*x | ADDR | UART Address Configure Register(UART0~3 corresponds to x: 0~3) |
| 0x4002 0014+0x400*x | BRT | UART Baud Rate Configure Register(UART0~3 corresponds to x: 0~3) |
| 0x4002 0018+0x400*x | CR | UART Control Register(UART0~3 corresponds to x: 0~3) |

22.4.1 UARTx Interrupt Flag Register (UARTx_FR)

Offset Address: 0x0000+0x400*x
Reset value: 0x0000 0006

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|----------|-----|-----|-----------|------------|-----|----------|-----|
| Reserved | | | | | | | | LBD C | PEC | FEC | RXO VC | TXC OLC | TCC | Reserved | |
| - | | | | | | | | WO | WO | WO | WO | WO | WO | - | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----------|-----|-----|-----|-----|-----|----|----|-----|----|----|----------|-----------|----|----|----|
| Reserved | | | | | | | | LBD | PE | FE | RXO V | TXC OL | TC | TI | RI |
| - | | | | | | | | RO | RO | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 24 | Reserved | - |
| 23 | LBDC | LIN synchronous interval interrupt flag clear bit 0: No operation 1: Clear LBD |
| 22 | PEC | UART parity error flag clear bit 0: No operation 1: Clear PE |
| 21 | FEC | UART frame error flag clear bit 0: No operation 1: Clear FE |
| 20 | RXOVC | UART receive overflow flag clear bit 0: No operation 1: Clear RXOV |
| 19 | TXCOLC | UART transmit collision flag clear bit 0: No operation 1: Clear TXCOL |
| 18 | TCC | UART all one frame data transmitting completion flag clear bit 0: No operation 1: Clear TC |
| 17 ~ 8 | Reserved | - |
| 7 | LBD | LIN synchronous interval interrupt flag bit 0: Synchronous interval interrupt signal is not received 1: Synchronous interval interrupt signal is received Cleared by writing LBDC |



| | | |
|---|-------|---|
| 6 | PE | UART parity error flag bit 0: Parity is right 1: Parity has error, and is set by hardware Note: Cleared by writing 1 to PEC |
| 5 | FE | UART frame error flag bit 0: No frame error 1: Frame error occurs, set by hardware Note: Cleared by writing 1 to FEC |
| 4 | RXOV | UART receive overflow flag bit 0: No receive overflow 1: Receive overflow occurs, set by hardware Note: Cleared by writing 1 to RXOVC |
| 3 | TXCOL | UART transmit collision flag bit 0: No transmit collision 1: Transmit collision occurs, set by hardware Note: cleared by writing 1 to TXCOLC |
| 2 | TC | UART all one frame data transmitting completion flag bit 0: All one frame data is being transmitted, no set 1: All one frame data transmitting is completed, set by hardware Note: TC is cleared by writing 1 to TCC. It can also be cleared by the operation sequence of writing TDR . |
| 1 | TI | UART one byte data transmit interrupt flag bit 0: One byte data is being transmitted, no set 1: One byte data transmitting is completed, set by hardware Note: TI can only be cleared by write operation to TDR. |
| 0 | RI | UART one byte data receive interrupt flag bit 0: One byte data is being received, no set 1: One byte data receiving is completed, set by hardware Note: RI can only be cleared by read operation to RDR. |

22.4.2 UARTx Transmit Data Register (UARTx_TDR)

Offset Address: 0x0008+0x400*x
Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|-----|----------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | TB8 | TDR[7:0] | | | | | | | |
| - | | | | | | | RW | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---------------------------------------|
| 31 ~ 9 | Reserved | - |
| 8 | TB8 | The 9th bit sent in UART mode 2 and 3 |
| 7 ~ 0 | TDR[7:0] | UART transmit data register |



22.4.3 UARTx receive data register (UARTx_RDR)

Offset Address: 0x000C+0x400*x
 Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | RB8 | RDR[7:0] | | | | | | | |
| - | | | | | | | RO | RO | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|---|
| 31 ~ 9 | Reserved | - |
| 8 | RB8 | The 9th bit received in UART mode 2 and 3 |
| 7 ~ 0 | RDR[7:0] | UART receive data register |

22.4.4 UARTx Address Configure Register (UARTx_ADDR)

Offset Address: 0x0010+0x400*x
 Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SMAR[7:0] | | | | | | | | SADDR[7:0] | | | | | | | |
| RW | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 8 | SMAR[7:0] | Bit mask register, which determines which bit of SADDR will be detected |
| 7 ~ 0 | SADDR[7:0] | UART hardware address |

22.4.5 UARTx Baud Rate Configure Register (UARTx_BRT)

Offset Address: 0x0014+0x400*x
 Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | BFINE[3:0] | | | |
| - | | | | | | | | | | | | RW | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | SBRT[14:0] | | | | | | | | | | | | | | |
| - | RW | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



SH30F9/SA0 Series

| Bit | Symbol | Description |
|---------|------------|--------------------------------------|
| 31 ~ 20 | Reserved | - |
| 19 ~ 16 | BFINE[3:0] | Baud rate generator fine-tuning data |
| 15 | Reserved | - |
| 14 ~ 0 | SBRT[14:0] | Baud rate generator adjusting data |

22.4.6 UARTx Control Register (UARTx_CR)

Offset Address: 0x0018+0x400*x
Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|---------|-------------|-----|----------|-------|------|------|------|------|------|-------|---------|------|-----|-------|
| Reserved | | | | | | | | FER | TNEG | RNEG | DMAT | DMAR | TEN | REN | LINEN |
| - | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SBK | SM[1:0] | MULTIE[1:0] | PS | Reserved | SMOD0 | LBDL | LBDI | TCIE | TIE | RIE | SMOD2 | SBR TEN | STOP | | |
| RW | RW | RW | RW | - | RW | RW | RW | RW | RW | RW | RW | RW | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 24 | Reserved | - |
| 23 | FER | Frame error reception enable bit 0: Stop bit confirmation check is prohibited, any stop bit will set RI 1: Allow stop bit confirmation check, only valid stop bit (1) can set RI |
| 22 | TNEG | UART transmit polarity control bit 0: Send data is not inverted 1: Send data inverted |
| 21 | RNEG | UART receive polarity control bit 0: Received data is not inverted 1: Received data is inverted |
| 20 | DMAT | UART transmit DMA channel enable bit 0: Send data without DMA transmission 1: Send data through DMA transmission |
| 19 | DMAR | UART receive DMA channel enable bit 0: The received data is not transmitted through DMA 1: The received data is transmitted through DMA |
| 18 | TEN | UART transmit function enable bit 0: Disable the sending function 1: Enable the sending function |
| 17 | REN | UART receive function enable bit 0: Disable the receiving function 1: Enable the receiving function |
| 16 | LINEN | LIN mode function enable bit 0: Disable LIN mode 1: Enable LIN mode |
| 15 | SBK | LIN synchronization interval enable bit 0: Not send synchronization interval 1: Send synchronization interval, automatically cleared after sending |
| 14 ~ 13 | SM[1:0] | UART serial mode control bit 00: Mode 0, synchronous mode, fixed baud rate 01: Mode 1, 8-bit asynchronous mode, variable baud rate 10: Mode 2, 9-bit asynchronous mode, fixed baud rate 11: Mode 3, 9-bit asynchronous mode, variable baud rate |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 12 ~ 11 | MULTIE[1:0] | Ninth function selection bit 00: Software customization 01: Multi-machine communication flag 1x: parity bit |
| 10 | PS | Parity selection bit 0: Odd parity 1: Even parity |
| 9 | Reserved | - |
| 8 | SMOD0 | UART mode 0 baud rate control bit 0: In mode 0, the baud rate is 1/48 of the bus clock 1: In mode 0, the baud rate is 1/12 of the bus clock |
| 7 | LBDL | Synchronization interval detection threshold length 0: Low level length is greater than 10BIT low level 1: Low level length is greater than 11BIT low level |
| 6 | LBDIE | LIN interrupt delimiter interrupt enable bit 0: Disable LIN interrupt delimiter interrupt function 1: Enable LIN to interrupt the delimiter interrupt function |
| 5 | TCIE | All data transmission complete interrupt enable bit 0: Disable the data transmission completion flag TC triggering the interrupt function 1: Enable the data transmission complete flag TC to trigger the interrupt function |
| 4 | TIE | Single byte transfer interrupt enable bit 0: Disable the transmission interrupt flag bit TI triggering interrupt 1: Enable the transmission interrupt flag bit TI to trigger an interrupt |
| 3 | RIE | Single byte receive interrupt enable bit 0: Disable receiving interrupt flag bit RI to trigger interrupt 1: Enable receiving interrupt flag bit RI to trigger interrupt |
| 2 | SMOD2 | UART mode 2 baud rate control bit 0: In mode 2, the baud rate is 1/256 of the bus clock 1: In mode 2, the baud rate is 1/128 of the bus clock |
| 1 | SBR TEN | UART baud rate generator enable control bit 0: Disable the baud rate generator 1: Enable the baud rate generator |
| 0 | STOP | Stop bit selection bit 0: 1 stop bit 1: 2 stop bits |



23. Serial Peripheral Interface (SPI)

23.1 Introduction

The serial peripheral interface (SPI) is a kind of high-speed serial communication interface that allows MCU to perform full duplex, synchronous serial communication with peripherals (including other MCUs).

The following figure shows a typical SPI bus network consisting of one master device and several slave peripherals. The master device controls 4 parallel ports of the slave device /SS pin to select one of the slave devices for communication.

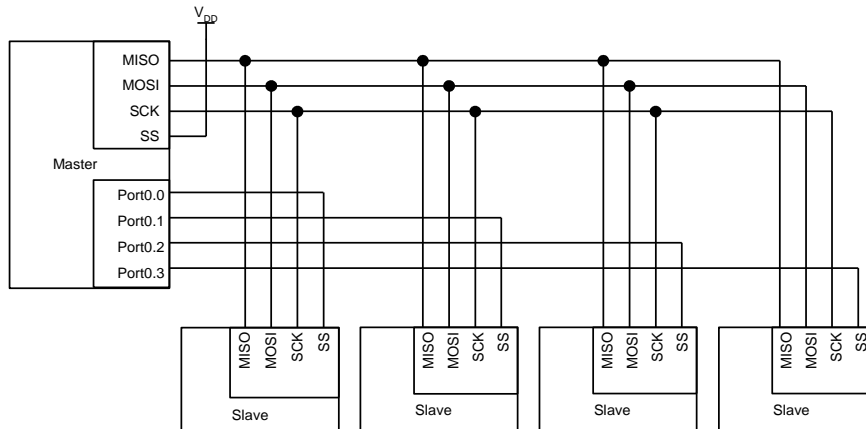


Figure 23-1 SPI Bus Network Diagram

23.2 Main Features

- Full duplex, three-wire synchronous transmitting
- Master & slave operation
- 10 programmable master clock frequencies
- Serial clock with programmable polar phase
- Optional data word width is 8 or 16 and 32
- Supports DMA communication
- Master mode fault error flag with MCU interrupt
- Write collision flag protection
- Can select LSB or MSB transmitting



23.3 Function Description

23.3.1 Signal Description

Main output slave input (MOSI)

This signal connects the master device and the slave device. Data is serially transferred from the master device to the slave device through MOSI, with the master device outputting and the slave device inputting.

Master input slave output (MISO)

This signal connects the slave device and the master device. Data is serially transferred from the slave device to the master device through MISO, with the slave device outputting, and the master device inputting. When SPI is configured as slave device not being selected (\overline{SS} pin is high level), MISO pin of the slave device is in high-resistance state.

SPI serial clock (SCK)

The SCK signal is used to control the synchronous movement of input and output data on MOSI and MISO lines. One byte is transferred every 8 clock period on the wire. If the slave device is not selected (\overline{SS} pin is high), the SCK signal is ignored by this slave device.

Slave device select pin (\overline{SS})

Each slave peripheral is selected by a slave select pin (\overline{SS} pin), and when the pin signal is low level, indicating that this slave device is selected. The master device can select each slave device by controlling the port level connected to the slave \overline{SS} pin by software. Obviously, only one master device can drive the communication network. In order to prevent MISO bus collision, only one slave device is allowed to communicate with the master device at the same time. In master device mode, the \overline{SS} pin state is associated with the MODF flag bit in the SPI flag register SPIx_FR to prevent multiple masters from driving MOSI and SCK.

The \overline{SS} pin can be used as a normal port or other functions in the following cases:

- (1) The device is configured as a master device and the SSDIS bit in the SPIx_CR register of the SPI control register is set to 1. This configuration only exists when there is only one master device in the communication network, so the MODF flag bit in the SPI status register SPSTA will not be set to 1.
- (2) The device is configured as a slave device and the CPHA bit and the SSDIS bit of the SPI control register SPIx_CR are set to 1. This configuration exists in a communication network with only one master device and one slave device. Therefore, the device is always selected, and the master device does not need to control the \overline{SS} pin of the slave device to select it as the communication destination.

Note: When $CPHA = 0$, \overline{SS} pin generates falling edge, indicating transmitting is initiated.

23.3.2 Baud Rate

In master mode, the SPI baud rate has ten selectable frequencies, which are 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions of the internal clock, which can be selected by setting the SPR[3:0] bit of the SPIx_CR register.

23.3.2.1 Principle Description

The detailed structure of the SPI module is shown in the following figure.

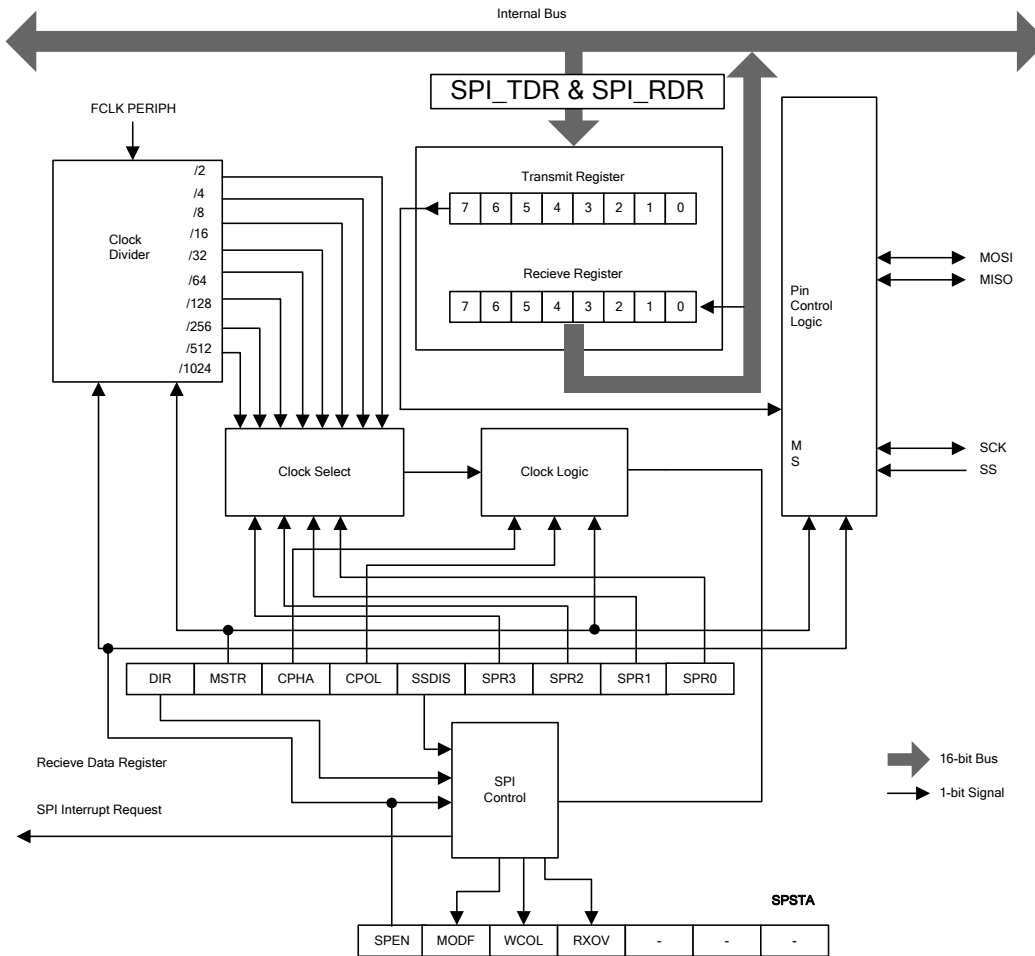


Figure 23-2 SPI Module Block Diagram

23.3.3 Working Mode

SPI can be configured as one mode from the master mode or the slave mode. The configuration and initialization of the SPI module is done by setting the SPIx_CR register (it is recommended to configure each control mode bit first, then enable SPEN@SPIx_CR). After the configuration is complete, data transfer is done by setting SPIx_CR, SPIx_TDR, SPIx_RDR (serial peripheral data register).

During SPI communication, data is synchronously and serially shifted in and out. The serial clock line (SCK) keeps the movement and sampling of data on the two serial data lines (MOSI and MISO) synchronized. The slave device select line (\overline{SS}) can independently select the SPI slave device; if the slave device is not selected, it cannot participate in the activity on the SPI bus.

When the SPI master transmits data to the slave via MOSI wire, the slave device transmits data to the master device via MISO wire in response, which enables simultaneous full duplex transfer of data transmitting and receiving at the same clock. Write to the SPI data send register SPIx_TDR, write to the sender when the cache is empty, and write to the send data register when there is data in the cache SPIx_TDR. Reading the SPIx_RDR register will result in the data in the received cache.

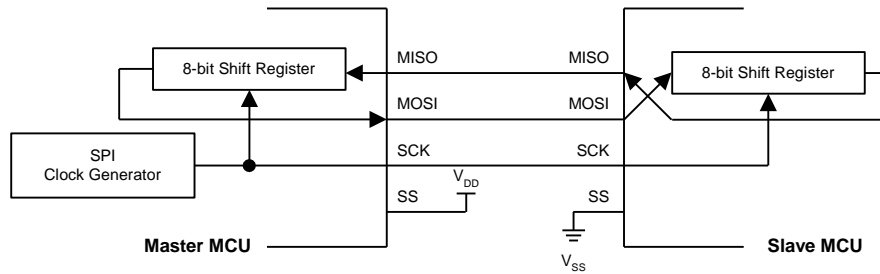


Figure 23-3 Full Duplex Master-slave Interconnection Diagram

Master mode

(1) Mode start-up

The SPI master device controls the start-up of all data transfers on the SPI bus. When the MSTR bit in the SPIx_CR register is set to 1, SPI is operating in master mode and only one master device can initiate the transfer.

(2) Transmit

In SPI master mode, write a byte of data to the SPI send data register SPIx_TDR, if the sending cache is empty, the data will be written to the sender cache, if the sending cache is not empty, the data will be placed in the send data register, when the sender is sent, the data will be automatically transferred from the sending data register into the send cache, and SPTI interrupt. If the data register is sent at full and the data is written to the SPIx_TDR, the master SPI generates a WCOL signal to indicate that the write is too fast. However, the data in the Send Data Register and send Cache is not affected and is not interrupted. Also, if the send cache is non-empty, the master device immediately moves the data from the sender serially to the MOSI line in accordance with the SPI clock frequency on the SCK. When the SPIx_TDR of the sending data register is empty, the SPTI bit in the SPIx_FR register is set 1. If the SPTIE interrupt is allowed, an interrupt occurs when SPTI position 1 occurs. SPTI@SPIx_FR will be automatically zeroed after the data is populated within the SPIx_TDR.

(3) Receive

When the primary device transmits data to the slave device through the MOSI line, the corresponding from the device also transmits the contents of its sending cache to the receiving cache of the master device via the MISO line, enabling full duplex operation. The data received from the device is stored in the receiving cache of the master device in the MSB or LSB priority transmission direction. When a byte of data is fully moved into the receive register, the data is automatically moved into the SPIx_RDR and an SPRI interrupt occurs. The processor can obtain this data through the read SPIx_RDR register. The receive interrupt flag bit SPRI is set 1 at full SPIx_RDR, and the SPRI is automatically zeroed when the data in the SPIx_RDR is read blank. If the excess occurs (the received data is not taken away in time, SPIx_RDR and the receiving cache have data, try to start the next reception), RXOV position 1, indicates that the data over limit occurs, at this time the receive shift register maintains the original data and SPRI position 1, so that until the SPRI bit is cleared 0, the SPI master device will not receive any data.

Slave mode

(1) Mode start-up

When the MSTR bit in the SPIx_CR register is cleared, the SPI is operating in slave mode. The \overline{SS} pin of the slave device must be set low before data transfer and must be kept low until one byte of data has been transferred.

(2) Transmit and Receive

In slave mode, according to the SCK signal controlled by the master device, the data moves in via the MOSI pin and the MISO pin is moved out. A bit counter records the number of edges of the SCK when the receiving shift register moves in 8 bits of data (one byte) and sends the shift register to move out of the 8-bit data (one byte). Data can be obtained by reading SPIx_RDR registers. If a SPRIE interrupt is allowed, an interrupt will also occur when the SPRI is set to 1.



SPI cannot initiate data transfer from the slave device, so the SPI slave device must write the data to the send register before the primary device starts a new data transfer SPIx_TDR (when only one data is written, write directly to the sender cache, and when writing two data, store it in the send cache and send data register SPIx_TDR). After the SPI module is reset, if data is not written from the device before it is first sent, the "0x00" byte is transferred from the device to the primary device. If the sending data register SPIx_TDR and the sending cache have data in the write SPIx_TDR operation, then the SPI is marked from the WCOL flag position 1 of the device, i.e. if the data register is already containing data SPIx_TDR the sending data register, the SPI is written from the WCOL location 1 of the device, indicating a write SPIx_TDR conflict. However, the sending data register and the data within the sending cache are not affected and the transfer is not interrupted.

23.3.3.1 Transfer Form

By setting the CPOL bit and the CPHA bit of the SPIx_CR register by software, user can select the four combinations of SPI clock polarity and phase. The CPOL bit defines the polarity of the clock, which means the level state at idle, and it has little effect on the SPI transmitting format. The CPHA bit defines the phase of the clock, which defines the edge of the clock that allows the data sampling shift. In the two devices of master-slave communication, the clock polarity phase settings should be the same.

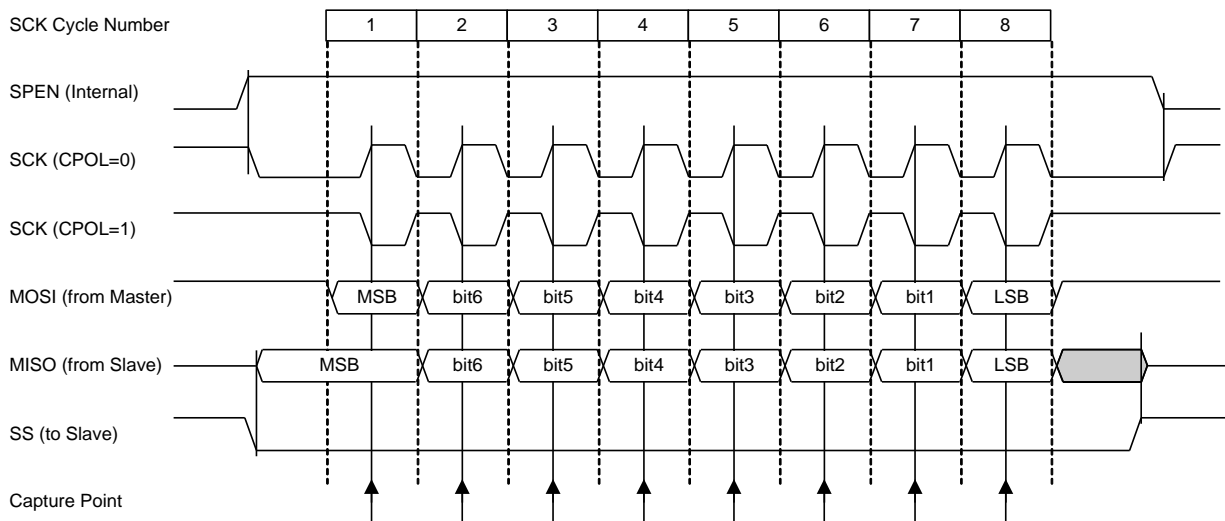


Figure 23-4 Data Transfer Form (CPHA = 0) Diagram

If CPHA = 0, the first edge of SCK captures data, the slave must prepare the data before the first edge of SCK, so in slave mode, the falling edge slave device of the \overline{SS} pin starts transmitting data. The \overline{SS} pin must be pulled high after finishing transmitting one byte each time, and reset to low level before the next byte is transmitted, so when CPHA = 0, the SSDIS masking function has no effect. In addition, for this mode, the SPI slave mode has also added the SPSFF@SPIx_CR control bit. After the control bit is set, the first byte data also needs to be loaded by the falling edge of the \overline{SS} pin. From the second byte on, data begins to be loaded using the last CLK edge of the previous byte to achieve continuous transmitting. At the same time, DMA method can also be used to transfer data after enabling this function.



SH30F9/SA0 Series

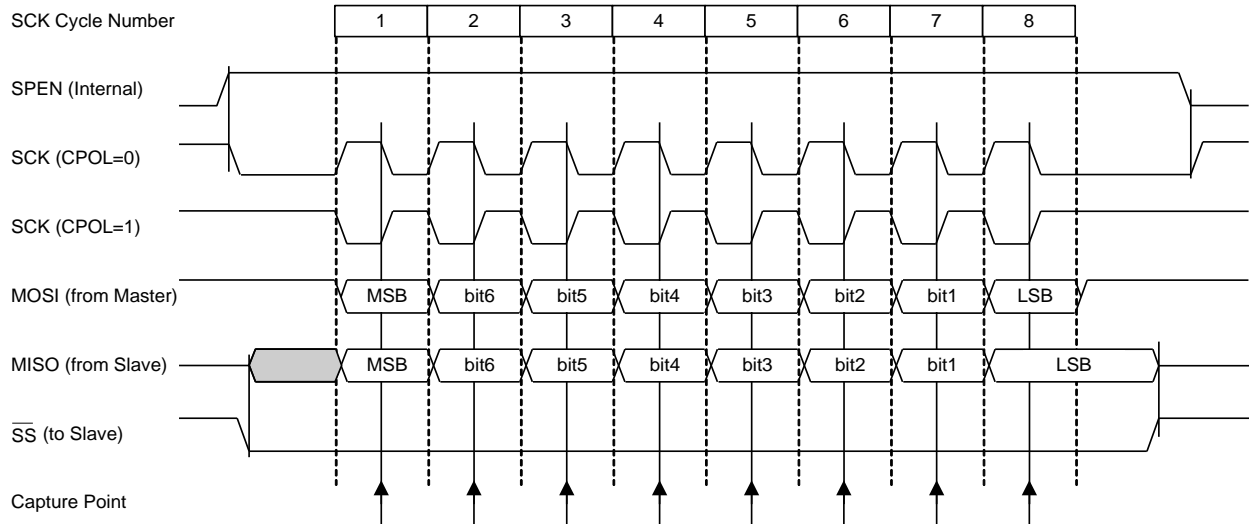


Figure 23-5 Data Transfer Form (CPHA = 1) Diagram

If CPHA = 1, the master device outputs data to the MOSI wire on the first edge of SCK, and the slave device takes the first edge of SCK as the start transmitting signal. User must complete the write operation on SPIx_TDR before the second edge of the first SCK. The \overline{SS} pin always keeps at low level during the transfer of each byte of data. This form of data transfer is the preferred form of communication between a master device and a slave device.

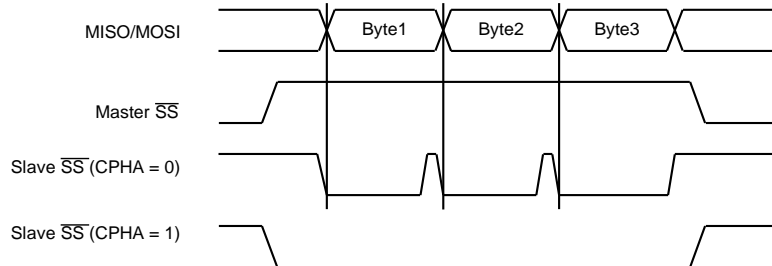


Figure 23-6 CPHA/NSS Timing Diagram

Note: When SPI is used as slave mode and the CPOL bit of the SPIx_CR register is cleared, the SCK port must turn on the pull-up resistor before data transfer.

23.3.3.2 Data Word 8-bit, 16-bit or 32-bit Configuration

The data word length for a single transfer can be configured as 8-bit, 16-bit or 32-bit via the SPDATL bit in the SPIx_CR register. When using 16-bit word length, both the shift register and the buffer register are expanded to 16-bit mode, writing and reading to both SPIx_TDR and SPIx_RDR are 16-bit operations. (Note: TDR, RDR's real hardware is 16 bits, 8 bit mode, the high 8 bits can be ignore) The timing diagram is as follows:



SH30F9/SA0 Series

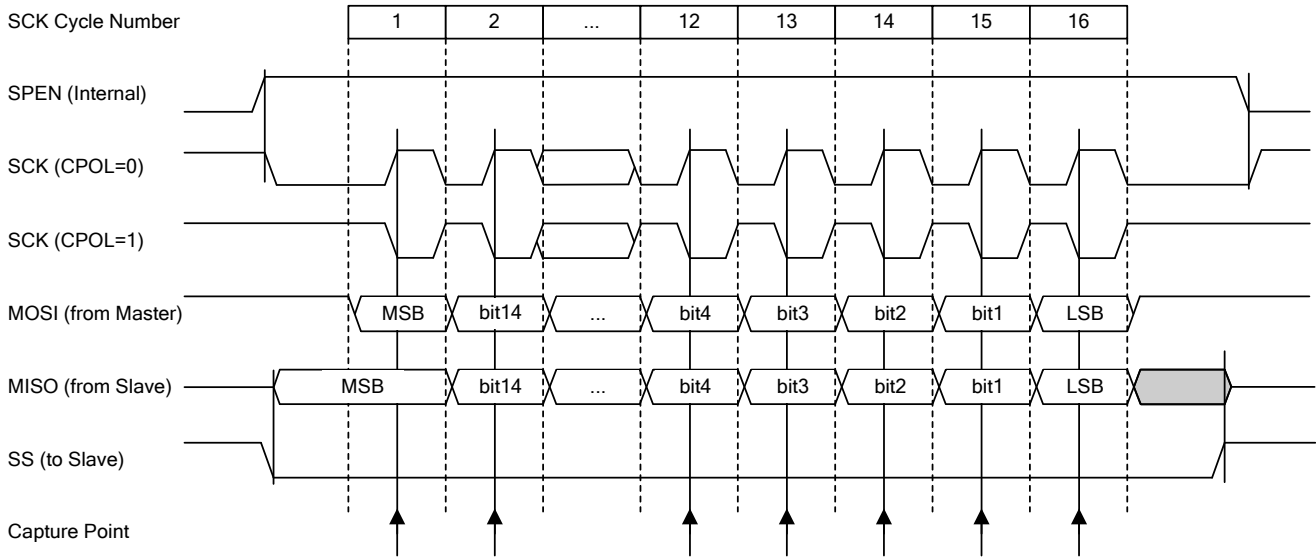


Figure 23-7 Data Transfer Form(CPHA = 0)

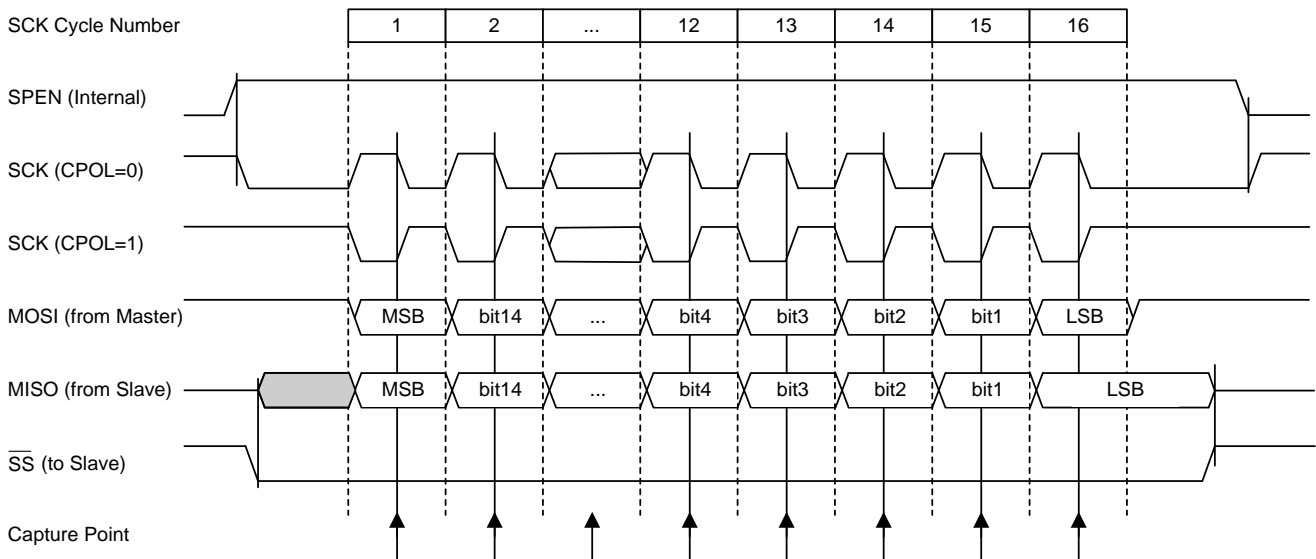


Figure 23-8 Data Transfer Form(CPHA = 1)

23.3.3.3 DMA Communication

In order to achieve maximum communication speed, it is necessary to fill the data into the SPI buffer in time, and the data in the receive buffer must also be read in time to prevent overflow. To facilitate high-speed data transfers, SPI uses a DMA mechanism that uses simple request/response.



The SPI module can issue DMA transfer requests when the corresponding enable bits DMAT, DMAR@SPIx_CR on the SPIx_CR register are set. The transmit buffer and the receive buffer have their own DMA requests. After selecting the DMA mode, it is recommended to disable the corresponding interrupt enable bit of the SPI, because in DMA mode, the DMA request can process data stream transmitting without the CPU participation.

SPI transmitting

When transmitting, DMA request is issued each time SPTI is set to 1, and the DMA controller writes data to the SPIx_TDR register, therefore the SPTI flag is cleared.

SPI receiving

When receiving, DMA request is issued each time SPRI is set to 1, and the DMA controller reads data from the SPIx_RDT register, therefore the SPRI flag is cleared.

When data is transmitted only using SPI, only the SPI transmit DMA channel needs to be enabled. At this time, if the received data is not read, RXOV is set to 1 (software does not need to pay attention to this flag).

23.3.3.4 SPSTA State Exception Case

The MODF, WCOL and RXOV flags in the SPIx_FR register indicate error cases in SPI communication:

(1) Mode Fault (MODF)

The mode fault error in SPI master mode indicates that the level state on the \overline{SS} pin is inconsistent with the actual device mode. After the MODF bit in the SPIx_FR register is set to 1, it indicates that the problem of multi-master collision exists in the system control. In this case, the SPI system is affected as follows:

The SPI receive/error CPU interrupt request is generated. In the interrupt, the software needs to complete the following operations: disable SPI, switch to the slave mode, and can re-enable the SPI as required.

When the \overline{SS} pin disable bit (SSDIS) in the SPIx_CR register is cleared and the \overline{SS} pin signal is low, the MODF flag is set to 1. However, for systems with only one master device, the \overline{SS} pin of the master device is pulled low, and that is definitely not the other master device trying to drive the network. In this case, to prevent MODF from being set to 1, the SSDIS bit in the SPIx_CR register can be set to 1, and the \overline{SS} pin can be used as a general purpose I/O port or other function pin.

When restarting serial communication, user must clear the MODF bit by software, set the MSTR bit and the SPIEN bit in the SPIx_CR register and restart the main mode.

(2) Write Collision (WCOL)

Write collision will be caused by writing to the SPIx_TDR register during the data transmitting sequence, and the WCOL bit in the SPIx_FR register is set to 1. Setting WCOL bit to 1 will not cause an interrupt and the transmitting will not be aborted. The WCOL bit needs to be cleared by software.

(3) Overrun Case (RXOV)

The overrun case occurs when the master device or the slave device has not cleared the SPRI bit and the master device or the slave device attempts to send several more data bytes. In this case, the receive shift register keeps the original data, SPRI is set to 1, and also the SPI device does not receive data until SPRI is cleared. The interrupt is called continuously until the SPRI bit is cleared, and the transmitting will not be aborted. Setting the RXOV bit to 1 will not cause an interrupt and the RXOV bit needs to be cleared by software.

23.3.3.5 Interrupt

Three SPI state flags, SPTI, SPRI & MODF, can generate a CPU interrupt request.

Serial peripheral data transmitting and receiving flag:

The SPI interrupt (SPTI, SPRI) is generated after the SPIx_TDR and SPIx_RDR buffers complete the transfer of one byte data (the transmit register is empty when transmitting and the receive register is full when receiving);

When the DMA channel is enabled, the DMA request signal of SPI is generated after the completion of one byte transfer (the transmitting register is empty when transmitting and the receiving register is full when receiving), and is directly transmitted to the DMA controller;

Mode fault flag MODF: setting this bit to 1 indicates that the level on the \overline{SS} pin is in a bus collision with the SPI multi-master mode. The SSDIS bit being 0 and setting MODF to 1 will generate an SPI receiver/error CPU interrupt request. When SSDIS is set to 1, no MODF interrupt request is generated.

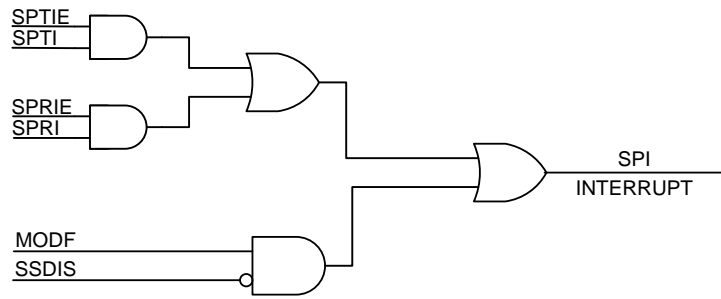


Figure 23-9 SPI Interrupt Control Diagram



23.4 SPI registers

SPI Module Register list (Base Address:0x4002 1000)

| Address | Register | Description |
|---------------------|----------|---|
| 0x4002 1000+0x400*x | FR | SPI interrupt flag register(SPI0~1 corresponds to x: 0~1) |
| 0x4002 1004+0x400*x | TDR | SPI transmit data register(SPI0~1 corresponds to x: 0~1) |
| 0x4002 1008+0x400*x | RDR | SPI receive data register(SPI0~1 corresponds to x: 0~1) |
| 0x4002 100C+0x400*x | CR | SPI control register(SPI0~1 corresponds to x: 0~1) |

23.4.1 SPI interrupt flag register (SPIx_FR)

Offset Address: 0x0000+0x400*x

Reset value: 0x0000 0002

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|-----------|-----------|----------|-----|-----|
| Reserved | | | | | | | | | | WC OLC | RXO VC | MOD FC | Reserved | | |
| - | | | | | | | | | | WO | WO | WO | - | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----------|----------|----------|----------|------|------|
| Reserved | | | | | | | | | | WC OL | RXO V | MOD F | BUS Y | SPTI | SPRI |
| - | | | | | | | | | | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 22 | Reserved | - |
| 21 | WCOLC | Write collision flag clear bit 0: Invalid 1: Clear |
| 20 | RXOVC | Receive overflow flag clear bit 0: Invalid 1: Clear |
| 19 | MODFC | Mode fault flag clear bit 0: Invalid 1: Clear |
| 18 ~ 6 | Reserved | - |
| 5 | WCOL | Write collision flag bit 0: There is no write collision or write collision has been dealt with, cleared by writing 1 to WCOLC bit by software 1: Write collision is detected, set by hardware |
| 4 | RXOV | Receive overflow flag bit 0: There is no receive overflow or receive overflow has been dealt with, cleared by writing 1 to RXOVC bit by software 1: Receive overflow is detected, set by hardware |
| 3 | MODF | Mode fault flag bit 0: No fault, cleared by writing 1 to MODFC bit by software 1: NSS pin level is inconsistent with the SPI mode, set by hardware |
| 2 | BUSY | master device BUSY state flag 0: Master device is in idle state 1: Master device is in transmitting state, BUSY |



| | | |
|---|-------------|---|
| 1 | SPTI | SPI one byte data transmit interrupt flag bit 0: Transmit data register TDR is in full state 1: Transmit data register TDR is in empty state, set by hardware Note: SPTI clears by writing to TDR |
| 0 | SPRI | SPI one byte data transmit interrupt flag bit 0: Transmit data register RDR is in empty state 1: Transmit data register RDR is in full state, set by hardware Note: SPRI clears by reading RDR |

23.4.2 SPI transmit data register (SPIx_TDR)

Offset Address: 0x0004+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| TDR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TDR[31:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | TDR[31:0] | Transmit buffer register 8-bit transmit buffer register when SPDATL=00 16-bit transmit buffer register when SPDATL=01 32-bit transmit buffer register when SPDATL=1X |

23.4.3 SPI receive data register (SPIx_RDR)

Offset Address: 0x0008+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| RDR[31:0] | | | | | | | | | | | | | | | |
| <i>RO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RDR[31:0] | | | | | | | | | | | | | | | |
| <i>RO</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 0 | RDR[31:0] | Receive buffer register 8-bit receive buffer register when SPDATL=00 16-bit receive buffer register when SPDATL=01 32-bit receive buffer register when SPDATL=1X |



SH30F9/SA0 Series

23.4.4 SPI control register (SPIx_CR)

Offset Address: 0x000C+0x400*x

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----------|------------|------------|-----------|-----------|--------------|-----|----------|----------|----------|-----------|----------|-----|-------------|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | SPDATL[1:0] | |
| | | | | | | | | | | | | | | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SPS FF | SPIE N | SPD MAT | SPD MAR | SPTI E | SPRI E | Rese rved | DIR | MST R | CPH A | CPO L | SSDI S | SPR[3:0] | | | |
| RW | RW | RW | RW | RW | RW | - | RW | RW | RW | RW | RW | RW | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|--|
| 31 ~ 18 | Reserved | - |
| 17 ~ 16 | SPDATL[1:0] | Data word length selection bit 00: Send 8bit data at a time 01: Send 16bit data at a time 1x: Send 32bit data at a time |
| 15 | SPSFF | Slave fast transmit mode control bit (valid when CPHA=0) 0: Normal mode, the SS pin of the salve must be pulled high after each time the transfer of one data (8,16 or 32 bits) is complete. Before transmitting the next data, SS must be pulled low to load data into the shift register. 1: Fast mode, the first data transmitted by the slave needs to be triggered with the falling edge of SS, and the second data is loaded to the shift register using the last CLK of the previous data (note: it will not be loaded if there is no data in the transmit buffer) |
| 14 | SPIEN | SPI enable 0: SPI is disabled 1: SPI is enabled |
| 13 | SPDMAT | SPI data transmit DMA channel enable bit 0: DMA is not used to transfer transmitted data of SPI 1: DMA is used to transfer transmitted data of SPI |
| 12 | SPDMAR | SPI data receive DMA channel enable bit 0: DMA is not used to transfer received data of SPI 1: DMA is used to transfer received data of SPI |
| 11 | SPTIE | SPI transmit interrupt flag enable bit 0: Transmit interrupt flag bit SPTI is disabled to trigger interrupt 1: Transmit interrupt flag bit SPTI is enabled to trigger interrupt |
| 10 | SPRIE | SPI receive interrupt flag enable bit 0: Receive interrupt flag bit SPRI is disabled to trigger interrupt 1: Receive interrupt flag bit SPRI is enabled to trigger interrupt |
| 9 | Reserved | - |
| 8 | DIR | Data transmit direction selection bit 0: MSB transmitting priority 1: LSB transmitting priority |
| 7 | MSTR | SPI master and slave devices configure bit 0: Configures SPI as slave device 1: Configures SPI as master device |
| 6 | CPHA | Clock phase control bit 0: The first edge of CK period collects data 1: The second edge of CK period collects data |



SH30F9/SA0 Series

| | | |
|-------|-----------------|--|
| 5 | CPOL | Clock polarity control bit 0: CK is at low level in IDLE state 1: CK is at high level in IDLE state |
| 4 | SSDIS | NSS pin control bit 0: In master/slave mode, NSS pin is turned on 1: In master/slave mode, NSS pin is turned off, no MODF interrupt request will be generated In slave mode, if CPHA=0, this bit has no effect |
| 3 ~ 0 | SPR[3:0] | Clock division control bit 0000: Bus clock /2 0001: Bus clock /4 0010: Bus clock /8 0011: Bus clock /16 0100: Bus clock /32 0101: Bus clock /64 0110: Bus clock /128 0111: Bus clock /256 1000: Bus clock /512 1001: Bus clock /1024 Other: Bus clock /2 |



24. Two-Wire Serial Interface (TWI)

24.1 Introduction

Two-wire serial interface (TWI) inherits and develops the I2C bus interface and is fully compatible with the I2C bus. It has the advantages of simple hardware implementation, convenient software design, reliable operation and low cost. TWI consists of a clock wire and a data wire, it is transmitting in bytes, and is compatible with the SMBus specification, automatically processing byte transfers and tracking serial communications.

SCL/SDA is the signal wire of the TWI bus. SDA is a bidirectional data wire and SCL is a clock wire. To transmit data on the TWI bus, first transmit the highest bit (MSB), the master device sends a start signal, and then the master device transmits one or more bytes of data. After the data is transmitted, the master device sends a stop signal to complete the simplest TWI transmitting.

Typical TWI bus application is shown below, supporting up to 128 different devices for communication.

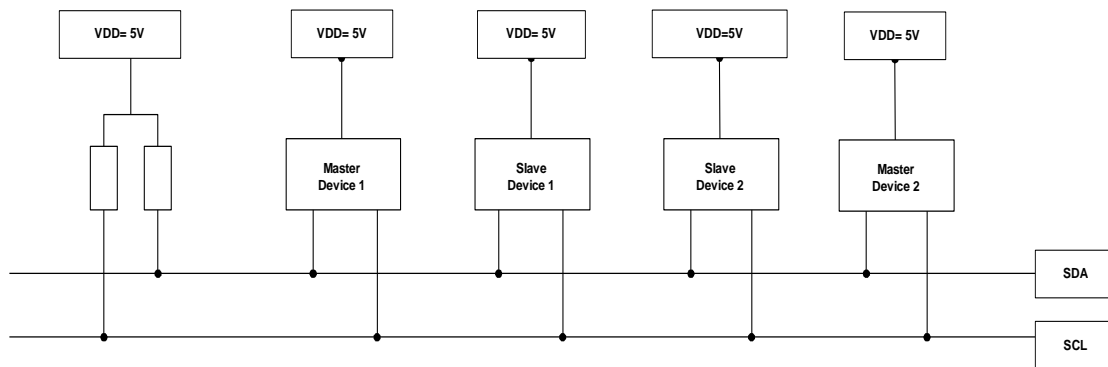


Figure 24-1 Typical TWI Bus Application Block Diagram

24.2 Main Features

- Two-wire mode
- OD output (needs to be configured in GPIO), signal level is not affected by VDD, and will not affect VDD
- Supports master mode and slave mode
- Allow to transmit and receive
- Supports arbitration function of multi-master device communication
- Has low level bus timeout judgement
- Can wake up the system in sleep mode, does not support stop mode wake-up
- Address is programmable, with multiple address mask bits, supports broadcast function
- Supports standard mode and fast mode

24.3 Function Description

24.3.1 Bus Signal Timing

The initial state of the bus is bus idle, which means both SDA and SCL signal wire are at high level, all devices on the bus release the bus, and the respective pull-up resistors of the two signal wires pull the level high.

In data transfer, transfer of each bit on the data wire requires a pulse on the clock wire. The data wire should remain stable when the clock is at high level and allow the wire line to change when the clock is at low level. However, the start condition and the stop condition are special, the former refers to the falling edge of the data wire during the high level time of the clock, and the latter refers to the rising edge of the data wire during the high level time of the clock.

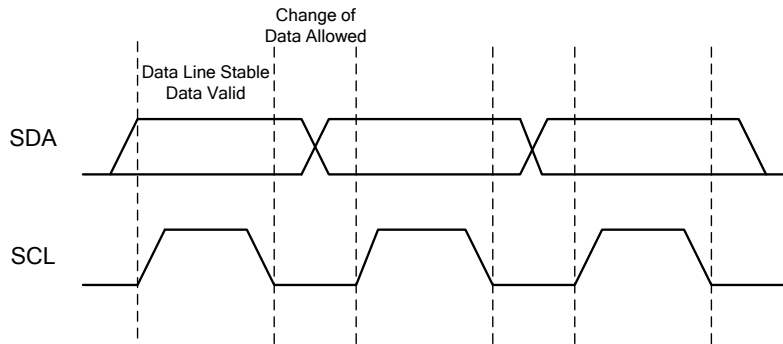


Figure 24-2 Basic Bit Transfer Characteristics of TWI Bus

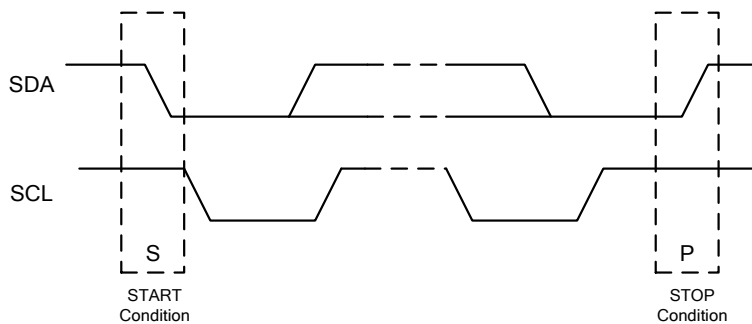


Figure 24-3 Start Condition (Start Signal) and Stop Condition (Stop Signal) of TWI Bus

Start condition and stop condition of TWI bus are all sent by the master device.

Master device can initiate and terminate a transfer. The master device starts a transfer when start condition is sent and this transfer will be ended if a stop condition is sent. The bus is defined as "busy" state between the start condition and the stop condition. Other master devices should not attempt to initiate a transfer. It is defined as "repeated start condition" that the master device in the "busy" state sends a start condition again, indicating that the master device wants to start a new transfer without giving up the bus. After the repeated start condition is sent, the bus is still in the "busy" state until a stop condition occurs on the bus. Since the nature of the repeated starting condition and the starting condition are exactly the same, unless otherwise specified, starting condition will be used instead of both.

All data packets (including address packets) consist of 9 bits, including 1 byte and an acknowledge bit. The master device sends clock, a start condition and a stop condition, and the receiver responds with an acknowledge signal. The receiver sends an "acknowledge (ACK)" signal by pulling the data line low at the ninth clock pulse; or a "not acknowledge (NACK)" signal by maintaining high level at the ninth pulse. When the receiver receives the last byte, or for some reason cannot continue to receive data, it should respond with a "not acknowledge (NACK)" signal.

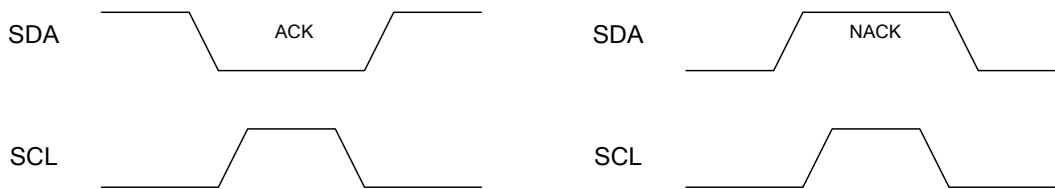


Figure 24-4 ACK and NACK Signals

When the slave does not respond to the master addressing signal for some reason (such as the slave is performing real-time processing and cannot respond to bus communication), it must return a NACK, and then the master generates a stop signal to end the bus's data transfer.

If the slave responds to the master but cannot continue to receive more data after the data transfer is on for a period of time, it can notify the master by returning a NACK, and the master should issue a stop signal to end the continuous transfer of the data.



SH30F9/SA0 Series

When the master receives the last data byte when receiving data, it must send a NACK to the slave to inform the slave to stop data transfer, and then the slave releases the SDA wire to allow the master to generate a stop signal.

Note: TWI is two-way data communication. When a node sends data through SDA, it requires other nodes to release the SDA wire, which means the SDA wire is in external pull-up state.

In the TWI bus, transfers are performed bit by bit from high to low.

A transfer usually includes a start condition, address + read/write, one or more data packets and a stop condition. Data formats that only contain start and stop conditions do not meet the communication rules. It is worth noting that the "wire and" structure provides convenience to the handshake signal between the master and the slave. When the master is relatively fast or the slave needs to handle other matters, the slave can pull down the clock wire to lengthen the low level time of the clock wire, thus reducing the communication frequency. The slave can stretch the low level period of the clock wire but does not affect the high level period of the clock wire.

When the response signal is generated, SH30F9/SA0 series pulls down the SDA signal wire. During the time the interrupt flag bit is set, SH30F9/SA0 series pulls the SCL signal wire low and releases the SDA signal wire. After the interrupt processing is completed, the TWINT flag is cleared and the SCL wire is released.

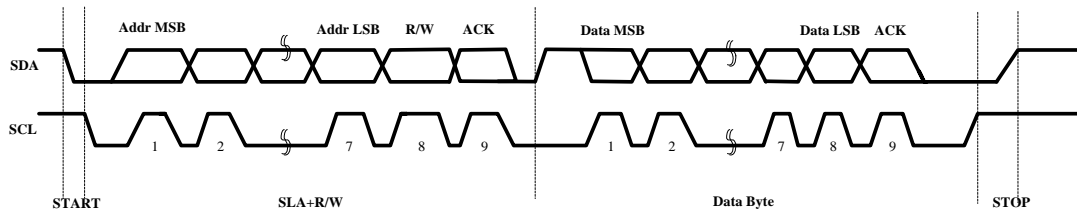


Figure 24-5 Diagram of a Typical TWI Data Transfer Timing

After the main controller completes one communication and wants to continue to occupy the bus for communication, and does not release the bus, the repeated start condition Rs needs to be used. It is both the end of the previous data transfer and the beginning of the next transfer.

In special cases, if all communication on the TWI bus needs to be disabled, it can be done by blocking or turning off the bus. The specific operation is to lock SCL at low level with any device on the bus.

24.3.2 Clock Synchronization

When multiple masters want to control the bus at the same time, the bus will determine whether the clock wire is at high or low level based on the "wire and" principle. For all masters involved in the transfer, it is important to clearly define the start of each clock pulse.

A high-to-low transition of the clock wire level will cause all devices participating in the transfer to start low level timing. Each device releases its clock wire when it reaches its own low level requirement, and enters a high level wait time before the clock wire goes high level; when all devices are fully counting low level time, the clock wire goes high level. All devices then begin timing high level, and the first device that fully counts high level time will pull the clock line wire, entering the next clock period.

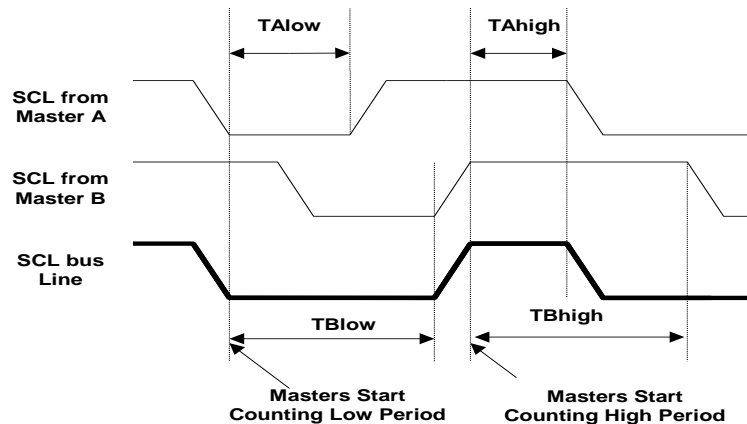


Figure 24-6 Clock Synchronization Timing Diagram



Note: When multiple masters simultaneously control the SCL wire, the “wire and” will cause the low level period of the SCL wire to be determined by the master with the longest low level time, while the high level period is determined by the master with the shortest high level time.

24.3.3 Bus Arbitration

The master can only start a transfer when the bus is in the "idle" state. Two or more masters may simultaneously send a start condition within the minimum duration ($t_{\text{HOLD}}: \text{STA}$). As a result, a normal start condition will be obtained on the bus, multiple masters will start transmitting, and subsequent transfers need to be arbitrated.

Since the master sending the start condition cannot know whether other masters are competing for the bus, it can only judge by the level detection of the SDA wire during the SCL high level period, which means the arbitration logic checks whether the logic 1 sent by each SDA actually appears on the bus, if there are other masters transmitting low level and pulling the SDA line low, the master transmitting the high level will lose arbitration and must abandon the bus. This is the basic arbitration logic.

Note: This is a "low level priority" arbitration principle that awards the bus to the master device that sends low level on SDA first. The master that has lost the arbitration will switch from the transmit state to the receive state and continue to transmit the clock pulse (on SCL) until the current transferred byte has been transmitted

Note: In addition to the master transmit mode, in the master receive mode, when the master returns a NACK signal (logic 1) to the slave, arbitration loss may occur. Since NACK appears at the end of the serial byte, the module will no longer send a clock pulse.

Arbitration is performed bit by bit. Its first phase is to compare the address bits. If multiple masters address the same device, they may successfully pass the address phase arbitration, and the arbitration will continue to compare the data bits (the master transmit mode), or response bits (the master receive mode). Because the address and data information on the bus is determined by the master that won the arbitration, no information is lost during the arbitration.

If this master has the slave mode enabled at the same time, it should check whether the address on the wire matches itself after losing the arbitration in the address transmitting phase; if this is an access to itself, it should immediately switch to the slave mode and receive the information.

In each transmitting, the “repeated start condition” on the wire still needs to be detected. When detecting that this is not the “repeated start condition” issued by itself, the current transmitting should be exited immediately.

When a master sends a repeated start condition or a stop condition, another master may still be transmitting data, which causes an undetermined state (the specification requires that this repeated start condition or stop condition be retransmitted in the same position of frame format). In order to achieve this, it is defined that in the following three cases the bus is not performing arbitration:

1. Master 1 sends a repeated start condition, master 2 sends the data.
2. Master 1 sends a stop condition, master 2 sends data.
3. Master 1 sends a repeated start condition and master 2 sends a stop condition.

In addition, arbitration only occurs between masters, and slaves do not participate in arbitration.

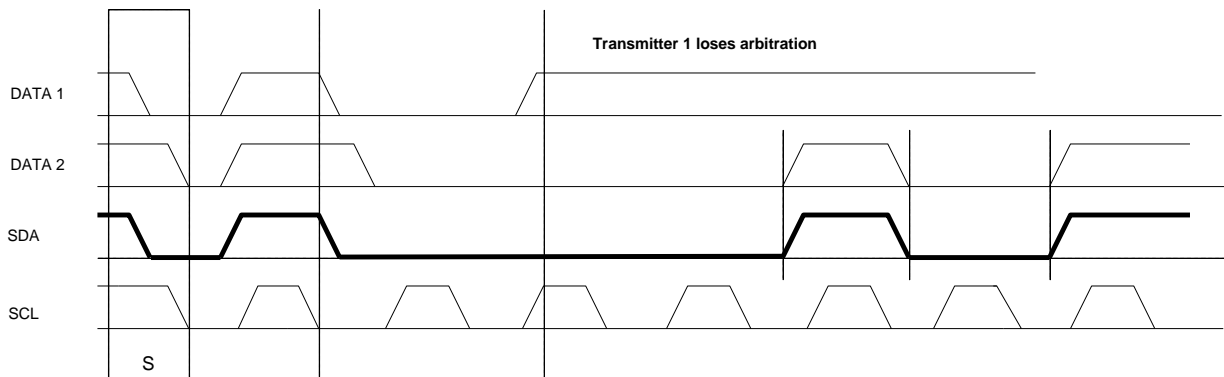


Figure 24-7 Data Arbitration Timing Diagram



24.3.4 Detailed Structure of TWI Communication Module

The following figure describes the detailed structure of TWI communication module.

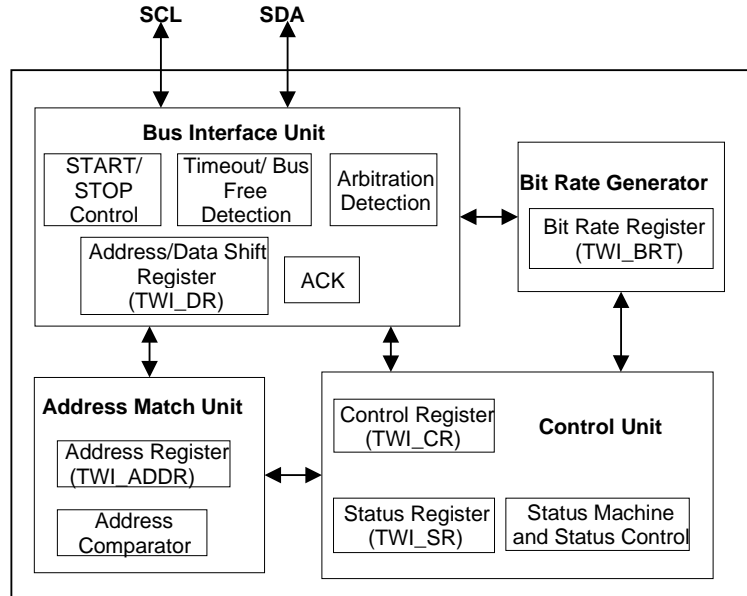


Figure 24-8 TWI Data Arbitration Timing Diagram

Bus interface unit

The bus interface unit includes data and address shift register (TWI_DR), start/stop condition controller, arbitration and bus timeout detection unit.

The register TWI_DR stores the data or address to be transmitted and the data and address received.

The start/stop condition controller is responsible for sending and detecting the start condition, the repeated start condition and the stop condition on the bus.

If SH30F9/SA0 series has started a transfer as master, the arbitration unit will always detect if arbitration has occurred.

When the arbitration is lost, the control unit can perform the appropriate action and generate the corresponding state code.

When transmitting data/address, SH30F9/SA0 series must maintain data stability before SCL jumps from low to high.

When transmitting ACK/NACK, SH30F9/SA0 series generates TWINT interrupt after SCL jumps from low to high, and pulls SCL low when SCL jumps from high to low, releases SCL when TWINT interrupt is cleared.

When SH30F9/SA0 series is transmitting ACK/NACK signal, if TWINT has been cleared and SCL is still at high level, SDA generates transition, and then TWINT interrupt will be generated again, with the state being 00H. The current communication of SH30F9/SA0 series is terminated. This state is consistent with the normal 00H status processing.

When SH30F9/SA0 series is transmitting ACK/NACK signal, if TWINT is not cleared and SCL is still at high level, SDA generates transition, and then the state will be switched directly to 00H, and no interrupt will be generated again. When SH30F9/SA0 series enters this state as a slave, the current communication is terminated, and may generate STA starting the master transfer or re-accept the STA+ADR access to its own address. When SH30F9/SA0 series enters this state as a master, the current communication is terminated, and the generatable STA may start the master transfer or re-accept the STA+ADR access to its own address.

SH30F9/SA0 series will not participate in the current transmitting after the current communication is terminated. If SH30F9/SA0 series exists as a master, enable the EFREE function to prevent from entering the logic dead zone.

SH30F9/SA0 series stipulates that when the bus maintains a high level exceeding the APB1 clock number defined by $TFREE = T_{APB1} * HOC * 1024$ (must ensure that $TFREE$ is greater than $t_{SCL} / 2$ (t_{SCL} is the period of the clock wire)) it is at the "idle" state,



releasing the bus (here the T_{APB1} is actually the TWI module operating clock). This feature is only available for the process of one packet transmitting (8+1 bits). This function is available when SH30F9/SA0 series is in slave transmit mode and the first byte transmitted is at low level. The start condition (STA, RSTA) does not apply to this function.

SH30F9/SA0 series generates an interrupt and the TFREE in the TWI_CR register will be set.

If the clock wire SCL is pulled low by the slave, the communication will be temporarily suspended; there is no way for the master to pull the clock wire high. To solve this problem, the TWI protocol stipulates that when all devices participating in the transfer maintain the clock wire at low level for more than the number of clocks defined by $N * T_{APB1}$ (the value of N is determined by the CNT register), it is "bus timeout". TOUT in the register TWI_FR will be set. Releases the bus by changes of this flag bit.

Frequency generation unit

In master mode, the communication frequency can be set by the register TWI_CR's CR[1:0] division factor and the TWI_BRT[7:0] register.

The SCL frequency is $f_{TWI} / (16 + 2 * CR * TWI_BRT)$.

Note: Here f_{TWI} is the working clock of the TWI module, which is the APB1 bus clock.

Address matching unit

The address matching unit verifies whether the received address matches the 7-bit address in TWIADR. If the corresponding bit in the address mask bit TWIAMR is set to 1, this bit address doesn't detect. If the general address enable bit GC is set, it will also check if it matches the general address 00H. When the addresses match, the control unit will generate the appropriate action and the corresponding state code.

Control unit

The control unit monitors the TWI bus and gives corresponding responds according to the setting of the control register TWI_CR. When the TWI bus has any event that requires attention from application layer, the TWI interrupt flag is set, indicating that the state code of the current event will be written to the state register TWI_STAT. The state register TWI_STAT only indicates the communication state information when the TWI communication interrupt is generated; in other cases, the state register is a state code indicating that there is no valid state code. The clock wire will remain at low level until the interrupt is cleared. The application software allows the TWI communication to continue after the task has been processed.

24.3.5 TWI Operation Modes

In specific application, the TWI module can be used as a master, a slave, or both master and slave. In slave mode, the TWI hardware looks up the slave address as well as the general address ($GC = 1$). If one of the addresses is detected, interrupt request is generated. If the processor wants to become the bus master, the hardware must wait until the bus is idle before entering master mode, so that slave operation is not aborted. If bus arbitration is lost in master mode, the TWI module enters a wait state (pure master), or immediately switches to slave mode and detects its own slave address (master slave) in the same serial transfer.

The four basic operating modes of TWI:

- Master transmit mode
- Master receive mode
- Slave transmit mode
- Slave receive mode

Detailed description about those four modes can be found below.

The TWI communication of SH30F9/SA0 series is performed by the underlying driver circuit and the interrupt-based application software. All bus events, such as receiving a byte or sending a start condition, each generate an interrupt. So during byte transfer, the application software can perform other operations. It should be noted that in addition to the basic TWI communication interrupt (TWINT), it is sometimes necessary to turn on the bus timeout (TOUT) and the SCL high level timeout (TFREE) detection. In both cases, interrupt is also generated, and the three interrupt sources share the same interrupt entry, see the "TWI Interrupt" part for details.

When the TWINT bit is set, it indicates that a TWI transmitting has been completed, waiting for the response of the application software, at that time the state register TWI_STAT contains the current state. The application software can determine which communication the TWI performs via the registers TWI_CR and TWI_STAT.



SH30F9/SA0 Series

The four main modes of TWI communication are described below, and all possible state codes are described. The following figures shall have the following abbreviations:

S : start condition
Rs : repeated start condition
R : read control bit
W : write control bit
A : acknowledge bit
Ā : no acknowledge bit
DATA : 8-bit data
P : stop condition
SLA : slave address

The circle in the figure is used to indicate that the interrupt flag has been set. The number in it indicates the state code in the current state register TWI_STAT.

Before TWINT is cleared, the TWI communication will be suspended and the application software must decide whether to continue the communication or terminate the current transfer. For each state code, the required software actions and subsequent transfer details are described.

● Master transmit mode

In the master transmit mode, the master sends a series of data to the slave. To enter the master transmit mode, a start condition followed by a slave address + write control word (SLA+W) address packet indicates entering the master transmit mode (MT).

STO and TWINT are cleared via setting TWIEN and STA in the control register TWI_CR, the TWI logic will detect the TWI bus and issue a START condition (STA) when enabled. After the start condition (STA) is transferred, the communication interrupt (TWINT) is set, the state register (TWI_STAT) is 08H, and the interrupt service routine should write the slave address and the write control word (SLA+W) to the data register TWI_DR. Clear the TWINT flag before initiating the next transfer.

After the slave address and the write control word are transferred and an "acknowledgement" message is received, the interrupt (TWINT) is set and there are several possible states in the state register TWI_STAT: 18H, 20H and 38H for the master mode, and 68H, 78H and B0H for the slave mode.



SH30F9/SA0 Series

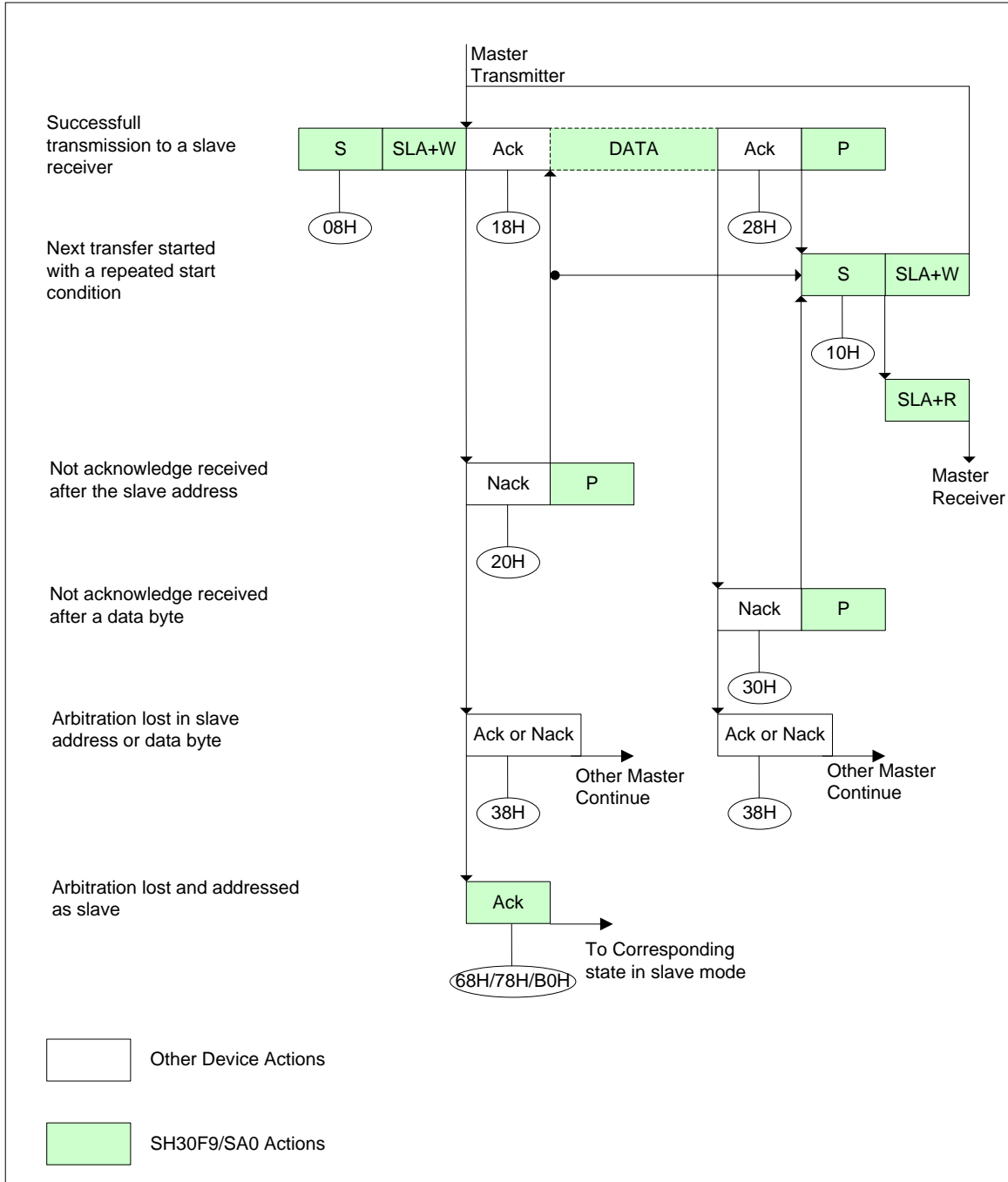


Figure 24-9 Master Transmit Mode State Diagram

Note: SH30F9/SA0 series is set as the master in the picture above



SH30F9/SA0 Series

Table 24-1 Status codes for master transmitter mode

| Status | TWI bus and serial interface hardware status | Application Software Response | | | | | The Next Action Performed by TWI |
|--------|--|---------------------------------------|-----------------------|-----|-------|----|---|
| | | R/W Data Register TWI_DR Operation | Control Bit Operation | | | | |
| | | | STA | STO | TWINT | AA | |
| 08H | A START condition has been transmitted | Load SLA+W | X | 0 | 0 | X | Transmit SLA+W, receive ACK or NACK |
| 10H | A repeated START condition has been transmitted | Load SLA +W | X | 0 | 0 | X | Transmit SLA+W, receive ACK or NACK |
| | | Load SLA +R | X | 0 | 0 | X | Transmit SLA+R, TWI will be switched to master receive mode |
| 18H | SLA+W has been transmitted; ACK has been received | Load data byte | 0 | 0 | 0 | X | Transmit data, receive ACK or NACK |
| | | No TWI_DR action | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |
| 20H | SLA+W has been transmitted; NACK has been received | Load data byte | 0 | 0 | 0 | X | Transmit data, receives ACK or NACK |
| | | No TWI_DR action | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |
| 28H | TWI_DR data has been transmitted; ACK has been received | Load data byte | 0 | 0 | 0 | X | Transmit data, receive ACK or NACK |
| | | No TWI_DR action | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |
| 30H | TWI_DR data has been transmitted; NACK has been received | Load data byte | 0 | 0 | 0 | X | Transmit data, receive ACK or NACK |
| | | No TWI_DR action | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |
| 38H | Arbitration lost in SLA+W or data bytes | No TWI_DR action | 0 | 0 | 0 | X | TWI bus is released; enter non-addressing slave mode |
| | | | 1 | 0 | 0 | X | Transmit start condition when the bus is idle |



● Master Receive Mode

In the master receive mode, the master receives a series of data from the slave. To enter the master receive mode, a start condition followed by a slave address + read control word (SLA+R) address packet indicates entering the master receive mode (MR).

STO and TWINT are cleared via setting TWIEN and STA in the control register TWI_CR, the TWI logic will detect the TWI bus and issue a start condition (STA) when enabled. After the start condition (STA) is transferred, the communication interrupt (TWINT) is set, and the state register (TWI_STAT) is 08H. The interrupt service routine should write the slave address and the read control word (SLA+R) to the data register TWI_DR. Clear the TWINT flag before initiating the next transfer.

After the slave address and the write control word are transferred and an "acknowledgement" message is received, the interrupt (TWINT) is set and there are several possible states in the state register TWI_STAT: 40H, 48H and 38H for the master mode, and 68H, 78H and B0H for the slave mode.

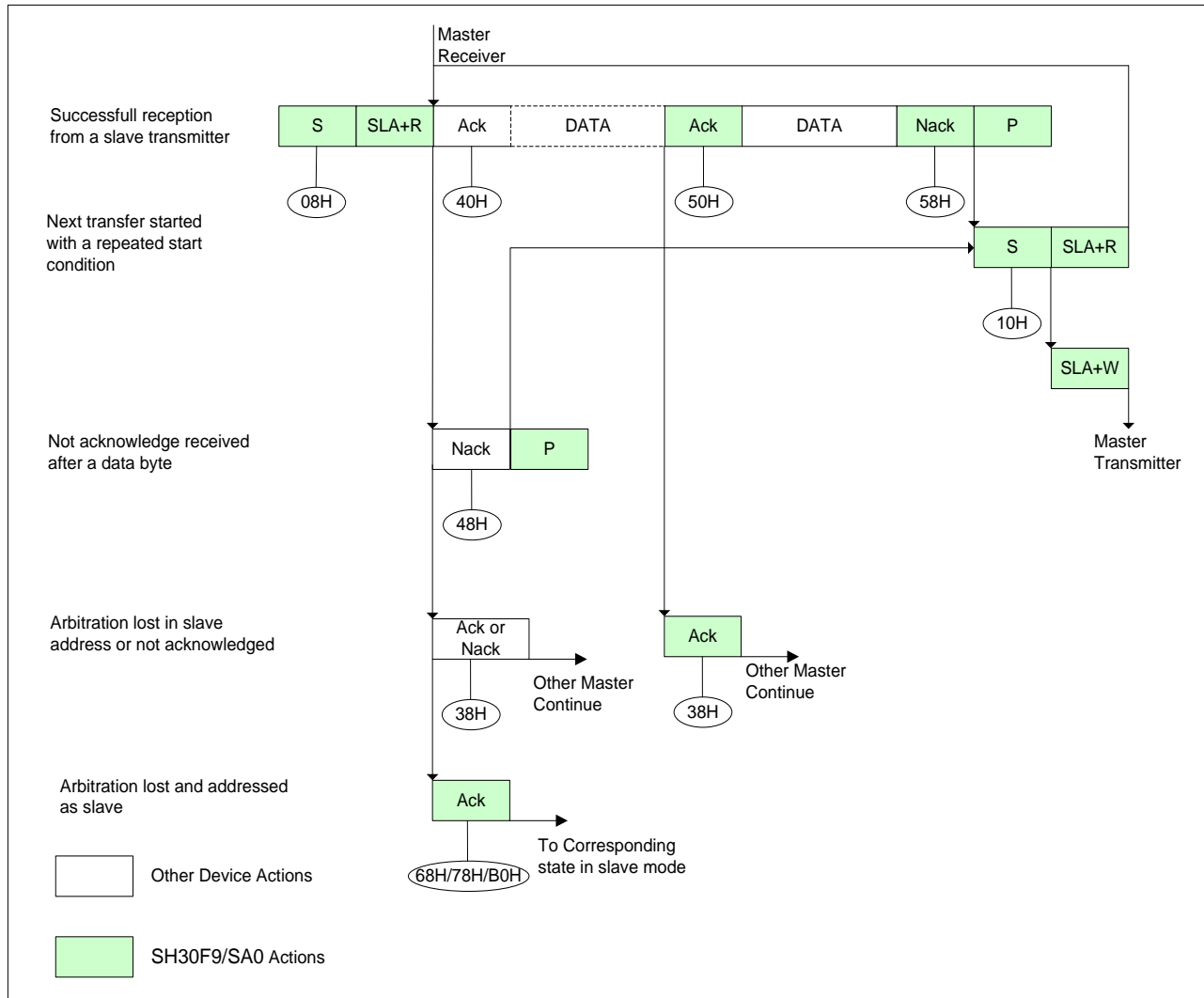


Figure 24-10 Master Receive Mode State Diagram

Note: SH30F9/SA0 series is set as the master in the picture above



SH30F9/SA0 Series

Table 24-2 Master Receive Mode State Code

| State Code | TWI Bus and Hardware Interface State | Application Software Response | | | | The Next Action Performed by TWI | |
|------------|--|---|-----------------------|-----|-------|----------------------------------|---|
| | | Read/write Data Register TWI_DR Operation | Control Bit Operation | | | | |
| | | | STA | STO | TWINT | | AA |
| 08H | A START condition has been transmitted | Load SLA+R | X | 0 | 0 | X | Transmit SLA+R, receive ACK or NACK |
| 10H | A repeated START condition has been transmitted | Load SLA+R | X | 0 | 0 | X | Transmit SLA+R, receive ACK or NACK |
| | | Load SLA+W | X | 0 | 0 | X | Transmit SLA+W, TWI will be switched to master transmit mode |
| 38H | SLA+R has been transmitted or arbitration lost in NACK | No TWI_DR action | 0 | 0 | 0 | X | TWI bus is released; enter non-addressing slave mode |
| | | | 1 | 0 | 0 | X | Transmit start condition when the bus is idle |
| 40H | SLA+R has been transmitted; ACK has been received | No TWI_DR action | 0 | 0 | 0 | 0 | Receive data, return NACK |
| | | | 0 | 0 | 0 | 1 | Receive data, return ACK |
| 48H | SLA+R has been transmitted; NACK has been received | No TWI_DR action | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |
| 50H | Data byte has been received; ACK has been returned | Read data byte | 0 | 0 | 0 | 0 | Receive data, return NACK |
| | | | 0 | 0 | 0 | 1 | Receive data, return ACK |
| 58H | Data byte has been received; NACK has been returned | Read data byte | 1 | 0 | 0 | X | Transmit repeated start condition |
| | | | 0 | 1 | 0 | X | Transmit stop condition; clear STO flag |
| | | | 1 | 1 | 0 | X | Transmit stop condition, after which start condition is transmitted; STO is cleared |



● Slave transmit mode

In the slave transmit mode, the slave sends a series of data to the master. To initialize the slave transmit mode, the control register TWI_CR and the address register TWI_ADDR must be initialized: TWIEN and AA in the control register TWI_CR are set, STA, STO, and TWINT are cleared; the address bit TWIADR prepares the corresponding address for SH30F9/SA0 series. If GC is set, SH30F9/SA0 series will also respond to the general address (00H); otherwise it will not respond to the general address.

After the TWIADR and TWI_CR are initialized, SH30F9/SA0 series will wait for the bus to respond to its own address or general address (if the GC is set). If the direction flag is "read", the TWI enters the slave transmit mode, otherwise it will enter the slave receive mode. After the address and read flag bits have been received, the interrupt flag (TWINT) is set and the state register TWI_STAT is valid.

During transmitting, if the acknowledge enable bit "AA" is cleared, TWI will transmit the last byte and enter the C0H or C8H state depending on the acknowledge or no acknowledge message bit sent by the master receiver. The bus will switch to non-addressing slave mode and not respond to master transfer any more. Thus the master receiver will receive a string of "1"s. After the last byte has been transmitted, if the master still needs additional data (transmitting the "acknowledge" signal), it enters the C8H state.

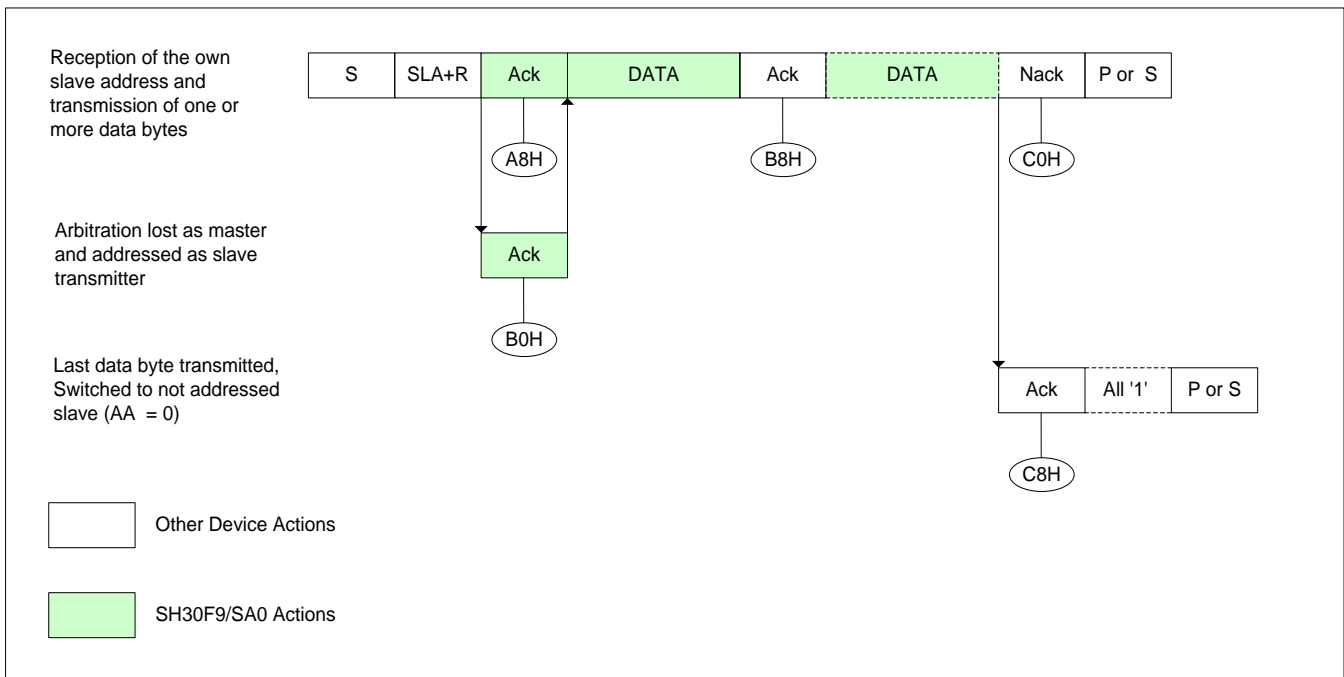


Figure 24-11 Slave Transmit Mode State Diagram

Note: SH30F9/SA0 series is set as the slave in the picture above



SH30F9/SA0 Series

Table 24-3 Slave Transmit Mode State Code

| State Code | TWI BUS AND HARDWARE INTERFACE STATE | Application Software Response | | | | THE NEXT ACTION PERFORMED BY TWI | |
|------------|---|---|-----------------------|-----|-------|----------------------------------|---|
| | | Read/write Data Register TWI_DR Operation | Control Bit Operation | | | | |
| | | | STA | STO | TWINT | | AA |
| A8H | Own SLA+R has been received; ACK has been returned | Load data byte | X | 0 | 0 | 0 | Transmit final data; wait for response from ACK or NACK |
| | | | X | 0 | 0 | 1 | Transmit data; wait for response from ACK or NACK |
| B0H | Arbitration lost in SLA+R/W as Master; own SLA+R has been received; ACK has been returned | Load data byte | X | 0 | 0 | 0 | Transmit final data; wait for response from ACK or NACK |
| | | | X | 0 | 0 | 1 | Transmit data; wait for response from ACK or NACK |
| B8H | Data byte in TWIDAT has been transmitted; ACK has been returned | Load data byte | X | 0 | 0 | 0 | Transmit final data; wait for response from ACK or NACK |
| | | | X | 0 | 0 | 1 | Transmit data; wait for response from ACK or NACK |
| C0H | Data byte in TWI_DR has been transmitted; NACK has been returned | No TWI_DR action | 0 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address |
| | | | 0 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR |
| | | | 1 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address; send start condition when the bus is idle |
| | | | 1 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR; send "start condition" when the bus is idle |
| C8H | Last data byte in TWI_DR register has been transmitted (AA=0); ACK has been returned; | No TWI_DR action | 0 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address |
| | | | 0 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR |
| | | | 1 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address; send "start condition" when the bus is idle |
| | | | 1 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR; send "start condition" when the bus is idle |



SH30F9/SA0 Series

● Slave receive mode

In the slave receive mode, the slave receives a series of data from the master. To initialize the slave receive mode, the control register TWI_CR and the address register TWI_ADDR must be initialized: TWIEN and AA in the control register TWI_CR are set, STA, STO, and TWINT are cleared; the upper 7-bit of the address register TWIADR prepares the corresponding address for SH30F9/SA0 series. If GC is set, SH30F9/SA0 series will also respond to the general address (00H); otherwise it will not respond to the general address.

After TWI_ADDR and TWI_CR are initialized, SH30F9/SA0 series will wait for the bus to respond to its own address or general address (if GC is set). If the direction flag is 'write', TWI enters the slave receive mode, otherwise it will enter the slave transmit mode. After the address and write flag bits have been received, the interrupt flag (TWINT) is set and the state register TWI_STAT is valid.

During transmitting, if the acknowledge enable bit "AA" is cleared, the TWI will receive the last byte and respond with a "no acknowledge" message. Responding with "no acknowledge" indicates that the current slave cannot receive more bytes. When AA=0, SH30F9/SA0 series cannot respond to access to its own address; however, it still monitors the bus state and can restore the response to the corresponding address to itself via AA=1. SH30F9/SA0 series can be temporarily isolated from the bus via AA=0.

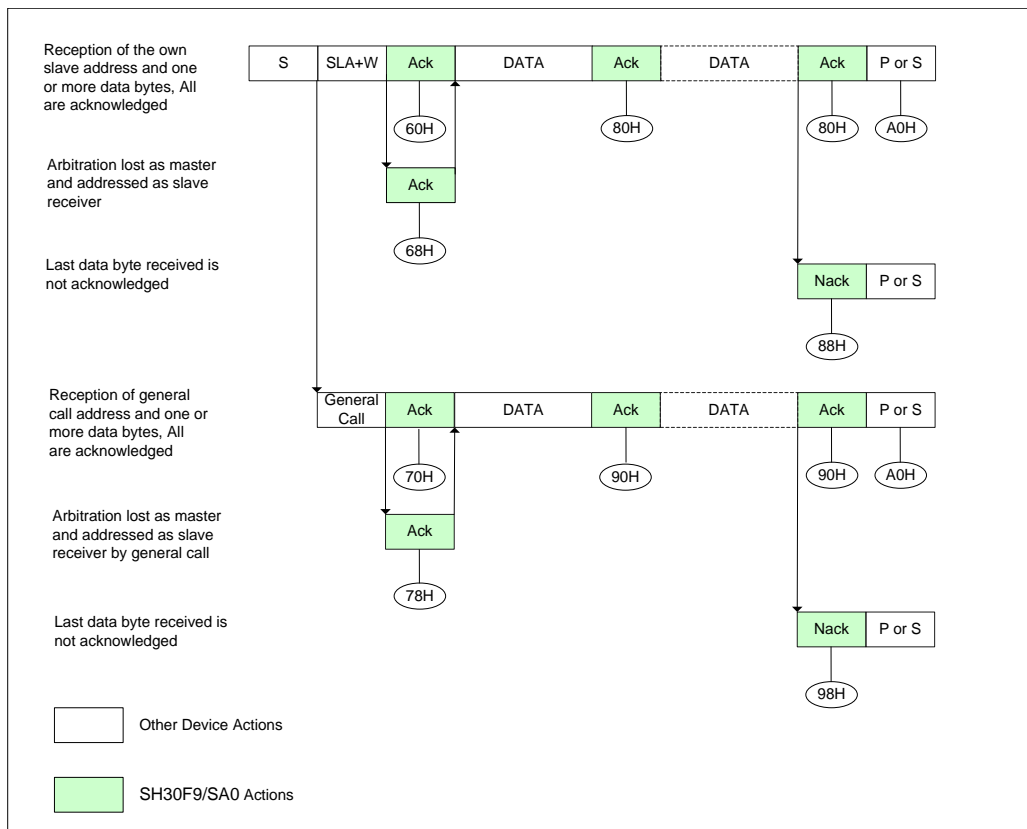


Figure 24-12 Slave Receive Mode Diagram

Note:SH30F9/SA0 series is set as the slave in the picture above



SH30F9/SA0 Series

Table 24-4 Slave Receive State Code

| State Code | TWI BUS AND HARDWARE INTERFACE STATE | Application Software Response | | | | THE NEXT ACTION PERFORMED BY TWI | |
|------------|---|---|-----------------------|-----|-------|----------------------------------|---|
| | | Read/write Data Register TWI_DR Operation | Control Bit Operation | | | | |
| | | | STA | STO | TWINT | | AA |
| 60H | Own SLA+W has been received; ACK has been returned | No TWI_DR action | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 68H | Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned | No TWI_DR action | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 70H | GCA has been received; ACK has been returned | No TWI_DR action | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 78H | Arbitration lost in SLA+R/W as Master; own GCA has been received; ACK has been returned | No TWI_DR action | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 80H | Previously addressed with own SLA+W; Data has been received; ACK has been returned | Reads data | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 88H | Previously addressed with own SLA+W; data has been received; NOT ACK has been returned | Reads data | 0 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address |
| | | | 0 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR |
| | | | 1 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address; send "start condition" when the bus is idle |
| | | | 1 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR; send "start condition" when the bus is idle |



SH30F9/SA0 Series

| | | | | | | | |
|-----|---|------------------|---|---|---|---|---|
| 90H | Previously addressed with general call; data has been received; ACK has been returned | Reads data | X | 0 | 0 | 0 | Receive data; transmit NACK response |
| | | | X | 0 | 0 | 1 | Receive data; transmit ACK response |
| 98H | Previously addressed with general call; data has been received; NOT ACK has been returned | Reads data | 0 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address |
| | | | 0 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depend on the setting of GC in register TWI_ADDR |
| | | | 1 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address; send "start condition" when the bus is idle |
| | | | 1 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR; send "start condition" when the bus is idle |
| A0H | A STOP condition or repeated START condition has been received while still addressed as Slave | No TWI_DR action | 0 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address |
| | | | 0 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR |
| | | | 1 | 0 | 0 | 0 | Switch to non-addressing slave mode; do not respond to its own address and general address; send "start condition" when the bus is idle |
| | | | 1 | 0 | 0 | 1 | Switch to non-addressing slave mode; respond to its own address, and whether to respond to general address depends on the setting of GC in register TWI_ADDR; send "start condition" when the bus is idle |

- **Other modes**

In addition to the above state codes, there are two state codes without explicit TWI state. State 0F8H indicates that there is no corresponding state information due to the interrupt flag TWINT not being set. When the interrupt TWINT is not set, it is filled by 0F8H after clearing a state and before a new state is established.



SH30F9/SA0 Series

State 00H indicates that error has occurred in the TWI bus communication, which means illegal start condition or stop condition occurs in the transfer. For example, a start or stop condition occurs when transferring an address, data or responding to an ACK response. The 00H state is also generated when the bus disturbs the internal logic. When illegal state occurs, the interrupt flag bit TWINT is set. The normal communication can be resumed by setting STO and clearing the TWINT flag. SH30F9/SA0 series will enter the non-addressing slave mode and automatically clear the STO flag. The data wire and clock wire will be released and there will be no stop condition transmitted on the wire.

Table 24-5 Other Modes State Code

| State Code | TWI BUS AND HARDWARE INTERFACE STATE | Application Software Response | | | | THE NEXT ACTION PERFORMED BY TWI | |
|------------|---|---|-----------------------|-----|-------|----------------------------------|---|
| | | Read/write Data Register TWI_DR Operation | Control Bit Operation | | | | |
| | | | STA | STO | TWINT | | AA |
| F8H | No valid state code; TWINT=0 | No TWI_DR action | No TWI_DR action | | | | Wait or deal with current transfer |
| 00H | Bus error due to an illegal START or STOP condition | No TWI_DR action | 0 | 1 | 0 | X | Only the internal hardware is affected, the bus will be released, and switches to the not addressed Slave mode. STO will be cleared. |

24.3.6 Supporting Fast Mode 400kbps

The TWI module is compatible with two common transfer rates: standard mode 100kHz, fast mode 400kHz. The two transfer rates are obtained by the application software configuring the division factor CR[1:0] and the bit rate configure bit BRT[7:0].

24.3.7 Interrupt

Bus timeout flag TOUT, SCL high level timeout flag TFREE, TWI communication interrupt flag TWINT, any one of those three flags will cause TWI interrupt, the three share the interrupt vector, and the flags must be cleared by software.

TOUT: Bus timeout flag. This bit is set when the TWI bus low level exceeds $N \cdot T_{APB1}$. The value of N is set by the CNT[1:0]@TWI_CR.

TFREE: SCL high level timeout flag. When participating in the bus transmission, the hardware is set when the SCL high level exceeds the number of system clocks defined by $TFREE = T_{APB1} \cdot HOC \cdot 1024$. HOC value is determined by HOC [7: 0]@TWI_HOC register bits.

TWINT: TWI basic function interrupt flag. TWINT is set when the TWI state is changed. However, entering state F8 does not set TWINT because the interrupt service routine does not work in that case.

When TWINT is set, the serial clock on the SCL wire expands with low level periods and the serial transfer is aborted. When SCL is high, the state of the TWINT flag is not affected.

TWINTEN: TWI interrupt enable bit, controls whether the TWI interrupt source generates an interrupt request. If the interrupt enable bit is turned off, the application software can only process TWI communication status by querying.

When entering TWI interrupt, the TWI status register TWI_STAT stores the status code, which can identify one of the multiple status services to be executed (typically 26 states for I2C).

24.3.8 TWI Communication Service Program Development

The TWI module SH30F9/SA0 series needs to be compatible with the user application to achieve a complete TWI communication protocol (communication protocols are different depending on the applications), ensuring maximum application flexibility. The user application is driven via the TWI interrupt without affecting the main flow and other controls. For details, please refer to the state diagram and state code of the four main modes for development. For the most basic cases of operations on the TWI/I2C interface peripherals of MCU, only the "Master Transfer Mode" needs to be referred for development.



SH30F9/SA0 Series

The operations which a complete TWI server program must perform include:

The TWI module is initialized after reset, including configuring the slave address and general address bit (GC), enabling interrupts, setting the serial clock frequency, and so on.

The TWI interrupt service, state codes are identified and corresponding state service programs are entered according to branches. Timeout conditions also need to be handled to limit invalid bus or lost service program.

Multiple state service programs supporting 4 TWI operation modes (26 states for I2C). If one or more modes are not used, the corresponding state services can be ignored (as long as they are handled carefully, those states will not appear).



24.4 TWI Registers

TWI Module Register list (Base Address:0x4002 1800)

| Address | Register | Description |
|-------------|----------|---|
| 0x4002 1800 | FR | TWI Interrupt Flag Register |
| 0x4002 1804 | STAT | TWI State Register |
| 0x4002 1808 | HOC | TWI High Level Timeout Detection Configure Register |
| 0x4002 180C | BRT | TWI Bit Rate Configure Register |
| 0x4002 1810 | DR | TWI data register |
| 0x4002 1814 | ADDR | TWI Address Configure Register |
| 0x4002 1818 | CR | TWI Control Register |

24.4.1 TWI Interrupt Flag Register (TWI_FR)

Offset Address: 0x0000

Reset value: 0x0000 0000

| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|------------|--------------|------------|
| Reserved | | | | | | | | | | | | TOU TC | TFR EEC | Rese rved | TWI NTC |
| - | | | | | | | | | | | | WO | WO | - | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----------|-----------|--------------|-----------|
| Reserved | | | | | | | | | | | | TOU T | TFR EE | Rese rved | TWI NT |
| - | | | | | | | | | | | | RO | RO | - | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 20 | Reserved | - |
| 19 | TOUTC | Bus timeout flag clear bit 0: Invalid 1: Clear |
| 18 | TFREEC | SCL high level timeout flag clear bit 0: Invalid 1: Clear |
| 17 | Reserved | - |
| 16 | TWINTC | TWI one byte data transfer interrupt flag clear bit 0: Invalid 1: Clear |
| 15 ~ 4 | Reserved | - |
| 3 | TOUT | Bus timeout flag bit 0: No timeout flag occurs 1: Position when the TWI bus low level exceeds $N \cdot T_{APB1}$, and the status machine jumps to the 0F8H state. The bit must be cleared by the software. When the TOUT flag interrupt is allowed, this position '1' will cause the CPU to turn to the TWI interrupt service program. The N value is determined by the CNT register |



| | | |
|---|-----------------|---|
| 2 | TFREE | SCL high level timeout flag bit 0: No timeout flag occurs 1: When participating in bus transmission, such as when the clock line high level exceeds the number of APB1 bus clocks defined by $TFREE = T_{APB1} * HOC * 1024$, the state machine jumps to the 0F8H state. The bit must be cleared by the software. When the TFREE flag interrupt is allowed, this position '1' will cause the CPU to turn to the TWI interrupt service program. Note: TFREE must be guaranteed to be larger than the bus's $t_{SCL}/2$ (t_{SCL} is the cycle of the clock line) |
| 1 | Reserved | - |
| 0 | TWINT | TWI one byte data transfer interrupt flag 0: During single-byte data transfer, write 1 to clear by software 1: Single-byte data transfer is completed, set by hardware |

24.4.2 TWI State Register (TWI_STAT)

Offset Address: 0x0004

Reset value: 0x0000 00F8

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|---------|-----|-----|-----|----------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | SR[4:0] | | | | Reserved | | | |
| - | | | | | | | | RO | | | | - | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|------------------------------------|
| 31 ~ 8 | Reserved | - |
| 7 ~ 3 | SR[4:0] | TWI serial communication state bit |
| 2 ~ 0 | Reserved | - |

24.4.3 TWI High Level Timeout Detection Configure Register (TWI_HOC)

Offset Address: 0x0008

Reset value: 0x0000 00FF

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | HOC[7:0] | | | | | | | |
| - | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Description |
|--------|----------|--|
| 31 ~ 8 | Reserved | - |
| 7 ~ 0 | HOC[7:0] | TWI SCL bus high level timeout detection configure bit |



SH30F9/SA0 Series

24.4.4 TWI Bit Rate Configure Register (TWI_BRT)

Offset Address: 0x000C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|----------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | BRT[7:0] | | | | | | | |
| - | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|----------------------------|
| 31 ~ 8 | Reserved | - |
| 7 ~ 0 | BRT[7:0] | TWI bit rate configure bit |

24.4.5 TWI data register (TWI_DR)

Offset Address: 0x0010

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|---------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | DR[7:0] | | | | | | | |
| - | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|----------|-------------------|
| 31 ~ 8 | Reserved | - |
| 7 ~ 0 | DR[7:0] | TWI data register |

24.4.6 TWI Address Configure Register (TWI_ADDR)

Offset Address: 0x0014

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|--------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | TWIAMR[6:0] | | | | | | | Rese rved |
| - | | | | | | | | RW | | | | | | | - |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|----|----|-----------|----|----|----|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | ADDR[6:0] | | | | | | | GC |
| - | | | | | | | | RW | | | | | | | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|-------------|
| 31 ~ 24 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|--|
| 23 ~ 17 | TWIAMR[6:0] | TWI address mask bit Each bit of the TWIAMR can mask the corresponding address bit in the TWIADR address register. If the mask bit is set to 1, the address matching logic ignores the compare result of the input address bits with the corresponding address |
| 16 ~ 8 | Reserved | - |
| 7 ~ 1 | ADDR[6:0] | TWI address configure bit |
| 0 | GC | General address enable bit 0: responding to general address is disabled 1: responding to general address is enabled |

24.4.7 TWI Control Register (TWI_CR)

Offset Address: 0x0018

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|---------------|-----------------|-----|-----|-----|----------------|-------------|--------------|-----------------|----------------|-----------------|------------|------------|-----------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TWI EN | Reserved | | | | TWI NTE | ETOT | EFREE | CNT[1:0] | CR[1:0] | Reserved | STA | STO | AA | | |
| <i>RW</i> | - | | | | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | <i>RW</i> | - | <i>RW</i> | <i>RW</i> | <i>RW</i> | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-----------------|---|
| 31 ~ 16 | Reserved | - |
| 15 | TWIEN | TWI enable bit 0: TWI is disabled 1: TWI is enabled |
| 14 ~ 11 | Reserved | - |
| 10 | TWINTEN | TWI transmit receive interrupt flag enable bit 0: TWI interrupt flag bit TWINT is disabled to trigger interrupt 1: TWI interrupt flag bit TWINT is enabled to trigger interrupt |
| 9 | ETOT | Bus timeout flag interrupts enable bits 0: Do not allow TOUT to trigger TWI interrupts, but bus timeout judgment is always valid 1: Allow TOUT to trigger a TWI interrupt when the TOUT bit is set to '1' |
| 8 | EFREE | The SCL high level timeout flag interrupts the enable bit 0: TFREE is prohibited from triggering TWI interrupts, but the SCL bus high timeout judgment is always valid 1: Allow TFREE to trigger TWI interrupts when the TFREE bit is placed '1' |
| 7 ~ 6 | CNT[1:0] | Bus timeout count control bit 00: N=25000 01: N=50000 10: N=100000 11: N=200000 Value N is used in bus timeout function |
| 5 ~ 4 | CR[1:0] | TWI division coefficient 00: 64 01: 16 10: 4 11: 1 |
| 3 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---|------------|--|
| 2 | STA | Start bit 0: Does not send start condition 1: Send start condition when the bus is idle Note: When STA=1, the start condition is sent and the main mode is entered. If it is already in the master mode, the repeated start condition is sent. When both STA and STO are set, if in the master mode, a stop condition is sent to the bus and then a start condition is sent. If the interface is in the slave mode, an internal stop condition is generated but not sent to the bus. |
| 1 | STO | Stop bit 0: Does not send stop condition 1: Send stop condition as a master; does not send stop condition to the bus as a slave, but the state is restored to the non-addressing slave state. The hardware will automatically clear this flag Note: In the master mode, when STO is 1, a stop condition is sent or it is recovered from the error state in the slave mode. When STO=1 in the master mode, stop condition is sent to the bus. When the bus detects a stop condition, STO is automatically cleared. In the slave mode, the STO bit is set to recover from the error state. In this case, no stop condition is sent to the bus. The hardware behaves as if it received a stop condition and switched to the non-addressing slave receive mode. The STO flag is automatically cleared by hardware. |
| 0 | AA | Acknowledge flag bit 0: Respond with “no acknowledge” signal (SDA high level) 1: Respond with “acknowledge” signal (SDA low level) |



25. Analog to Digital Converter

25.1 Introduction

SH30F9/SA0 series contains a single-ended 12-bit high-speed successive approximation analog-to-digital converter (ADC, Analog-to-Digit Converter), the block diagram is shown in Figure 25-1. The ADC reference voltage uses VDD by default after the chip is reset. The user can also select the input of the external VREF port or the built-in VREF as the reference voltage.

The ADC has 28 analog input channels (ADC.AN0-ADC.AN26, V_{CC}). The ADC can be programmed into the sequence for automatic conversion, and the result is stored in the corresponding result register ADDR_x (x = 0-7). Each time the sequence is converted, the value of the result register is updated once and a comparison is made. The mapping relationship between the result register and the analog input channel can be programmed at will to form a conversion sequence, and a certain analog input channel can be re-programmed in the sequence, so that the results of continuous multiple conversions of this analog input channel can be obtained in the result register.

When the continuous conversion function is activated, the sequence automatically starts the conversion cycle, and the result register is also continuously updated, and each updated value is compared with the limit value.

For a single channel, the conversion rate can reach up to 1MSPS, and the ADC clock rate and sampling time can be set by the register TS[3:0]@ADCON2. The time interval between adjacent channels in the sequence can also be set by register TGAP[2:0]@ADCON2.

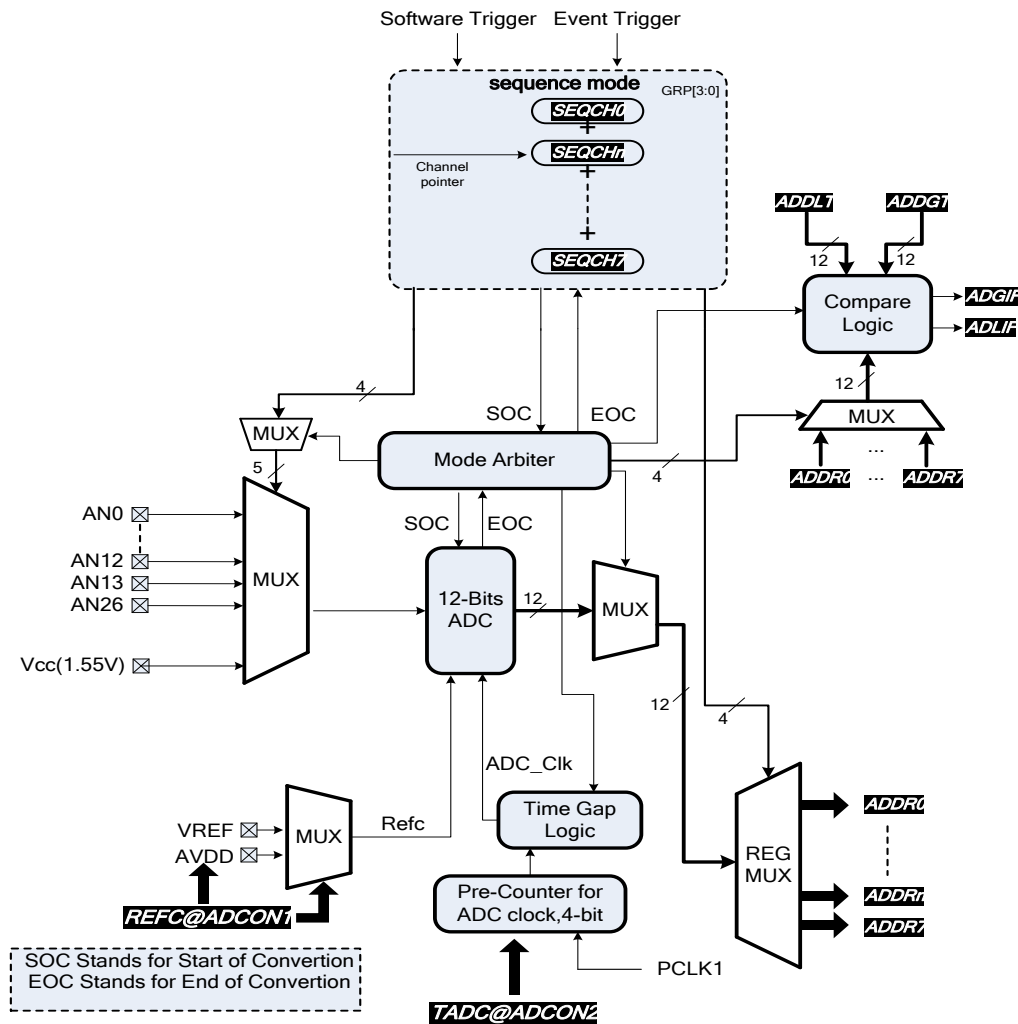


Figure 25-1 ADC System Block Diagram



25.2 Main Features

- SH30F9/SA0 series built-in a 12-bit A/D converter with successive approximation method
- Reference voltage can be built-in VREF, external VREF or VDD
- 28 ADC analog input channels
- Start the ADC once to automatically complete the multi-channel conversion (sequence), and each channel can be configured as any one of the multi-channel analog input
- The sequence can be configured as a single channel or multiple channels, a sequence can contain up to 8 channels, and the conversion results are stored in 8 result registers
- There are three conversion modes: sequence conversion mode, intermittent conversion mode and continuous conversion mode
- The time interval between adjacent channel conversions during sequence conversion can be set by software
- Four PWM modules (PWM0/1/2/3), four PCA timers (PCA0/1/2/3), a common timer (TIM2) and external interrupt (EXTI0/1/2/3) The trigger signal automatically starts AD conversion
- Three conversion methods are equipped with comparison function
- Support DMA function

25.3 Function Description

25.3.1 Single Conversion Method

When ADCTU[1:0] = 00, the ADC is configured to single-sequence conversion method, converting one sequence at a time. A sequence consists of a single channel or multiple channels. Converting a sequence means converting the channels in the sequence one by one. In hardware, it can be achieved by making multiple signals convert at the same time point (the shortest sampling interval between 2 channels is 0.5us, which can be approximated as simultaneous).

The result of the conversion is stored in the corresponding result registers. At the same time as the sequence conversion is completed, the value of a specified result register is compared with the values of ADDGT and ADDLT, and the result of the comparison is indicated by a flag.

In the single sequence conversion mode, after ADSOC is set to 1, conversion starts from channel 0. After each single-channel conversion is completed, the ADPCH register is automatically incremented by one, and then the conversion of the next channel is performed. When the number of single-sequence conversion channels reaches the setting value (ADMAXCH[2:0]), the current sequence conversion ends and the ADSOC bit is cleared, indicating that the sequence conversion is completed and the ADPCH register is automatically reset to 0. At the same time, the ADIF bit in ADINTF is set by hardware. At this time, if the ADCIE bit in ADCON1 is '1' (Note: the ADC interrupt is not masked), the sequence conversion completion interrupt will be triggered and the ADIF flag needs to be cleared by software.

Example: There are 4 channels that need to be converted. They are AN0, AN2, AN5, AN7 in order of priority, then set ADMAXCH[2:0] = 3, SEQCH0 = 0, SEQCH1 = 2, SEQCH2 = 5, SEQCH3 = 7, Then every time AD conversion is started (ADSOC=1), the 4 channels starting from channel 0 will be converted sequentially, and the results will be stored in ADDR0, ADDR1, ADDR2, and ADDR3 in sequence.

CASE1: ADMAXCH=3, SEQCH0=0, SEQCH1=2, SEQCH2=5, SEQCH3=7, ..., SEQCH7=3

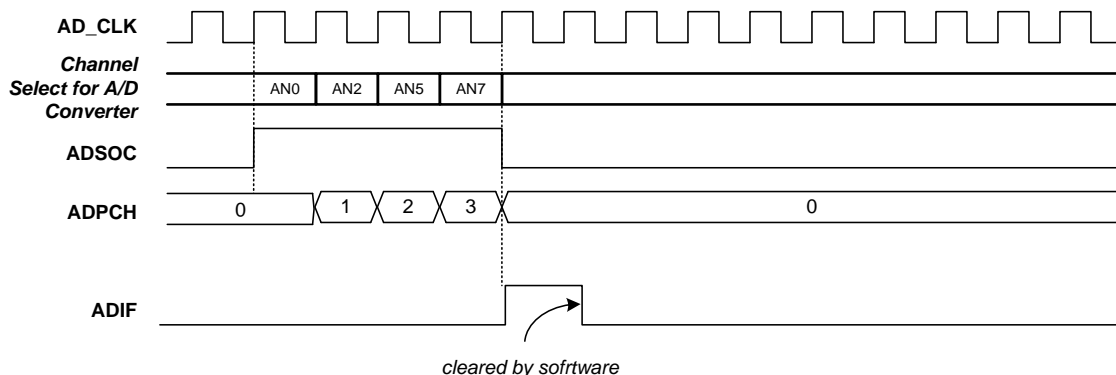


Figure 25-2 Single Conversion Method Waveform Example Diagram



Note: In the single conversion mode, ADSOC is set to 1 to start the sequence conversion, and remains as 1 during the conversion process, and the hardware is automatically cleared to 0 after the conversion is completed (If this bit is cleared during the conversion period, the sequence conversion will be terminated immediately.)

25.3.2 Intermittent Conversion Method

When ADCTU[1:0] = 01, the ADC operates in the intermittent conversion method. The ADC directly converts from the channel specified by the ADPCH register. When the number of conversions reaches the (ADMAXCH[2:0]) setting, this sequence conversion ends. The maximum ADC sampling channel number is 7, so when the conversion channel pointer register ADPCH exceeds 7, it will automatically return to 0. After a sequence conversion is completed, if the ADPCH software is rewritten, the next conversion channel starts from n at this time, the next conversion starts from channel n. If the ADPCH is not rewritten, the next conversion channel starts from the ADPCH value at the end of the last sampling. The ending of hardware sampling ends will not clear the value of ADPCH, so the program rewriting of the ADPCH register needs to be treated carefully (during ADC conversion period (ADSOC = 1), ADPCH cannot be modified).

For example, SEQCH0=2, SEQCH1=8, SEQCH2=7, SEQCH3=9, SEQCH4=5, SEQCH5=6, SEQCH6=1, SEQCH7=0, when ADMAXCH=3, if ADPCH=6 at the end of the previous conversion, In the discontinuous conversion mode, SEQCH6, SEQCH7, SEQCH0, and SEQCH1 are valid channel selection registers. The resulting channel sequence is AN1, AN0, AN2 and AN8, and the sampling results will be stored in ADDR6, ADDR7, ADDR0, ADDR1 in sequence. After the conversion, the ADPCH value should be 2. When the AD conversion is started again, AN7, AN9, AN5, and AN6 will be converted in sequence, and the results will be stored in ADDR2, ADDR3, ADDR4, and ADDR5 in sequence. After the conversion, the ADPCH value should be 6. And so on. In this example, the timing relationship of each key signal is shown in Figure 25-3:

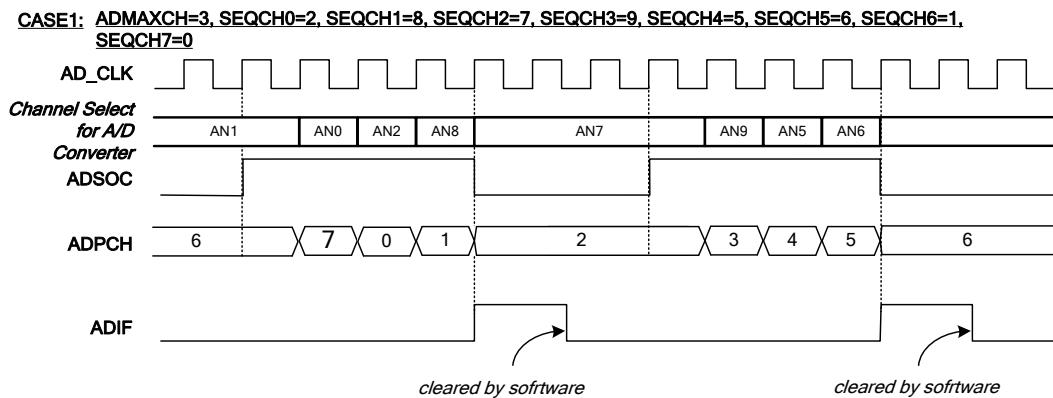


Figure 25-3 Intermittent Conversion Method Waveform Example Diagram

Note 1: ADPCH can only be written by software when the ADC is not sampling (ADSOC is equal to 0), and can only be written by software in intermittent mode.

Note 2: ADPCH is modified when each channel conversion is completed and the sampling result is sent out, which means the ADPCH modification moment coincides with the moment when the ADC result is sent out.

25.3.3 Continuous Conversion Method

ADCTU[1:0] = 1x selects the continuous conversion mode, the sequence is cyclically converted, and the next conversion of this sequence is performed after a sequence is converted. For some register settings of continuous conversion, please refer to "single conversion mode". Unlike sequence conversion, this mode will cyclically convert the sequence. The time from the completion of one sequence conversion to the start of the next sequence is also controlled by TGAP[3:0]. When each sequence conversion is completed, the value of a specified result register is compared with the values of ADDGT and ADDLT, and a flag indicates the result of the comparison. When the software clears the ADSOC bit, AD conversion stops immediately. For example, SEQCH0=2, SEQCH1=5, SEQCH2=0, SEQCH3=7,..., SEQCH6=1, SEQCH7=6, when ADMAXCH=3, after the conversion starts, SEQCH0, SEQCH1, SEQCH2 and SEQCH3 are valid channels Select the register, the channel sequence formed by this is AN2, AN5, AN0, AN7. In this example, the timing relationship of each key signal is shown in Figure 25-4:

Note: The value of ADPCH is cleared at the end of sampling.

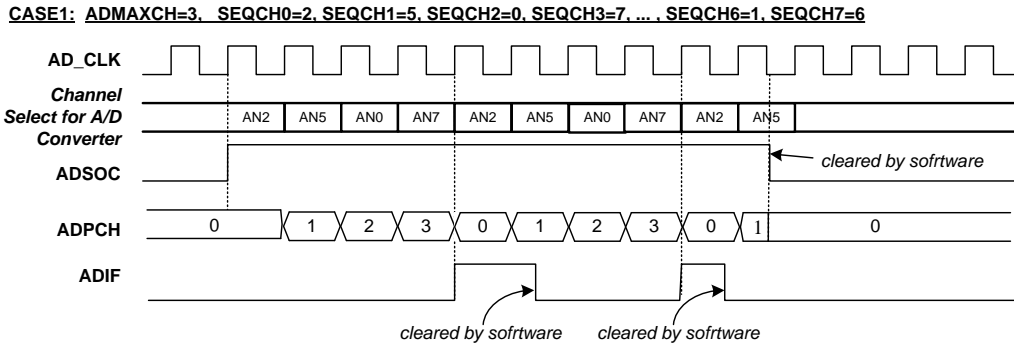


Figure 25-4 Continuous Conversion Method Waveform Example Diagram

25.3.4 Other Function Description

25.3.4.1 Channel Selection Settings in Sequence Conversion

A sequence can contain single or multiple channels. Store the channel numbers to be converted in the channel register SEQCHx (x = 0-7). There are 8 groups in the SEQCHx register. Therefore, when the sequence is converted, up to 8 channels can be converted at a time. The number of channels converted each time is determined by the value of ADMAXCH[2:0] in the register ADCON2. Example: ADMAXCH[2:0] = 0, that is, single-channel conversion, convert the channel stored in SEQCH0; ADMAXCH[2:0] = 3, then there are 4 channels in the sequence, and set the channels stored in SEQCH0 to SEQCH3 Convert in turn.

The channel register SEQCHx (x = 0-7) stores the channel number to be converted. Example: There are 3 channels that need to be converted. They are AN0, AN2, AN7 in order of priority, then set ADMAXCH[2:0] = 2, SEQCH0 = 0, SEQCH1 = 2, SEQCH2 = 7, and they will be converted sequentially.

Note: The channel to be converted must be configured as an AD channel function. The same channel number can also be set in SEQCHx, for example, all the values in SEQCHx are set to AN3, and the result register will store the converted value of AN3 in different time periods.

25.3.4.2 Sequence conversion mode ADC result register ADDR_x (x = 0-7)

The result of the sequence conversion is sequentially stored in the result register ADDR_x (x = 0-7), and the result register is a read-only register. The result register is fixed in a right-justified manner, with a sign bit. For example, when the ADC reference voltage selects VDD, the maximum value of the positive sampling result 0x0FFF represents $(4095/4096) \cdot VDD$, as shown in the figure below. After a sequence is converted, the value in the result register ADDR_x is also updated once.

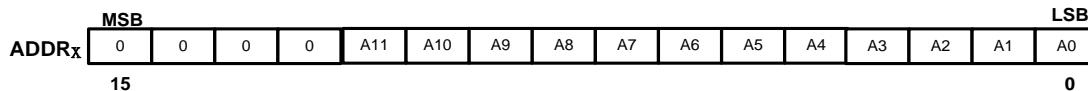


Figure 25-5 Example image of result register

25.3.4.3 Gap Time setting between channels during sequence conversion

During sequence conversion, the time from the completion of the previous channel conversion to the beginning of the sampling time of the next channel can be set by the register TGAP@ADCON2 bit segment. When TGAP[2:0] = 0, the sampling of the next channel starts immediately after the conversion of one channel is completed, and there is no waiting time in between.

25.3.4.4 Start and stop of sequence conversion mode

The startup of the sequence conversion mode can be divided into software startup and hardware startup.

The software starts, by setting the ADON bit in ADCON1 to 1, so that the clock AD_CLK of the ADC module can be enabled, and at the same time setting the ADON bit to 1 can power up the analog circuit in the ADC module. When the ADSOC bit in the ADCON1 register is 1, the conversion is started. When a sequence conversion is completed, ADSOC is cleared to 0 by the hardware, and comparison is performed at the same time (explained in detail in Section 25.3.5). When ADSOC is read as 1, the flag conversion is not completed. If the ADSOC bit is cleared during the conversion process, the conversion is terminated.



Note: After setting the ADON bit in ADCON1 to 1, you need to wait for $5\mu s + 5T_{con}$ before sampling.

There are 4 ways to start hardware, external interrupt (EXTI0~3) trigger signal, common timer (TIM2) module trigger signal, PWM module trigger signal trigger start and PCA module trigger signal trigger start. Specifically by setting the ADSTRS[6:0] bits in the ADCON1 register and the related registers of the ordinary timers.

When the software trigger is in the sequence conversion process, the hardware trigger signal cannot terminate the software trigger, and the hardware trigger request will be ignored.

Note 1: In any hardware trigger single conversion, intermittent conversion and continuous conversion process, if another hardware trigger occurs, the subsequent hardware trigger conversion request will be ignored.

Note 2: During the conversion process, that is, when ADSOC is 1, writing to all ADC registers except ADCON1, ADCMPCON, ADDGT, ADDLT will be regarded as an invalid operation.

25.3.4.5 Sequence conversion complete interrupt

After the sequence conversion is completed, ADIF@ADINTF will be set by the hardware. At this time, if the corresponding ADIE@ADCON1 is 1, it will trigger the sequence conversion completion interrupt, and the ADIFC@ADINTF bit can only be cleared by software.

25.3.5 Comparison function

25.3.5.1 Designation of result register for comparison

The register CSEL@ADCMPCON specifies the result register to be compared. If CSEL = n, the result register ADDRn will be compared with the values of ADDGT and ADDLT. Take the configuration of Figure 25-6 as an example. If CSEL = 2 is set, the value of ADDR2 is compared with the values of ADDGT and ADDLT at the same time when each sequence conversion is completed.

It should be noted that if you specify a result register that is not used by the sequence conversion, no comparison action will occur. As in the above example, CSEL = 3, the result register ADDR3 is not used after conversion, so no comparison action takes place, and the values of the flag bits ADGIF and ADLIF will not change.

Note: When writing the comparison values ADDGT and ADDLT, pay attention to whether the storage mode of the result register is left-aligned or right-aligned, and the data writing format should be consistent with the storage format of the result register. The written values of ADDGT and ADDLT have immediate effect, and the last updated value of ADDGT and ADDLT will be used for comparison.

25.3.5.2 Comparison process

Set CSEL = n, ADDGT = Max, ADDLT = Min, when the sequence conversion is completed, immediately compare the value of the designated result register ADDRn with Max and Min. If ADDRn \geq Max, the ADGIF bit in the ADINTF register will be set to 1, if the ADGIE bit is 1, a comparison interrupt can be triggered (share the interrupt vector with the sequence conversion completion), and the ADGIF bit will remain at 1 until the software clears it ; If ADDRn \leq Min, the ADLIF bit in the ADINTF register will be set to 1, if the ADLIE bit is 1, it can also trigger a comparison interrupt (shared with the sequence conversion complete interrupt vector), the ADLIF bit will remain 1 until the software Clear.

Set ADDGT = 0x3DF, ADDLT = 0x8FE, CSEL = n, then the relationship between the value of ADDRn and the value of the ADGIF and ADLIF bits is shown in the left half of Figure 25-6; set ADDGT = 0x8FE, ADDLT = 0x3DF, CSEL = n, The relationship between the value of ADDRn and the median value of ADGIF and ADLIF bits is shown in the right half of Figure 25-6.

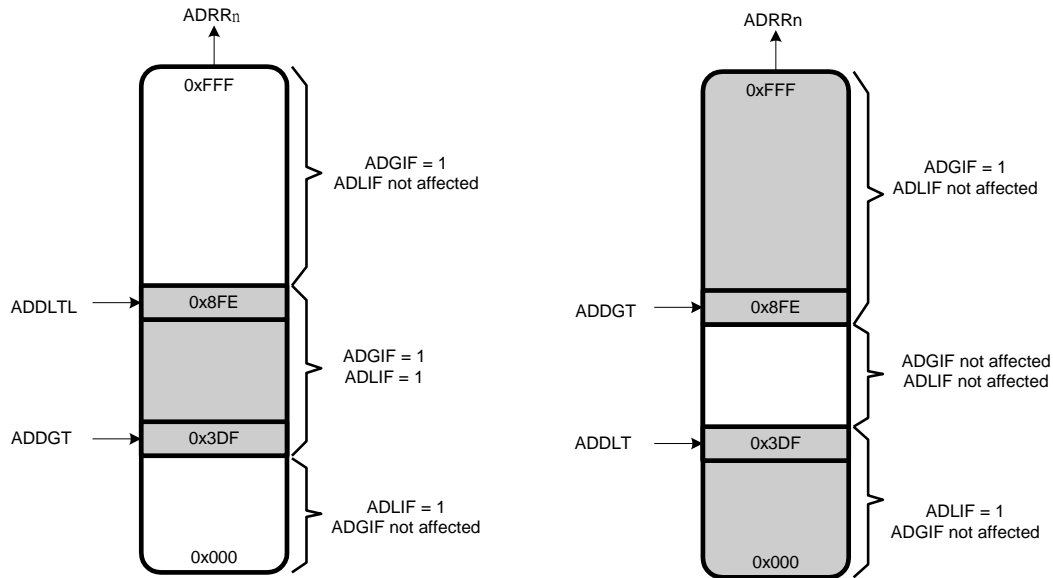


Figure 25-6 Comparison process diagram

25.3.5.3 Compare interrupt

Each time the sequence conversion is completed, the value of the designated result register will be compared. If it exceeds the limit, a comparison interrupt can be generated. Because the comparison and sequence conversion are completed at the same time, the interrupt is generated at the same time.

25.3.6 ADC conversion time setting

The ADC clock and sampling time can be set through the register ADCON2. The ADC clock can be set by setting the TADC[3:0] bit segment in ADCON2.

The TS[3:0], TGAP[2:0] and GAPENy@ADGAPON bits in the ADCON2 register can set the sampling time t_{SAMP} of each channel, $t_{SAMP} = (TS[3:0] + TGAP[2:0] * GAPEN + 1) * t_{AD}$, see ADC register chapter for details.

For 12BIT mode AD conversion, the AD conversion time of each channel is fixed at $15 * t_{AD}$. Therefore, the total conversion time of each channel = $t_{SAMP} + 15 * t_{AD}$. For 10BIT mode AD conversion, the AD conversion time of each channel is fixed at $13 * t_{AD}$. Therefore, the total conversion time of each channel = $t_{SAMP} + 13 * t_{AD}$.

Note: The fastest ADC clock is set to 24MHz.

25.3.7 ADC module reference voltage setting

The input voltage on the chip VDD or VREF pin can be selected as the reference voltage of the ADC module. After the chip is reset, the reference voltage of the ADC module is the chip's VDD. Setting the REFC[1:0] position in the ADCON1 register to 1 will make the ADC module use the voltage on the VREF pin as the reference voltage.

25.3.8 ADC channel and IO port function settings

Note 1: ADC input I/O port is set as analog port.

Note 2: The former I/O port configuration register is in the ADC module, and now it is in the GPIO module.

25.3.9 DMA request

The ADC sampling results can be stored through DMA, and after each channel sampling is over, the DMA function is triggered to achieve continuous conversion without software intervention. You can select whether to enable this function through the ADDE@ADNCON1 bit.



25.3.10 Requirements for sensor output impedance during AD conversion

As shown in Figure 25-7, in order to ensure that the AD conversion can convert accurate AD results under a certain conversion rate, it is necessary to charge the internal sampling capacitor C of the SH30F9/SA0 series within a certain period of time (sampling time TS). If the sampling capacitor C fails to be fully charged within the specified sampling time, that is, $V_0 < V_1$, the AD conversion result will produce accuracy errors. Because R0 (SH30F9/SA0 series internal resistance) and C (internal sampling capacitor) are constants and have been determined by the chip, whether the conversion of AD results within the allowable accuracy range depends on the sensor's internal equivalent resistance Rx. Suppose the resolution of the AD conversion result is M (the resolution of the 12bit result is 4096). If the accuracy requirement of the AD conversion is N, then the selection of the sensor resistance has the following requirements:

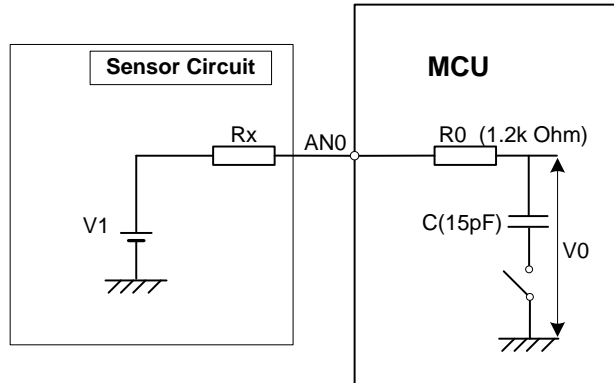


Figure 25-7 Output impedance

$$R_x \leq -\frac{T}{C * \ln \frac{V_1 - V_0}{V_1}} - R_0$$

The above formula can also be expressed as:

$$R_x \leq -\frac{T}{C * \ln \frac{N}{M}} - R_0$$

For 12bit mode that is 4096 resolution, the standard sampling time is 2us, and the result accuracy error is controlled within 0.1LSB, according to the above formula:

$$R_x \leq -\frac{2 \times 10^{-6}}{15 \times 10^{-12} \ln \frac{0.1}{4096}} - 1.2 \times 10^3 \approx 11.3K\Omega$$

That is, the internal equivalent resistance during sensor selection must be less than 11.3KΩ.

25.3.11 Matters needing attention

ADC conversion, it is best to turn on ADON@ADCON1 for 10us and then set ADSOC@ADCON1(GO/DONE) bit to 1 for conversion. Because there is a stabilization time after the ADC module is turned on.

When converting the analog channel, the port must be set to the AD channel function to be able to convert smoothly.

When ADIE, ADLIE, and ADGIE are all turned on, setting any one of ADIF, ADLIF, and ADGIF can cause an interrupt, and share an interrupt vector. By judging which bit ADIF, ADLIF, or ADGIF is 1, to determine the specific interrupt source and Customize specific operations.

Note: During the conversion process, when ADSOC is 1, all writes to ADC registers except for ADCON1, ADCMPCON, ADDGT, and ADDLT will be considered invalid operations.



25.4 Registers

ADC Module Register list (Base Address:0x4000 4000)

| Address | Register | Description |
|-------------|-----------|--|
| 0x4000 4000 | ADCON1 | ADC control register 1 |
| 0x4000 4004 | ADCON2 | ADC control register 2 |
| 0x4000 4008 | ADPCH | ADC sampling conversion channel pointer register |
| 0x4000 400C | ADDR0 | ADC result register 0 |
| 0x4000 4010 | ADDR1 | ADC result register 1 |
| 0x4000 4014 | ADDR2 | ADC result register 2 |
| 0x4000 4018 | ADDR3 | ADC result register 3 |
| 0x4000 401C | ADDR4 | ADC result register 4 |
| 0x4000 4020 | ADDR5 | ADC result register 5 |
| 0x4000 4024 | ADDR6 | ADC result register 6 |
| 0x4000 4028 | ADDR7 | ADC result register 7 |
| 0x4000 402C | ADCMPCON | ADC compare control register |
| 0x4000 4030 | ADDGT | ADC upper limit compare register |
| 0x4000 4034 | ADDLT | ADC lower limit compare register |
| 0x4000 4038 | SEQCHSEL0 | ADC channel select register 0 |
| 0x4000 403C | SEQCHSEL1 | ADC channel select register 1 |
| 0x4000 4040 | ADGAPON | ADC channel GAP control register |
| 0x4000 4044 | ADINTF | ADC interrupt flag and clear register |

25.4.1 ADC control register 1 (ADC_ADCON1)

Offset Address: 0x0000

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|------|-------------|-----|-----|-----|-----|----|------|------|-----------|----|-----|------------|----|-------|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ADON | ADSTRS[6:0] | | | | | | ADIE | ADDE | REFC[1:0] | | MOD | ADCTU[1:0] | | ADSOC | |
| RW | RW | | | | | | RW | RW | RW | | RW | RW | | RW | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|--|
| 31 ~ 16 | Reserved | - |
| 15 | ADON | ADC enable bit 0: ADC module is disabled 1: ADC module is enabled |
| 14 ~ 8 | ADSTRS[6:0] | ADC trigger source selection bit The start-up conversion enable bit for the channel sequence of ADC. Please refer to Table 25-1 for specific content. |
| 7 | ADIE | Channel sequence conversion end interrupt enable bit 0: ADC conversion end interrupt is disabled 1: ADC conversion end interrupt is enabled |
| 6 | ADDE | Channel sequence conversion end DMA trigger enable bit 0: ADC conversion end DMA trigger interrupt is disabled 1: ADC conversion end DMA trigger interrupt is enabled |



| | | |
|-------|-------------------|---|
| 5 ~ 4 | REFC[1:0] | Reference voltage selection 00: Select VDD as reference voltage 01: Select external VREF port input as reference voltage 1x: Select VDD as reference voltage |
| 3 | MOD | ADC operation mode selection bit 0: 12bit 1: 10bit |
| 2 ~ 1 | ADCTU[1:0] | Conversion mode control bit of ADC channel sequence 00: Single sequence conversion mode 01: Intermittent sequence conversion mode 10: Continuous conversion mode 11: Reserved Note: Modifying this control bit will generate effect when the next time the AD conversion process starts. |
| 0 | ADSOC | ADC channel sequence start-up AD conversion request bit 0: Sequence conversion/continuous conversion is not done or has been completed. Writing 0 can cancel the conversion that is currently in progress. 1: In the single and intermittent conversion modes, the sequence conversion starts when this bit is set to 1 and remains at 1 during the conversion process. This is automatically cleared by hardware after the conversion is completed (if cleared during conversion, this bit will immediately terminate the sequence conversion.) In continuous conversion mode, set 1 to start continuous conversion, then this bit will not be cleared by hardware, and can be cleared and terminated immediately by software. |

25.4.2 ADC control register 2 (ADC_ADCON2)

Offset Address: 0x0004

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | TGAP [2:0] | | |
| | | | | | | | | | | | | | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|----------|-----|----|--------------|--------------|----|----|-----------|----|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | TS [3:0] | | | Rese rved | ADMAXCH[2:0] | | | TADC[3:0] | | | | |
| | | | | RW | | | - | RW | | | RW | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|------------|--|
| 31 ~ 19 | Reserved | - |
| 18 ~ 16 | TGAP [2:0] | Time interval between adjacent channels while in sequence setting bit segment In a sequence conversion, the time interval between the completion of one channel conversion and the start of sampling of the next channel 000: no wait time 001: 2ADC clock period 010: 4ADC clock period 011: 8ADC clock period 100: 16ADC clock period 101: 32ADC clock period 110: 64ADC clock period 111: 128ADC clock period |
| 15 ~ 12 | Reserved | - |



SH30F9/SA0 Series

| | | |
|--------|---------------------|---|
| 11 ~ 8 | TS [3:0] | <p>Sampling time setting: $1 t_{AD} \leq (TS [3:0]+1) * t_{AD} \leq 16 t_{AD}$ Note 1: The total sampling time of a channel is composed of two parts: TS + TGAP, the TS part is set here; Note 2: Total sampling time range: $1 t_{AD} \leq ((TS[3:0])+1+TGAP[2:0]*GAPEN)* t_{AD} \leq 144 t_{AD}$, TGAP is controlled by GAPEN whether to enable; Note 3: Before setting TS[3:0], it is necessary to estimate the series resistance connected to the ADC input pin to obtain the best sampling accuracy. When $1 * t_{AD}$ is selected as the sampling time, make sure to connect to the ADC input pin The series resistance meets the input impedance requirement, see 25.3.10 for details; Note 4: TADC settings need to ensure that the ADC clock period $T_{adc} \geq 25ns$ Note 5: The total sampling conversion time of one channel in 12bit mode = $15t_{AD} + \text{sampling time}$; the total sampling conversion time of one channel in 10bit mode = $13t_{AD} + \text{sampling time}$</p> |
| 7 | Reserved | - |
| 6 ~ 4 | ADMAXCH[2:0] | <p>Total length setting bit of ADC channel sequence (set value 0~7) The total number of analog channels for one AD conversion process is $ADMAXCH+1$</p> |
| 3 ~ 0 | TADC[3:0] | <p>ADC clock cycle selection 0000: ADC clock cycle $t_{AD} = 1$ bus clock cycle 0001: ADC clock cycle $t_{AD} = 2$ bus clock cycles 0010: ADC clock cycle $t_{AD} = 3$ bus clock cycles 0011: ADC clock cycle $t_{AD} = 4$ bus clock cycles 0100: ADC clock cycle $t_{AD} = 5$ bus clock cycles 0101: ADC clock cycle $t_{AD} = 6$ bus clock cycles 0110: ADC clock cycle $t_{AD} = 8$ bus clock cycles 0111: ADC clock cycle $t_{AD} = 12$ bus clock cycles 1000: ADC clock cycle $t_{AD} = 16$ bus clock periods 1001: ADC clock cycle $t_{AD} = 24$ bus clock cycles 1010: ADC clock cycle $t_{AD} = 32$ bus clock cycles 1011: ADC clock cycle $t_{AD} = 48$ bus clock periods 1100: ADC clock cycle $t_{AD} = 64$ bus clock cycles 1101: ADC clock cycle $t_{AD} = 128$ bus clock cycles 1110: ADC clock cycle $t_{AD} = 256$ bus clock cycles 1111: ADC clock cycle $t_{AD} = 320$ bus clock periods</p> |



25.4.3 ADC sampling conversion channel pointer register (ADC_ADPCN)

Offset Address: 0x0008

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | ADPCN[2:0] | | |
| - | | | | | | | | | | | | | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|------------|--|
| 31 ~ 3 | Reserved | - |
| 2 ~ 0 | ADPCN[2:0] | <p>ADC sampling conversion channel pointer register</p> <p>000: ADC current sampling channel is SEQCH0 001: ADC current sampling channel is SEQCH1 010: ADC current sampling channel is SEQCH2 011: ADC current sampling channel is SEQCH3 100: ADC current sampling channel is SEQCH4 101: ADC current sampling channel is SEQCH5 110: ADC current sampling channel is SEQCH6 111: ADC current sampling channel is SEQCH7</p> <p>Note1: ADPCN can only be written by software when ADC is not sampling (ADSOC=0) and only in break mode.</p> <p>Note2: ADPCN changes after the last conversion is completed and the sampling results are sent out. That is to say, the change time of ADPCN is consistent with the sending time of ADC results.</p> |

25.4.4 ADC result register n (ADC_ADDRn)(n=0..7)

Offset Address: 0x000C

:0x0010

:0x0014

:0x0018

:0x001C

:0x0020

:0x0024

:0x0028

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ADDRn[15:0] | | | | | | | | | | | | | | | |
| RO | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|-------------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | ADDRn[15:0] | <p>After a channel is converted, the data is immediately updated and stored in ADDR_x (x = 0). All the result registers are updated once after the sequence conversion.</p> |



25.4.5 ADC compare control register (ADC_ADCMPCON)

Offset Address: 0x002C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----------|-----------|-----------|-----------|--------------|-----------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | ADG IE | ADG DE | ADLI E | ADL DE | Rese rved | CSEL[2:0] | | |
| - | | | | | | | | RW | RW | RW | RW | - | RW | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|-----------|--|
| 31 ~ 8 | Reserved | - |
| 7 | ADGIE | Upper limit compare interrupt enable bit 0: Upper limit compare interrupt is disabled 1: Upper limit compare interrupt is enabled |
| 6 | ADGDE | Upper limit compare DMA enable bit 0: Upper limit compare trigger DMA is disabled 1: Upper limit compare trigger DMA is enabled |
| 5 | ADLIE | Lower limit compare interrupt enable bit 0: Lower limit compare interrupt is disabled 1: Lower limit compare interrupt is enabled |
| 4 | ADLDE | Lower limit compare DMA enable bit 0: Lower limit compare trigger DMA is disabled 1: Lower limit compare trigger DMA is enabled |
| 3 | Reserved | - |
| 2 ~ 0 | CSEL[2:0] | Result register used for comparison selection bit segment 0: Select the value in ADDR0 to compare with ADDGT, ADDLT 1: Select the value in ADDR1 to compare with ADDGT, ADDLT 2: Select the value in ADDR2 to compare with ADDGT, ADDLT 3: Select the value in ADDR3 to compare with ADDGT, ADDLT 4: Select the value in ADDR4 to compare with ADDGT, ADDLT 5: Select the value in ADDR5 to compare with ADDGT, ADDLT 6: Select the value in ADDR6 to compare with ADDGT, ADDLT 7: Select the value in ADDR7 to compare with ADDGT, ADDLT |

25.4.6 ADC upper limit compare register (ADC_ADDGT)

Offset Address: 0x0030

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| - | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| GT[15:0] | | | | | | | | | | | | | | | |
| RW | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | GT[15:0] | ADC upper limit compare register When a sequence conversion is completed, the value of the result register specified by CSEL bit segment in the ADCMPCON register is immediately compared with the value in ADDGT. If it is greater than or equal to the value in ADDGT, the ADGIF bit in ADCMPCON is set to 1, and the ADGIF bit will remain at 1 until it is cleared by software. |

25.4.7 ADC lower limit compare register (ADC_ADDLT)

Offset Address: 0x0034

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LT[15:0] | | | | | | | | | | | | | | | |
| <i>RW</i> | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|---|
| 31 ~ 16 | Reserved | - |
| 15 ~ 0 | LT[15:0] | ADC lower limit compare register When a sequence conversion is completed, the value of the result register specified by CSEL bit segment in the ADCMPCON register is immediately compared with the value in ADDLT. If it is smaller than or equal to the value in ADDLT, the ADLIF bit in ADCMPCON is set to 1, and the ADLIF bit will remain at 1 until it is cleared by software. |

25.4.8 ADC channel select register 0 (ADC_SEQCHSEL0)

Offset Address: 0x0038

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-------------|-----|-----|-----|----------|-----|-----|-----|-------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | SEQCH3[4:0] | | | | Reserved | | | | SEQCH2[4:0] | | | |
| - | | | | <i>RW</i> | | | | - | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | SEQCH1[4:0] | | | | Reserved | | | | SEQCH0[4:0] | | | |
| - | | | | <i>RW</i> | | | | - | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|-------------|
| 31 ~ 29 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 28 ~ 24 | SEQCH3[4:0] | Analog channel 3 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 23 ~ 21 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 20 ~ 16 | SEQCH2[4:0] | Analog channel 2 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 15 ~ 13 | Reserved | - |



SH30F9/SA0 Series

| | | |
|--------|--------------------|---|
| 12 ~ 8 | SEQCH1[4:0] | Analog channel 1 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 7 ~ 5 | Reserved | - |



| | | |
|-------|--------------------|---|
| 4 ~ 0 | SEQCH0[4:0] | Analog channel 0 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
|-------|--------------------|---|

25.4.9 ADC channel select register 1 (ADC_SEQCHSEL1)

Offset Address: 0x003C

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-------------|-----|-----|-----|----------|-----|-----|-----|-------------|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | SEQCH7[4:0] | | | | Reserved | | | | SEQCH6[4:0] | | | |
| | | | | <i>RW</i> | | | | | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-------------|-----|----|----|----------|----|----|----|-------------|----|----|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | SEQCH5[4:0] | | | | Reserved | | | | SEQCH4[4:0] | | | |
| | | | | <i>RW</i> | | | | | | | | <i>RW</i> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|-------------|
| 31 ~ 29 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 28 ~ 24 | SEQCH7[4:0] | Analog channel 7 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 23 ~ 21 | Reserved | - |



SH30F9/SA0 Series

| | | |
|---------|--------------------|---|
| 20 ~ 16 | SEQCH6[4:0] | Analog channel 6 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 15 ~ 13 | Reserved | - |



SH30F9/SA0 Series

| | | |
|--------|--------------------|---|
| 12 ~ 8 | SEQCH5[4:0] | Analog channel 5 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
| 7 ~ 5 | Reserved | - |



SH30F9/SA0 Series

| | | |
|-------|--------------------|---|
| 4 ~ 0 | SEQCH4[4:0] | Analog channel 4 select register group, specifying the corresponding analog input channel: 00000 - AN0 00001 - AN1 00010 - AN2 00011 - AN3 00100 - AN4 00101 - AN5 00110 - AN6 00111 - AN7 01000 - AN8 01001 - AN9 01010 - AN10 01011 - AN11 01100 - AN12 01101 - AN13 01110 - AN14 01111 - AN15 10000 - AN16 10001 - AN17 10010 - AN18 10011 - AN19 10100 - AN20 10101 - AN21 10110 - AN22 10111 - AN23 11000 - AN24 11001 - AN25 11010 - AN26 11011 - V _{CC} Other - AN0 |
|-------|--------------------|---|



25.4.10 ADC channel GAP control register (ADC_ADGAPON)

Offset Address: 0x0040

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | GAPENy(y=7~0) | | | | | | | |
| - | | | | | | | | RW | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|--------|---------------|---|
| 31 ~ 8 | Reserved | - |
| 7 ~ 0 | GAPENy(y=7~0) | GAP time enable bit of ADC channel 0: Disabled 1: Enabled |

25.4.11 ADC interrupt flag and clear register (ADC_ADINTF)

Offset Address: 0x0044

Reset value: 0x0000 0000

| | | | | | | | | | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|------------|------------|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| Reserved | | | | | | | | | | | | | ADIF C | ADG IFC | ADLI FC |
| - | | | | | | | | | | | | | WO | WO | WO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| Reserved | | | | | | | | | | | | | ADIF | ADG IF | ADLI F |
| - | | | | | | | | | | | | | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|---------|----------|--|
| 31 ~ 19 | Reserved | - |
| 18 | ADIFC | Conversion end interrupt flag clear bit 0: Invalid 1: Clear |
| 17 | ADGIFC | Upper limit compare interrupt flag clear bit 0: Invalid 1: Clear |
| 16 | ADLIFC | Lower limit compare interrupt flag clear bit 0: Invalid 1: Clear |
| 15 ~ 3 | Reserved | - |
| 2 | ADIF | Channel sequence conversion end interrupt flag bit 0: No channel sequence conversion end interrupt occurs 1: Channel sequence conversion end interrupt occurs |
| 1 | ADGIF | Upper limit compare interrupt flag bit 0: No upper limit interrupt occurs, and the latest updated value in the result register specified by the CSEL bit segment is smaller than the value in ADDGT 1: Upper limit interrupt occurs, and the latest updated value in the result register specified by the CSEL bit segment is greater than or equal to the value in ADDGT. This flag needs to be cleared by software after it is set. |



SH30F9/SA0 Series

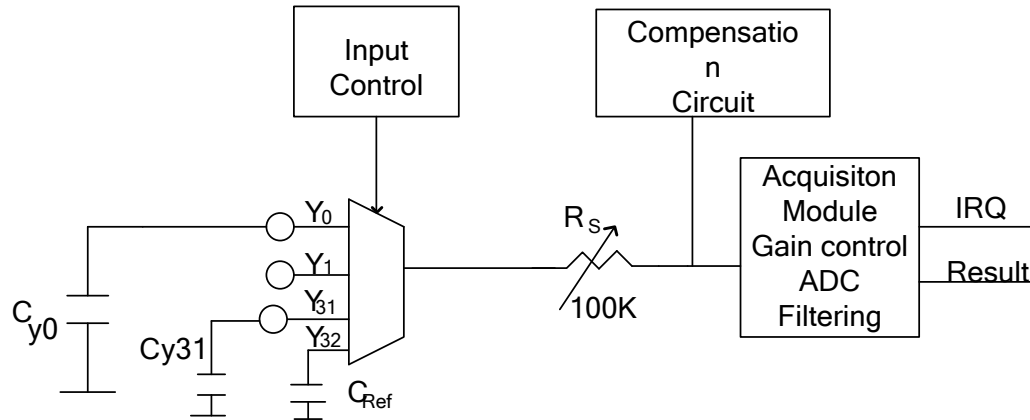
| | | |
|---|--------------|--|
| 0 | ADLIF | <p>Lower limit compare interrupt flag bit</p> <p>0: no lower limit interrupt occurs, and the last updated value in the result register specified by the CSEL bit segment is greater than the value in ADDLT</p> <p>1: lower limit interrupt occurs, and the latest updated value in the result register specified by the CSEL bit segment is smaller than or equal to the value in ADDLT.</p> <p>This flag needs to be cleared by software after it is set.</p> |
|---|--------------|--|

Table 25-1 ADC trigger source selection table

| ADSTRS[6:0] | | | | | Occurrence source | AD start source | Start condition |
|-------------|---|---|---|---|--------------------|-----------------|--------------------------------------|
| - | - | - | - | - | software | ADSOC | Software start |
| 000 | x | x | x | 1 | External Interrupt | EXTI0 | External interrupt 0 |
| | x | x | 1 | x | | EXTI1 | External interrupt 1 |
| | x | 1 | x | x | | EXTI2 | External interrupt 2 |
| | 1 | x | x | x | | EXTI3 | External interrupt 3 |
| 001 | x | x | x | x | Timer 2 | TIMTRG | Timer 2 overflow |
| 010 | x | x | x | x | PWM0 | PWM0TRG | PWM0 overflow |
| 011 | x | x | x | x | PWM1 | PWM1TRG | PWM1 overflow |
| 100 | x | x | x | x | PWM2 | PWM2TRG | PWM2 overflow |
| 101 | x | x | x | x | PWM3 | PWM3TRG | PWM3 overflow |
| 110 | x | x | x | 1 | PCA0 | PCA0TRG | PCA0 interrupt flag overflow trigger |
| | x | x | 1 | x | PCA1 | PCA1TRG | PCA1 interrupt flag overflow trigger |
| 111 | x | x | x | 1 | PCA2 | PCA2TRG | PCA2 interrupt flag overflow trigger |
| | x | x | 1 | x | PCA3 | PCA3TRG | PCA3 interrupt flag overflow trigger |



26. Touch Key



System Block Diagram

Function description

SH30F9/SA0 series has a built-in touch key function module, which can connect up to 32 keys, and a built-in calibration circuit. The touch detection function can be realized without external capacitor in application.

Steps to start scanning with Touch Key function:

- (1) Select the key channel to be scanned;
- (2) Bit of the TKCON register is 1, allowing the Touch Key module to work;
- (3) Set the key sampling times, scan sequence, and initial value of C_s , with a delay of at least 5.4us (warm-up time);
- (4) Register TKGO/DONE is set to 1, start the key scan, and start the calibration process (after TKGO is set to 1, Before it is cleared, it is not allowed to modify the value of any TK module register);
- (5) When an interrupt is generated, TKGO hardware is automatically cleared to 0
- (6) Calculate and adjust the size of C_s based on the first set of data;
- (7) Repeat steps (4)-(6) until the detection error is within a reasonable range in the non-touch state;
- (8) Register TKGO/DONE is set to 1, start key scan, start the normal scan process;
- (9) IFAVE = 1, the program saves the data results, and a group of key scans are completed.

Note: For details, please refer to the touch instruction document (including library files and instructions)



27. Device Electronic Signature

27.1 Unique device ID register (96 bits)

The unique device identifier is ideally suited:

- For use as serial numbers (for example, the product serial number meets the terminal application)
- For use as security keys in order to increase the security of code in Flash memory while using and combining this unique ID with software cryptographic primitives and protocols before programming the internal Flash memory
- Activate secure boot processes, etc.

The 96-bit unique device identifier provides a reference number which is unique for any device and in any SH30F9071 / 30F9871 context. These bits can never be altered by the user.

The 96-bit unique device identifier can also be read in single bytes/half-words/words in different ways.

The 96-bit identifier is located in the Flash address area 0x0FFF E110 to 0x0FFF E11B. There is no definite rule. You are advised to use all 96 bits as the unique product ID.



28. Debug interface

28.1 Main Feature

SH30F9/SA0 series uses Cortex™-M0+ core, this core contains hardware debug module, supports complex debug operation. Hardware debug module allows core stop when fetching instruction (instruction bread point) or accessing data (data break point). When core stops, both core state and system external state can be queried. After the query is completed, the core and peripheral can be restored, the program will continue to execute.

SH30F9/SA0 series supports SW debug interface, when SH30F9/SA0 series microcontroller connects to debugger and starts to debug, the debugger will use hardware debug module of core for debug operation.

The ARM Cortex™-M0+ core provides integrated on-chip debug function. It consists of the following parts:

- SW-DP: Serial Wire debug port
- AHP-AP: AHB access port
- FPB: Flash patch breakpoint
- DWT: Data watch point trigger

For more information about the debugging interface, please refer to the ARM official manual and the ARM development tool set technical reference manual.

28.2 SW Debug Port (Serial Wire)

The SH30F9/SA0 series core integrates the serial debug interface (SW-DP), which is a simplified version of the standard ARM Core Sight debug interface. The serial debug interface (SW-DP) provides a 2-pin (clock + data) interface for the AHP-AP module.

28.3 SW debug port pin

Two pins are used GPIO from the SH30F9/SA0 series for the SW-DP port pin. These pins are present in all packages.

Table 28-1 SW Debug Interface Pins

| SW-DP Interface pin name | SW Debug Interface | | Pin Assignment |
|--------------------------|--------------------|-------------------------------|----------------|
| | Type | Debug Function | |
| SWDIO | Input/output | Serial wire data input/output | PA4 |
| SWCLK | Input | Serial wire clock | PA5 |

28.4 Using serial wire and releasing the unused debug pin as GPIOs

After reset (SYSRESET or POWRESET), all 2 pins belonging to SW-DP are immediately initialized as dedicated pins that can be used by the debugger. However, the SH30F9/SA0 series microcontroller can reuse the SW interface as a GPIO port. For details, please refer to the relevant introduction in the "SYSCFG" chapter.

For user software design, it should be noted that after reset, these dedicated pins are still in the state of input with pull-up (SWDIO) and input (SWCLK) with pull-down, and continue for a period of time until the user code releases these pins.

Table 28-2 SW_DP pin assignment

| SWJCFG | The dedicated pins configured as debug | SW debug I/O pin assignment | | Whether SW Debug Interface is Available |
|--------|--|-----------------------------|-----------------------------|---|
| | | PA4 /SWDIO | PA5 /SWCLK | |
| 0 | SW-DP pin Reset state | Debugging dedicated | Debugging dedicated | Available |
| 1 | SW-DP port is disabled | Available for user (Note 1) | Available for user (Note 1) | Available |

Note 1: When in debug mode, the SWDIO and SWCLK pins are still configured as debug interfaces, and it is invalid to modify the corresponding common I/O port configuration registers.



28.5 MCU debug module

MCU Debug module assists the debugger to provide the following functions:

- Low power mode
- Provide clock control of PWM, PCA, TIMER, UART, SPI, TWI, DMA, IWDWT, WWDT, LED and LCD during breakpoints

28.5.1 Debug Support for Low Power Mode

Low power mode can be entered by using WFI and WFE. MCU supports a variety of low-power modes that can turn off the CPU clock or reduce the power consumption respectively. However, in order to enable the user to debug the code in low power mode, FCLK and HCLK cannot be turned off during debugging, which means setting DBG_STOP=0. Therefore, after entering the low power mode in debug mode, FCLK and HCLK are working by default. If user needs to enter the real low power mode during debugging, DBG_STOP=1 can be set by the debugger or software.

Note that DBG_STOP=0 only guarantees that the registers of each peripheral can be read after STOP in debug mode, but the peripheral's working clock is still off. In addition, HCLK is clocked by the internal HSI in this case.

28.5.2 Module Debug Support with Timing Function

When generating breakpoints, it is necessary to select the operating mode of the corresponding counters according to the different uses of each module. For modules like PWM, PCA, TIMER, UART, SPI, TWI, DMA, IWDWT, WWDT, LED and LCD, when breakpoint is generated, user can select whether the counter of each module continues to count.

28.5.3 Debug MCU configuration register

For details, please refer to the "System Configuration Module" chapter.



29. Customer Options

FLASH_DIVDE:

- 0: Flash is not divided into blocks (Default)
- 1: Flash is divided into two blocks, Main Memory and Backup Memory

FLASH_SELECT: (Only valid when FLASH_DIVDE be set as 1)

- 0: PC pointer addresses from Main Memory (Default)
- 1: PC pointer addresses from Backup Memory



SH30F9/SA0 Series

30. Electrical characteristics

Absolute rating *

DC supply voltage. -0.3V to +6.0V
 Input/output voltage GND-0.3V to VDD+0.3V
 Working environment temperature. -40°C to +105°C
 Storage temperature. -55°C to +125°C
 FLASH memory write/erase operation . . . -40°C to +105°C
 Pin internal protection diode current. . . . -2mA to +2mA

* Note

If the working conditions of the device exceed the range of the "absolute rating" in the left column, it will cause permanent damage to the device. The function is only guaranteed if the device is operating within the limits specified in the instructions. Working under the conditions listed in the limit parameters will affect the reliability for the operation of the device.

DC electrical characteristics(VDD = 2.0– 5.5V, GND = 0V, TA = 25°C, unless otherwise indicated)

| Parameter | Symbol | Minimum value | Typical value | Maximum value | Unit | Condition |
|------------------------------|------------------|---------------|---------------|---------------|------|--|
| Working voltage | V _{DD} | 2.0 | 5.0 | 5.5 | V | 4MHz ≤ f _{sys} ≤ 48MHz |
| Operating current | I _{OP1} | - | 10 | 25 | mA | f _{sys} =48MHz, PLL is on, V _{DD} = 5.0V, using crystal oscillator(8MHz). All output pins are not loaded and all input pins are not floating. CPU is on (executing local loop jump), IWDT is on, LVR is on, and all other functions are turned off. |
| | I _{OP2} | - | 5 | 10 | mA | f _{sys} =24MHz, PLL is close, V _{DD} = 5.0V, using internal RC oscillator (24MHz). All output pins are not loaded and all input pins are not floating. CPU is on (executing local loop jump), IWDT is on, LVR is on, and all other functions are turned off. |
| | I _{OP3} | - | 0.5 | 1 | mA | f _{sys} =32kHz, PLL is off, High frequency oscillator is shut down, V _{DD} = 5.0V, using crystal oscillator (32.768kHz). All output pins are not loaded and all input pins are not floating. CPU is on (executing local loop jump), IWDT is on, LVR is on, and all other functions are turned off. |
| Standby current (Sleep mode) | I _{SB1} | - | 5 | 18 | mA | f _{sys} = 48MHz, PLL is on, V _{DD} = 5.0V, using crystal oscillator(8MHz). All output pins are not loaded and all input pins are not floating. CPU is off (executing WFI to enter sleep mode), IWDT is on, LVR is on, and all other functions are turned off. |
| | I _{SB2} | - | 2 | 7 | mA | f _{sys} = 24MHz, PLL is off, V _{DD} = 5.0V, using internal RC oscillator (24MHz). All output pins are not loaded and all input pins are not floating. CPU is off (executing WFI to enter sleep mode), IWDT is on, LVR is on, and all other functions are turned off. |



SH30F9/SA0 Series

| | | | | | | |
|-----------------------------|-------------------|---|-----|----|----|---|
| | I _{SB3} | - | 0.5 | 1 | mA | f _{sys} =32kHz, PLL is off, High frequency oscillator is shut down, V _{DD} = 5.0V, using crystal oscillator (32.768kHz). All output pins are not loaded and all input pins are not floating. CPU is off (executing WFI to enter sleep mode), IWDT is on, LVR is on, and all other functions are turned off. |
| Standby current (Stop mode) | I _{SB4} | - | 12 | 22 | uA | High frequency oscillator is shut down, PLL is off, V _{DD} = 5.0V, All output pins are not loaded and all input pins are not floating. CPU is off (executing WFI to enter stop mode), IWDT is on, LVR is off, and all other functions are turned off. |
| | I _{SB5} | - | 14 | 25 | uA | f _{sys} = 32kHz, High frequency oscillator is shut down, PLL is off, V _{DD} = 5.0V, All output pins are not loaded and all input pins are not floating. CPU is off (executing WFI to enter stop mode), IWDT is on, LVR is off, and all other functions are turned off. |
| LVR current | I _{LVR1} | - | 1 | 3 | uA | LVR is on, LVR level=4.1V/3.7V/2.8V/2.3V |
| IWDT current | I _{IWDT} | - | - | 1 | uA | All output pins are not loaded, IWDT is on, V _{DD} =5.0V |

Flash features (V_{DD} = 2.0 - 5.5V, G_{ND} = 0V, T_A = 25°C, unless otherwise stated)

| Parameter | Symbol | Minimum value | Typical value | Maximum value | Unit | Condition |
|-----------------------------|---------------------------------|---------------|---------------|---------------|----------------|---|
| Read operation speed | T _{READ} | 40 | - | - | ns | 32-bit read |
| Burn write operation speed | T _{PROG} | - | - | 20 | us | 32-bit write |
| The sector erase | T _{SECTOR_ERASE} | - | - | 2 | ms | A single sector |
| Total erasure speed | T _{TOTAL_ERASE} | - | - | 10 | ms | |
| Number of programs/erasures | N _{END} ⁽¹⁾ | | | | | |
| Program area | - | 100 | - | - | Thousand times | Complies with JESD22-A117 testing standards |
| EEPROM-like area | - | 100 | - | - | Thousand times | Complies with JESD22-A117 testing standards |
| Data storage life | t _{RET} ⁽¹⁾ | 20 | - | - | Year | |

(1) Guaranteed by design, not tested in production



SH30F9/SA0 Series

Brownout detection (BOD) electrical characteristics ($V_{DD}=2.0-5.5V$, $GND = 0V$, $T_A = +25^\circ C$, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|---------------------|--------------------|---------------|---------------|---------------|---------|----------------------------|
| BOD voltage range | V_{BOD} | 2.4 | - | 4.65 | V | |
| BOD threshold level | $ \Delta V_{BOD} $ | - | 150 | - | mV | $V_{BOD}= 2.4V \sim 4.65V$ |
| BOD debounce time | T_{BOD} | 40 | - | 80 | μs | |
| BOD voltage 0 | V_{BOD0} | 2.30 | 2.40 | 2.50 | V | $VBOD[3:0]=0000$ |
| BOD voltage 1 | V_{BOD1} | 2.45 | 2.55 | 2.65 | V | $VBOD[3:0]=0001$ |
| BOD voltage 2 | V_{BOD2} | 2.60 | 2.70 | 2.80 | V | $VBOD[3:0]=0010$ |
| BOD voltage 3 | V_{BOD3} | 2.75 | 2.85 | 2.95 | V | $VBOD[3:0]=0011$ |
| BOD voltage 4 | V_{BOD4} | 2.90 | 3.00 | 3.10 | V | $VBOD[3:0]=0100$ |
| BOD voltage 5 | V_{BOD5} | 3.05 | 3.15 | 3.25 | V | $VBOD[3:0]=0101$ |
| BOD voltage 6 | V_{BOD6} | 3.20 | 3.30 | 3.40 | V | $VBOD[3:0]=0110$ |
| BOD voltage 7 | V_{BOD7} | 3.35 | 3.45 | 3.55 | V | $VBOD[3:0]=0111$ |
| BOD voltage 8 | V_{BOD8} | 3.50 | 3.60 | 3.70 | V | $VBOD[3:0]=1000$ |
| BOD voltage 9 | V_{BOD9} | 3.65 | 3.75 | 3.85 | V | $VBOD[3:0]=1001$ |
| BOD voltage 10 | V_{BOD10} | 3.80 | 3.90 | 4.00 | V | $VBOD[3:0]=1010$ |
| BOD voltage 11 | V_{BOD11} | 3.95 | 4.05 | 4.15 | V | $VBOD[3:0]=1011$ |
| BOD voltage 12 | V_{BOD12} | 4.10 | 4.20 | 4.30 | V | $VBOD[3:0]=1100$ |
| BOD voltage 13 | V_{BOD13} | 4.25 | 4.35 | 4.45 | V | $VBOD[3:0]=1101$ |
| BOD voltage 14 | V_{BOD14} | 4.40 | 4.50 | 4.60 | V | $VBOD[3:0]=1110$ |
| BOD voltage 15 | V_{BOD15} | 4.55 | 4.65 | 4.75 | V | $VBOD[3:0]=1111$ |

Low voltage reset (LVR) electrical characteristics ($V_{DD}=2.0-5.5V$, $GND = 0V$, $T_A = +25^\circ C$, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|--|-------------|---------------|---------------|---------------|---------|-------------|
| LVR voltage 1 | V_{LVR1} | 3.95 | 4.1 | 4.25 | V | LVR1 enable |
| LVR voltage 2 | V_{LVR2} | 3.55 | 3.7 | 3.85 | V | LVR2 enable |
| LVR voltage 3 | V_{LVR3} | 2.7 | 2.8 | 2.9 | V | LVR3 enable |
| LVR voltage 4 | V_{LVR4} | 2.2 | 2.3 | 2.4 | V | LVR4 enable |
| LVR hysteresis window | V_{SMTLV} | - | 50 | - | mv | |
| LVR debounce time (equal to LVR reset width) | T_{LVR} | 40 | - | 80 | μs | |



SH30F9/SA0 Series

Power on reset electrical characteristics (V_{DD} = 2.0 - 5.5V, GND = 0V, T_A = 25°C, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|------------------------------------|---------------------------------|---------------|---------------|---------------|------|---|
| Power on reset voltage | V _{POR} | 1.82 | 1.90 | 1.98 | V | |
| Rise slope of power supply voltage | S _{VDD} ⁽¹⁾ | 0.005 | - | 1000 | V/ms | LVR enable, V _{DD} = 2.0V - 5.5V |

(1) Design guarantee, production does not do the full slope range test.

AC electrical characteristics (V_{DD} = 2.0 - 5.5V, GND = 0V, T_A = +25°C, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|----------------------------|--------------------|---------------|---------------|---------------|------|---|
| Reset pulse width | t _{RESET} | 100 | - | - | μs | Low level is valid |
| Reset pin pull-up resistor | R _{RPH} | 15 | 30 | 45 | kΩ | V _{DD} = 5.0V, V _{IN} = GND |
| Frequency stability (HSI) | $ \Delta F1 /F1$ | - | - | 0.3 | % | Internal high frequency RC oscillator accuracy: (average of 1024 periods) F _{RC1} - 24MHz /24MHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = 25°C) |
| | $ \Delta F1 /F1$ | - | - | 0.5 | % | Internal high frequency RC oscillator accuracy: (average of 1024 periods) F _{RC1} - 24MHz /24MHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = -10°C ~ 60°C, Design guarantee, do not test in production) |
| | $ \Delta F1 /F1$ | - | - | 1.0 | % | Internal high frequency RC oscillator accuracy: (average of 1024 periods) F _{RC1} - 24MHz /24MHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = -40°C ~ +105°C, Design guarantee, do not test in production) |
| Frequency stability (LSI) | $ \Delta F2 /F2$ | - | - | 4.0 | % | Internal low frequency RC oscillator accuracy: (average of 1024 periods) F _{RC2} - 128kHz /128kHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = 25°C) |
| | $ \Delta F2 /F2$ | - | - | 7.0 | % | Internal low frequency RC oscillator accuracy: (average of 1024 periods) F _{RC2} - 128kHz /128kHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = -10°C ~ 60°C, Design guarantee, do not test in production) |
| | $ \Delta F2 /F2$ | - | - | 15.0 | % | Internal low frequency RC oscillator accuracy: (average of 1024 periods) F _{RC2} - 128kHz /128kHz X 100% (V _{DD} = 2.0 - 5.5V, T _A = -40°C ~ +105°C, Design guarantee, do not test in production) |



SH30F9/SA0 Series

PLL characteristics ($V_{DD} = 2.0 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|----------------------------|----------------|---------------|---------------|---------------|------|---|
| PLL input clock frequency | f_{PLL_IN} | 4 | - | 24 | MHz | |
| PLL input clock Duty | D_{PLL_IN} | 40 | - | 60 | % | |
| PLL output clock frequency | f_{PLL_OUT} | - | - | 48 | MHz | |
| PLL setup time | T_{PLL} | - | 2 | - | ms | |
| PLL frequency stability | - | - | ± 0.05 | ± 0.1 | % | (Average of 1024 periods $V_{DD}=2.0 - 5.5V$, $T_A = 25^\circ C$) |

GPIO Electrical characteristics ($V_{DD}=2.0-5.5V$, $GND=0V$, $T_A=25^\circ C$, unless otherwise indicated)

| Parameter | Symbol | Minimum value | Typical value | Maximum value | Unit | Condition |
|-----------------------|--------------------|----------------------------|---------------|----------------------|-----------|--|
| Input low voltage 1 | V_{IL1} | GND | - | $0.3 \times V_{DD}$ | V | I/O ports (all ports are schmitt input) $V_{DD}=3.0-5.5V$ |
| | | GND | - | $0.2 \times V_{DD}$ | V | I/O ports (all ports are schmitt input) $V_{DD}=2.0-3.0V$ |
| Input high voltage 1 | V_{IH1} | $0.6 \times V_{DD}$ | - | V_{DD} | V | I/O ports (all ports are schmitt input) $V_{DD}=3.0-5.5V$ |
| | | $0.7 \times V_{DD}$ | - | V_{DD} | V | I/O ports (all ports are schmitt input) $V_{DD}=2.0-3.0V$ |
| Input low voltage 2 | V_{IL2} | GND | - | $0.2 \times V_{DD}$ | V | Only SWDIO、SWCLK pin, $V_{DD}=2.0-5.5V$ |
| Input high voltage 2 | V_{IH2} | $0.8 \times V_{DD}$ | - | V_{DD} | V | Only SWDIO、SWCLK pin, $V_{DD}=2.0-5.5V$ |
| Input low voltage 3 | V_{IL3} | GND | - | 0.8 | V | I/O ports (all ports are TTL input) (Input high/low voltage window is 0.4V) $V_{DD}=4.5-5.5V$, TTL function is on (Note2) |
| | | GND | - | $0.15 \times V_{DD}$ | V | I/O ports (all ports are TTL input) (Input high/low voltage window is 0.4V) $V_{DD}=2.0-4.5V$, TTL function is on (Note2) |
| Input high voltage 3 | V_{IH3} | 2.0 | - | V_{DD} | V | I/O ports (all ports are TTL input) (Input high/low voltage window is 0.4V) $V_{DD}=4.5-5.5V$, TTL function is on (Note2) |
| | | $0.25 \times V_{DD} + 0.8$ | - | V_{DD} | V | I/O ports (all ports are TTL input) (Input high/low voltage window is 0.4V) $V_{DD}=2.0-4.5V$, TTL function is on (Note2) |
| Input leakage current | I_{IL} | -1 | | 1 | μA | Input has no pull-up, $V_{IN}=V_{DD}$ or GND |
| Pull-up resistor | R_{PH1} | 15 | 30 | 45 | $k\Omega$ | $V_{DD} = 5.0V$, $V_{IN}=GND$ |
| IO output current | I_{OH1} | -10 | -12 | - | mA | I/O port, $V_{OH} = 4.3V$, $V_{DD} = 5.0V$ |
| | $I_{OH1MAX}^{(1)}$ | - | - | -30 | mA | I/O port, $V_{DD} = 5.0V$, $T_A = +25^\circ C$ |
| IO sink current | I_{OL1} | 15 | 20 | - | mA | I/O port, $V_{OL} = 0.6V$, $V_{DD} = 5.0V$, $GPIOx_ODRVR_SINKy = 0b$ (Note3) |
| | $I_{OL1MAX}^{(1)}$ | - | - | 30 | mA | I/O port, $V_{DD} = 5.0V$, $T_A = +25^\circ C$, $GPIOx_ODRVR_SINKy = 0b$ (Note3) |



SH30F9/SA0 Series

| | | | | | | |
|--|--------------------|-----|-----|------|----|---|
| IO sink current capability of large drive port | I_{OL2} | 200 | 230 | - | mA | I/O port, $V_{DD} = 5.0V, V_{OL} = 0.5V$ $GPIOx_ODRVR_SINKy = 1b$ (Note3) Only PB2~PB9 (LED_C0~7) |
| | $I_{OL2MAX}^{(1)}$ | - | - | 250 | mA | I/O port, $V_{DD} = 5.0V, TA = +25^{\circ}C$ $GPIOx_ODRVR_SINKy = 1b$ (Note3) Only PB2~PB9 (LED_C0~7) |
| VDD current | I_{VDD1} | - | - | 200 | mA | $V_{DD} = 5.0V, TA = +25^{\circ}C$ |
| GND current | I_{VSS1} | - | - | -250 | mA | $V_{DD} = 5.0V, TA = +25^{\circ}C$ |

Note:

- (1). The project is only a design guarantee, and there is no test for mass production
- (2). The TTL input function is defined with reference to the $GPIOx_CFG_TTLEN$ register of the GPIO module.
- (3). The drive capability selection is referenced to the $GPIOx_CFG_ODRVR$ register definition of the GPIO module.

LED current source accuracy

| Parameter | Symbol | Min | Typical | Max | Unit | conditions |
|--|----------------|-----|---------|---------|------|---|
| Accuracy of current (Separate calibration) | $ \Delta I /I$ | - | ± 1 | ± 3 | % | The constant current source is selected to output 20mA current: $(I-20) / 20$ ($V_{DD} = 5V, TA = +25^{\circ}C$) |
| Accuracy of current (Unified calibration) | $ \Delta I /I$ | - | - | ± 5 | % | The constant current source is selected to output 20mA current: $(I-20) / 20$ ($V_{DD} = 5V, TA = +25^{\circ}C$) |
| Linear adjustment | $\Delta I1$ | - | ± 1 | ± 3 | %/V | The constant current source is selected to output 20mA current, When $V_{DD}=5V$, the current is $I1$, When $V_{DD}=4.5V$, the current is $I2$: $((I1-I2)/I1)*2$ ($TA = +25^{\circ}C$) |
| Load regulation | $\Delta I2$ | - | ± 1 | ± 3 | %/V | When the constant current source is selected to output 20mA current, the current is $I1$ at $V_{DD}-1$ and the current is $I2$ at voltage $V_{DD}-2$: $(I1-I2)/I1$ ($V_{DD} = 4.5 - 5.5V, TA = +25^{\circ}C$) |
| Temperature drift | | - | - | ± 7 | % | ($TA = -40^{\circ}C$ to $+125^{\circ}C$) |

LCD Electrical characteristics ($V_{DD} = 2.0 - 5.5V, GND = 0V, TA = +25^{\circ}C$, Unless otherwise stated)

| Parameter | Symbol | Min | Typical | Max | Unit | Conditions |
|----------------------|------------|-----|---------|-----|---------|--|
| LCD Working current1 | I_{LCD1} | - | 1.5 | 2.5 | μA | Traditional LCD mode, $V_{DD}=5V$, Bias resistor sum is 2.4M, $VOL[2:0] = 001$ (LCD panel not included) |
| LCD Working current2 | I_{LCD2} | - | 3 | 4 | μA | Fast charge mode, $V_{DD}=5V$, Bias resistor sum is 2.4M, $VOL[2:0] = 001$, 1/16 LCD com period, $VOL[2:0] = 001$ (LCD panel not included) |
| LCD Working current3 | I_{LCD3} | - | - | 3.3 | μA | $V_{DD}=5V, VLCD=V_{DD}, 1/8 DUTY, 1/4 BIAS$, All the dots are on (no external LCD screen), $R=RH$ |



SH30F9/SA0 Series

| | | | | | | |
|---|-----------------|---|-----|---|----|---------------------------------------|
| Total value of LCD divider resistance (high-resistance) | RH | - | 2.4 | - | MΩ | |
| Total value of LCD divider resistance (low-resistance) | RL | - | 30 | - | KΩ | |
| LCD output resistor | R _{ON} | - | 10 | - | KΩ | COM0-7, SEG0-35, V _{DD} = 5V |

High-speed 12BIT analog-to-digital converter electrical characteristics 1 (1LSB = V_{DD}/4096)

The conversion rate is up to 1MSPS. (V_{DD} = 2.7V~5.5V, GND = 0V, T_A = +25°C, unless otherwise stated)

| Parameter | Symbol | Minimum Value | Typical Value | Maximum Value | Unit | Condition |
|--|-------------------|---------------|---------------|------------------|-----------------|--|
| Working voltage range | V _{AD} | 2.7 | 5.0 | 5.5 | V | |
| Accuracy | N _R | - | 12 | - | bit | V _{REF} = 5.0V |
| A/D input voltage | V _{AIN} | GND | - | V _{REF} | V | |
| A/D input resistance* | R _{AIN} | 0.5 | - | - | MΩ | V _{IN} = 5.0V |
| External analog reference voltage | V _{REF} | 2.5 | - | V _{DD} | V | |
| Recommended impedance of analog voltage source | Z _{AIN} | - | 2 | - | kΩ | The typical value is measured by 1MSPS conversion rate and error 2LSB, and the specific reference "chapter 25.3.10" formula. |
| | | - | - | 0.5 | MΩ | The lower the sampling rate, the greater the support impedance. The 1KSPS conversion rate and the error 1LSB are measured. |
| A/D conversion current | I _{AD} | - | 1.5 | 3 | mA | ADC module is working, V _{DD} = 5.0V |
| Differential nonlinearity error | D _{LE} | - | ±0.5 | ±2 | LSB | V _{DD} = 5.0V, V _{REF} = V _{DD} , ADC CLK = 24MHz |
| Integral nonlinearity error | I _{LE} | - | ±1 | ±3 | LSB | V _{DD} = 5.0V, V _{REF} = V _{DD} , ADC CLK = 24MHz |
| Full scale error | E _F | - | ±3 | ±5 | LSB | V _{DD} = 5.0V, V _{REF} = V _{DD} , ADC CLK = 24MHz |
| Offset error | E _Z | - | ±3 | ±7 | LSB | V _{DD} = 5.0V, V _{REF} = V _{DD} , ADC CLK = 24MHz |
| Total absolute error | E _{AD} | - | - | ±8 | LSB | V _{DD} = 5.0V, V _{REF} = V _{DD} , ADC CLK = 24MHz |
| ADC working clock | f _{ADC} | - | - | 30 | MHz | V _{DD} = 5.0V, V _{REF} = V _{DD} |
| ADC sampling time | t _{SAMP} | 0.375 | - | - | μs | V _{DD} = 5.0V, V _{REF} = V _{DD} |
| ADC conversion rate | F _{CON} | - | - | 1 | MSPS | V _{DD} = 5.0V, V _{REF} = V _{DD} |
| Total conversion time | T _{CON} | 16 | - | 159 | t _{AD} | (1 t _{AD} ~ 16 t _{AD}) + t _{GAP} + t _{COMP} |
| V _{CC} voltage | V _{CC} | 1.50 | 1.55 | 1.60 | V | V _{DD} = 5.0V |

Note:

- (1) "*" indicates that the input resistance of ADC is the input resistance of ADC itself under DC conditions.
- (2) When the external reference source impedance r_{ref} is greater than 30 Ω, it is recommended to add uF capacitor to improve the driving ability.



SH30F9/SA0 Series

ESD characteristics

| Parameter | Symbol | Test values | Unit | Condition |
|------------------|------------------|-------------|------|--|
| HBM | V _{HBM} | 4000 | V | TA = 25 °C, Conform to the JEDEC JS-001-2017 standards |
| CDM | V _{CDM} | 1000 | V | TA = 25 °C, Conform to the JEDEC JS-002-2018 standards |
| MM | V _{MM} | 200 | V | TA = 25 °C, Conform to the JESD22-A115C standards |
| Latch Up Current | I _{LU} | 200 | mA | TA = 25 °C, Conform to the JESD78E standards |

(1)Based on features, not tested in production.

EFT characteristics

| Parameter | Symbol | Level/Class | Condition |
|-----------|------------------|-------------|--|
| EFT | V _{EFT} | 4A | VDD=5.0V,TA = 25 °C, Conform to the GB/T17626.4-2008 standards |

(1)Based on features, not tested in production.



SH30F9/SA0 Series

31. Ordering Information

| Product Number | Package |
|--------------------|---------------|
| SH30F9871P/064PR | LQFP64(10X10) |
| SH30F9871P/044PR | LQFP44 |
| SH30F9071P/064PR | LQFP64(10X10) |
| SH30F9071U/048UR | TQFP48 |
| SH30F9071P/044PR | LQFP44 |
| SH30F9820P/064PR | LQFP64(10X10) |
| SH30F9820U/048UR | TQFP48 |
| SH30F9820P/044PR | LQFP44 |
| SH30F9020P/064PR | LQFP64(10X10) |
| SH30F9020S/064SR | LQFP64(7X7) |
| SH30F9020U/048UR | TQFP48 |
| SH30F9020P/044PR | LQFP44 |
| SH30F9821P/044PR | LQFP44 |
| SH30F9821P/032PR | LQFP32 |
| SH30F9021P/044PR | LQFP44 |
| SH30F9021P/032PR | LQFP32 |
| SH30F9621S/032SR | LQFP32 |
| SH30F9621Q3/032Q3Y | QFN32(5X5) |

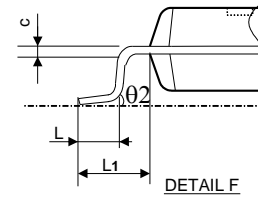
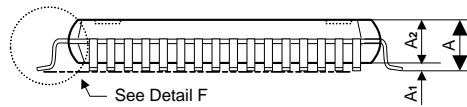
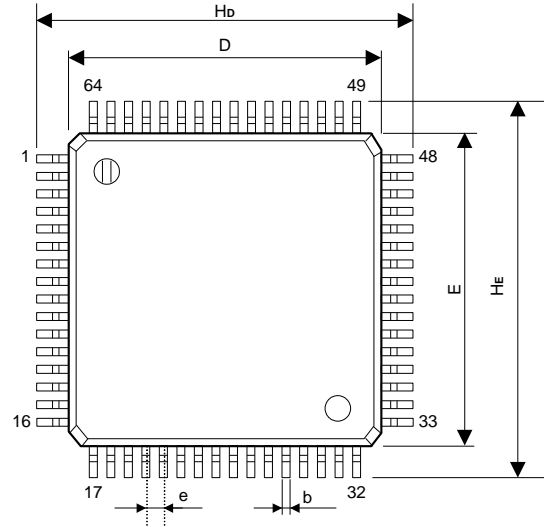


SH30F9/SA0 Series

32. Package Information

LQFP64(10X10) Outline Dimensions

Unit: inch/millimeter



| Symbol | Dimensions in inches | | Dimensions in mm | |
|----------------|----------------------|-------|------------------|-------|
| | MIN | MAX | MIN | MAX |
| A | --- | 0.063 | --- | 1.6 |
| A1 | 0.002 | 0.006 | 0.05 | 0.15 |
| A2 | 0.053 | 0.057 | 1.35 | 1.45 |
| D | 0.388 | 0.400 | 9.85 | 10.15 |
| E | 0.388 | 0.400 | 9.85 | 10.15 |
| H _D | 0.465 | 0.480 | 11.8 | 12.2 |
| H _E | 0.465 | 0.480 | 11.8 | 12.2 |
| b | 0.007 | 0.011 | 0.17 | 0.27 |
| e | 0.016BSC | | 0.50BSC | |
| c | 0.004 | 0.008 | 0.09 | 0.20 |
| L | 0.018 | 0.030 | 0.45 | 0.75 |
| L1 | 0.033 | 0.045 | 0.850 | 1.150 |
| θ ₂ | 0° | 10° | 0° | 10° |

Notice:

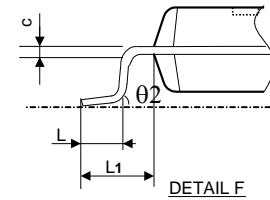
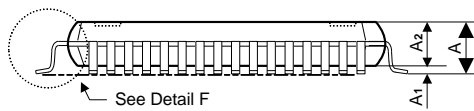
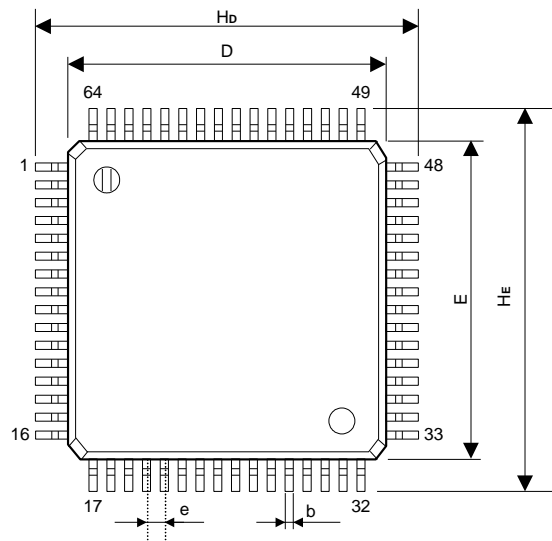
- Both package length and width do not include mold flash.
- Tolerance is ± 0.1 mm if not specified.
- Coplanarity: 0.1mm max
- Controlling dimension: mm



SH30F9/SA0 Series

LQFP64(7X7) Outline Dimensions

Unit: inch/millimeter



| Symbol | Dimensions in inches | | Dimensions in mm | |
|----------------|----------------------|-------|------------------|-------|
| | MIN | MAX | MIN | MAX |
| A | --- | 0.065 | --- | 1.65 |
| A1 | 0.000 | 0.008 | 0.01 | 0.21 |
| A2 | 0.051 | 0.059 | 1.30 | 1.50 |
| D | 0.272 | 0.280 | 6.90 | 7.10 |
| E | 0.272 | 0.280 | 6.90 | 7.10 |
| H _D | 0.346 | 0.362 | 8.80 | 9.20 |
| H _E | 0.346 | 0.362 | 8.80 | 9.20 |
| b | 0.005 | 0.010 | 0.13 | 0.25 |
| e | 0.016BSC | | 0.400BSC | |
| c | 0.004 | 0.008 | 0.090 | 0.200 |
| L | 0.016 | 0.031 | 0.400 | 0.800 |
| L1 | 0.033 | 0.045 | 0.850 | 1.150 |
| $\theta 2$ | 0° | 10° | 0° | 10° |

Notice:

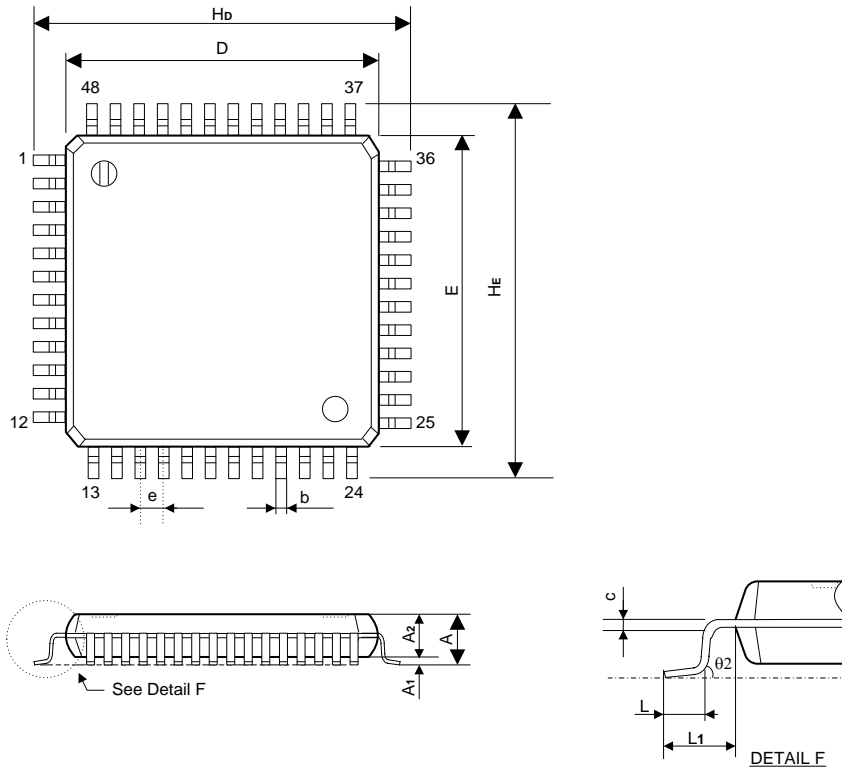
- Both package length and width do not include mold flash.
- Tolerance is ± 0.1 mm if not specified.
- Coplanarity: 0.1mm max
- Controlling dimension: mm



SH30F9/SA0 Series

TQFP48 Outline Dimensions

Unit: inch/millimeter



| Symbol | Dimensions in inches | | Dimensions in mm | |
|------------|----------------------|-------|------------------|-------|
| | MIN | MAX | MIN | MAX |
| A | --- | 0.047 | --- | 1.2 |
| A1 | 0.002 | 0.006 | 0.05 | 0.15 |
| A2 | 0.035 | 0.041 | 0.9 | 1.05 |
| D | 0.270 | 0.281 | 6.85 | 7.15 |
| E | 0.270 | 0.281 | 6.85 | 7.15 |
| H_b | 0.346 | 0.362 | 8.8 | 9.2 |
| H_E | 0.346 | 0.362 | 8.8 | 9.2 |
| b | 0.005 | 0.011 | 0.15 | 0.27 |
| e | 0.020 TYP | | 0.500 TYP | |
| c | 0.004 | 0.008 | 0.090 | 0.200 |
| L | 0.018 | 0.030 | 0.45 | 0.75 |
| L1 | 0.033 | 0.045 | 0.85 | 1.15 |
| θ_2 | 0° | 10° | 0° | 10° |

Notice:

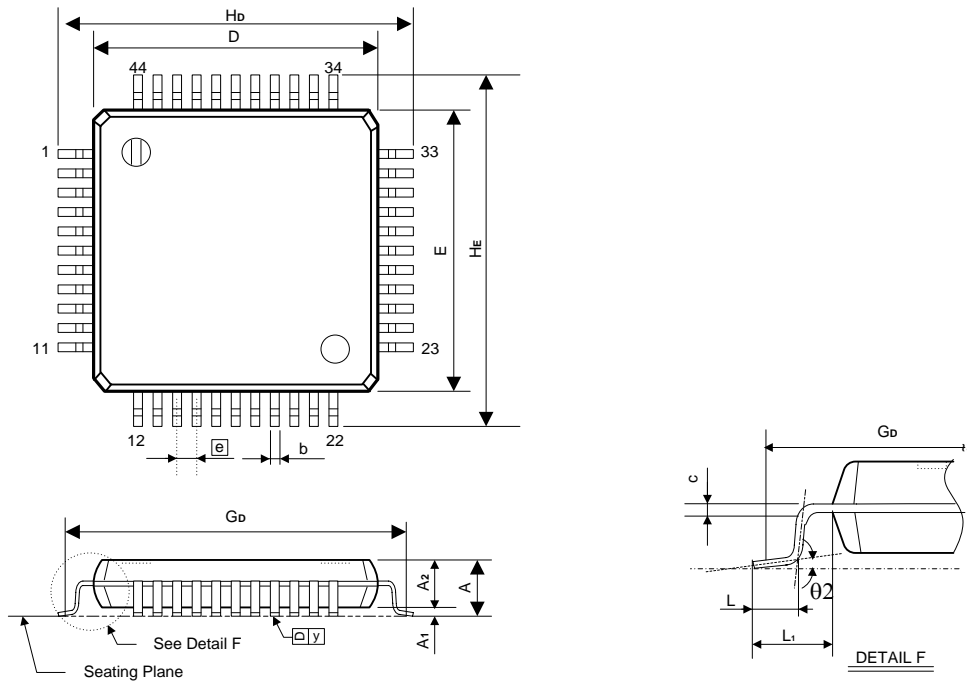
1. Both package length and width do not include mold flash.
2. Tolerance is ± 0.1 mm if not specified.
3. Coplanarity: 0.1mm max
4. Controlling dimension: mm



SH30F9/SA0 Series

LQFP44 Outline Dimensions

Unit: inch/millimeter



| Symbol | Dimensions in inches | | Dimensions in mm | |
|----------------|----------------------|-------|------------------|-------|
| | MIN | MAX | MIN | MAX |
| A | 0.057 | 0.065 | 1.45 | 1.65 |
| A1 | 0.000 | 0.001 | 0.01 | 0.21 |
| A2 | 0.051 | 0.059 | 1.3 | 1.5 |
| D | 0.388 | 0.400 | 9.85 | 10.15 |
| E | 0.388 | 0.400 | 9.85 | 10.15 |
| H _b | 0.465 | 0.480 | 11.8 | 12.2 |
| H _E | 0.465 | 0.480 | 11.8 | 12.2 |
| b | 0.010 | 0.018 | 0.25 | 0.45 |
| e | 0.031 TYP | | 0.8 TYP | |
| c | 0.004 | 0.008 | 0.09 | 0.20 |
| L | 0.017 | 0.031 | 0.42 | 0.78 |
| L1 | 0.037 | 0.045 | 0.95 | 1.15 |
| θ ₂ | 0° | 10° | 0° | 10° |

Notice:

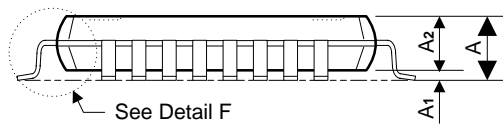
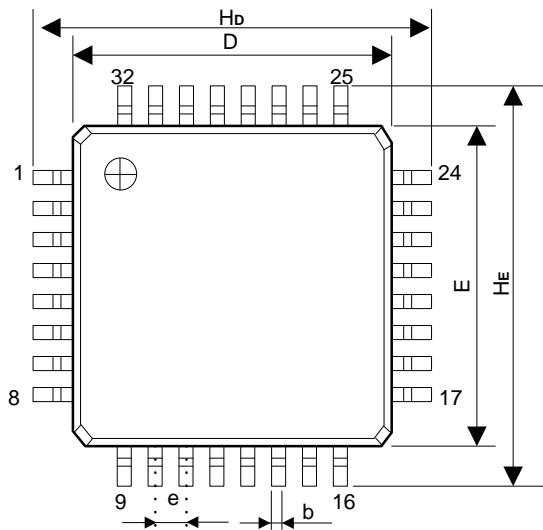
- Both package length and width do not include mold flash.
- Tolerance is $\pm 0.1\text{mm}$ if not specified.
- Coplanarity: 0.1mm max
- Controlling dimension: mm



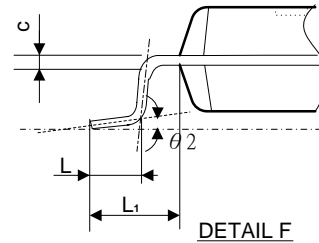
SH30F9/SA0 Series

LQFP32 Outline Dimensions

Unit: inch/millimeter



See Detail F



DETAIL F

| Symbol | Dimensions in inches | | Dimensions in mm | |
|------------|----------------------|-------|------------------|------|
| | MIN | MAX | MIN | MAX |
| A | 0.057 | 0.065 | 1.45 | 1.65 |
| A1 | 0.000 | 0.008 | 0.01 | 0.21 |
| A2 | 0.051 | 0.059 | 1.30 | 1.50 |
| D | 0.268 | 0.281 | 6.80 | 7.15 |
| E | 0.268 | 0.281 | 6.80 | 7.15 |
| H_d | 0.346 | 0.362 | 8.80 | 9.20 |
| H_E | 0.346 | 0.362 | 8.80 | 9.20 |
| b | 0.010 | 0.018 | 0.25 | 0.45 |
| e | 0.031 TYP | | 0.8TYP | |
| c | 0.004 | 0.009 | 0.09 | 0.22 |
| L | 0.016 | 0.031 | 0.40 | 0.78 |
| L_1 | 0.035 | 0.043 | 0.90 | 1.10 |
| θ_2 | 0° | 10° | 0° | 10° |

Notice:

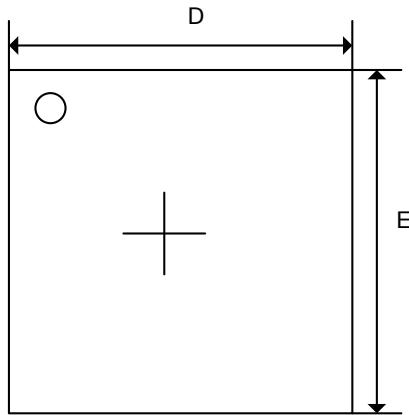
- Both package length and width do not include mold flash.
- Tolerance is $\pm 0.1\text{mm}$ if not specified.
- Coplanarity: 0.1mm max
- Controlling dimension: mm



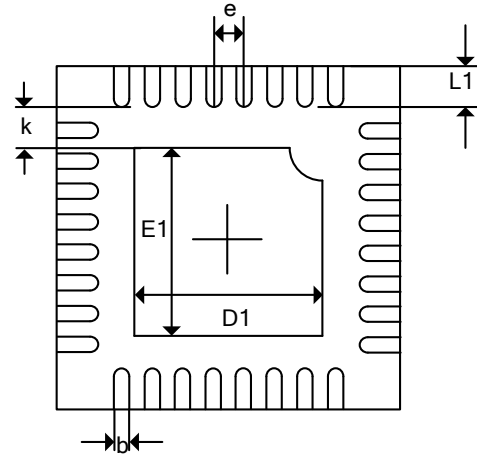
SH30F9/SA0 Series

QFN 32L-D (5 X 5) (P0.50 T 0.75) Outline Dimensions

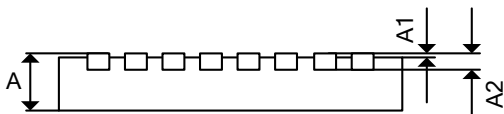
Unit: inch/millimeter



Top View



Bottom View



Side View

| Symbol | Dimensions in mm | |
|--------|------------------|------|
| | MIN | MAX |
| A | 0.70 | 0.80 |
| A1 | 0 | 0.05 |
| A2 | 0.20REF | |
| D | 5.00BSC | |
| E | 5.00BSC | |
| D1 | 3.40 | 3.60 |
| E1 | 3.40 | 3.60 |
| k | 0.20 | --- |
| b | 0.18 | 0.30 |
| e | 0.50BSC | |
| L1 | 0.35 | 0.45 |



SH30F9/SA0 Series

33. Specification Change Record

| Version | Record | Data |
|---------|---|-----------|
| 2.2 | <ol style="list-style-type: none">1. Update SH30F9071U/048UR pins diagram and product information table2. Add limit parameter: Internal protection diode current of pins3. Add HCLK configurable frequencies and corresponding LAGENCY [1:0], PRFTEN optional configuration table4. Update clerical errors | May. 2024 |
| 2.1 | <ol style="list-style-type: none">1. Update the specification architecture2. Add SH30F9621 product information3. Update clerical errors | Oct. 2023 |
| 2.0 | <ol style="list-style-type: none">1. SH30F9071 added TQFP48 packaging2. SH30F9020 added LQFP64(7X7) packaging3. Update the power on preheating time4. Add the definition of V_{POR} and delete the definition of auxiliary LVR5. Added V_{CC} voltage accuracy electrical characteristics6. Update clerical errors | Aug. 2023 |
| 1.0 | Original | Nov. 2022 |



IMPORTANT NOTICE

This manual is the property of Sino Wealth Electronic Ltd. and its affiliates ("Company"). This manual, including all of the Company's products ("Products") described herein, is owned by the Company according to relevant laws or treaties. The Company reserves all rights under such laws and treaties, and does not grant you any license to use its patents, copyrights, trademarks and other intellectual property rights.

Any technical information in this manual, including functional descriptions and schematic diagrams, shall not be construed as a license to use or exercise any intellectual property rights. The names and brands (if any) of the relevant third parties are quoted in this manual are the property of their respective owners, for identification purposes only.

The Company does not make any warranty, expressed or implied, about this manual or any Product, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. The Company shall not be liable for any use of any of Product described in this manual out of its specifications or the Company's standard. Except for the special Products specified in the applicable agreement, the Products are designed, developed and manufactured for general commercial, industrial, personal and/or domestic applications only. In particular, the Products are prohibited to be used in following fields, such as military, national defense, nuclear energy, medical treatment and others that may cause personal injury, death or environmental damage. Users shall take all actions to ensure that the Products are used and sold in accordance with applicable laws and regulations.

The failure or malfunction of semiconductor products may occur. Users of the Products should therefore give thorough consideration to safety design, including redundancy, fire-prevention measures and etc., to prevent any accidents, fire or community damage that may ensue. Especially, the reference application circuit is not guaranteed to be suitable for the specific application of mass production.

In the event that users breach the above statement, the Company shall not be liable in whole or in part, and users shall hereby release the Company and its -agents and/or distributors from any claim, damage or other liability; At the same time, users shall indemnify the Company and its agents and/or distributors against all claims, costs, damages and other liabilities, including claims for personal injury or death, arising out of any unintended use of the Product.

The information in this manual is related to the Product only. The Company reserves the right to make changes, modifications or improvements to this manual, the Products and services described herein at any time without prior notice. Users are advised to contact our sales department before purchase.

The Company has the final right to interpret this manual.



| | |
|---|-----------|
| 34. Directory | |
| 1. Features | 2 |
| 2. Overview | 3 |
| 3. Block Diagram | 4 |
| 4. Product information | 5 |
| 5. Definitions Pins | 7 |
| <i>LQFP 64 pins (10X10) package</i> | 7 |
| <i>LQFP 64 pins (7X7) package</i> | 8 |
| <i>TQFP 48 pins package</i> | 9 |
| <i>TQFP 48 pins package</i> | 10 |
| <i>LQFP 44 pins package</i> | 11 |
| <i>LQFP 44 pins package</i> | 12 |
| <i>LQFP 44 pins package</i> | 13 |
| <i>LQFP 32 pins package</i> | 14 |
| <i>QFN 32 pins (5X5) package</i> | 15 |
| <i>Alternate Function Mapping Table</i> | 16 |
| 6. Descriptions of Pins | 18 |
| 7. Core | 20 |
| 7.1 ARM® Cortex™-M0+ Core | 20 |
| 7.2 Register Group | 20 |
| 8. Memories and Bus Architecture | 21 |
| 8.1 System Architecture | 21 |
| 8.2 Memory Organization | 22 |
| 8.2.1 Register Image | 22 |
| 8.2.2 Data Memory (SRAM) | 23 |
| 8.2.3 Program Memory | 23 |
| 8.3 Flash Program Memory | 24 |
| 8.3.1 Features | 24 |
| 8.3.2 Introduction | 24 |
| 8.3.3 Flash Memory Function Description | 25 |
| 8.4 Startup Configuration (BootLoader) | 35 |
| 8.5 In System Programming (ISP) | 38 |
| 8.6 Flash Access Acceleration Control | 38 |
| 8.7 Flash Block Division Function | 38 |
| 8.8 Registers | 40 |
| 8.8.1 Flash access control register (FLASH_ACR) | 40 |



SH30F9/SA0 Series

| | | |
|------------|--|-----------|
| 8.8.2 | Flash main program storage area unlock control register (FLASH_MKYR)..... | 41 |
| 8.8.3 | EEPROM-like storage area unlock control register (FLASH_E2KYR)..... | 42 |
| 8.8.4 | Flash status and clear register (FLASH_SR)..... | 42 |
| 8.8.5 | Flash control register (FLASH_CR)..... | 44 |
| 8.8.6 | Flash control register 1 (FLASH_CR1)..... | 45 |
| 8.8.7 | Flash read protection register (FLASH_RPR)..... | 45 |
| 8.8.8 | Flash write protection register (FLASH_WRPR)..... | 46 |
| 8.8.9 | Flash operation timer count register (FLASH_CNTR)..... | 46 |
| 8.8.10 | Flash operation timer upper limit (FLASH_UPCNTR)..... | 46 |
| 8.8.11 | Flash operation timer start register (FLASH_CNTCR)..... | 47 |
| 8.8.12 | Flash special information storage area unlock control register (FLASH_IKYR)..... | 47 |
| 8.8.13 | Memory map select register (FLASH_MEMRMP)..... | 48 |
| 9. | Power Control (PWR) | 49 |
| 9.1 | Warm-up Timer..... | 49 |
| 9.2 | Brownout Detection (BOD)..... | 50 |
| 9.3 | Low-voltage Reset (LVR)..... | 51 |
| 9.4 | Low Power Consumption Management..... | 51 |
| 9.4.1 | Sleep Mode..... | 52 |
| 9.4.2 | Stop Mode..... | 53 |
| 9.4.3 | Debug in stop mode..... | 55 |
| 9.5 | Registers..... | 56 |
| 9.5.1 | Power control register (SYSCFG_PWRCR)..... | 56 |
| 9.5.2 | Power Status Register (SYSCFG_PWRSR)..... | 57 |
| 10. | Reset and Clock Control (RCC) | 58 |
| 10.1 | Reset..... | 58 |
| 10.1.1 | System Rest..... | 58 |
| 10.1.2 | Power Reset..... | 59 |
| 10.2 | Clock..... | 60 |
| 10.2.1 | HSE Clock..... | 61 |
| 10.2.2 | LSE Clock..... | 62 |
| 10.2.3 | HSI Clock..... | 62 |
| 10.2.4 | LSI Clock..... | 63 |
| 10.2.5 | WDT clock..... | 63 |
| 10.2.6 | PLL clock..... | 63 |
| 10.2.7 | System Clock (SYSCLK) Selection..... | 64 |
| 10.2.8 | Clock Safety Monitoring (CSM)..... | 64 |
| 10.2.9 | System Tick Timer (SysTick)..... | 67 |
| 10.3 | Registers..... | 68 |
| 10.3.1 | Clock Control Register (RCC_CR)..... | 68 |
| 10.3.2 | Clock Configuration Register (RCC_CFGR)..... | 70 |
| 10.3.3 | Clock interrupt enable register (RCC_CIENR)..... | 72 |
| 10.3.4 | Clock Interrupt Status Register (RCC_CISTR)..... | 72 |
| 10.3.5 | Clock Interrupt Flag Clear Register (RCC_CICLR)..... | 73 |
| 10.3.6 | AHB Peripheral Reset Register (RCC_AHBSTR)..... | 74 |
| 10.3.7 | APB1 Peripheral Reset Register (RCC_APB1RSTR)..... | 75 |
| 10.3.8 | APB0 Peripheral Reset Register (RCC_APB0RSTR)..... | 76 |
| 10.3.9 | AHB Peripheral Reset Register (RCC_AHBENR)..... | 77 |
| 10.3.10 | APB1 Peripheral Reset Register (RCC_APB1ENR)..... | 78 |
| 10.3.11 | APB0 Peripheral Reset Register (RCC_APB0ENR)..... | 79 |



| | | |
|------------|--|------------|
| 10.3.12 | RESET Status Register (RCC_RSTSTR) | 80 |
| 10.3.13 | RESET Status Clear Register (RCC_RSTCLR) | 81 |
| 10.3.14 | The RCC configuration lock register (RCC_RCCLOCK) | 82 |
| 10.3.15 | Selection of feedback resistance of external high frequency crystal oscillator / ceramic oscillator driving circuit (HSERFB) | 82 |
| 11. | Watchdog Timer (WDT) | 83 |
| 11.1 | Independent Watchdog Timer (IWDT) | 83 |
| 11.2 | Window Watchdog Timer (WWDT) | 83 |
| 11.3 | Registers | 85 |
| 11.3.1 | Independent Watchdog Control Register (IWDT_CR) | 85 |
| 11.3.2 | Independent Watchdog Feeding Register (IWDT_CLR) | 86 |
| 11.3.3 | WWDT Control Register (WWDT_CR) | 86 |
| 11.3.4 | WWDT Status Register (WWDT_SR) | 87 |
| 11.3.5 | WWDT Feeding Register (WWDT_CLR) | 88 |
| 11.3.6 | WWDT Window Register (WWDT_WTR) | 88 |
| 12. | Code and Data Checking | 89 |
| 12.1 | CRC Calculation Unit | 89 |
| 12.1.1 | Introduction | 89 |
| 12.1.2 | Main Features | 89 |
| 12.1.3 | CRC function description | 89 |
| 12.1.4 | CRC Register | 94 |
| 12.2 | SRAM Detection Module (RAMBIST) | 96 |
| 12.2.1 | SRAM Detection Module Introduction | 96 |
| 12.2.2 | SRAM detection module feature description | 96 |
| 12.2.3 | RAMBIST Registers | 100 |
| 13. | General-Purpose I/O (GPIO) | 102 |
| 13.1 | Introduction | 102 |
| 13.2 | Main Features | 102 |
| 13.3 | Function Description | 103 |
| 13.3.1 | Default Configuration of I/O Pins | 103 |
| 13.3.2 | I/O Control Register | 104 |
| 13.3.3 | I/O Data Register | 104 |
| 13.3.4 | I/O Bit Set And Bit Reset | 104 |
| 13.3.5 | I/O Locking Mechanism | 104 |
| 13.3.6 | I/O Configured as External Interrupt/Wake-up Line | 105 |
| 13.3.7 | I/O Alternate Function | 105 |
| 13.3.8 | GPIO Input Function | 106 |
| 13.3.9 | GPIO Output Function | 107 |
| 13.3.10 | Digital Alternate Function | 107 |
| 13.3.11 | Analog Alternate Function | 108 |
| 13.3.12 | Basic application of I/O pins | 109 |
| 13.4 | Registers | 111 |
| 13.4.1 | Port x Mode Configuration register (GPIOx_MODER) | 111 |
| 13.4.2 | Port x Input Data Register (GPIOx_IDR) | 112 |
| 13.4.3 | Port x Output Data Register (GPIOx_ODR) | 113 |
| 13.4.4 | Port x Reset/Set Register (GPIOx_BSRR) | 113 |
| 13.4.5 | Port x Configuration Lock Register (GPIOx_LCKR) | 114 |
| 13.4.6 | Port x input enable configuration register (GPIOx_IE) | 115 |



| | | |
|------------|---|------------|
| 13.4.7 | Port x output type configuration register (GPIOx_OTYPER) | 117 |
| 13.4.8 | Port x output driver configuration register (GPIOx_ODRVR) | 118 |
| 13.4.9 | Port x Pull-up Configuration register (GPIOx_PUPDR)..... | 119 |
| 13.4.10 | Port x Partial Pin TTL Level Selection Register (GPIOx_TTLEN) | 121 |
| 13.4.11 | Port x Alternate Function Register Low Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (GPIOx_AFRL) | 122 |
| 13.4.12 | Port x Alternate Function Register High Bits(Refer to "Alternate Function Mapping Table", which will not be specified in REG) (GPIOx_AFRH)..... | 125 |
| 14. | System Configuration Module (SYSCFG)..... | 129 |
| 14.1 | <i>System Configuration Correlation</i> | 129 |
| 14.1.1 | SWD and XTAL pin alternate function definition | 129 |
| 14.1.2 | Debug Interface Register..... | 129 |
| 14.2 | <i>Registers</i> | 130 |
| 14.2.1 | System Configuration Register (SYSCFG_SAFR)..... | 130 |
| 14.2.2 | SRAM Sector Lock Control Register (SYSCFG_SRAMLOCK) | 131 |
| 14.2.3 | Debug Interface Control Register (SYSCFG_DBGCR) | 131 |
| 14.2.4 | GPIO module mount bus configuration register (SYSCFG_GPIOBCR)..... | 133 |
| 15. | Interrupts and events..... | 134 |
| 15.1 | <i>Nestable Interrupt Vector Controller (NVIC)</i> | 134 |
| 15.1.1 | Introduction | 134 |
| 15.1.2 | Main Features | 134 |
| 15.1.3 | Interrupts Summary | 134 |
| 15.2 | <i>External Interrupt/Event Controller (EXTI)</i> | 137 |
| 15.2.1 | Main Features..... | 137 |
| 15.2.2 | Wake-up Event Management | 138 |
| 15.2.3 | Function Description | 138 |
| 15.2.4 | External Interrupt/Event Line Mapping | 140 |
| 15.3 | <i>Registers</i> | 141 |
| 15.3.1 | Interrupt Mask Register (EXTI_IMR)..... | 141 |
| 15.3.2 | Event Enable Register (EXTI_EMR)..... | 141 |
| 15.3.3 | Trigger Mode Selection Register (EXTI_TMSR) | 142 |
| 15.3.4 | Rising Edge/High Level Trigger Mode Selection Register (EXTI_RTISR) | 142 |
| 15.3.5 | Falling Edge/Low Level Trigger Mode Selection Register (EXTI_FTISR)..... | 143 |
| 15.3.6 | Software Trigger Interrupt Register (EXTI_SWIER) | 143 |
| 15.3.7 | Pending Register (EXTI_PR)..... | 144 |
| 15.3.8 | External Interrupt Configuration Low Register (EXTI_CFGL)..... | 144 |
| 15.3.9 | External Interrupt Configuration High Register (EXTI_CFGH)..... | 146 |
| 15.3.10 | EXTI Sample Control Low Register (EXTI_SAMP) | 148 |
| 15.3.11 | DMA Enable Register (EXTI_DMR)..... | 149 |
| 16. | DMA Controller | 150 |
| 16.1 | <i>Introduction</i> | 150 |
| 16.2 | <i>Main Features</i> | 150 |
| 16.3 | <i>Function Description</i> | 151 |
| 16.3.1 | DMA Processing | 151 |
| 16.3.2 | Arbitrator..... | 152 |
| 16.3.3 | DMA Channel | 153 |
| 16.3.4 | Source and Destination Data Transfer Format | 156 |
| 16.3.5 | Error Management..... | 158 |
| 16.3.6 | DMA Interrupt | 158 |



| | | |
|------------|--|------------|
| 16.3.7 | DMA Request Image..... | 159 |
| 16.3.8 | Precautions | 159 |
| 16.4 | <i>Registers</i> | 161 |
| 16.4.1 | DMA Interrupt Status Register (DMA_IFSR) | 161 |
| 16.4.2 | DMA Interrupt Flag Clear Register (DMA_IFCR) | 162 |
| 16.4.3 | DMA Control Status Register (DMA_CSR) | 163 |
| 16.4.4 | DMA Channel n Configuration Register (DMA_CCRn)(n=0..3) | 164 |
| 16.4.5 | DMA Channel n Transmitting Quantity Register (DMA_NPKTn)(n=0..3)..... | 165 |
| 16.4.6 | DMA Channel n Transmitting Count Register (DMA_CPKTn)(n=0..3)..... | 166 |
| 16.4.7 | DMA Channel n Source Address Register (DMA_SARn)(n=0..3)..... | 166 |
| 16.4.8 | DMA Channel n Target Address Register (DMA_DARn)(n=0..3)..... | 167 |
| 17. | LED Driver (constant current) | 168 |
| 17.1 | <i>Overview</i> | 168 |
| 17.2 | <i>SEG width control</i> | 171 |
| 17.3 | <i>Continuous scan SEG output control (if and only if AUEN@LED_AUCR = 1)</i> | 171 |
| 17.4 | <i>Matters attention</i> | 172 |
| 17.4.1 | LED output status in stop mode | 172 |
| 17.5 | <i>Registers</i> | 173 |
| 17.5.1 | LED COM control register (LED_COMCR) | 173 |
| 17.5.2 | LED interrupt flag and clear register (LED_FR)..... | 174 |
| 17.5.3 | LED constant current switch control register (LED_CCCR) | 175 |
| 17.5.4 | LED data register (LED_CCDR) | 175 |
| 17.5.5 | LED continuous scan control register (LED_AUCR) | 178 |
| 17.5.6 | LED mode control register (LED_SEGMOD) | 178 |
| 17.5.7 | LED SEG width control register (LED_SDR0) | 181 |
| 17.5.8 | LED SEG width control register (LED_SDR1) | 181 |
| 17.5.9 | LED SEG width control register (LED_SDR2) | 182 |
| 17.5.10 | LED SEG width control register (LED_SDR3)..... | 183 |
| 17.5.11 | Continuous scan of the SEG output control register (LED_SOR0)..... | 183 |
| 17.5.12 | Continuous scan of the SEG output control register (LED_SOR1)..... | 184 |
| 17.5.13 | Continuous scan of the SEG output control register (LED_SOR2)..... | 184 |
| 17.5.14 | Continuous scan of the SEG output control register (LED_SOR3)..... | 185 |
| 18. | LCD driver (Resistance type) | 186 |
| 18.1 | <i>Introduction</i> | 186 |
| 18.2 | <i>LCD RAM CONFIGURATION</i> | 189 |
| 18.3 | <i>Registers</i> | 190 |
| 18.3.1 | LCD control register (LCD_CR)..... | 190 |
| 18.3.2 | LCD data register (LCD_BUFS0)..... | 192 |
| 18.3.3 | LCD data register (LCD_BUFS1)..... | 192 |
| 18.3.4 | LCD data register (LCD_BUFS2)..... | 193 |
| 18.3.5 | LCD data register (LCD_BUFS3)..... | 193 |
| 18.3.6 | LCD data register (LCD_BUFS4)..... | 194 |
| 18.3.7 | LCD data register (LCD_BUFS5)..... | 194 |
| 18.3.8 | LCD data register (LCD_BUFS6)..... | 195 |
| 18.3.9 | LCD data register (LCD_BUFS7)..... | 195 |
| 18.3.10 | LCD data register (LCD_BUFS8) | 196 |
| 18.3.11 | LCD data register (LCD_BUFS9) | 196 |
| 19. | Basic Timer (TIM0~TIM3) | 197 |



SH30F9/SA0 Series

| | | |
|------------|--|------------|
| 19.1 | Introduction | 197 |
| 19.2 | Main Features..... | 197 |
| 19.3 | Function Description | 197 |
| 19.3.1 | Operation of Counter/Timer | 197 |
| 19.3.2 | External input clock source filtering | 198 |
| 19.4 | Registers | 199 |
| 19.4.1 | TIMx Control Register (TIMx_CR)..... | 199 |
| 19.4.2 | TIMx Count Register (TIMx_TCNT) | 200 |
| 19.4.3 | TIMx period register (TIMx_TPR) | 200 |
| 19.4.4 | TIMx Pre-division Register (TIMx_PSQ)..... | 201 |
| 19.4.5 | TIMx Interrupt Flag and Clear Register (TIMx_TIMINTF) | 201 |
| 19.4.6 | TIM3 Control Register (TIMER32_CR)..... | 202 |
| 19.4.7 | TIM3 Count Register (TIMER32_TCNT) | 203 |
| 19.4.8 | TIM3 period register (TIMER32_TPR) | 203 |
| 19.4.9 | TIM3 Pre-division Register (TIMER32_PSQ)..... | 203 |
| 19.4.10 | TIM3 Interrupt Flag and Clear Register (TIMER32_TIMINTF) | 204 |
| 20. | 16 bit Pulse Width Modulation (PWM0/1/2/3)..... | 205 |
| 20.1 | Feature..... | 205 |
| 20.2 | PWM Enable Control..... | 205 |
| 20.3 | PWM protection register..... | 205 |
| 20.4 | 16 bit PWM timer..... | 205 |
| 20.5 | PWMxB complementary output (x = 0,1,2,3) | 206 |
| 20.6 | Dead Time..... | 206 |
| 20.7 | Registers | 208 |
| 20.7.1 | PWMx control register (PWMx_CR)..... | 208 |
| 20.7.2 | PWMx protection register (PWMx_PWMLOCK)..... | 209 |
| 20.7.3 | PWMx cycle control register (PWMx_PWMPR) | 209 |
| 20.7.4 | PWMx duty cycle control register (PWMx_PWMDR)..... | 210 |
| 20.7.5 | PWMx dead time control register (PWMx_PWMDTR) | 210 |
| 20.7.6 | PWMx interrupt flag and clear register (PWMx_PWMINTF)..... | 210 |
| 21. | Programmable counter array (PCA, PCA0~PCA3) | 212 |
| 21.1 | Introduction | 212 |
| 21.2 | Feature..... | 212 |
| 21.3 | Function description..... | 212 |
| 21.3.1 | Counting unit..... | 212 |
| 21.3.2 | Counter clock source..... | 213 |
| 21.4 | Working mode..... | 214 |
| 21.4.1 | Input Capture Mode | 214 |
| 21.4.2 | Compare match output | 215 |
| 21.4.3 | Frequency output mode..... | 216 |
| 21.4.4 | PWM output | 217 |
| 21.5 | Cascade operation..... | 222 |
| 21.6 | Interrupt..... | 222 |
| 21.7 | Immediate update of registers..... | 223 |



SH30F9/SA0 Series

| | | |
|------------|---|------------|
| 21.8 | Trigger DMA | 223 |
| 21.9 | Register Lock | 223 |
| 21.10 | matters needing attention..... | 223 |
| 21.10.1 | PCA output status in debug mode..... | 223 |
| 21.10.2 | PCA output status in stop mode..... | 223 |
| 21.11 | Registers | 224 |
| 21.11.1 | PCAx enable register (PCAx_CR) | 224 |
| 21.11.2 | PCAx mode register (PCAx_CFGR) | 224 |
| 21.11.3 | PCAx flag register (PCAx_SR) | 226 |
| 21.11.4 | PCAx forced output control register (PCAx_FORCE)..... | 227 |
| 21.11.5 | PCAx counter register (PCAx_CNT) | 228 |
| 21.11.6 | Prescaler (PCAx_PSC)..... | 228 |
| 21.11.7 | PCAx counting period value (PCAx_PR)..... | 229 |
| 21.11.8 | PCAx capture/compare module 0 control register (PCAx_CCMR0) | 229 |
| 21.11.9 | PCAx capture/compare module 1 control register (PCAx_CCMR1) | 230 |
| 21.11.10 | PCAx compare/capture module 0 register (PCAx_CCR0) | 231 |
| 21.11.11 | PCAx compare/capture module 1 register (PCAx_CCR1) | 232 |
| 21.11.12 | PCAx lock register (PCAx_LCKR) | 232 |
| 22. | Universal Asynchronous Receiver/Transmitter (UART) | 233 |
| 22.1 | Introduction | 233 |
| 22.2 | Main Feature | 233 |
| 22.3 | Function Description | 233 |
| 22.3.1 | UART Work Mode..... | 233 |
| 22.3.2 | Fine-tunable Baud Rate..... | 240 |
| 22.3.3 | Multiprocessor Communication | 241 |
| 22.3.4 | Frame Error Detection..... | 241 |
| 22.3.5 | Parity | 242 |
| 22.3.6 | LIN (Local Area Internet) bus mode | 242 |
| 22.3.7 | Use DMA communication..... | 243 |
| 22.3.8 | Interrupt control..... | 243 |
| 22.4 | UART registers | 244 |
| 22.4.1 | UARTx Interrupt Flag Register (UARTx_FR) | 244 |
| 22.4.2 | UARTx Transmit Data Register (UARTx_TDR) | 245 |
| 22.4.3 | UARTx receive data register (UARTx_RDR) | 246 |
| 22.4.4 | UARTx Address Configure Register (UARTx_ADDR)..... | 246 |
| 22.4.5 | UARTx Baud Rate Configure Register (UARTx_BRT)..... | 246 |
| 22.4.6 | UARTx Control Register (UARTx_CR)..... | 247 |
| 23. | Serial Peripheral Interface (SPI) | 249 |
| 23.1 | Introduction | 249 |
| 23.2 | Main Features..... | 249 |
| 23.3 | Function Description | 250 |
| 23.3.1 | Signal Description..... | 250 |
| 23.3.2 | Baud Rate | 250 |
| 23.3.3 | Working Mode | 251 |
| 23.4 | SPI registers..... | 258 |
| 23.4.1 | SPI interrupt flag register (SPIx_FR) | 258 |
| 23.4.2 | SPI transmit data register (SPIx_TDR) | 259 |
| 23.4.3 | SPI receive data register (SPIx_RDR) | 259 |



| | | |
|------------|---|------------|
| 23.4.4 | SPI control register (SPIx_CR) | 260 |
| 24. | Two-Wire Serial Interface (TWI) | 262 |
| 24.1 | Introduction | 262 |
| 24.2 | Main Features..... | 262 |
| 24.3 | Function Description | 262 |
| 24.3.1 | Bus Signal Timing..... | 262 |
| 24.3.2 | Clock Synchronization | 264 |
| 24.3.3 | Bus Arbitration..... | 265 |
| 24.3.4 | Detailed Structure of TWI Communication Module | 266 |
| 24.3.5 | TWI Operation Modes | 267 |
| 24.3.6 | Supporting Fast Mode 400kbps..... | 278 |
| 24.3.7 | Interrupt..... | 278 |
| 24.3.8 | TWI Communication Service Program Development..... | 278 |
| 24.4 | TWI Registers..... | 280 |
| 24.4.1 | TWI Interrupt Flag Register (TWI_FR) | 280 |
| 24.4.2 | TWI State Register (TWI_STAT) | 281 |
| 24.4.3 | TWI High Level Timeout Detection Configure Register (TWI_HOC)..... | 281 |
| 24.4.4 | TWI Bit Rate Configure Register (TWI_BRT) | 282 |
| 24.4.5 | TWI data register (TWI_DR) | 282 |
| 24.4.6 | TWI Address Configure Register (TWI_ADDR) | 282 |
| 24.4.7 | TWI Control Register (TWI_CR) | 283 |
| 25. | Analog to Digital Converter | 285 |
| 25.1 | Introduction | 285 |
| 25.2 | Main Features..... | 286 |
| 25.3 | Function Description | 286 |
| 25.3.1 | Single Conversion Method..... | 286 |
| 25.3.2 | Intermittent Conversion Method | 287 |
| 25.3.3 | Continuous Conversion Method..... | 287 |
| 25.3.4 | Other Function Description | 288 |
| 25.3.5 | Comparison function..... | 289 |
| 25.3.6 | ADC conversion time setting | 290 |
| 25.3.7 | ADC module reference voltage setting | 290 |
| 25.3.8 | ADC channel and IO port function settings | 290 |
| 25.3.9 | DMA request..... | 290 |
| 25.3.10 | Requirements for sensor output impedance during AD conversion | 291 |
| 25.3.11 | Matters needing attention | 291 |
| 25.4 | Registers | 292 |
| 25.4.1 | ADC control register 1 (ADC_ADCON1) | 292 |
| 25.4.2 | ADC control register 2 (ADC_ADCON2) | 293 |
| 25.4.3 | ADC sampling conversion channel pointer register (ADC_ADPCH) | 295 |
| 25.4.4 | ADC result register n (ADC_ADDRn)(n=0..7)..... | 295 |
| 25.4.5 | ADC compare control register (ADC_ADCMPCON) | 296 |
| 25.4.6 | ADC upper limit compare register (ADC_ADDGT) | 296 |
| 25.4.7 | ADC lower limit compare register (ADC_ADDLT) | 297 |
| 25.4.8 | ADC channel select register 0 (ADC_SEQCHSEL0)..... | 297 |
| 25.4.9 | ADC channel select register 1 (ADC_SEQCHSEL1)..... | 301 |
| 25.4.10 | ADC channel GAP control register (ADC_ADGAPON) | 306 |
| 25.4.11 | ADC interrupt flag and clear register (ADC_ADINTF) | 306 |
| 26. | Touch Key..... | 308 |



SH30F9/SA0 Series

| | |
|---|------------|
| 27. Device Electronic Signature | 309 |
| 27.1 Unique device ID register (96 bits)..... | 309 |
| 28. Debug interface | 310 |
| 28.1 Main Feature | 310 |
| 28.2 SW Debug Port (Serial Wire) | 310 |
| 28.3 SW debug port pin..... | 310 |
| 28.4 Using serial wire and releasing the unused debug pin as GPIOs..... | 310 |
| 28.5 MCU debug module..... | 311 |
| 28.5.1 Debug Support for Low Power Mode | 311 |
| 28.5.2 Module Debug Support with Timing Function | 311 |
| 28.5.3 Debug MCU configuration register..... | 311 |
| 29. Customer Options | 312 |
| 30. Electrical characteristics | 313 |
| Absolute rating *..... | 313 |
| DC electrical characteristics(V _{DD} = 2.0– 5.5V, GND = 0V, T _A = 25°C, unless otherwise indicated)..... | 313 |
| Flash features (V _{DD} = 2.0 - 5.5V, GND = 0V, T _A = 25 °C, unless otherwise stated)..... | 314 |
| Brownout detection (BOD) electrical characteristics (V _{DD} =2.0-5.5V, GND = 0V, T _A = +25 °C, unless otherwise stated) | 315 |
| Low voltage reset (LVR) electrical characteristics (V _{DD} =2.0-5.5V, GND = 0V, T _A = +25°C, unless otherwise stated) | 315 |
| Power on reset electrical characteristics (V _{DD} = 2.0 - 5.5V,GND = 0V,T _A = 25°C, unless otherwise stated)..... | 316 |
| AC electrical characteristics (V _{DD} = 2.0 - 5.5V, GND = 0V, T _A = +25°C, unless otherwise stated) | 316 |
| PLL characteristics (V _{DD} = 2.0 - 5.5V, GND = 0V, T _A = +25°C, unless otherwise stated) | 317 |
| GPIO Electrical characteristics (V _{DD} =2.0-5.5V, GND= 0V, T _A =25°C, unless otherwise indicated)..... | 317 |
| LED current source accuracy..... | 318 |
| LCD Electrical characteristics (V _{DD} = 2.0 - 5.5V,GND = 0V,T _A = +25°C,Unless otherwise stated) | 318 |
| High-speed 12BIT analog-to-digital converter electrical characteristics 1 (1LSB = V _{DD} /4096)..... | 319 |
| ESD characteristics..... | 320 |
| EFT characteristics..... | 320 |
| 31. Ordering Information..... | 321 |
| 32. Package Information..... | 322 |
| 33. Specification Change Record | 328 |
| 34. Directory | 330 |