# HT502X
# User manual

Tel: 021-51035886
Fax: 021-50277833
Email: sales@hitrendtech.com
Web: http://www.hitrendtech.com

# Version updating instructions

| Version Num | Modication time | Content changed |
|---|---|---|
| V1.0 | 2014-11-12 | First edition |
| V1.1 | 2018-3-12 | 1.Correct the error an example of timing maximum cycle formula at the timer section<br>2. Correct DC PRFCFG[1:0] defined<br>3. Correct the errors of other chapters |
| V1.11 | 2018-4-2 | 1. Add write protect register list of gpio in 2.5 chapter<br>2. Add gpio chapter write protect register note |
| V1.12 | 2018-7-25 | 1.Delete the change of V1.11<br>2.Correct SCANIN1 to JTAGWDTEN in 1.4 chapter<br>3.Delete the bit2 of the JTAGSTA register |
| V1.13 | 2018-8-21 | 1.Correct the condition of HFCONST calibration in 22.4 chapter<br>2.Correct RANDSTR bit15 trng_en to RANDEN |
| V1.14 | 2019-6-27 | 1.Unify UdetCyc and OVCyc for OVCyc<br>2. Add HT5029B pin diagram |
| V1.15 | 2019-7-29 | 1. Add HT5029 pin diagram<br>2. Add 48pin Package |
| V1.16 | 2019-8-8 | 1、Correct RTC module about the describe of DFi,23bit to 21bit<br>2、Add the consumption range of hold and sleep mode<br>3、Add tbs open describe |
| V1.17 | 2020-4-21 | 1. Correct JTAGWDTEN to SCANIN1 in 1.4 chapter<br>2. Delete HT5029B pin diagram |
| V1.18 | 2021-3-12 | 1.Fixed section format |

# Contents

# 1 HT502X overview

## 1.1 Brief introduction

HT502X is a low-power and high-performance single-phase energy measuring SoC chip to provide solution to single-phase multifunction、energy stealing prevention and highly integrated chips. Which integrated internal 32-bit ARM kernel、256K flash、32K SRAM, hardware EMU modules which support missing phase to prevent electricity power stealing、 precise RTC block with self-temperature compensating function and LCD driving function.

## 1.2 Features

### 1.2.1 General features

Operating voltage range：2.2V～5.5V
Operating temperature range：-45℃～85℃
Unleaded green packaging：LQFP100： HT5023
LQFP80：HT5025
LQFP64：HT5027

### 1.2.2 Processor and peripherals

- Based on ARM 32-bit Cortex-M0 CPU Core designing，support Thumb-2 instruction set,including nested vectored interrupt controller NVIC and extensible debugging technology
- Memory resource：256K Flash+1Kbytes Information Block, 32K SRAM
- High-speed system clock: use PLL frequency multiplication
  Insert waiting CPU’s maximum operation frequency 39.32MHz

No waiting CPU's maximum operating frequency 19.66MHz

- System Low-power:lowest-power in Hold Mode 3.7uA
  lowest-power in sleep mode 2.9uA
- External power detection function, external system source VSYS and battery input VBAT can switch internally
- Internal 8.8MHz HRC, 32KHz LRC, multiple clock detection function
- Support SW debug protocol
- RTC block：
  External 32K low frequency crystal oscillator, integrated resistor and capacitor need for oscillator temperature self-compensation：With RTC temperature curve digital compensation factor, no need for users to involve in RTC compensation at all temperature range
  Auxiliary RTC：System can switch to internal low frequency RC to get the counting time while external OSC oscillator does not run
- High-accuracy temperature sensor TPS：temperature sensor's coherence is less than ±1 ℃ at the range of -45 ℃~85 ℃
- LCD：support display of LCD 4COM，6COM，8COM，SEG interface can support 54 section at most.
- Support 6 branches of UART at most, of which two support 7816 protocol function
- WDT block is unable to shut down to ensure reliable operation system in normal mode
- Internal hardware-based AES/GHASH encryption/decryption algorithm
- Internal ECC encryption/decryption algorithm, support ECC256/224/192
- Internal DMA function
- Internal keys scanning function, available for 1*4, 2*4, 3*4, 4*4
- Support I2C，SPI，TIMER CC & PWM

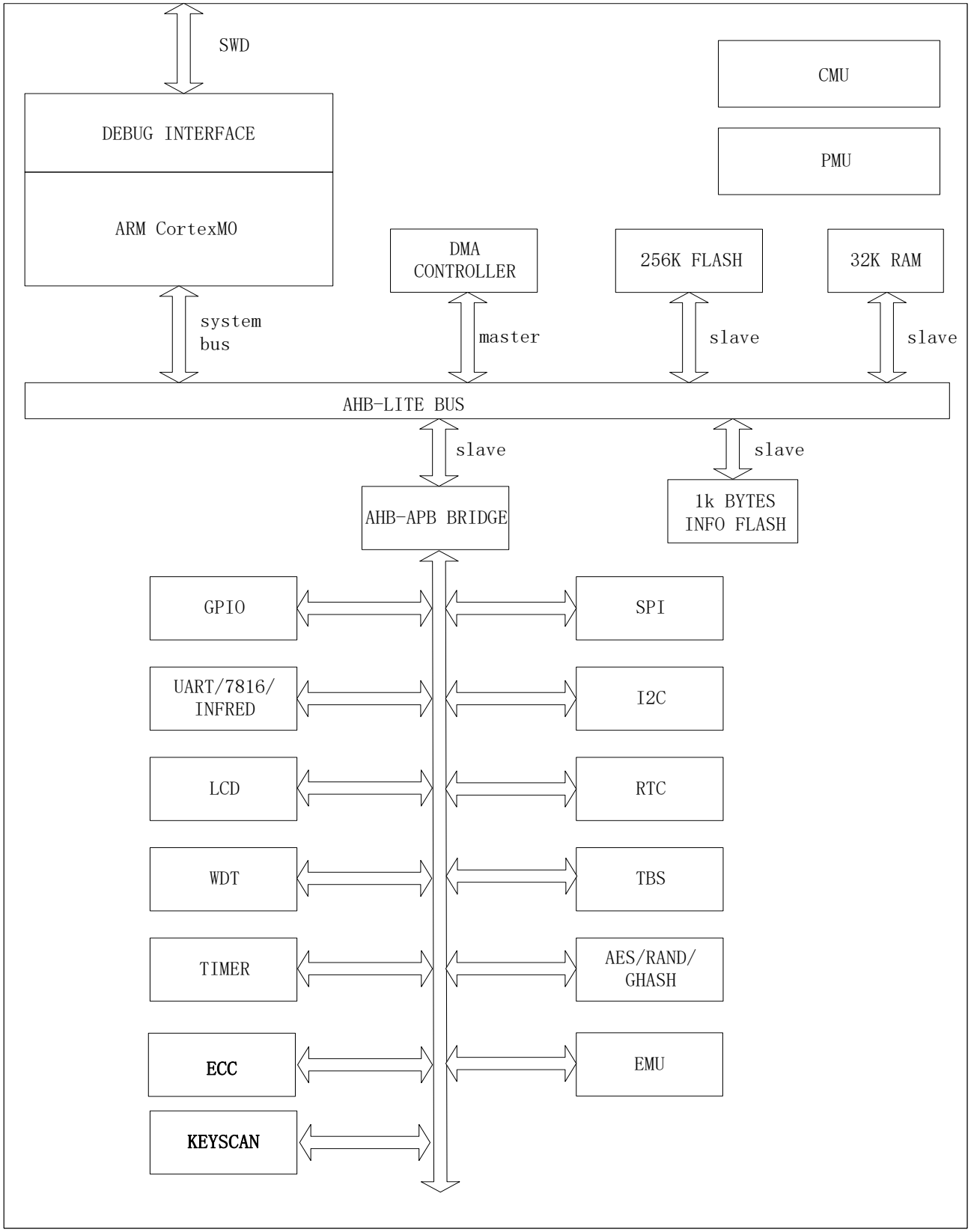
### 1.2.3 Energy measurement

- Active power measurement error is less than 0.1%, dynamic range is more than 5000:1, support IEC62053-21、IEC62053-22、IEC62053-23
- Three ∑-ΔADC，can output parameters of two measuring channels at same time, support flexible anti-stealing power function
- Provide active power, reactive power, apparent power,active energy,reactive power
- support active power, reactive power, apparent power pulse output,pulse counting register is available
- high precise effective value,frequency,ADC waveform data can be measured
- support SAG、PEAK function
- multiple ways of energy accumulation is accessible
- gain error,phase error software correction table is available
- anti-creep function of two measuring channels are supported respectively
- support EMU low power mode：EMU low-power operating frequency can be configured to be 204.8 KHz or 32KHz, support earth line and live line electricity power stealing prevention measuring, power is less than 650uA while continuously measuring in this mode. DC measuring is support
- support DC measuring
- support single-phase 3 wires measuring

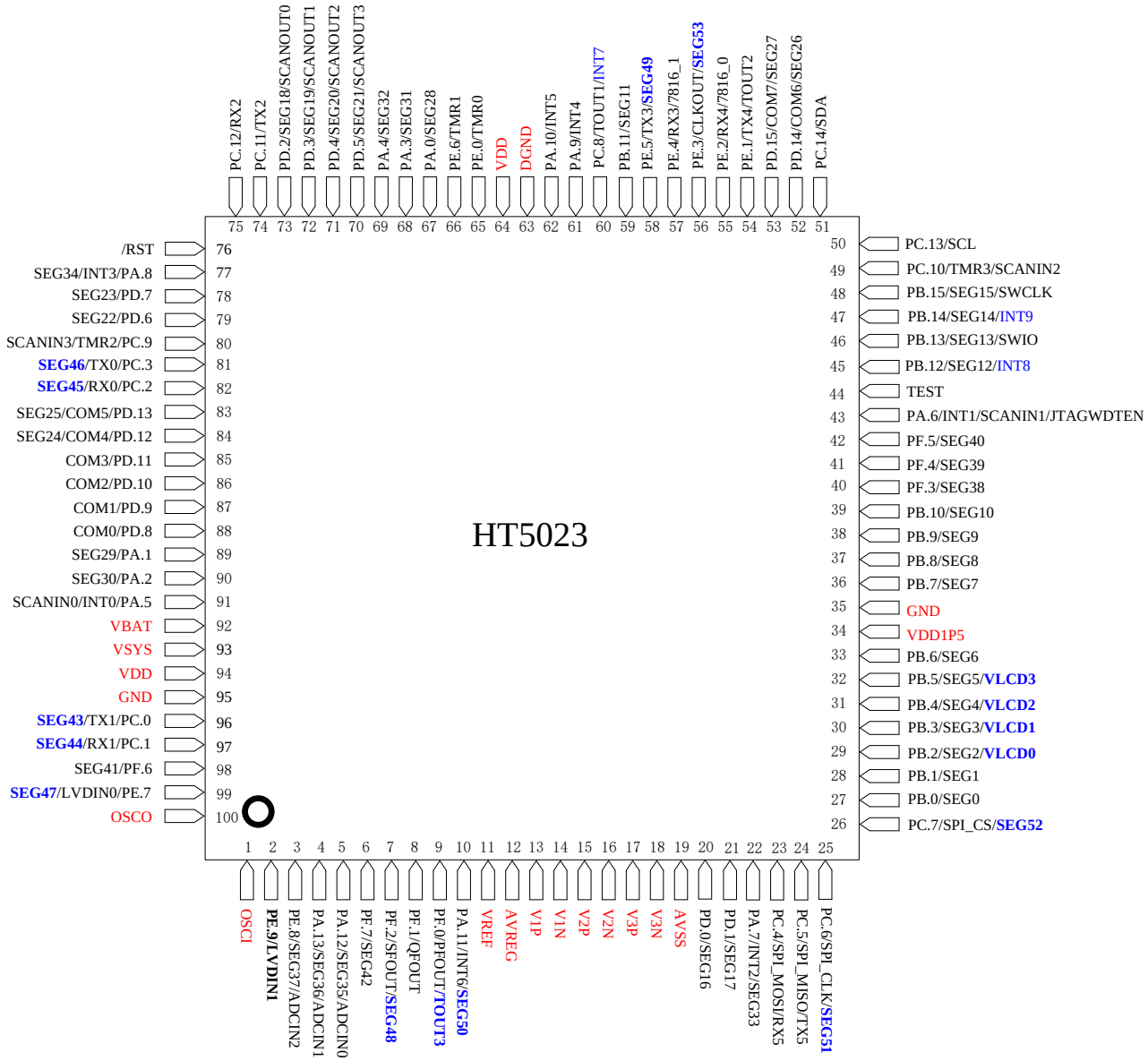
- internal reference voltage：1.2V，temperature coefficient ±10ppm/℃

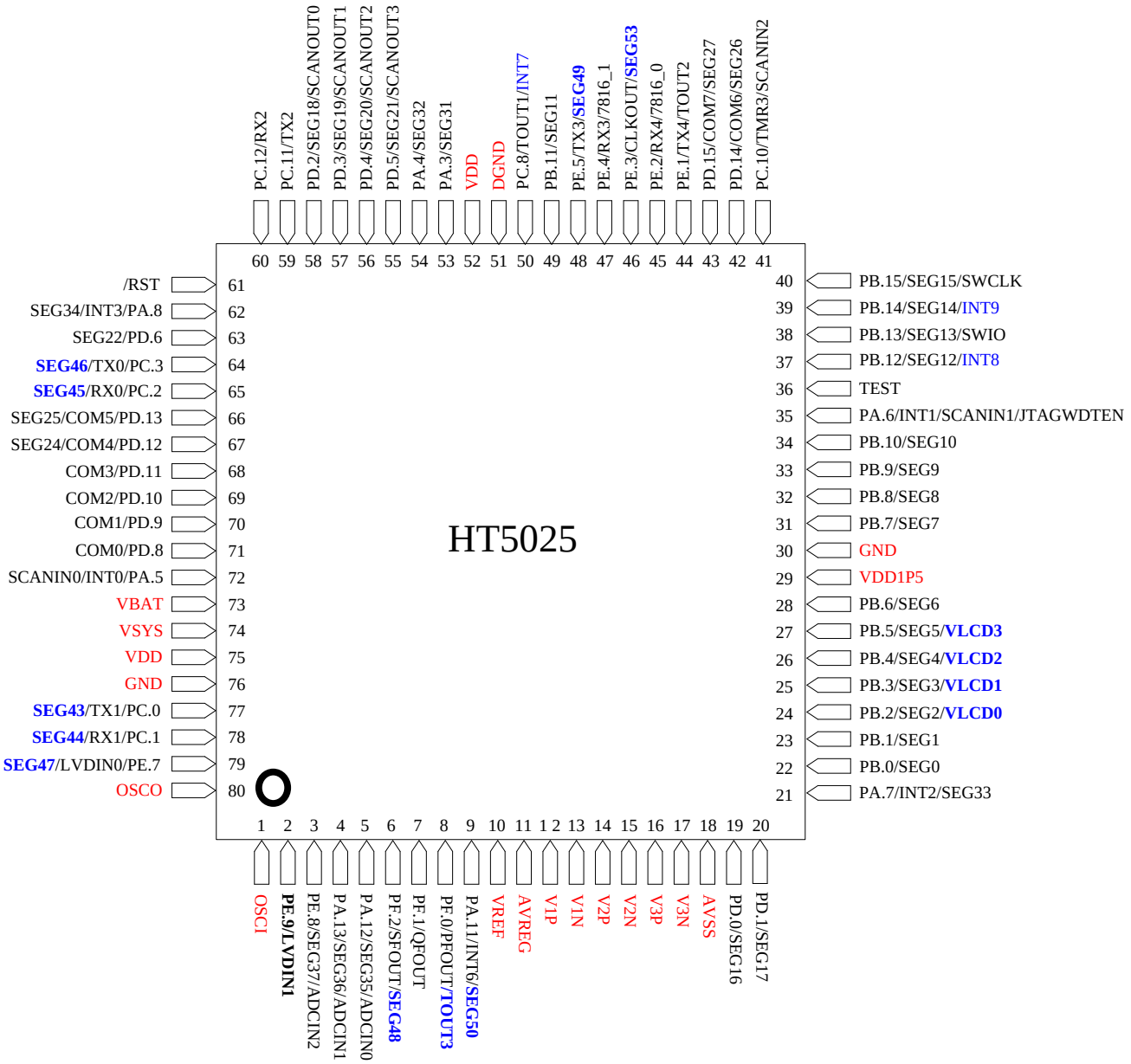
## 1.3 Abbreviations

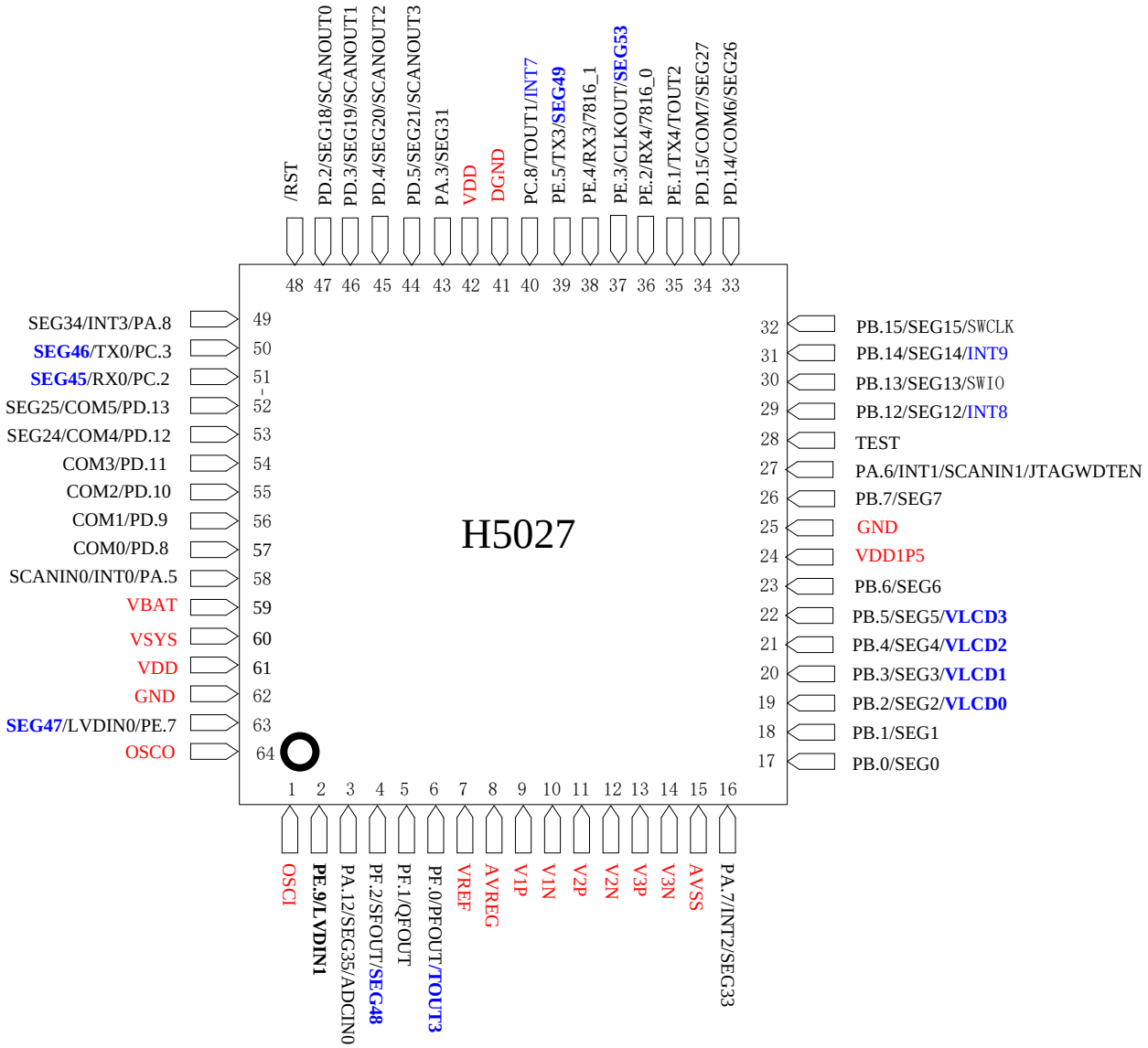
| Abbr. | English |
|---|---|
| WDT | Watch Dog Timer |
| GPIO | General Purpose IO |
| TBS | Temperature Battery Sensor |
| LVD | Low Voltage Detect |
| POR | Power On Reset |
| BOR | Brown Out Reset |
| WKR | Wake-up Reset |
| EMU | Energy Measurement Unit |
| PMU | Power Management Unit |
| CMU | Clock Management Unit |
| RTC | Real Time Clock |
| RSRV | Reserved |

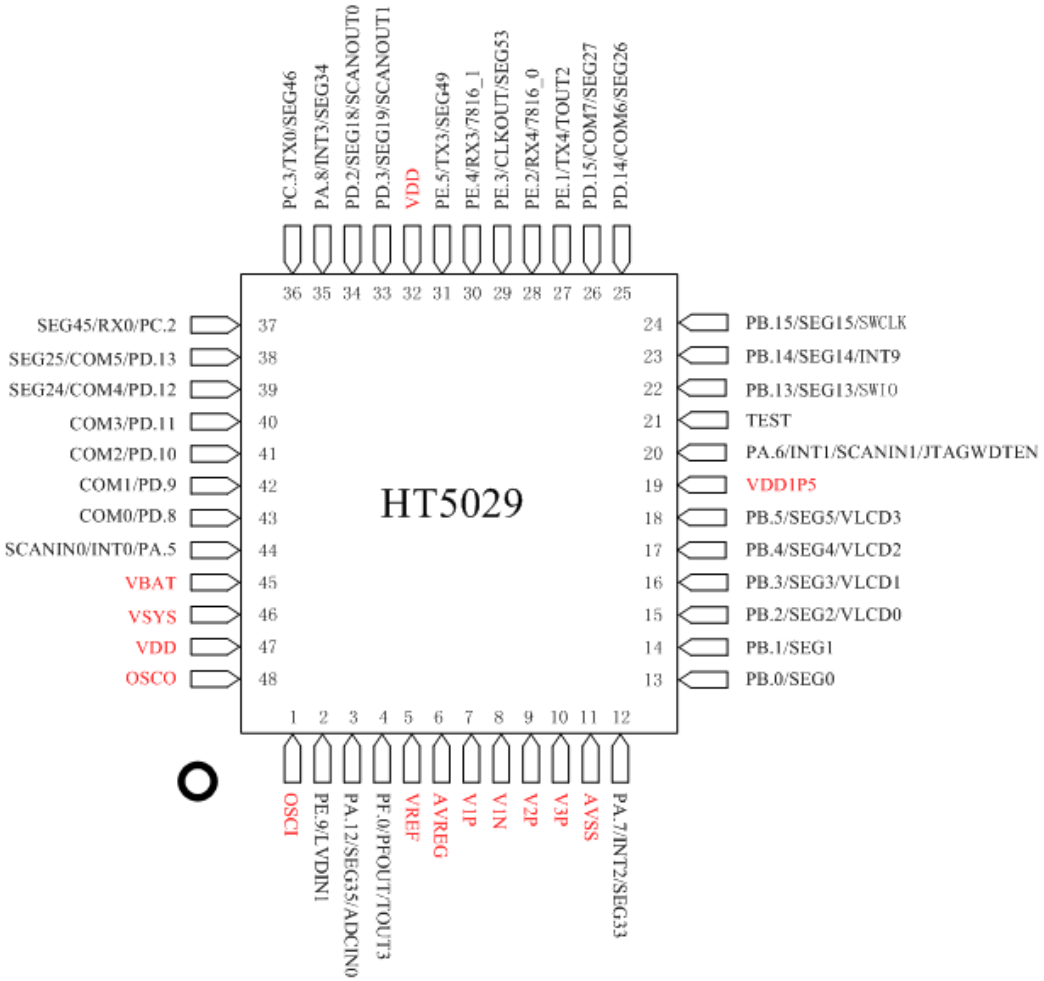


# 1.4Pin configuration









## 1.5 Pin definition

| 100 PIN | 80 PIN | 64 PIN | 48 PIN（HT5029） | Identification (IOCFG=0 AFCFG=0) | Pin type | First multiplex function (IOCFG=1 AFCFG=0) | Second multiplex function (IOCFG=1 AFCFG=1) | Pin description |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | OSCI | I | | | 32K clock input |
| 2 | 2 | 2 | 2 | PE.9 | I/O | LVDIN1 | LVDIN1 | GPIO\External power detection1 |
| 3 | 3 | | | PE.8 | I/O | SEG37 | ADCIN2 | GPIO\LCD_seg\External voltage signal sampling input 2，filter 2us |
| 4 | 4 | | | PA.13 | I/O | SEG36 | ADCIN1 | GPIO\LCD_seg\External voltage signal sampling input 1，filter 2us |
| 5 | 5 | 3 | 3 | PA.12 | I/O | SEG35 | ADCIN0 | GPIO\LCD_seg\External voltage signal sampling input 0，filter 2us |
| 6 | | | | PF.7 | I/O | SEG42 | SEG42 | GPIO\LCD Segment output drive |
| 7 | 6 | 4 | | PF.2 | I/O | SFOUT | SEG48 | Default Apparent power pulse output,5mA drive capacity, can be configured as other types of pulse output through EMCON \GPIO\LCD-seg |
| 8 | 7 | 5 | | PF.1 | I/O | QFOUT | QFOUT | Default reactive power pulse output,5mA drive capacity, can be configured as other types of pulse output through EMCON\GPIO |
| 9 | 8 | 6 | 4 | PF.0 | I/O | PFOUT | TOUT3 | Default active power pulse output,5mA drive capacity, can be configured as other types of pulse output through EMCON \GPIO\RTC second pulse output |
| 10 | 9 | | | PA.11 | I/O | INT6 | SEG50 | External interrupt\GPIO, I/O filter 2us,output drive capacity 5mA |
| 11 | 10 | 7 | 5 | VREF | OUT | | | 1.2V measuring reference voltage output, need external filter capacity 0.1uF&1uF |
| 12 | 11 | 8 | 6 | AVCC | P | | | 2.8V internal analog power output, need external filter |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | capacity 0.1uF&10uF |
| 13 | 12 | 9 | 7 | V1P | I | | | Current channel 1 analog signal input(positive), internal ESD protect, maximum signal 800mVp |
| 14 | 13 | 10 | 8 | V1N | I | | | Current channel 1 analog signal input(negative), internal ESD protect, maximum signal 800mVp |
| 15 | 14 | 11 | 9 | V2P | I | | | Current channel 2 analog signal input(positive), internal ESD protect, maximum signal 800mVp |
| 16 | 15 | 12 | | V2N | I | | | Current channel 2 analog signal input(negative), internal ESD protect, maximum signal 800mVp |
| 17 | 16 | 13 | 10 | V3P | I | | | voltage channel analog signal input(positive), internal ESD protect, maximum signal 800mVp |
| 18 | 17 | 14 | | V3N | I | | | voltage channel analog signal input(negative), internal ESD protect, maximum signal 800mVp |
| 19 | 18 | 15 | 11 | AVSS | G | | | Analog ground |
| 20 | 19 | | | PD.0 | I/O | SEG16 | | GPIO\LCD SEG,<br>Output drive 5mA |
| 21 | 20 | | | PD.1 | I/O | SEG17 | SEG17 | GPIO\LCD Segment |
| 22 | 21 | 16 | 12 | PA.7 | I/O | INT2 | SEG33 | GPIO\External interrupt\LCD Seg, Filter 2us，output drive 30mA |
| 23 | | | | PC.4 | I/O | SPI_MOSI | RX5 | GPIO\SPI data line\UART communication port, Filter 2us，output drive 5mA |
| 24 | | | | PC.5 | I/O | SPI_MISO | TX5 | GPIO\SPI data line\UART communication port, output drive 5mA |
| 25 | | | | PC.6 | I/O | SPI_CLK | SPI_CLK | GPIO\SPI clock line, output drive 5mA |
| 26 | | | | PC.7 | I/O | SPI_CS | SPI_CS | GPIO\SPI control line, output drive 5mA |
| 27 | 22 | 17 | 13 | PB.0 | I/O | SEG0 | SEG0 | GPIOLCD Segment |
| 28 | 23 | 18 | 14 | PB.1 | I/O | SEG1 | SEG1 | GPIO\LCD Segment |
| 29 | 24 | 19 | 15 | PB.2 | I/O | SEG2 | VLCD0 | GPIO\LCD Segment\LCD resistor divider mode bias voltage output (external 0.47uF filter capacitor) |
| 30 | 25 | 20 | 16 | PB.3 | I/O | SEG3 | VLCD1 | GPIO\LCD Segment\LCD resistor |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | divider mode bias voltage output (external 0.47uF filter capacitor) |
| 31 | 26 | 21 | 17 | PB.4 | I/O | SEG4 | VLCD2 | GPIO\LCD Segment\LCD resistor divider mode bias voltage output (external 0.47uF filter capacitor) |
| 32 | 27 | 22 | 18 | PB.5 | I/O | SEG5 | VLCD3 | GPIO\LCD Segment\LCD resistor divider mode bias voltage output (external 0.47uF filter capacitor) |
| 33 | 28 | 23 | | PB.6 | I/O | SEG6 | SEG6 | GPIO\LCD Segment |
| 34 | 29 | 24 | 19 | VDD1P5 | P | | | Internal 1.5V output, need external 0.1uF filter capacity |
| 35 | 30 | 25 | | GND | G | | | GND |
| 36 | 31 | 26 | | PB.7 | I/O | SEG7 | SEG7 | GPIO\LCD Segment |
| 37 | 32 | | | PB.8 | I/O | SEG8 | SEG8 | GPIO\LCD Segment |
| 38 | 33 | | | PB.9 | I/O | SEG9 | SEG9 | GPIO\LCD Segment |
| 39 | 34 | | | PB.10 | I/O | SEG10 | SEG10 | GPIO\LCD Segment |
| 40 | | | | PF.3 | I/O | SEG38 | SEG38 | GPIO\LCD Segment |
| 41 | | | | PF.4 | I/O | SEG39 | SEG39 | GPIO\LCD Segment |
| 42 | | | | PF.5 | I/O | SEG40 | SEG40 | GPIO\LCD Segment |
| 43 | 35 | 27 | 20 | PA.6 | I/O | INT1 | SCANIN1 | High power GPIO (output 30mA), external interrupt (2us filter), and keyboard scan line scan input line. Simulation control interface JTAGWDTEN: when TEST, PIN and this PIN are low at the same time, the system enters the simulation mode. |
| 44 | 36 | 28 | 21 | TEST | I | | | Test pin, filter 2us. When this PIN is lower than JTAGWDTEN, the system goes into the test mode. |
| 45 | 37 | 29 | | PB.12 | I/O | SEG12 | INT8 | GPIO\LCD Segment\External interrupt |
| 46 | 38 | 30 | 22 | PB.13 | I/O | SEG13 | SW-IO | GPIO\LCD Segment\SW-IO |
| 47 | 39 | 31 | 23 | PB.14 | I/O | SEG14 | INT9 | GPIO\LCD Segment \ External interrupt |
| 48 | 40 | 32 | 24 | PB.15 | I/O | SEG15 | SW-CLK | GPIO\LCD Segment \SW-CLK |
| 49 | 41 | | | PC.10 | I/O | TMR3 | SCANIN2 | GPIO\Timer input \Output drive 5mA |
| 50 | | | | PC.13 | I/O | SCL | SCL | GPIO\IIC-CLK |
| 51 | | | | PC.14 | I/O | SDA | SDA | GPIO\IIC-IO |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 52 | 42 | 33 | 25 | PD.14 | I/O | COM6 | SEG26 | GPIO\LCD comment\Segment |
| 53 | 43 | 34 | 26 | PD.15 | I/O | COM7 | SEG27 | GPIO\LCD comment\Segment |
| 54 | 44 | 35 | 27 | PE.1 | I/O | TX4 | TOUT2 | GPIO\UART port\RTC second pulse output. Output drive 5mA |
| 55 | 45 | 36 | 28 | PE.2 | I/O | RX4 | 7816_0 | GPIO\UART port with 7816 function , Filter 2us |
| 56 | 46 | 37 | 29 | PE.3 | I/O | CLKOUT | SEG53 | GPIO\ESAM/CARD_CLK，clock out\LCD segment |
| 57 | 47 | 38 | 30 | PE.4 | I/O | RX3 | 7816_0 | GPIO\UART port with 7816 function, Filter 2us |
| 58 | 48 | 39 | 31 | PE.5 | I/O | TX3 | SEG49 | GPIO\UART port \LCD segment |
| 59 | 49 | | | PB.11 | I/O | SEG11 | SEG11 | GPIO\LCD Segment |
| 60 | 50 | 40 | | PC.8 | I/O | TOUT1 | INT7 | GPIO\Second pulse output 1\External interrupt |
| 61 | | | | PA.9 | I/O | INT4 | INT4 | GPIO\External interrupt 4, filter 2us |
| 62 | | | | PA.10 | I/O | INT5 | INT5 | GPIO\External interrupt 5, filter 2us |
| 63 | 51 | 41 | | DGND | G | | | Chip digital ground |
| 64 | 52 | 42 | 32 | VDD | P | | | Internal source output |
| 65 | | | | PE.0 | I/O | TMR0 | TMR0 | GPIO\TIMER clock in |
| 66 | | | | PE.6 | I/O | TMR1 | TMR1 | GPIO\TIMER clock in |
| 67 | | | | PA.0 | I/O | SEG28 | SEG28 | GPIO\LCD Segment |
| 68 | 53 | 43 | | PA.3 | I/O | SEG31 | SEG31 | GPIO\LCD Segment |
| 69 | 54 | | | PA.4 | I/O | SEG32 | SEG32 | GPIO\LCD Segment |
| 70 | 55 | 44 | | PD.5 | I/O | SEG21 | SCANOUT 3 | GPIO\LCD Segment \KEY board scanning. Output drive 5mA |
| 71 | 56 | 45 | | PD.4 | I/O | SEG20 | SCANOUT 2 | GPIO\LCD Segment \KEY board scanning. Output drive 5mA |
| 72 | 57 | 46 | 33 | PD.3 | I/O | SEG19 | SCANOUT 1 | GPIO\LCD Segment \ Key board scanning |
| 73 | 58 | 47 | 34 | PD.2 | I/O | SEG18 | SCANOUT 0 | GPIO\LCD Segment \ Key board scanning |
| 74 | 59 | | | PC.11 | I/O | TX2 | TX2 | GPIO\UART port, Output drive 5mA |
| 75 | 60 | | | PC.12 | I/O | RX2 | RX2 | GPIO\UART port,Filter 2us |
| 76 | 61 | 48 | | /RST | I | | | Reset signal(active low), filter 2us |
| 77 | 62 | 49 | 35 | PA.8 | I/O | INT3 | SEG34 | GPIO\External interrupt \LCD segment, Filter 2us，output |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | drive 30mA |
| 78 | | | | PD.7 | I/O | SEG23 | SEG23 | GPIO\LCD segment, output drive 5mA |
| 79 | 63 | | | PD.6 | I/O | SEG22 | SEG22 | GPIO\LCD segment, output drive 5mA |
| 80 | | | | PC.9 | I/O | TMR2 | SCANIN3 | GPIO\Timer input\ key board scanning, output drive 5mA |
| 81 | 64 | 50 | 36 | PC.3 | I/O | TX0 | SEG46 | GPIO\UART port\ LCD segment |
| 82 | 65 | 51 | 37 | PC.2 | I/O | RX0 | SEG45 | GPIO\UART port \ LCD segment, Filter 2us |
| 83 | 66 | 52 | 38 | PD.13 | I/O | COM5 | SEG25 | GPIO\LCD comment\segment |
| 84 | 67 | 53 | 39 | PD.12 | I/O | COM4 | SEG24 | GPIO\LCD comment\segment |
| 85 | 68 | 54 | 40 | PD.11 | I/O | COM3 | COM3 | GPIO\LCD comment |
| 86 | 69 | 55 | 41 | PD.10 | I/O | COM2 | COM2 | GPIO\LCD comment |
| 87 | 70 | 56 | 42 | PD.9 | I/O | COM1 | COM1 | GPIO\LCD comment |
| 88 | 71 | 57 | 43 | PD.8 | I/O | COM0 | COM0 | GPIO\LCD comment |
| 89 | | | | PA.1 | I/O | SEG29 | | GPIO\LCD segment |
| 90 | | | | PA.2 | I/O | SEG30 | | GPIO\LCD segment |
| 91 | 72 | 58 | 44 | PA.5 | I/O | INT0 | SCANIN0 | GPIO\External interrupt\ key board scanning , Filter 2us |
| 92 | 73 | 59 | 45 | VBAT | P | | | Battery input |
| 93 | 74 | 60 | 46 | VSYS | P | | | System power input |
| 94 | 75 | 61 | 47 | VDD | P | | | System power output |
| 95 | 76 | 62 | | GND | G | | | Analog ground |
| 96 | 77 | | | PC.0 | I/O | TX1 | SEG43 | GPIO\UART port\LCD segment, output drive 30mA |
| 97 | 78 | | | PC.1 | I/O | RX1 | SEG44 | GPIO\UART port\LCD segment, Filter 2us |
| 98 | | | | PF.6 | I/O | SEG41 | SEG41 | GPIO\ LCD segment, |
| 99 | 79 | 63 | | PE.7 | I/O | LVDIN0 | SEG47 | GPIO\External power detection \LCD segment |
| 100 | 80 | 64 | 48 | OSCO | O | | | Clock drive output |

Note：1．I=input；O=output；P=power；G=ground；

2．Digital output pin can be configured to support open drain.

3．Digital input pins(except RST/TEST/JTAGWDTEN who are always pulled up) can be configured to pull up.

4．PA.6 pin is a special pin, it function as JTAG_WDTEN input while TEST=0.

5. Two VDD pins should be connected externally.

6．After the PB13/PB15 power on, the default is the debug port, that is, multiplexing function 2.

7. When TEST=0 and JTAGWDTEN=0, the PB13 (SWIO) /PB15 (SWCLK) is constant to the SW debug port, the open drain function is off, and the configuration open drain is invalid.

8. After power on, PF.0 default to 1 multiplexing function of PFOUT function, PF.0 can be configured for multiplexing function 2 TOUT3 (second pulse output); PF.1/PF.2 power on the default for multiplexing function 1, and the default is QFOUT/SFOUT function respectively.

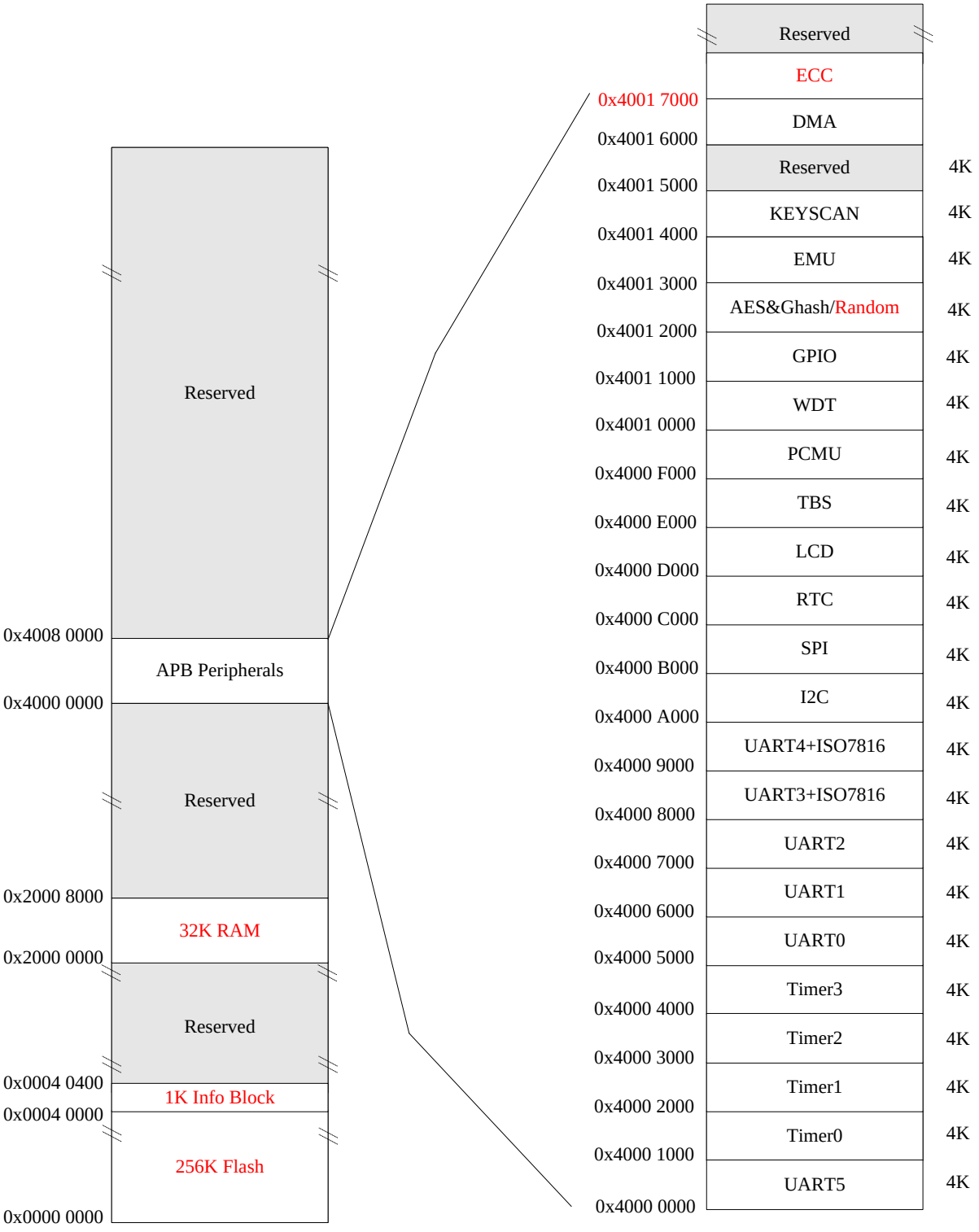
# 2 Memory block

## 2.1 Brief introduction

HT502X contains internal programmable reliable 256K Flash + 1K bytes Information Block and 32K RAM。Of which Flash has read-protect function and can be read, written, page erased and all erased, features of Flash are discussed below:

Flash byte read time：40ns
Flash byte write time：20us（max）
Flash page erase time：2ms（max）
Flash all erase time：10ms（max）
Code Flash page size：1K bytes/page
Information Block：1K bytes/page,1page
Rewrite cycle：100,000 times
Data preserving time：20 years （min）
Operating temperature:-45 ℃ ~+105 ℃

## 2.2Memory map



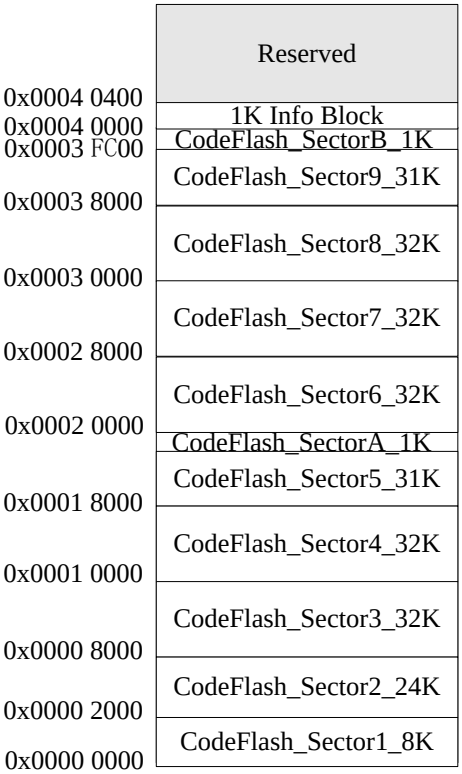
# 2.3 Flash operation

## 2.3.1 Read-protect ion of Flash

Flash has read-protection function and can protect user's code from being read.
write a value not 0FFH to 0000FC1H of Flash to enable read-protect function, data in 256K Flash cannot be read while data in Information Block can be read.（it is necessary to read data in Flash in real time while simulating in circuit, therefore, it cannot simulate while in read-protected.

## 2.3.2 Code Flash operation description

The 256K CodeFlash erasable and divided into 11 blocks according to the following figure:




Code Flash operation in detail is following:

| Register operation | Flash operation description |
|---|---|
| FLASHLOCK = 0x7A68 | You can make the whole 256K code flash entire erase or page erased, no need to consider whether any section FSnLOCK is unlocked |
| **The following to unlock 256K Flash partition block after block, only to unlock the page erase, wipe all invalid.** | |

| FLASHLOCK = 0x7A68The write enable command for the highest coverage level, need to partition the unlock operation on Flash; FLASHLOCK must be set to a non 0x7A68. | |
|---|---|
| FS1LOCK = 0x7161 | Only page erase & write on 8K CodeFlash 0x0 0000 – 0x0 1FFF |
| FS2LOCK = 0x7262 | Only page erase & write on 24K CodeFlash 0x0 2000- 0x0 7FFF |
| FS3LOCK = 0x7363 | Only page erase & write on 32K CodeFlash 0x0 8000- 0x0 FFFF |
| FS4LOCK = 0x7464 | Only page erase & write on 32K CodeFlash 0x10000- 0x1 7FFF |
| FS5LOCK = 0x7565 | Only page erase & write on 31K CodeFlash 0x1 8000- 0x1 FBFF |
| FSALOCK = 0x7A6A | Only page erase & write on 1K CodeFlash 0x1 FC00 - 0x1 FFFF |
| FS6LOCK = 0x7666 | Only page erase & write on 32K CodeFlash 0x2 0000- 0x2 7FFF |
| FS7LOCK = 0x7767 | Only page erase & write on 32K CodeFlash 0x2 8000- 0x2 FFFF |
| FS8LOCK = 0x7868 | Only page erase & write on 32KCodeFlash 0x3 0000- 0x3 7FFF |
| FS9LOCK = 0x7969 | Only page erase & write on 31K CodeFlash 0x3 8000- 0x3 FBFF |
| FSBLOCK = 0x7B6B | Only page erase & write on 1K CodeFlash 0x3 FC00 - 0x3 FFFF |

### 2.3.3 256K Code Flash Operation Description

256K Code Flash support write/page erase/full erase operation, described as blowing,(take pseudo-code for example)

It is recommended to use macro definition to write to certain address of Memory. HT502X support byte operation, half-word operation, word operation, but users need to use word alignment access otherwise a HardFault will be triggered.

Macro definition manner：

```
#define M8(adr) (*((uint8_t *) (adr)))        //need to concern about word alignment
#define M16(adr) (*((uint16_t *) (adr)))      //need half word alignment, which means bit0 of adr must be 0
#define M32(adr) (*((uint32_t *) (adr)))      //need word alignment, which means bit0 and bit1 of adr must be 0
```

Macro definition above can be used to take the address of adr of Flash

**Process of writing byte to 256K Code Flash：**

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;                 //unlock flash memory

FLASHCON = 0x01;                    //program
M32(prog_address) = prog_data;   //prog_data is the data to be written（32bit），
                                    //prog_address is the address of flash to write at
M16(prog_address) = prog_data;  //prog_data is the data to code（16bit），
                                    //prog_address is the address of flash to write at
M8(prog_address) = prog_data;       //prog_data is the data to code（8bit），
                                    //prog_address is the address of flash to write at
while (FLASHCON.BUSY);              //wait until flash writing is finished, 20us at most
```

Note：

8

While writing on words(32bit), ithe prog_address should be increased by multiple of 4 if necessary.
While writing on words(16bit), ithe prog_address should be increased by multiple of 4 if necessary.
While writing on words(8bit), ithe prog_address should be increased by multiple of 1 if necessary.

**Process of page erasing to 256K Code Flash：**

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;                 //unlock flash memory

FLASHCON = 0x02;                    //page erase
M32(prog_address) = prog_data;   //prog_data can be arbitrary data（32bit），
                                    //prog_address is a address of flash where a data need to be erased
while (FLASHCON.BUSY);          //wait until flash page erasing is finished, 2ms at most
```

**Process of all erasing to 256K Code Flash：**

```
WPREG = 0xA55A;
FLASHLOCK = 0x7A68;                 //unlock flash memory
FLASHCON = 0x03;                    // mass erase
M32(prog_address) = prog_data;      //prog_data can be arbitrary data（32bit），
                                    // prog_address is the arbitrary address of 128K Flash
while (FLASHCON.BUSY);              //wait until flash all erasing is finished,10ms at most
                                    //the user code to be executed will be erased if all erased
```

### 2.3.4 Information Block operation instructions

Information Block has 1K bytes（0x00040000~0x000403FF）, one page, 1024 bytes/page. The memory chip factory information, write operation is not recommended.

## 2.4Flash control function

In the HT5X2X microcontroller, the 0FC0H~0FC3H region of the Flash memory is the Flash control option byte region. When the chip turns on the power or restarts from the reset state, the system automatically refers to the option byte and functions as specified in its configuration settings. When using HT5X2X, you must use the option byte to set the following functions.

- Flash encryption control function
- POR/LBOR reset RTC enable control
- Info Block parameter auto loading enable control

The HT502X Flash option word acts as follows:

| Control By Flash | | | Base Address :<br>Offset Address: | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Flash Address** | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **FC1H** | FLASH[7:0] | | | | | | | |
| **Reset value** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **Flash Address** | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **FC0H** | X | X | X | X | RTCRST | AUTORELOAD | X | X |
| **Reset value** | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

| Bit | Function Description |
|---|---|
| FLASH[7:0] | =0xFF, then Flash is not encrypted.<br>Other values: Flash read protection, only low 256 bytes space can be read out. |
| RTCRST | **RTC reset control bit**<br>=1, LBOR and POR can reset RTC timing register<br>=0, LBOR and POR cannot reset RTC timing register, can be writed |
| AUTORELOAD | **Auto-load enable bit (for RTC compensation coefficients, etc.)**<br>=1, auto-load function is enabled<br>=0, auto-load function is blocked |

**NOTE: Other bits cannot be modified and remain default.**

## 2.5 Write protect register list

The read-protected registers lays in CMU block, PMU block, RTC block,details list as following:

| Base address of CMU block register：0x4000F000 | | | |
|---|---|---|---|
| Offset address | Register | Reset value | Function description |
| 0x00 | WPREG | 0x0000 | Write protect control register |
| 0x04 | SYSCLKCFG | 0x0002 | System clock configuration register(write protect) |
| 0x0C | LRCADJ | 0x0009 | Low frequency RC adjust register(write protect) |
| 0x10 | HRCADJ | 0x003D | High frequency RC adjust register(write protect) |
| 0x1C | SYSCLKDIV | 0x0001 | System clock division register(write protect) |
| 0x24 | CLKOUTSEL | 0x0000 | CLKOUT clock selection register(write protect) |
| 0x28 | CLKOUTDIV | 0x0000 | CLKOUT clock division register(write protect) |
| 0x2C | CLKCTRL0 | 0x04E0 | Internal block enable register0(write protect) |

| | | | |
|---|---|---|---|
| 0x30 | CLKCTRL1 | 0x0000 | Internal block enable register1(write protect) |
| 0x34 | FLASHCON | 0x0000 | Flash access control register(write protect) |
| 0x90 | FLASHCON2 | 0x0000 | Flash access control register 2 (write protect) |

| Base address of PMU block register Base address：0x4000F400 | | | |
|---|---|---|---|
| Offset address | Register | Reset value | Function description |
| 0x00 | PMUCON | 0x0017 | PMU configuration register(write protect) |

| Base address of RTC block register Base address：0x4000C000 | | | |
|---|---|---|---|
| Offset | Register | Reset value | Function description |
| 0x18 | SECR | 0x0000 | Second register(write protect) |
| 0x1C | MINR | 0x0000 | Minute register(write protect) |
| 0x20 | HOURR | 0x0000 | Hour register(write protect) |
| 0x24 | DAYR | 0x0001 | Day register(write protect) |
| 0x28 | MONTHR | 0x0001 | Month register(write protect) |
| 0x2C | YEARR | 0x0000 | Year register(write protect) |
| 0x30 | WEEKR | 0x0001 | Week register(write protect) |
| 0x200 | SECR2 | 0x0000 | Second register(write protect) |
| 0x204 | MINR2 | 0x0000 | Minute register(write protect) |
| 0x208 | HOURR2 | 0x0000 | Hour register(write protect) |
| 0x20C | DAYR2 | 0x0001 | Day register(write protect) |
| 0x210 | MONTHR2 | 0x0001 | Month register(write protect) |
| 0x214 | YRR2 | 0x0000 | Year register(write protect) |
| 0x218 | WEEKR2 | 0x0001 | Week register(write protect) |

## 2.6 Special function register list

| Base address of CMU block registerBase address：0x4000F000 | | | | |
|---|---|---|---|---|
| Offset | Register | Write or read | Reset value | Function description |
| 0x00 | WPREG | R/W | 0x0000 | Write protect control register |
| 0x08 | JTAGSTA | R | 0x0000 | JTAG status register(read only) |
| 0x34 | FLASHCON | R/W | 0x00 | Flash control register(write protect) |
| 0x38 | FLASHLOCK | R/W | 0x0000 | Flash lock register |
| 0x50 | INFOLOCK | R/W | 0x0000 | Information Block lock register |
| 0x60 | FS1LOCK | R/W | 0x0000 | FlashSector1 lock register |
| 0x64 | FS2LOCK | R/W | 0x0000 | FlashSector2 lock register |
| 0x68 | FS3LOCK | R/W | 0x0000 | FlashSector3 lock register |
| 0x6C | FS4LOCK | R/W | 0x0000 | FlashSector4 lock register |

| | | | | |
|---|---|---|---|---|
| 0x70 | FS5LOCK | R/W | 0x0000 | FlashSector5 lock register |
| 0x74 | FS6LOCK | R/W | 0x0000 | FlashSector6 lock register |
| 0x78 | FS7LOCK | R/W | 0x0000 | FlashSector7 lock register |
| 0x7C | FS8LOCK | R/W | 0x0000 | FlashSector8 lock register |
| 0x80 | FS9LOCK | R/W | 0x0000 | FlashSector9 lock register |
| 0x84 | FSALOCK | R/W | 0x0000 | FlashSectorA lock register |
| 0x88 | FSBLOCK | R/W | 0x0000 | FlashSectorB lock register |
| 0x90 | FLASHCON2 | R/W | 0x0000 | Flash control register 2 (write protect) |

## 2.7 Special function register instruction

| WPREG (write protect register) | | | Base address： 0x4000F000 Offset： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | WPREG[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | WPREG[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bitsbit | Function description |
|---|---|
| WPREG[15:0] | Write 0xA55A to WPREG to disable write protect function and users can write to operation-protected register WPREG<br>Write non-0xA55A to WPREG to enable write protect function and users cannot write to operation-protected register WPREG<br>Read this register<br>0x0001： close write protect,users can write to operation-protected register<br>0x0000： open write protect,users cannot write to operation-protected register |

| FLASHCON （write protect） Flash control register | | | Base address： 0x4000F000 Offset address： 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | X | X | X | X | X | BUSY | FOP[1:0] | |
| Write: | | | | | | X | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|

| | |
|---|---|
| BUSY | FLASH busy flag<br>0：free flash, operable<br>1：Flash is being written or erased.<br>Read only |
| FOP[1:0] | FLASH operate mode selection |

| FOP1 | FOP0 | FLASH operate |
|---|---|---|
| 0 | 0 | Flash in read-only mode |
| 0 | 1 | Write to address of flash pointed by STR/STRH |
| 1 | 0 | Erase the page of flash pointed by STR/STRH |
| 1 | 1 | Full erase the page of flash pointed by STR/STRH |

| FLASHLOCK (Flash lock register) | | | Base address： 0x4000F000 Offset address： 38H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | KEY[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | KEY[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bitsbit | Function description |
|---|---|
| KEY[15:0] | Flash lock control bit<br>Write 0x7A68 to this register to unlock Flash and users can write Flash<br>Write non-0x7A68 to lock Flash and users cannot write to Flash<br>Default locked, flash cannot be written/page erased/all erased<br>Write 0x7A68 and value read is 1, write non-0x7A68 and value read is 0 |

| INFOLOCK (InfoFlash lock register) | | | Base address： 0x4000F000 Offset address： 50H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | KEY[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | KEY[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| KEY[15:0] | Information Block lock control<br>Write 0Xf998 to this register to unlock Information block and users can write to Information block.<br>Write non-0Xf998 to this register to lock Information block and users cannot write to Information block.<br>Default locked, Information Block cannot be written/page erased/all erased<br>Write 0xF998 and value read is 1, write non-0xF998 and value read is 0 |

| **FS1LOCK (FlashSector1 lock register)** | | | **Base address: 0x4000F000 Offset address: 60H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **bit** | **Function description** |
|---|---|
| KEY[15:0] | **FlashSector1 lock control bit**<br>After writing 0x7161 to the register, 8K CodeFlash 0x00000 – 0x01FFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7161 data in this register, CodeFlash 0x0 0000 – 0x0 1FFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7161, read out 1; write to non 0x7161, read out 0. |

| **FS2LOCK (FlashSector2 lock register)** | | | **Base address: 0x4000F000 Offset address: 64H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |

| Write: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector2 lock control bit**<br>After writing 0x7262 to the register, 24K Code Flash 0x0 2000 – 0x0 7FFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7262 data in this register, Code Flash 0x0 2000 – 0x0 7FFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7262, read out 1; write to non 0x7262, read out 0. |

| FS3LOCK (FlashSector3 lock register) | | | Base address: 0x4000F000 Offset address: 68H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: / Write: | KEY[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: / Write: | KEY[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector3 lock control bit**<br>After writing 0x7363 to the register, 32K Code Flash 0x0 8000 – 0x0 FFFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7363 data in this register, Code Flash 0x0 8000 – 0x0 FFFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7363, read out 1; write to non 0x7363, read out 0. |

| FS4LOCK (FlashSector4 lock register) | | | Base address: 0x4000F000 Offset address: 6CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: / Write: | KEY[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector4 lock control bit**<br>After writing 0x7464 to the register, 32K Code Flash 0x1 0000 – 0x1 7FFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7464 data in this register, Code Flash 0x1 0000 – 0x1 7FFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7464, read out 1; write to non 0x7464, read out 0. |

| **FS5LOCK (FlashSector5 lock register)** | | | **Base address: 0x4000F000 Offset address: 70H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector5 lock control bit**<br>After writing 0x7565 to the register, 31K Code Flash 0x1 8000 – 0x1 FBFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7565 data in this register, Code Flash 0x1 8000 – 0x1 FBFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7565, read out 1; write to non 0x7565, read out 0. |

| **FS6LOCK (FlashSector6 lock register)** | | | **Base address: 0x4000F000 Offset address: 74H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |

| Write: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| Read: | KEY[7:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector6 lock control bit**<br>After writing 0x7666 to the register, 32K Code Flash 0x2 0000 – 0x2 7FFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7666 data in this register, Code Flash 0x2 0000 – 0x2 7FFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7666, read out 1; write to non 0x7666, read out 0. |

| **FS7LOCK (FlashSector7 lock register)** | | | **Base address: 0x4000F000**<br>**Offset address: 78H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| Read: | KEY[15:8] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| Read: | KEY[7:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector7 lock control bit**<br>After writing 0x7767 to the register, 32K Code Flash 0x2 8000 – 0x2 FFFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7767 data in this register, Code Flash 0x2 8000 – 0x2 FFFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7767, read out 1; write to non 0x7767, read out 0. |

| **FS8LOCK (FlashSector8 lock register)** | **Base address: 0x4000F000**<br>**Offset address: 7CH** |
|---|---|

| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
|---|---|---|---|---|---|---|---|---|
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector8 lock control bit**<br>After writing 0x7868 to the register, 32K Code Flash 0x3 0000 – 0x3 7FFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7868 data in this register, Code Flash 0x3 0000 – 0x3 7FFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7868, read out 1; write to non 0x7868, read out 0. |

| **FS9LOCK (FlashSector9 lock register)** | | **Base address: 0x4000F000**<br>**Offset address: 80H** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector9 lock control bit**<br>After writing 0x7969 to the register, 31K Code Flash 0x3 8000 – 0x3 FBFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7969 data in this register, Code Flash 0x3 8000 – 0x3 FBFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7969, read out 1; write to non 0x7969, read out 0. |

| FSALOCK (FlashSectorA lock register) | | | Base address: 0x4000F000 Offset address: 84H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function description |
|---|---|
| KEY[15:0] | **FlashSector A lock control bit**<br>After writing 0x7A6A to the register, 1K Code Flash 0x1 FC00 – 0x1 FFFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7A6A data in this register, Code Flash 0x1 FC00 – 0x1 FFFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7A6A, read out 1; write to non 0x7A6A, read out 0. |

| FSBLOCK (FlashSectorB lock register) | | | Base address: 0x4000F000 Offset address: 88H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | KEY[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | KEY[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | BFunction description |
|---|---|
| KEY[15:0] | **FlashSectorB lock control bit**<br>After writing 0x7B6B to the register, 1K Code Flash 0x3 FC00 – 0x3 FFFF is unlocked, user can write in page for this sector.(Note: Only page erase is available at this time, and total erase is invalid )<br>After writing to non 0x7B6B data in this register, Code Flash 0x3 FC00 – 0x3 FFFF is locked, user is forbidden to erase flash of this sector。<br>**Default locked status**<br>Write to 0x7B6B, read out 1; write to non 0x7B6B, read out 0. |

| FLASHCON2（write protect）<br>(Flash control register) | | | Base address: 0x4000F000<br>Offset address: 90H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | CON[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | CON[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | Function discreption |
|---|---|
| CON[15:0] | **Flash read/write control**<br>When FLASHCON2= 0xCA53, Flash is in the write state<br>When FLASHCON2= 0xAC35, Flash is in the page erase state<br>When FLASHCON2 is other value，**Flash is in the read state**<br><br>Note: write FLASHCON2 has no total erase state, this register is for data transmit function. The read value is 0. If you want to determine the BUSY status, read the BUSY flag bit of FLASHCON. |

# 3 Clock unit

## 3.1 Clock classify

Test temperature range: -40℃~85℃

| name | frequency | accuracy | power | | |
|---|---|---|---|---|---|
| | | | MIN | TYP | MAX |
| Internal low frequency clock(Flrc) | 32.768KHz | 13KHz ~50KHz | | 1uA | |
| Internal high frequency clock(Fhrc) | 9.8MHz | +- 3% | | 130uA | |
| External low frequency oscillator(Fosc) | 32768Hz | | | | 1uA |
| Internal PLL(Fpll) | 19.660800MHz/ 39.321600MHz | | | 320uA | |

Note: the frequency of the internal high frequency RC clock is the nominal value after the HRCADJ load Info Block adjustment value, the normal temperature accuracy is + 1%, and the full temperature range (-40 DEG C, ~+85 DEG C) is + 3%

## 3.2 Clock block diagram

Clock symbols description：

Flrc： Internal low frequency RC clock(32768Hz),clock source of watch dog.

Fhrc： Internal high frequency RC clock(9.5 MHz).

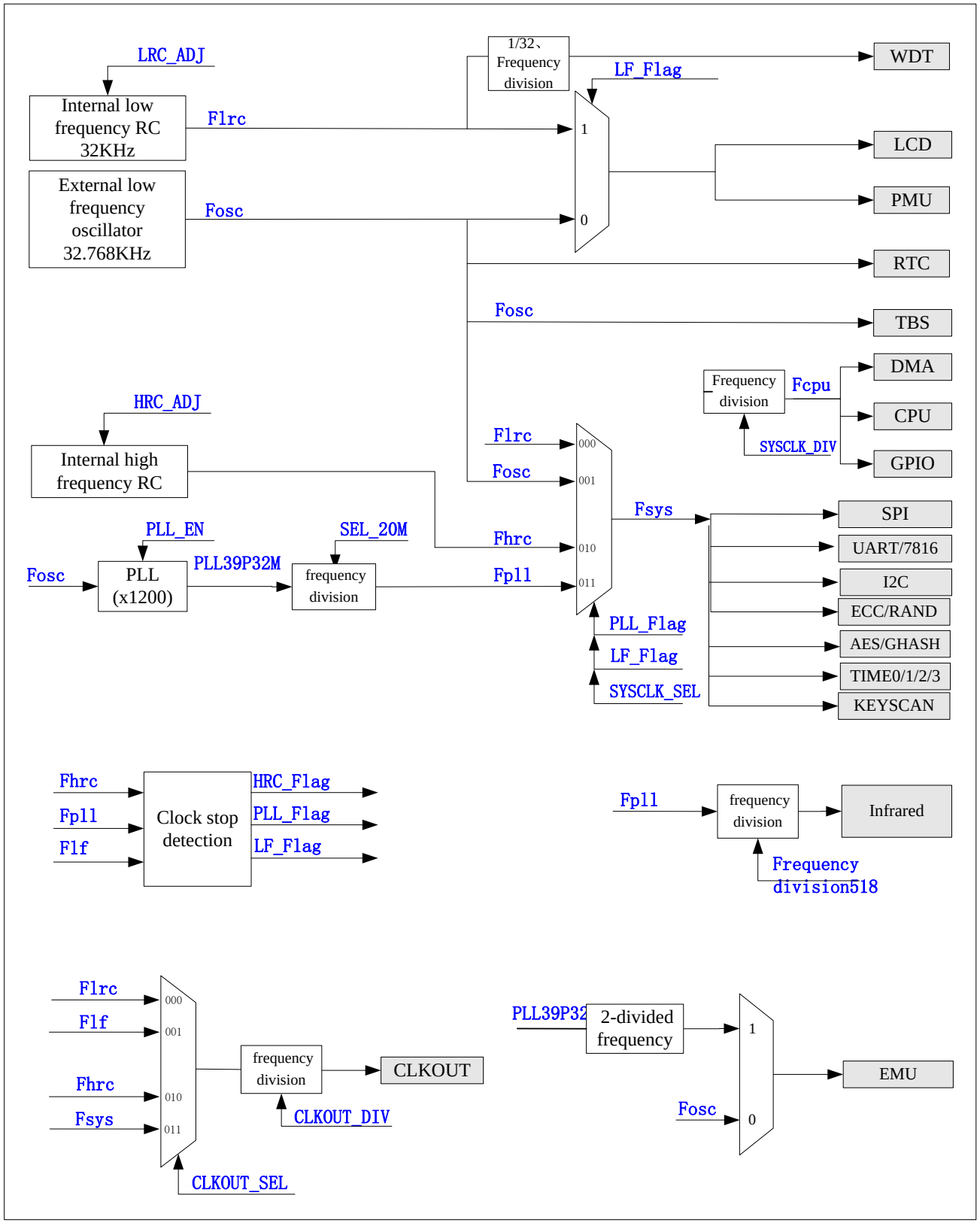
Fosc： External low frequency oscillator(32768Hz).

Fpll： High frequency clock source (39.321600MHz) ,produced by Internal PLL,comes from Fosc

Flf： internal selected low frequency clock（32.768KHz）,same as Fosc

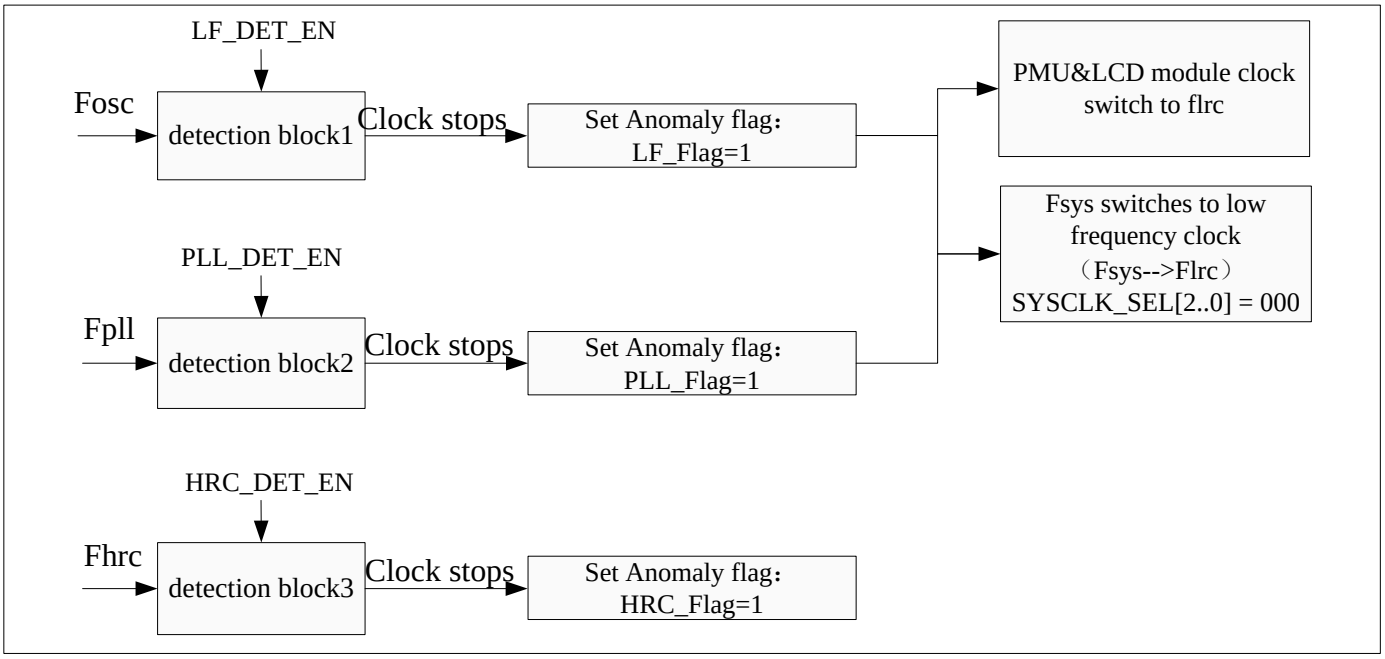
Fsys：System clock, mainly provides clock for all kinds of peripheral

Fcpu：Provides clock for CPU\GPIO\DMA




Note: in this diagram, the internal select low frequency clock Flf is an external low frequency clock Fosc.

## 3.3 Clock stop detection block diagram




## 3.4 Clock instruction

### 3.4.1 Internal low frequency RC clock（Flrc）

Internal low frequency RC clock is 32768Hz, for use of watchdog,low frequency RC clock can function as system clock(SYSCLK_SEL[2:0]=000)

Internal low frequency RC clock can be calibrated by register LRCDAJ to achieve high accuracy.

Internal low frequency RC can be stopped by software by users in sleep or hold mode, control bit is LRC_CTRL of ControlByFlash.

### 3.4.2 Internal high frequency RC clock（Fhrc）

Internal high frequency RC clock is 9.8Mhz and division frequency of this clock can function as system clock(SYSCLK_SEL[2:0]=010).

Internal high frequency RC clock error is less than 1% after calibrated through register HRCADJ
Internal high frequency RC clock sampling error less than 1% at all temperature range

Default system clock is high frequency RC clock (SYSCLK_SEL[2:0]=010) after system resetting

Internal high frequency RC clock is unable to be shut down while chosen to be system clock,and write 0 operation will be invalid.

External low frequency oscillator clock

System clock frequency Fosc=32768Hz while external 32K low power oscillator is connected. and related resistances and capacitors are integrated, the matching capacitance is about 12.5pF.

### 3.4.3 Internal PLL clock（Fpll）

External low frequency oscillator clock Fosc（32768Hz）can be frequency multiplication with internal PLL (frequency multiplication number=1200) to provide system clock signals Fpll of 39.321600MHz at most.PLL_LOCK signals will be available when PLL is steady.

### 3.4.4 Clock security mechanism

There are 3 integrated independent clock stop detection blocks for external low frequency oscillator clock Fosc, PLL output clock Fpll and internal high frequency clock Fhrc respectively. The block is enabled by defaulted and can be shut down by software by users, control bits are LF_DET_EN, PLL_DET_EN, HRC_DET_EN respectively.

Clock source of clock stop detection block is internal low frequency RC clock Flrc.

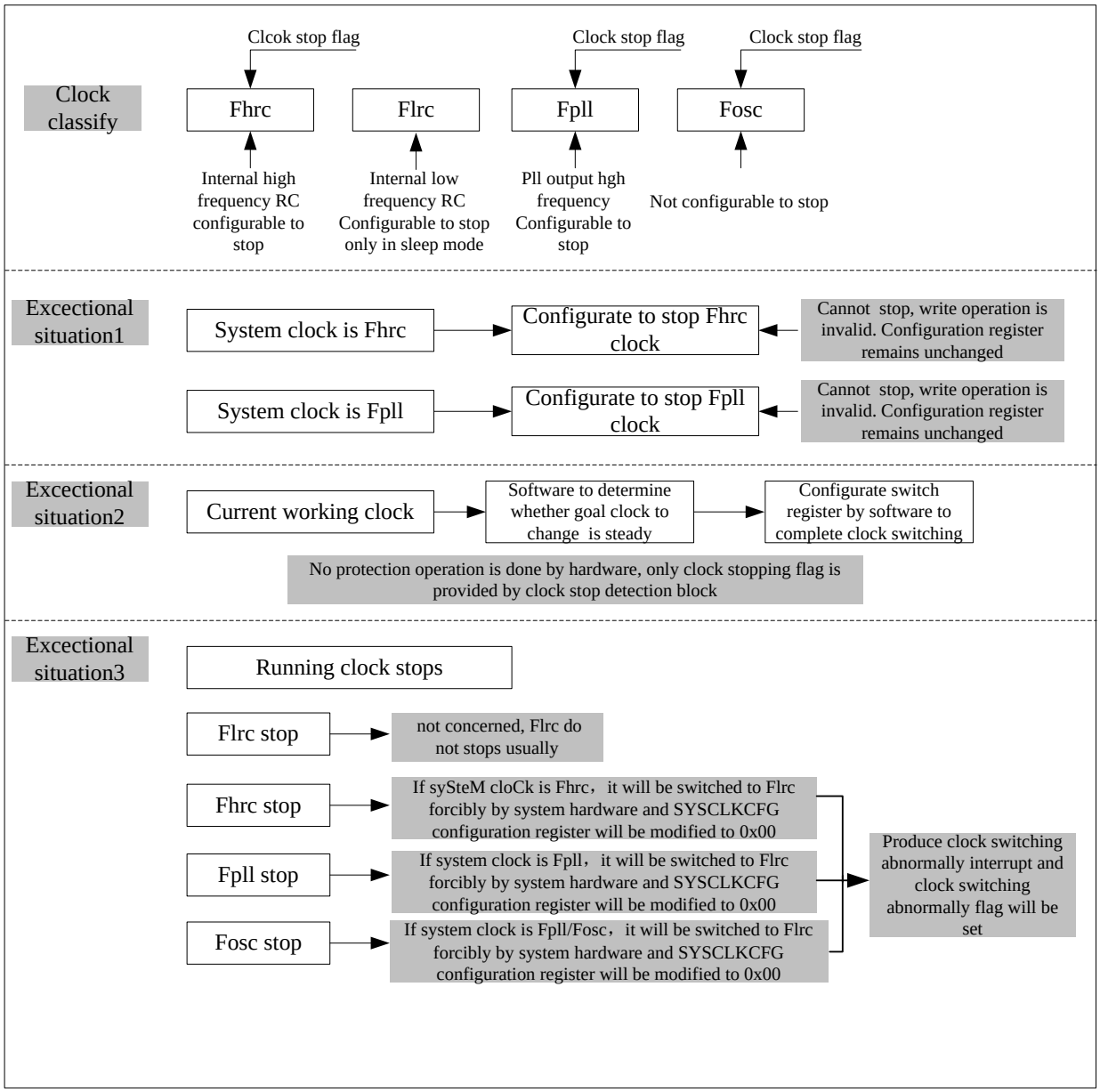
When corresponding clock stop detection block is enabled, external low frequency oscillator clock Fosc stops,PLL clock Fpll stops or internal high frequency RC clock Fhrc stops, corresponding Clock fault flag （LF_FLAG，PLL_FLAG, HRC_FLAG） will be set.

While Fosc stopping is detected, clock stopping flag LF_FLAG will be set.if system clock Fsys is Fosc or Fpll(Fosc is the clock source of Fpll), system clock will be forced to switch to internal low frequency RC by hardware and a interrupt will be produced(NMI) and register SYSCLK_SEL[2:0] will be set to 000.

While Fpll stopping is detected, clock stopping flag PLL_FLAG will be set.if system clock Fsys is Fpll, system clock will be forced to switch to internal low frequency RC by hardware and a interrupt will be produced(NMI) and register SYSCLK_SEL[2:0] will be set to 000.

While Fhrc stopping is detected, flag HRC_FALG will be set.if system clock Fsys is Fhrc, system clock will not be forced to change and system will stop to wait watch dog to be reset.

### 3.4.5 Clock abnormal state dealing




1. When System clock is low frequency clock Fosc, if PLL stop, no action will be taken by hardware and PLL_FLAG will be set.
2. When System clock is low frequency clock Fosc, if clock source of Fosc or Fmems stops,system clock will be forced to switch to Flrc and a NMI interrupt will be produced.
3. When System clock is Fpll, if Fosc or Fpll stops,system clock will be forced to switch to Flrc and a NMI interrupt will be produced.

## 3.5 Special function register list

| Base address of CMU register：0x4000F000 | | | | |
|---|---|---|---|---|
| Offset address | name | Write or read | Reset value | Function description |
| 0x00 | WPREG | R/W | 0x0000 | Write protect control register |
| 0x04 | SYSCLKCFG | R/W | 0x0002 | System clock configuration register(wirte protect) |
| 0x08 | JTAGSTA | R/W | 0x0001 | JTAG state register(write protect) |
| 0x0C | LRCADJ | R/W | 0x0009 | LRC clock adjust register(write protect) |
| 0x10 | HRCADJ | R/W | 0x003D | HRC clock adjust register(write protect) |
| 0x18 | CLKSTA | R | 0x0000 | Clock state register(read protect) |
| 0x1C | SYSCLKDIV | R/W | 0x0001 | System clock frequency division register(write protect) |
| 0x24 | CLKOUTSEL | R/W | 0x0002 | CLKOUT clock selection register(write protect) |
| 0x28 | CLKOUTDIV | R/W | 0x0000 | CLKOUT clock frequency division register(write protect) |
| 0x2C | CLKCTRL0 | R/W | 0x24E0 | Internal block enable register0(write protect) |
| 0x30 | CLKCTRL1 | R/W | 0x0000 | Internal block enable register1(write protect) 内 |
| 0x34 | FLASHCON | R | 0x00 | Flash control register(write register) |
| 0x38 | FLASHLOCK | W | 0x0000 | Flash lock register |
| 0x3C | FLASHDLY | R/W | 0x0000 | Flash late access address register |
| 0x60 | FS1LOCK | R/W | 0x0000 | FlashSector1 lock register (refer to chapt 2 “memory module” in detail) |
| 0x64 | FS2LOCK | R/W | 0x0000 | FlashSector2 lock register (refer to chapt 2 “memory module” in detail) |
| 0x68 | FS3LOCK | R/W | 0x0000 | FlashSector3 lock register (refer to chapt 2 “memory module” in detail) |
| 0x6C | FS4LOCK | R/W | 0x0000 | FlashSector4 lock register (refer to chapt 2 “memory module” in detail) |
| 0x70 | FS5LOCK | R/W | 0x0000 | FlashSector5 lock register (refer to chapt 2 “memory module” in detail) |
| 0x74 | FS6LOCK | R/W | 0x0000 | FlashSector6 lock register (refer to chapt 2 “memory module” in detail) |
| 0x78 | FS7LOCK | R/W | 0x0000 | FlashSector7 lock register (refer to chapt 2 “memory module” in detail) |
| 0x7C | FS8LOCK | R/W | 0x0000 | FlashSector8 lock register (refer to chapt 2 “memory module” in detail) |
| 0x80 | FS9LOCK | R/W | 0x0000 | FlashSector9 lock register |

| | | | | |
|---|---|---|---|---|
| | | | | (refer to chapt 2 “memory module” in detail) |
| 0x84 | FSALOCK | R/W | 0x0000 | FlashSectorA lock register<br>(refer to chapt 2 “memory module” in detail) |
| 0x88 | FSBLOCK | R/W | 0x0000 | FlashSectorB lock register<br>(refer to chapt 2 “memory module” in detail) |
| 0x90 | FLASHCON2 | R/W | 0x0000 | Flash control register2（write protect）<br>(refer to chapt 2 “memory module” in detail) |
| 0xF00 | CHIPID | R | | Chip version register,read-only |

## 3.6 Special function register description

| WPREG<br>（write protect register） | | | Base address： 0x4000F000<br>Offset address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | WPREG[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | WPREG[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| WPREG[15:0] | Write 0xA55A to WPREG to disable write protect function,users can write to operation-protected register.<br>Write non-0xA55A to WPREG to enable write protect function,users cannot write to operation-protected register.<br>Read this register：<br>0x0001： Write protect is disabled, users can write to operation-protected register.<br>0x0000： Write protect is enabled, users cannot write to operation-protected register. |

| SYSCLKCFG（write protect）<br>（ system clock configuration register） | | | Base address： 0x4000F000<br>Offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | WCLKE | X | X | X | X | SYSCLK_SEL[2:0] | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Write: | N | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| bits | Function description |
|---|---|
| WCLKEN | clock configuration register write protection bit<br>users must set WCLKEN to 1 while changing system clock, for example, 'b1xxxxxxx'users can write to system clock selection bits SYSCLK_SEL[2:0]. |
| SYSCLK_SEL[2:0] | System clock selction control bits：<br>SYSCLK_SEL[2:0] / System clock selection Fsys<br>0 0 0 Flrc<br>0 0 1 Fosc<br>0 1 0 Fhrc(Default)<br>0 1 1 Fpll<br>System clock can function as clock resource of internal hardware peripheral, and can be clock resource of CPU and GPIO after frequency division.<br>System clock is internal high frequency RC clock Fhrc by default after resetting （SYSCLK_SEL[2:0]=010）.<br>When external low frequency oscillator clock Fosc function as system clock and is detected to be stopped,Fsys will be forced to switch to internal RC low speed clock Flrc by hardware and system clock control bits SYSCLK_CLK[2:0] will be set to 000.<br>When PLL output clock Fpll function as system clock and Fosc or PLL is detected to be stopped,Fsys will be forced to switch to internal RC low speed clock Flrc by hardware and system clock control bits SYSCLK_CLK[2:0] will be set to 000. |

| JTAGSTA （chip status indicate register） | | | Base address： 0x4000F000<br>Offset address： 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | X | X | X | FLASHSTA |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| bits | Function description |
|---|---|
| FLASHSTA | This bit is internal state reserve,read only. |

| LRCADJ（write protect）<br>（LRC clock adjust register） | | | Base address： 0x4000F000<br>Offset address： 0CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | LRC_ADJ[3:0] | | | |
| Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| bits | Function description |
|---|---|
| LRC_ADJ[3:0] | LRC output frequency adjust control bit |

Note: to ensure the accuracy of the LRC output frequency, it is recommended that the customer only load the factory test values stored in the Info Block, and do not automatically modify them to other values.

| HRCADJ（write protect）<br>（HRC clock adjust register） | | | Base address： 0x4000F000<br>Offset address： 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | HRC_ADJ[5:0] | | | | | |
| Reset: | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

| bits | Function description |
|---|---|
| HRC_ADJ[5:0] | HRC output frequency adjust control bit |

Note: to ensure the accuracy of the HRC output frequency, it is recommended that the customer only load the factory test values stored in the Info Block, and do not automatically modify them to other values.

| CLKSTA<br>（clcok status register） | | | Base address： 0x4000F000<br>Bas address： 18H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Reset | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Read: | X | X | PLL_LOCK | PLL_FLAG | HRC_FLAG | X | X | LF_FLAG |
| Write: | | | X | X | X | | X | X |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: this register is read only status register

| bitsbit | Function description |
|---|---|
| PLL_LOCK | PLL clock lock status flag register PLL<br>0：PLL clock lock abnormal.<br>1：PLL clock lock normal.<br>Note: to indicate internal PLL of chip steady working-status.<br>PLLLOCK is always 1 in JTAG mode. |
| PLL_FLAG | PLL clock Fpll stop flag<br>0：normal<br>1：stop |
| HRC_FLAG | Internal high frequency RC clock Fhrc stop flag<br>0：normal<br>1：stop |
| LF_FLAG | External low frequency oscillator clock Fosc stop flag<br>0：normal<br>1：stop<br>(if internal low frequency RC clock Flrc stop in sleep or hold mode,external low frequency oscillator clock Fosc stop detection block LF_DEF default output is that LF clock is normal,namely LF_FLAG=0;only when Fsys is Fosc or output clock of PLL,Fpll,and corresponding clock is detected to be stopped,system clock will switch to internal low frequency RC clock Flrc forcibly and value of register SYSCLK_SEL[2:0] will be modified to 000.) |

| SYSCLKDIV（write rotect）（ system clock frequency division register） | | | Base address： 0x4000F000<br>Offset address： 1CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | SEL_20M | SYSCLK_DIV[2:0] | | |
| Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| | Function description |
|---|---|

| | |
|---|---|
| SEL_20M | PLL output clock pre-frequency-division register control bits：<br>=0：PLL output 19.66MHz clock<br>=1：PLL output 39.32MHz clock |
| SYSCLK_DIV[2:0] | Divide system clock to get Fcpu,setup as below： |

| SYSCLK_DIV [2:0] | | | System clcok select Fcpu |
|---|---|---|---|
| 0 | 0 | 0 | Fsys |
| 0 | 0 | 1 | Fsys/2(default) |
| 0 | 1 | 0 | Fsys/4 |
| 0 | 1 | 1 | Fsys/8 |
| 1 | 0 | 0 | Fsys/16 |
| 1 | 0 | 1 | Fsys/32 |
| 1 | 1 | 0 | Fsys/64 |
| 1 | 1 | 1 | Fsys/128 |

| CLKOUTSEL（write protect）（ CLKOUT clock selection register） | | | Base address： 0x4000F000<br>Offset address： 24H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | X | CLKOUT_SEL[2:0] | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bits | Function description |
|---|---|
| CLKOUT_SEL[2:0] | CLKOUT clock output pins configuration |

| CLKOUT_SEL[2:0] | | | CLKOUT clock select |
|---|---|---|---|
| 0 | 0 | 0 | Flrc |
| 0 | 0 | 1 | Fosc |
| 0 | 1 | 0 | Fhrc(Default) |
| 0 | 1 | 1 | Fsys |
| 1 | X | X | Fosc |

1，user can get internal clock source from CLKOUT pin to observe it

2，user can use CLKOUTDIV register to divide internal clock for clock source of external divice.

| CLKOUTDIV（write protect）（CLKOUT clock frequency division reister） | | | Base address： 0x4000F000<br>Offset address： 28H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | CLKOUT_DIV[3:0] | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CLKOUT_DIV[3:0] | CLKOUT output frequency $= \frac{clock\ souce\ selected\ by\ CLKOUT}{2\times(CLKOUT\_DIV[3...0]+1)}$ |

| CLKCTRL0 （write protect）（internal block enable control register0） | | | Base address： 0x4000F000<br>Offset address： 2CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | ECC_EN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | EMU_EN | AES_EN | 1P5LBOR_EN | CLKOUT_EN | KEY_EN | OSC_SLP | HRC_DET_EN | PLL_DET_EN |
| **Reset:** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | LF_DET_EN | PLL_LOCK_EN | HRC_EN | PLL_EN | I2C_EN | SPI_EN | LCD_EN | X |
| **Reset:** | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| ECC | **ECC Module enable signal**<br>0：disabled（default）<br>1：enabled |
| EMUEN | EMU block enable signal<br>0：disabled（default） |

| | |
|---|---|
| | 1：enabled |
| AES_EN | AES block enable signal<br>0：disabled（default）<br>1：enabled |
| 1P5LBOR_EN | Low power LDO_1P5 internal LBOR enable signal<br>0：disabled<br>1：enabled（default）recommend leaving this default configuration. |
| CLKOUT_EN | CLKOUT enable signal<br>0：disabled（default）<br>1：enabled |
| KEY_EN | Keys scanning function enable<br>0：disable<br>1：enable |
| OSC_SLP | OSC low power enable control signal<br>0：high power<br>1：low power（default） |
| HRC_DET_EN | HRC clock detection block control signal<br>0：shut down HRC clock stop detection unit（default）<br>1：enable HRC clock stop detection unit |
| PLL_DET_EN | PLL clock detection block control signal<br>0：shut down PLL clock stop detection unit（default）<br>1：enable PLL clock stop detection unit |
| LF_DET_EN | LF clock detection block control signal（Fosc may comes from external OSC or MEMS）<br>0：shut down LF clock stop detection unit<br>1：enable LF clock stop detection unit（default） |
| PLL_LOCK_EN | PLL forced lock enable control bit<br>0：PLL not forced lock<br>1：PLL force lock（default）<br>note：Enable this bit and corresponding PLL lock flag PLL_LOCK is always 1,otherwise PLL_LOCK will be the working status of PLL. |
| HRC_EN | HF RC clock oscillator enable bit<br>0：shut down high frequency RC clock block；<br>1：enable high frequency RC clock block；（default）<br>Note: while Fsys is Fhrc,HRC_EN cannot be shut down,it is invalid to write this bit. |
| PLL_EN | PLL block clock enable bit<br>0：shut down PLL block(default）<br>1：enable PLL block<br>Note: while Fsys is Fpll,PLL_EN cannot be shut down,it is invalid to write this bit. |
| I2C_EN | I2C block clock enable bot<br>0：shut down I2C block（default）<br>1：enable I2C block |
| SPI_EN | SPI block clock enable bit |

| | |
|---|---|
| | 0：shut down SPI block（default）<br>1：enable SPI block |
| LCD_EN | LCD block clock enable bit<br>0：shut down LCD block（default）<br>1：enable LCD block |

| CLKCTRL1 （write protect）（internal block enable control register1） | | | Base address： 0x4000F000<br>Offset address： 30H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | LRC_RTC2_EN | LRC_TFREQ_RTC2 | X | X | X | X | UART5_EN | UART4_7816_EN |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | UART3_7816_EN | UART2_EN | UART1_EN | UART0_EN | TMR3_EN | TMR2_EN | TMR1_EN | TMR0_EN |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| LRC_RTC2_EN | auxiliary RTC working enable bit<br>0：shut down<br>1：enable |
| LRC_TFREQ_RTC2 | Auxiliary RTC frequency measurement enable bit<br>0: shut down<br>1：enable |
| | |
| UART5_EN | UART5 clock enable bit<br>0：shut down<br>1：enable |
| UART4_7816_EN | UART4 clock enable bit<br>0：shut down<br>1：enable |
| UART3_7816_EN | UART3 clock enable bit<br>0：shut down；<br>1：enable |
| UART2_EN | UART2 clock enable bit<br>0：shut down<br>1：enable |
| UART1_EN | UART1 clock enable bit<br>0：shut down |

| | |
|---|---|
| | 1：enable |
| UART0_EN | UART0 clock enable bit<br>0：shut down<br>1：enable |
| TMR3_EN | Timer3 clock enable bit<br>0：shut down<br>1：enable |
| TMR2_EN | Timer2 clock enable bit<br>0：shut down<br>1：enable |
| TMR1_EN | Timer1 clock enable bit<br>0：shut down<br>1：enable |
| TMR0_EN | Timer0 clock enable bit<br>0：shut down<br>1：enable |

| FLASHCON （write protect）<br>(Flash control register) | | | Base address： 0x4000F000<br>Offset address： 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | X | X | X | X | X | BUSY | FOP[1:0] | |
| Write: | | | | | | X | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| BUSY | FLASH bust flag<br>0：Flash busy，operable。<br>1：Flash in writing/erasing operation。<br>Read-only status register,invalid writing operation. |
| FOP[1:0] | FLASH operation mode select |

| FOP1 | FOP0 | FLASH operation |
|---|---|---|
| 0 | 0 | In Flash read-only mode |
| 0 | 1 | Write to section of FLASH pointed by STR/STRH |
| 1 | 0 | Page erase section of FLASH pointed by STR/STRH |
| 1 | 1 | All erase section of FLASH pointed by STR/STRH |

| FLASHLOCK<br>(Flash lock register) | | | Base address： 0x4000F000<br>Offset address： 38H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | KEY[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | KEY[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| KEY[15:0] | Flash lock control bits<br>Write 0x7A68 to this register to unlock FLASH and users can write to FLASH<br>Write non-0x7A68 to this register to lock FLASH and users cannot write to FLASH<br>In lock status by default,unable to write/page erase/all erase FLASH<br>Writing to this register is unavailable,but it can be written. |

| FLASH_DLY （write protect）<br>(Flash control register) | | | Base address： 0x4000F000<br>Offset address： 3CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | X | X | X | X | X | X | X | FLASH_<br>DLY |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| FLASH_DLY | This bit should be set to 1 if system clock is 39.32M.<br>0：no waiting while fetching instructions in Flash<br>1：need waiting while fetching instructions in Flash（goes for it when PLL output clock is 39.32M） |

| CHIPID<br>Chip version register | | | Base address： 0x4000F000<br>Offset address： F00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | CHIPID[15…8] | | | | | | | |
| Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | CHIPID[7…0] | | | | | | | |
| Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CHIPID[15…0] | Chip version：0x5001 |

# 4 Power unit

## 4.1 General description

PMU is power manage unit of chip,of which functions are shown as blow:

- Operating voltage range：2.2V～5.5V
- Internal comparator VSYS_DET monitors system power VSYS,LVDIN_DER detect external low voltage detect input LVDIN0 and LVDIN1,an interrupt signals is generated when supply voltage is blow or exceed the threshold.
- Automatically switch VSYS and VBAT,power status is updated in PMSR register in real time .
- Supervise chip’s internal operating voltage VDD， interrupt signals can be generated according to the threshold;
- Supervise chip’s internal operating voltage VDD，reset signals like BOR、LBOR、POR can be generated according to the threshold;
- Provide 2.8V power for measurement block.
- Provide 1.5V power for internal digital blocks.

## 4.2 Block diagram

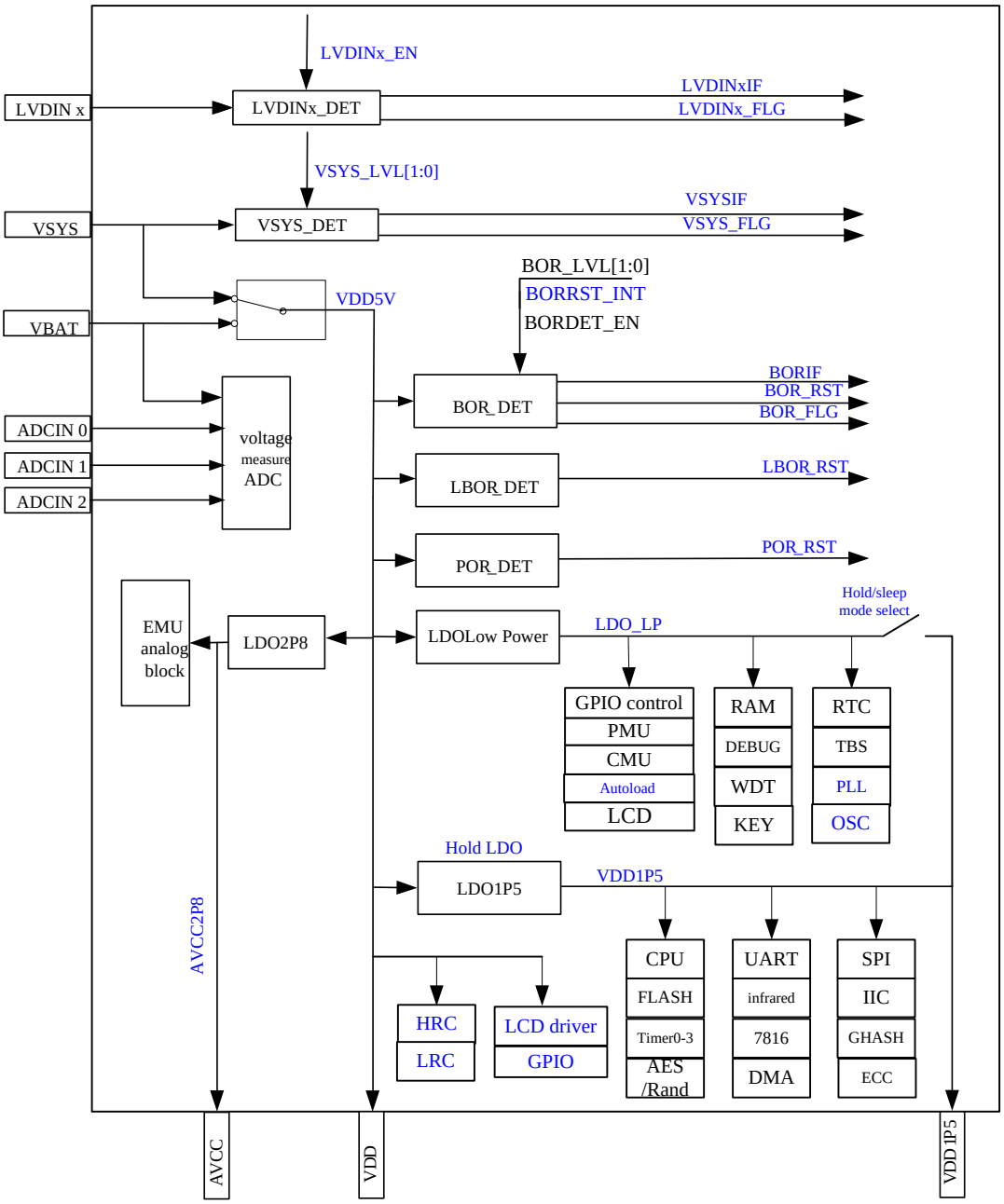



Fig 4-1 chip internal power-supply diagram

# 4.3 Power unit function detailed description

## 4.3.1 Power switch

External power source switching is done automatically by internal hardware.VSYS_DET will keep working after the system is powered on.VSYS_DET block detect external power VSYS voltage,and switching between system power source VSYS and battery power source is automatically done according to the voltage. And corrent VSYS voltage status will be updated at status bits VSYS_FLG in PMUSTA.

If battery power source VBAT voltage exceed 1.65V and VSYS_DET detect that VSYS voltage below threshold,it will VBAT that supply the power,namely VDD connect to VBAT; If battery power source VBAT voltage exceed 1.65V and VSYS_DET detect that VSYS voltage exceed threshold,it will VSYS that supply the power,namely VDD connect to VSYS.

If battery power source VBAT voltage below 1.65V,system power remains to be VSYS and VSYS_DET will has no impact on it.

## 4.3.2 Power detection in real-time

PMU unit has 5 built-in power detection block to detect operating power status respectively in real time and feedback it to users through 3 interrupt signals and 3 reset signals.

- **LVD_DET block:**
  Detect external pins voltage of LVDIN0 or LVDIN1. If voltage below or exceed 1.21V,LVDIN0IF/LVDIN1IF flag will be set, and LVDIN interrupt will be generated if LVDIN0IF/LVDIN1IF interrupt is enabled.
- **VSYS_DET block:**
  Detect system power source VDD voltage,if it exceed or below the threshold,VSYSIF will be set, and VSYS interrupt will be generated if VSYSIE interrupt is enabled;detection threshold can be set by bits VSYS_LVL[3:0] of VDETCFG.
- **BOR_DET block:**
  Detect system power source VDD voltage,if it exceed or below the threshold, BORIF will be set,;if BOR reset(BORRSET=1) is set,BOR will reset;if BOR reset(BORRSET=0) is not set but BORIE is enabled,BOR interrupt will be generated; detection threshold can be set by bits BOR_LVL[1:0] of VDETCFG.
- **LBOR_DET block:**
  Detect system power source VDD voltage,if it below threshold of 1.9v, it will undergo LBOR power-off reset.
- **POR_DET block:**
  Detect system power source VDD voltage,if it rise from below threshold 0.3V to above threshold,it will undergo POR power-on reset.

### 4.3.3 Built-in 1.5V power source

Employ VREG block to modulate voltage of VDD to 1.5V for chip internal operation and VDD output.An external 0.1uF capacitor should connect to output pin VDD for the steady of 1.5V internal digital power source.

### 4.3.4 BOR detection function BOR

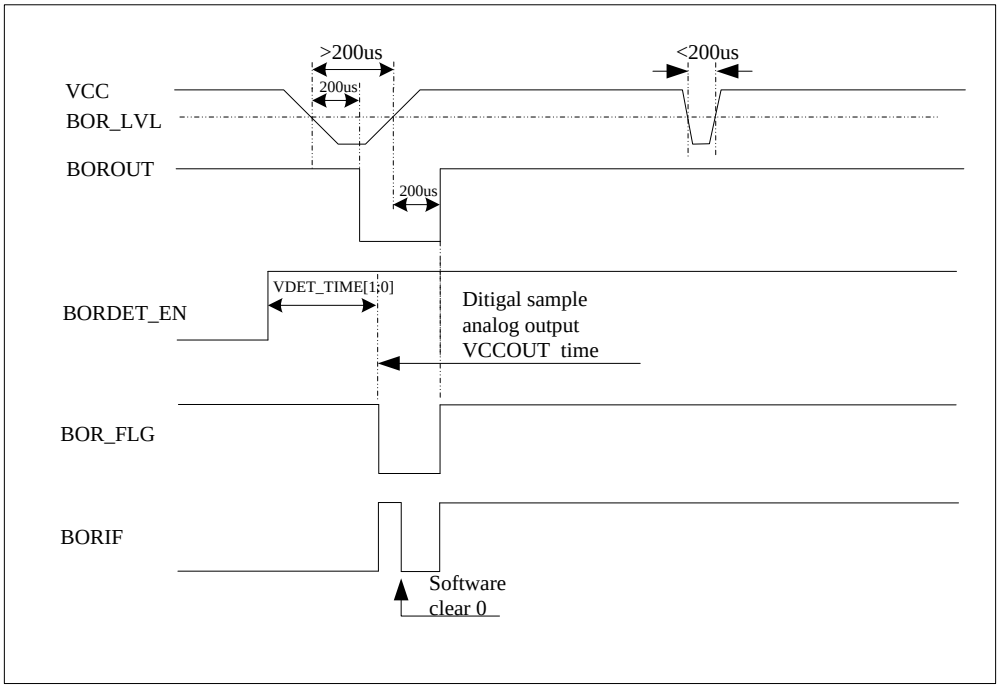
If BOR detection block detects that system power source below preset voltage Vbor1,BOR detection block internal signal BOROUT outputs the low level, internal reset signal IRST will turn to be low level and reset status register RSTSR BOR flag will be set to 1. If power-down detection circuit detect that system power source VDD voltage exceed preset voltage Vborh,BOR detect block internal signal BOROUT output the high level,and internal reset signal IRST will turn to be high level after it lasting 1024 Flrc cycles.

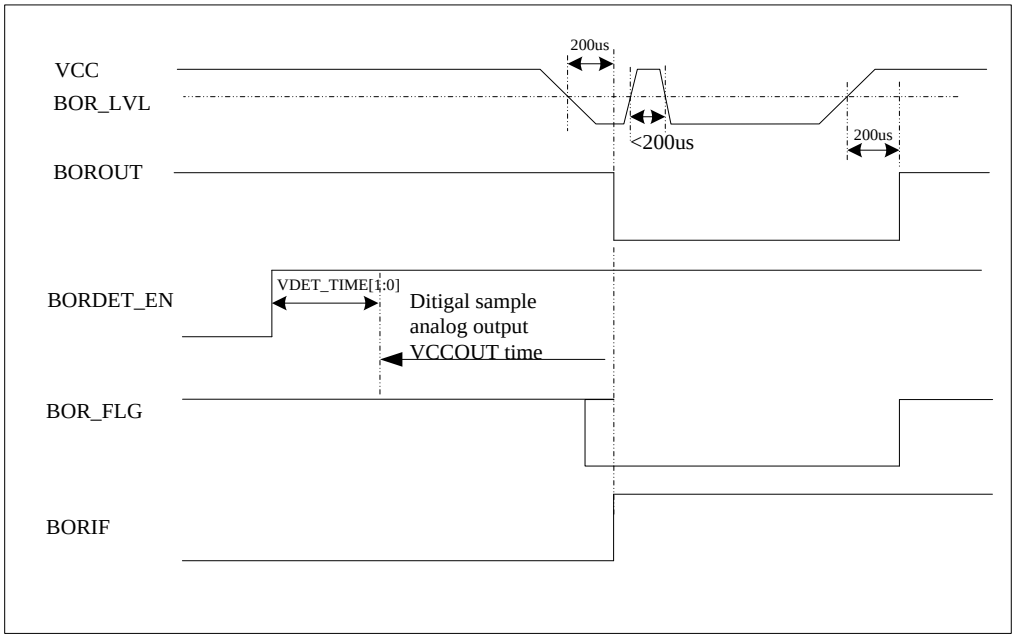
Three is 200mV delay between Vborh and Vborl,BOR block detecting threshold can be set by BOR_LVL[1:0] of VDETCFG.

Situation below will happen after Power-down reset BOR occurs:

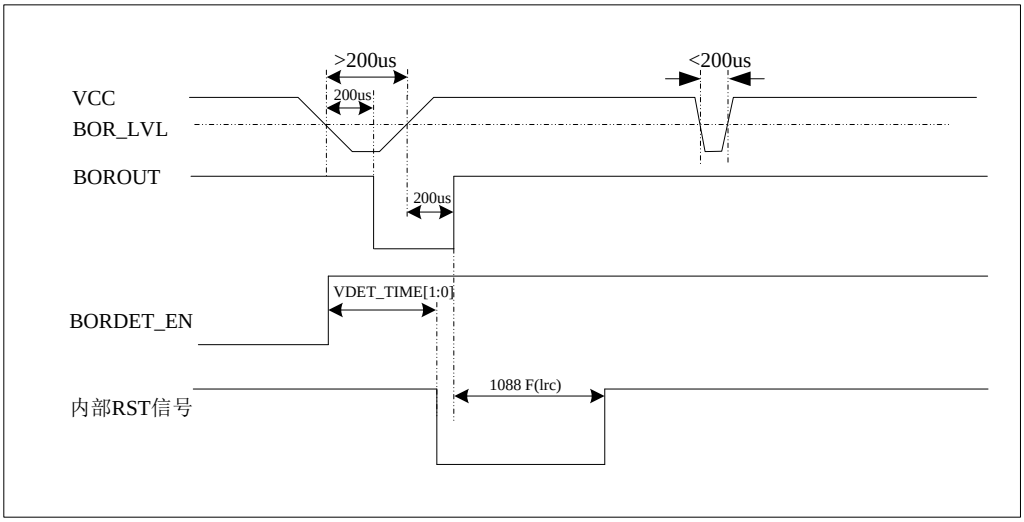
- Generate a BOR pulse
- Internal reset signal IRST will be valid
- Count 1088 Flrc
- Power-down reset flag of reset status register RSTSR BOR will be set to 1
- CPU start execute progress from 0000H

LBOR_DET has the generally same detection process as BOR_DET



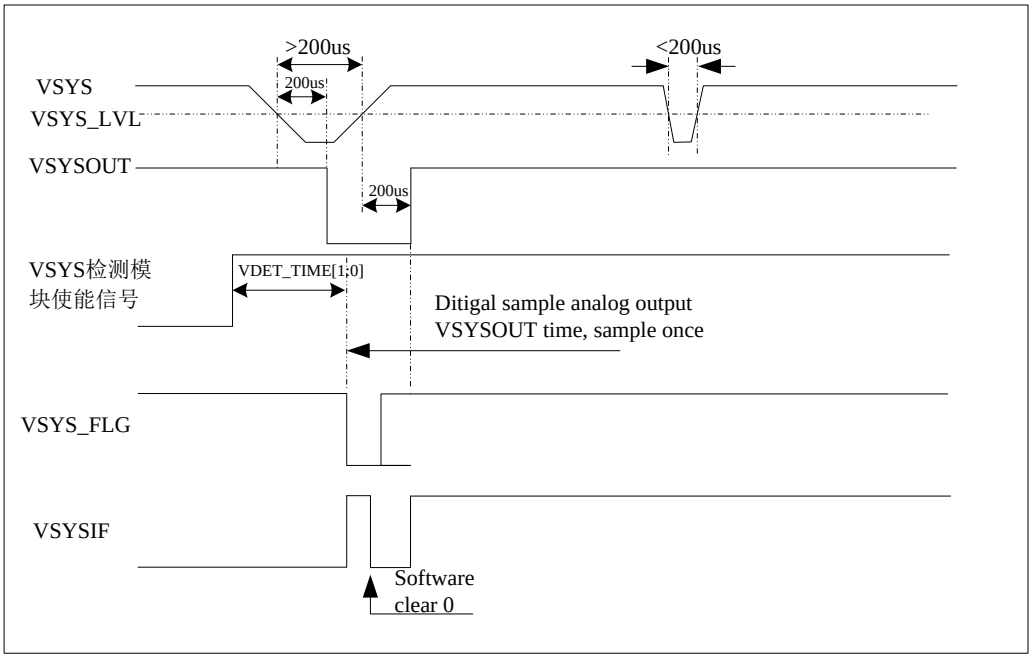



BOR interrupt signal generation diagram

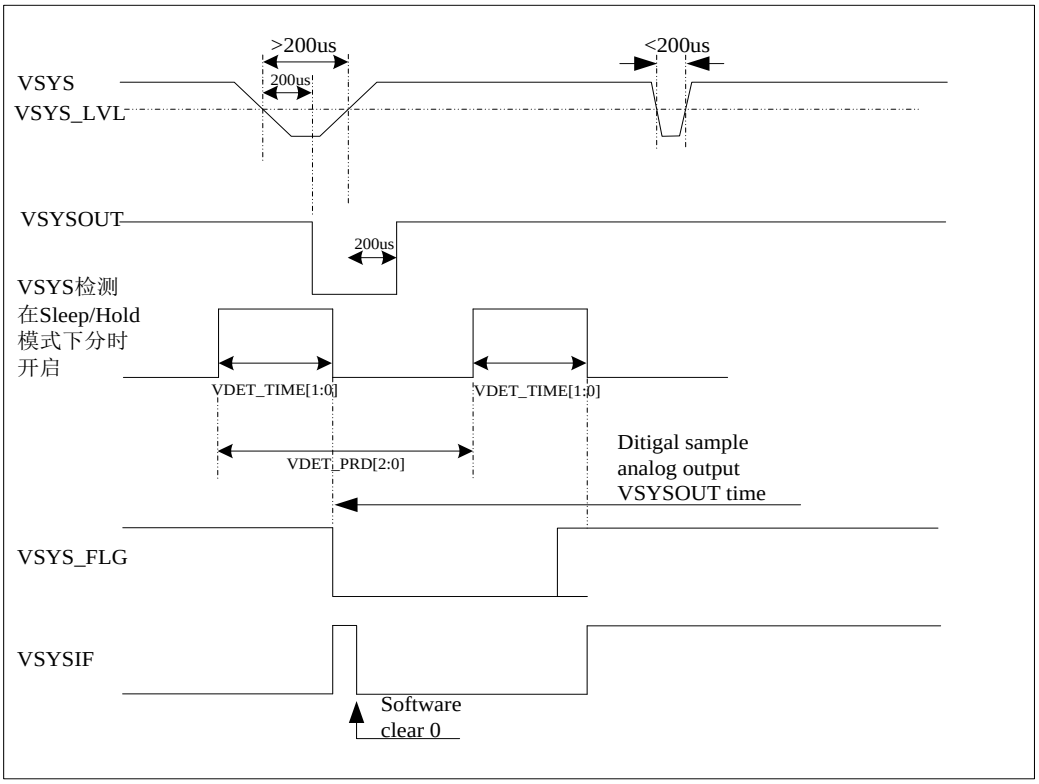



BOR reset diagram

### 4.3.5 System power source detect function

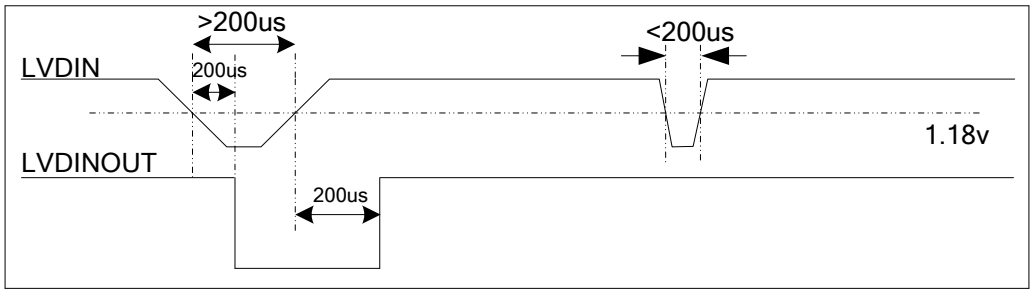



Normal mode Vsys detection




Sleep mode Vsys detect in time division

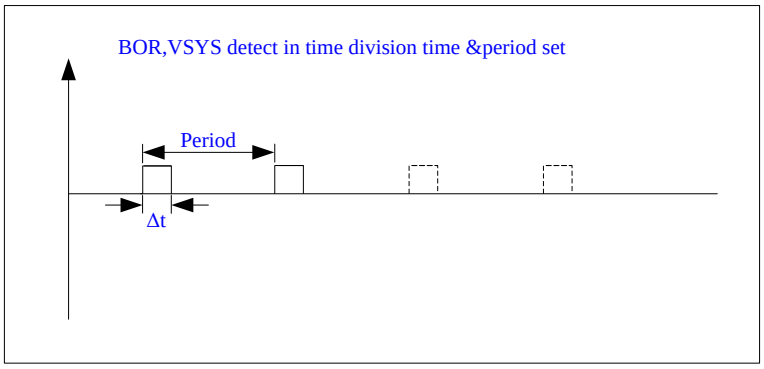
### 4.3.6 Low voltage detect function(LVD_DET)




LVDIN detect signal diagram

### 4.3.7 VSYS_DET, BOR_DET detect in time division

To reduce power consumption when system is in Hold or Sleep mode,LVD_DET、VSYS_DET、BOR_DET adopt the operation mode of opening in time division:




Where Period is the period of opening in time division of LVD_DET、VSYS_DET and BOR_DET detection block and can be configurated by bits VDET_PRD[2:0] in VDETPCFG register. Δt is the waiting time from enabling detection in time division to LVD_DET、VSYS_DET and BOR_DET starting sampling and can be configurated by bits VDET_TIME[1:0] in VDETPCFG register.

**Suggestions in application:**

Use VSYS_DET or BOR_DET block detect system power VSYS voltage when power on,namely detect power-amp status of power source to guarantee that system can shift to a stable status.

Use LVD_DET detect external pin LVDIN voltage when power off,namely detect pre-amp of power source to find the abnormal state of power source and deal with it accordingly.

### 4.3.8 Battery passivation-prevented function

If VBATPin connects to external battery,set DISCHARGE_EN control bit to enable battery passivation

prevented function,and corresponding current flows in VBAT Pin. Users can configure DICHARGE_CTRL to control size of passivation prevented current.

## 4.4 Special function register list

| PMU register base address：0x4000F400 | | | | |
|---|---|---|---|---|
| offset address | name | Read/write | Reset value | Function description |
| 0x00 | PMUCON | R/W | 0x001F | PMU configuration register(write protect) |
| 0x04 | VDETCFG | R/W | 0x0005 | Power detect threshold configuration register |
| 0x08 | VDETPCFG | R/W | 0x0022 | Power detect time period configuration register |
| 0x0C | PMUIE | R/W | 0x0000 | PMU interrupt enable register |
| 0x10 | PMUIF | R/W | 0x0000 | PMU interrupt flag register |
| 0x14 | PMUSTA | R | 0x0000 | PMU status indicate register |
| 0x18 | WAKEIF | R/W | 0x00000000 | wake-up source flag register |

## 4.5 Special function register introduction

| **PMUCON（write protect）（PMU configuration register）** | | | **Base address: 0x4000F400 Offset address: 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | DISCHARGE_EN | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | Hold_LDO | LVDIN1_EN | LVDIN0_EN | BORRST | BORDET_EN |
| **Reset:** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| **Bits** | **Function description** |
|---|---|
| DISCHARGE_EN | **Battery passivation prevented function enable:**<br>0：shut down battery passivation prevented function<br>1：turn on battery passivation prevented function |
| Hold_LDO | **To shut down/turn on high-power LD0 in Hold mode（default enabled）**<br>0：shut down high power<br>1：turn on high power LD0（default） |
| LVDIN1_EN | **LVD_DET enable signal,monitor LVDIN1 input pin** |

| | |
|---|---|
| | 0：shut down LVDIN1_DET<br>1：turn on LVDIN1_DET（default） |
| LVDIN0_EN | **LVD_DET enable signal,monitor LVDIN0 input pin**<br>0：shut down LVDIN0_DET<br>1：turn on LVDIN0_DET（default） |
| BORRST | **BOR reset/interrupt selection bit**<br>0：generate BOR interrupt if VCC voltage below/exceed threshold<br>1：generate BOR reset if VCC voltage below threshold set by VDETCFG[1:0]（default） |
| BORDET_EN | **BORDET_EN enable signal**<br>0：shut down BOR_DET<br>1：turn on BOR_DET（default） |

| **VDETCFG（power source detect threshold configuration register）** | | | **Base address：0x4000F400**<br>**Offset address：04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | DISCHARGE_CTRL | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | RESERVED | X | VSYS_LVL[3:0] | | | | BOR_LVL[1:0] | |
| **Reset:** | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| **Bits** | **Function description** |
|---|---|
| DISCHARGE_CTRL | **Battery passivation prevented function,select size of current：**<br>0：1mA<br>1：2mA |
| BIT7 | **Internal reserved bit, keep value as 1.** |
| VSYS_LVL [3:0] | **VSYS_DET detect threshold control bits** |

| VSYS_LVL[3:0] | | | | Detect voltage |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2.4V |
| 0 | 0 | 0 | 1 | 2.6V |
| 0 | 0 | 1 | 0 | 2.8V（default） |
| 0 | 0 | 1 | 1 | 3.0V |
| 0 | 1 | 0 | 0 | 3.2V |
| 0 | 1 | 0 | 1 | 3.4V |
| 0 | 1 | 1 | 0 | 3.6V |

| | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 1 | 1 | 3.8V |
| | 1 | 0 | 0 | 0 | 4.0V |
| | 1 | 0 | 0 | 1 | 4.2V |
| | 1 | 0 | 1 | 0 | 4.4V |
| | 1 | 0 | 1 | 1 | 4.6V |
| | 1 | 1 | 0 | 0 | 4.8V |
| | 1 | 1 | 0 | 1 | 5V |
| | 1 | 1 | 1 | X | 5V |

BOR_LVL[1:0]

BOR_DET detect threshold control bits

| BOR_LVL[1:0] | | Detect voltage |
|---|---|---|
| 0 | 0 | 2.4V |
| 0 | 1 | 2.2V（default） |
| 1 | 0 | 2.8V |
| 1 | 1 | 2.6V |

| **VDETPCFG（power source detect period configuration register）** | | | **Base address：0x4000F400**<br>**Offset address：08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read**<br>**Write** | X | X | RSRV | VDET_TIME[1:0] | | VDET_PRD[2:0] | | |
| **Reset:** | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

| **Bits** | **Function description** |
|---|---|
| RSRV | Internal reserved bit，POR reset value is 1 by default,other reset cannot change its value.<br>Default value 1 can remain unchanged and can be set to 0 as well |
| VDET_ TIME [1:0] | **Hold&Sleep mode SYS_DET,BOR_DET detect in time division time set** |

| VDET_Time[1:0] | | Detect time |
|---|---|---|
| 0 | 0 | 1068us |
| 0 | 1 | 1068us |
| 1 | 0 | 1068us |
| 1 | 1 | 1068us |

| | |
|---|---|
| VDET_PRD[2:0] | **Hold&Sleep mode VSYS_DET,BOR_DET detect in time division period set** |

| VDET_PRD[2:0] | | | Detect period |
|---|---|---|---|
| 0 | 0 | 0 | 16.5mS |
| 0 | 0 | 1 | 33mS |
| 0 | 1 | 0 | 67mS(default) |
| 0 | 1 | 1 | 134mS |
| 1 | 0 | 0 | 268mS |
| 1 | 0 | 1 | 536mS |
| 1 | 1 | 0 | 1072mS |
| 1 | 1 | 1 | 2144mS |

| **PMUIE (PMU interrupt enable register)** | | | **Base address: 0x4000F400 Offset address: 0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read** **Write** | X | X | X | X | LVDIN1IE | LVDIN0IE | BORIE | VSYSIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:this interrupt will be valid if PMUIE is enabled as well.**

| **Bits** | **Function description** |
|---|---|
| LVDIN1IE | **LVDIN1 detect interrupt enable bit**<br>0：shut down<br>1：enable |
| LVDIN0IE | **LVDIN0 detect interrupt enable bit**<br>0：shut down<br>1：enable |
| BORIE | **BOR detect interrupt enable bit**<br>0：shut down<br>1：enable |
| VSYSIE | **VSYS detect interrupt enable bit**<br>0：shut down<br>1：enable |

Note: setting the interrupt of PMU requires the kernel function to enable the PMU interrupt at the same time.
NVIC_EnableIRQ(PMU_IRQn)

| PMUIF （PMU interrupt flag register） | | | Base address： 0x4000F400 Offset address： 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read** **Write** | X | X | X | X | LVDIN1IF | LVDIN0IF | BORIF | VSYSIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:this register cannot be wake up reset by Wake_UP.**

| **Bits** | **Function description** |
|---|---|
| LVDIN1IF | **LVDIN1 detect interrupt flag** If external pin LVDIN voltage fall below 1.18V or raise above 1.18V,this bit will be set to 1,software clear 0. |
| LVDIN0IF | **LVDIN0 detect interrupt flag** If external pin LVDIN voltage fall below 1.18V or raise above 1.18V,this bit will be set to 1,software clear 0. |
| BORIF | **BOR detect interrupt flag** If internal operating voltage VDD fall below threshold or raise above threshold and BORRST=1,this bit will be set to 1,software clear 0. |
| VSYSIF | **VSYS detect interrupt flag** If system power source VSYS voltage fall below threshold or raise above threshold,this bit will be set to 1,software clear 0. |

Note: this register cannot be reset by Wake_UP wakeup.

| PMUSTA （PMU status register） | | | Base address： 0x4000F400 Offset address： 14H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read** | X | X | X | X | LVDIN1_FLG | LVDIN0_FLG | BOR_FLG | VCC_FLG |
| **Write** | | | | | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0X | 0 | 0 | 0 |

**Note: this register cannot be wake-up reset by Wake_UP.**

| **bits** | **Function description** |
|---|---|
| **LVDIN1_FLG** | **LVDIN1 pin voltage status** |

| | |
|---|---|
| | 0：indicate that LVDIN1 pin voltage is less than threshold 1.18V<br>1：indicate that LVDIN1 pin voltage is greater than threshold 1.18V |
| **LVDIN0_FLG** | **LVDIN0 pin voltage status**<br>0：indicate that LVDIN0 pin voltage is less than threshold 1.18V<br>1：indicate that LVDIN0 pin voltage is greater than threshold 1.18V |
| **BOR_FLG** | **internal operating voltage VDD voltage status**<br>0：indicate that VDD is less than threshold（BOR_LVL[1:0]）<br>1：indicate that VDD is greater than threshold（BOR_LVL[1:0] |
| **VSYS_FLG** | **System power VSYS voltage state**<br>0：indicate that VSYS is less than threshold（VSYS_LVL[1:0]）<br>1：indicate that VSYS is greater than threshold（VSYS_LVL[1:0]） |

Note: this register cannot be reset by Wake_UP wakeup

| **WAKEIF**<br>**（wake-up flag register）** | | | **Base address：** **0x4000F400**<br>**Offset address：** **18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31** | **30** | **29** | **28** | **27** | **26** | **25** | **Bit24** |
| **Read:**<br>**Write:** | NMIWKIF | INT9WKIF | INT8WKIF | INT7WKIF | KEYWKIF | DMAWKIF | EMUWKIF | RSRV |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:**<br>**Write:** | X | SPIWKIF | I2CWKIF | RTCWKIF | TBSWKIF | TMR3WKIF | TMR2WKIF | TMR1WKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | TMR0WKI | RX5WKIF | RX4WKIF | RX3WKIF | RX2WKIF | RX1WKIF | RX0WKIF | INT6WKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | INT5WKIF | INT4WKIF | INT3WKIF | INT2WKIF | INT1WKIF | INT0WKIF | ARGEWKIF | PMUWKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note: Sleep wake-up and Hold wake-up share this flag bit**

| **Bits** | **Function description** |
|---|---|
| NMIWKIF | **NMI wake-up flag**<br>When system wakes up, an optional interrupt NMI is set, the flag bit is set to 1. |
| INT9WKIF | **INT9 wake-up flag**<br>Under SLEEP/HOLD mode, when INT9 wakes up, the flag bit is set to 1. |
| INT8WKIF | **INT8 wake-up flag**<br>Under SLEEP/HOLD mode, when INT8 wakes up, the flag bit is set to 1. |
| INT7WKIF | **INT7 wake-up flag** |

| | |
|---|---|
| | Under SLEEP/HOLD mode, when INT7 wakes up, the flag bit is set to 1. |
| KEYWKIF | **Key scan wake-up flag**<br>Under SLEEP/HOLD mode, when KEY wakes up, the flag bit is set to 1. |
| DMAWKIF | **DMA wake-up flag**<br>Under HOLD mode, when DMA wakes up, the flag bit is set to 1. |
| EMUWKIF | **EMU wake-up flag**<br>Under HOLD mode, when EMU wakes up, the flag is set to 1. |
| RSRV | This bit constant 0 |
| SPIWKIF | **SPI wake-up flag**<br>Under HOLD mode, when SPI wakes up, the flag is set to 1. |
| I2CWKIF | **I2C wake-up flag**<br>Under HOLD mode, when I2C wakes up, the flag is set to 1. |
| WDTWKIF | **WDT interrupt wake up**<br>This flag will be set to 1 if WDT interrupt wake-up occurs. |
| RTCWKIF | **RTC wake-up flag**<br>RTC wake up will occur and this bit will be set to 1 if RTC interrupt is generated in SLEEP mode.(particular RTC wake-up source indicated by RTCIF register) |
| TBSWKIF | **TBS wake-up flag**<br>TBS wake up will occur and this bit will be set to 1 if TBS interrupt is generated in SLEEP mode.(particular TBS wake-up source indicated by TBSIF register) |
| TMR3WKIF | **TMR3 wake-up flag**<br>Under HOLD mode, when TMR3 wakes up, the flag is set to 1. |
| TMR2WKIF | **TMR2 wake-up flag**<br>Under HOLD mode, when TMR2 wakes up, the flag is set to 1. |
| TMR1WKIF | **TMR1 wake-up flag**<br>Under HOLD mode, when TMR1 wakes up, the flag is set to 1. |
| TMR0WKIF | **TMR0 wake-up flag**<br>Under HOLD mode, when TMR0 wakes up, the flag is set to 1. |
| RX5WKIF | **RX5 wake-up flag**<br>If RX5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX4WKIF | **RX4 wake-up flag**<br>If RX4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX3WKIF | **RX3 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX2WKIF | **RX2 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX1WKIF | **RX1 wake-up flag**<br>If RX1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared |

| | |
|---|---|
| | if it happens after hardware is in sleep mode. |
| RX0WKIF | **RX0 wake-up flag**<br>If RX0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI6WKIF | **INT6 wake-up flag**<br>If INT6 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI5WKIF | **INT5 wake-up flag**<br>If INT5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI4WKIF | **INT4 wake-up flag**<br>If INT4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI3WKIF | **INT3 wake-up flag**<br>If INT3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI2WKIF | **INT2 wake-up flag**<br>If INT2 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI1WKIF | **INT1 wake-up flag**<br>If INT1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI0WKIF | **INT0 wake-up flag**<br>If INT0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| ARGEWKIF | **AES/RAND/GHASH/ECC wake-up flag**<br>Under HOLD mode when AES/RAND/GHASH/ECC wakes up, the flag is set to 1. |
| PMUWKIF | **PMU wake-up flag**<br>If PMU occurs in SLEEP mode, PMU wake-up will occur and this flag will be set to 1, otherwise previous flag will be cleared if it happens after hardware is in sleep mode. (particular PMU wake-up source indicated by PMUIF register) |

Note:

1. Sleep wakeup and Hold wakeup share this flag bit
2. DMAWKIF/EMUWKIF/SPIWKIF/I2CWKIF/TMR3WKIF/TMR2WKIF/TMR1WKIF/TMR0WKIF /ARGEWKIF can only wake up in Hold, sleep mode cannot wake up.
3. registers are read-only register, it will always keep on a chip lead to wake source, when a new wake-up event generated by the hardware, generate new wake-up source symbol, while the 0 mark before awakening. This register can only be reset by POR.

# 5 Debugging support

## 5.1 General description

HT502X employ Cortex-M0 kernel which contains hardware debugging module.Processor Cortex-M0 support debugging features shown as below:

- pause、resume and single stepping of progress
- access to processor kernel register and special register
- hardware breakpoint(4 at most)
- software breakpoint(BKPT instruction)
- data monitor point(2 at most)
- dynamic memory access
- support SW debugging protocol

SW adopts 2-pin interface,the information it contains shown as below:

| SW signal | Description |
|---|---|
| SWIO | Data signal |
| SWCLK | Clock signal |

## 5.2 SW pins distribution

| SW debugging interface | | Pins description |
|---|---|---|
| type | description | |
| Input/output | SW data | PB.13/ SEG13/ SWIO |
| output | SW clock | PB.15/ SEG15/ SWCLK |

## 5.3 SW pin function introduction

| Operating mode | Pin function | | |
|---|---|---|---|
| | PA.6/INT1/JTAGWDTEN | PB.13/ SEG13/ SWIO | PB.15/ SEG15/ SWCLK |
| Test mode | X | X | X |
| Debugging mode | JTAGWDTEN | SWIO (OD close) | SWCLK (OD close) |
| Normal mode | PA.6/INT1 | PB.13/SEG13 | PB.15/SEG15 |

Note:

1, X indicates unavailability

2, the system is silent, the test tube foot is second multiplexing function, SW mode, these two ports can enter the 8

debug mode directly without any configuration

3, TEST, JTAGWDTEN feet low, the simulation port will be forced to configure the simulation function, OD function closed, SWTCK fixed input, SWIO output or input (determined by the execution of instructions, dynamic changes).

4, the TEST=0, JTAGWDTEN=1 case, the chip goes into the test mode, and the user should avoid entering this mode.

## 5.4 Debug module reset description

In order to improve the reliability of the simulation and erase operation, the chip debug module design are as follows:

1. In addition to Debug Reset, the debug module can be restored by any other complex Bit ;

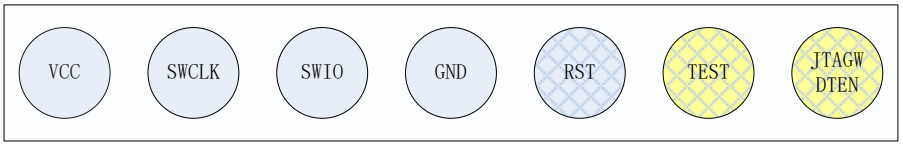
2. When the PB13/PB15 is not used as the SW port (as long as one is not), the debug module is in the complex Bit state;

3. When the TEST and JTAGWDTEN are switched from non 0 (two pins are not all 0) to 0 (two pins are 0), the JTAG module will have a repeat Bit.

4. From the non-Normal mode (TEST pulls low, no matter the other pins) back to the Normal mode (TEST pull up), the chip has once completely re Bit (except the RTC timing register, all other Bit).

## 5.5 Debug Port

The recommended debug ports are as follows:




Explain：

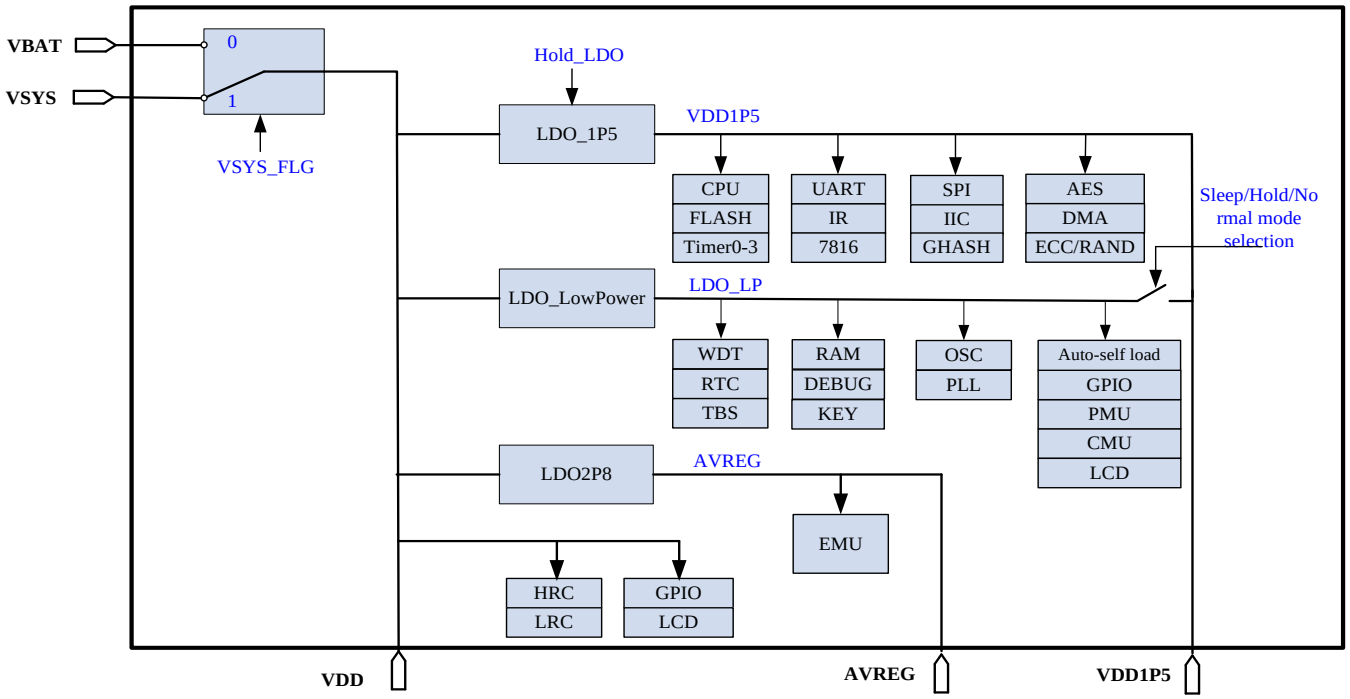
1, the first 4 ports are necessary for program download (if the target board is powered, VCC can be omitted);

2, the role of RST is to provide a reliable compound BitSW state machine when the SW state machine is out of order;

3, the role of TEST and JTAGWDTEN is to configure PB13/PB15 to be the corresponding SW port when users configure SWCLK/SWIO for other functions (such as GPIO or SEG capabilities).

# 6 Operation mode

## 6.1 Distribution of chip power source




## 6.2 Operation mode

There are 5 modes in total:normal mode,debug mode,test mode,Sleep mode,Hold mode

| TEST | JTAGWDTEN | Operation mode |
|---|---|---|
| 0 | 0 | **Debug mode**<br>This mode is mainly used for simulating downloading program<br>PB13 and PB15 are fixed as SW interfaces. |
| 0 | 1 | **Test mode** |
| 1 | X | **Normal mode**<br>All internal modules are normally power supplied,system clock and module shut down/open depends on user software.<br>Watch dog cannot be disabled,unless to turn off internal low frequency RC clock in Sleep/Hold mode.<br>In this mode, users can enter into debug mode by configuring the PB13/15 to be a multiplexed function 2 (i.e., SW port function). |

| | | |
|---|---|---|
| | | **Sleep mode**<br>By executing instruction shown as below, CPU enter Sleep mode in normal mode：<br>CB->SCR = 0x0004;<br>_WFI(); |
| | | **Hold mode**<br>By executing instruction shown as below, CPU enter Hold mode in normal mode：<br>CB->SCR = 0x0000;<br>__WFI(); |

Chip can enter two low power mode，Sleep mode and Hold mode, by software configuration in normal mode.They differ from each other in two main aspects shown as below:

- Sleep mode use lower power
- Wake-up and reset function identically in Sleep mode,while in Hold mode it pick up to execute the code if waked-up

# 6.3 Sleep mode

## 6.3.1 Status of all modules in SLEEP mode

- Turn off digital power source LD0_1P5 and it power supply module will be shut down accordingly.
- RTC related oscillator circuit,TBS module,frequency divide compensate circuit will always working unless RTC power supply module is shut down.
- If users configure to open BOR and VCC detection function in sleep mode,BOR_DET、LBOR_DET and VCC_DET will open in time division by hardware to reduce power.
- WDT is open by default.if WDT counter overflow in sleep mode,system will reset. Configure WDT_EN=0 to shut down WDT in sleep mode(detailed information in CONTROL0 register).
- Set LCD、TBS on before entering sleep mode and LCD static display ,temperature and battery voltage measurement function will be available after entering sleep mode
- To reduce power in sleep mode,users can configure the state of GPIO(detailed in GPIO chapter) to control the chip and peripheral status and prevent GPIO from leaking away before enter sleep mode.
- To achieve the lowest power:
  （1） Clear CLKCTRL0 and CLKCTRL1 register
  （2） Clear LRC_EN of CTRLBYFLASH, shut down internal low frequency RC
- Chip will close LD0_1P5 (high power) output after entering sleep mode,LD0_LowPower(low power) keep 1.5V output.

## 6.3.2 Wake up in SLEEP mode

Waking up CPU from sleep mode has the same function with resetting:so it will not enter interrupt vector and

no interrupt source processor will be executed and process start running from reset address 000H.

Reset signal,generated by POR，BOR，LBOR，external RESET PIN and internal WDT reset signal included, cannot be masked. If these reset signals are generated after system entering SLEEP mode,chip will be reset and processor will be executed from 000H.

To achieve the wake-up function in sleep mode,some function should be configured before entering sleep mode:

(1)External INT interrupt and serial port RX pin wake up:Corresponding PIN need to configured to be INT and RX function(detailed in GPIO unit) and enable external interrupt and UART interrupt. If RX pin wake up and high/low level flip and keeps it for at least 2 Fcpu on corresponding PIN,CPU can wake up from SLEEP mode;if external INT wake up and trigger edge emerges and high/low level keeps for at least 2 Fcpu on corresponding PIN,CPU can wake up from SLEEP mode,trigger edge type is configured by RIE and FIE control bits of EXTIE. Note：where Fcpu clock is the output of clock frequency division register. If the division value of CPU clock frequency division SYSCLVDIV is too great before enter SLEEP mode, a long period of low level given to external wake up PIN will be necessary to wake up chip from sleep mode.

(2)RTC interrupt wake up: configure RTC main interrupt and corresponding RTC sub-interrupt source(only for corresponding enable bits of RTCIE), if RTC interrupt or RTC enabled alarm and timer time is up,CPU can be woken up from SLEEP.

(3)PWU interrupt and TBS interrupt wake up: Configure PMU,enable TBS main interrupt and corresponding TBS sub-interrupt source(configure corresponding interrupt enable bits of PMUIE and TBSIE).if power source or temperature is detected to exceed the threshold; CPU can be woken up from sleep mode.

(4) KEY interrupt wakeup: the corresponding PIN should be configured as the KEY scan function, configure the button scan module, interrupt enable (NVIC_EnableIRQ (KEY_IRQn)), and when the key is pressed, CPU wakes from SLEEP mode.

### 6.3.3 Wake up mode confirmation after SLEEP mode wakeup

Reset flag(WakeupRST and Sleep_Flag bits of register RSTSTA) can be queried after waking up from SLEEP mode. If both of the flags are 1,it can be show that a wake-up reset happens indeed,then wake up flag register WAKEIF can be checked to find out the specific wake-up source,where:

1) If RTCWKIF of WAKEIF is 1,it can be shown that RTC interrupt signals trigger the wake-up of CPU. Specific interrupt sources ,of which 8 can trigger it,can be confirmed by checking corresponding bits of RTCIF .detailed in RTC unit.
2) If RTCIF of WAKEIF is 1,it can be shown that PMU interrupt signals trigger the wake-up of CPU. Specific interrupt sources ,of which 4 can trigger it,can be confirmed by checking corresponding bits(BORIF、VCCIF、LVDIF、POWIF) of PMUIF .detailed in PMU unit.
3) If TBSWKIF of WAKEIF is 1,it can be shown that TBS interrupt signals trigger the wake-up of CPU. Specific interrupt sources ,of which 9 can trigger it,can be confirmed by checking corresponding bits(TPSIF、VbatIF、VccIF、ADC0IF、ADC1IF、ADC2IF、VbatCMPIF、ADC0CMPIF、ADC1CMPIF) of TBSIF. Detailed in TBS unit.
4) If INTxWKIF(x=0~6) of WAKEIF is 1,it can be shown that corresponding external PIN signals trigger the wake-up of CPU.
5) If RXxWKIF(x=0~5) of WAKEIF is 1,it can be shown that corresponding RX pin signals trigger the

wake-up of CPU.

### 6.3.4 Enter sleep mode

By calling CortexM0 system-provided instructions system can enter Sleep mode and it goes for it when system is in debugging mode. Specific instructions are shown as below:

```
SCB->SCR = 0x0004;
__WFI();
```

## 6.4 Hold mode

Hold mode and Sleep mode differ from each other in the following aspects.In Hold mode, LD0_1P5 and LD0_LowPower are both control by users and LD0_LowPower power keeps supplying,but many digital function modules are not available due to its low output drive(20-30uA) and users can configure to open LD0_1P5 of high power output drive.

To reduce power in Hold mode,BOR_DET,VCC_DET modules are turning on in time division by hardware.

If interrupt enable is configured before system entering Hold mode,interrupt event happens after entering Hold mode will lead to chip wake-up from Hold mode and corresponding interrupt processing program will be executed.

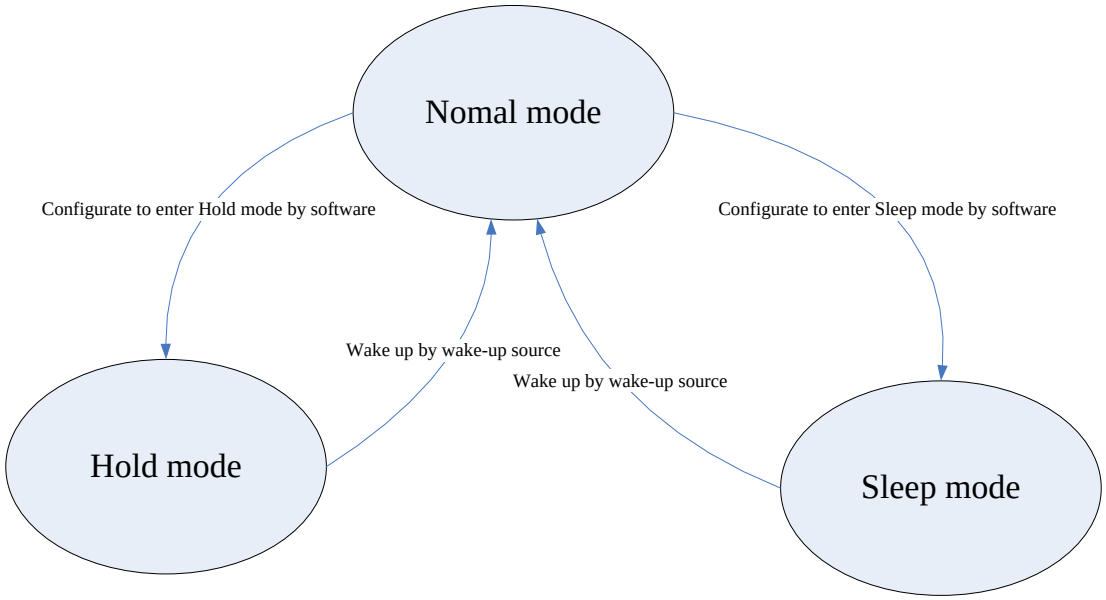
WDT is open by default. If WDT counter overflows in Hold mode, system will be reset by WDT.Users can configure LRC_EN=0 to shut down WDT in Hold mode(detailed in CTRLBYFLASH register).

### Enter Hold mode

By calling CortexM0 system-provided instructions WFI, system can enter Hold mode.

Specific instructions are shown as below:

```
SCB->SCR = 0x00;
__WFI();
```




## 6.5 Special function register list

| **Base address：0xE000ED00** | | | | |
|---|---|---|---|---|
| Offset address | name | Write/read | Reset value | Function description |
| 0x10 | SCR | R/W | 0x0000 | System control register |
| **Module register base address：0x4000F400** | | | | |
| Offset address | name | Write/read | Reset value | Function description |
| 0x18 | WAKEIF | R/W | 0x0000 | Wake-up source flag register |

## 6.6 Special function register introduction

| **SCR（system control register）** | | | **Base address：0xE000ED10<br>Offset address：10H** | | | | |
|---|---|---|---|---|---|---|---|
| | **Bit31** | **30** | **29** | **28…11** | **10** | **9** | **Bit8** |
| **Read:<br>Write:** | SCR[31:8] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:<br>Write:** | | | | RESERVED | | SLEEPDEEP | RESERVED | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|

| | |
|---|---|
| SLEEPDEEP | =1：enter Sleep mode after call WFI instruction<br>=0：enter Hold mode after call WFI instruction |

| **WAKEIF**<br>**（wake-up flag register）** | | | **Base address：** **0x4000F400**<br>**Offset address：** **18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31** | **30** | **29** | **28** | **27** | **26** | **25** | **Bit24** |
| **Read:**<br>**Write:** | NMIWKIF | INT9WKIF | INT8WKIF | INT7WKIF | KEYWKIF | DMAWKIF | EMUWKIF | RSRV |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:**<br>**Write:** | X | SPIWKIF | I2CWKIF | RTCWKIF | TBSWKIF | TMR3WKIF | TMR2WKIF | TMR1WKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | TMR0WKI | RX5WKIF | RX4WKIF | RX3WKIF | RX2WKIF | RX1WKIF | RX0WKIF | INT6WKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | INT5WKIF | INT4WKIF | INT3WKIF | INT2WKIF | INT1WKIF | INT0WKIF | ARGEWKIF | PMUWKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note: Sleep wake-up and Hold wake-up share this flag bit**

| **Bits** | **Function description** |
|---|---|
| NMIWKIF | **NMI wake-up flag**<br>When system wakes up, an optional interrupt NMI is set, the flag bit is set to 1. |
| INT9WKIF | **INT9 wake-up flag**<br>Under SLEEP/HOLD mode, when INT9 wakes up, the flag bit is set to 1. |
| INT8WKIF | **INT8 wake-up flag**<br>Under SLEEP/HOLD mode, when INT8 wakes up, the flag bit is set to 1. |
| INT7WKIF | **INT7 wake-up flag**<br>Under SLEEP/HOLD mode, when INT7 wakes up, the flag bit is set to 1. |
| KEYWKIF | **Key scan wake-up flag**<br>Under SLEEP/HOLD mode, when KEY wakes up, the flag bit is set to 1. |
| DMAWKIF | **DMA wake-up flag**<br>Under HOLD mode, when DMA wakes up, the flag bit is set to 1. |
| EMUWKIF | **EMU wake-up flag**<br>Under HOLD mode, when EMU wakes up, the flag is set to 1. |
| RSRV | This bit constant 0 |
| SPIWKIF | **SPI wake-up flag**<br>Under HOLD mode, when SPI wakes up, the flag is set to 1. |

| | |
|---|---|
| I2CWKIF | **I2C wake-up flag**<br>Under HOLD mode, when I2C wakes up, the flag is set to 1. |
| WDTWKIF | **WDT interrupt wake up**<br>This flag will be set to 1 if WDT interrupt wake-up occurs. |
| RTCWKIF | **RTC wake-up flag**<br>RTC wake up will occur and this bit will be set to 1 if RTC interrupt is generated in SLEEP mode.(particular RTC wake-up source indicated by RTCIF register) |
| TBSWKIF | **TBS wake-up flag**<br>TBS wake up will occur and this bit will be set to 1 if TBS interrupt is generated in SLEEP mode.(particular TBS wake-up source indicated by TBSIF register) |
| TMR3WKIF | **TMR3 wake-up flag**<br>Under HOLD mode, when TMR3 wakes up, the flag is set to 1. |
| TMR2WKIF | **TMR2 wake-up flag**<br>Under HOLD mode, when TMR2 wakes up, the flag is set to 1. |
| TMR1WKIF | **TMR1 wake-up flag**<br>Under HOLD mode, when TMR1 wakes up, the flag is set to 1. |
| TMR0WKIF | **TMR0 wake-up flag**<br>Under HOLD mode, when TMR0 wakes up, the flag is set to 1. |
| RX5WKIF | **RX5 wake-up flag**<br>If RX5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX4WKIF | **RX4 wake-up flag**<br>If RX4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX3WKIF | **RX3 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX2WKIF | **RX2 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX1WKIF | **RX1 wake-up flag**<br>If RX1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX0WKIF | **RX0 wake-up flag**<br>If RX0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI6WKIF | **INT6 wake-up flag**<br>If INT6 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI5WKIF | **INT5 wake-up flag**<br>If INT5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |

| | |
|---|---|
| EXTI4WKIF | **INT4 wake-up flag**<br>If INT4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI3WKIF | **INT3 wake-up flag**<br>If INT3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI2WKIF | **INT2 wake-up flag**<br>If INT2 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI1WKIF | **INT1 wake-up flag**<br>If INT1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI0WKIF | **INT0 wake-up flag**<br>If INT0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| ARGEWKIF | **AES/RAND/GHASH/ECC wake-up flag**<br>Under HOLD mode when AES/RAND/GHASH/ECC wakes up, the flag is set to 1. |
| PMUWKIF | **PMU wake-up flag**<br>If PMU occurs in SLEEP mode, PMU wake-up will occur and this flag will be set to 1, otherwise previous flag will be cleared if it happens after hardware is in sleep mode. (particular PMU wake-up source indicated by PMUIF register) |

Note:

1. Sleep wakeup and Hold wakeup share this flag bit
2. DMAWKIF/EMUWKIF/SPIWKIF/I2CWKIF/TMR3WKIF/TMR2WKIF/TMR1WKIF/TMR0WKIF /ARGEWKIF can only wake up in Hold, sleep mode cannot wake up.
3. registers are read-only register, it will always keep on a chip lead to wake source, when a new wake-up event generated by the hardware, generate new wake-up source symbol, while the 0 mark before awakening. This register can only be reset by POR.

# 7 GPIO module

## 7.1 General introduction

HT5023 provide **PA[0..13], PB[0..15], PC[0..14], PD[0..15], PE[0..9]， PF[0..7]** parallel port, support 79 bidirectional I/O pins which can be configured to output or input port. And internal 88K pull-up resistor is available when it functions as input.

HT5025 provide **PA[3..8]，PA[11..13], PB[0..15], PC[0..3]，PC8, PC[10..12]， PD[0..6], PD[8..15]，PE[1..5], PE[7..9], PF[0..2]** parallel port, support 59 bidirectional I/O pins which can be configured to output or input port. And internal 88K pull-up resistor is available when it functions as input.

HT5027 provide **PA3，PA[5..8]，PA12, PB[0..7], PB[12..15]，PC[2..3]，PC8，PD[2..5], PD[8..15],PE[1..5], PE7， PE9, PF[0..2]**parallel port, support 43 bidirectional I/O pins which can be configured to output or input port. And internal 88K pull-up resistor is available when it functions as input.

All I/O has 3mA drive capacity at least,part of whom has high drive capacity as introduced.

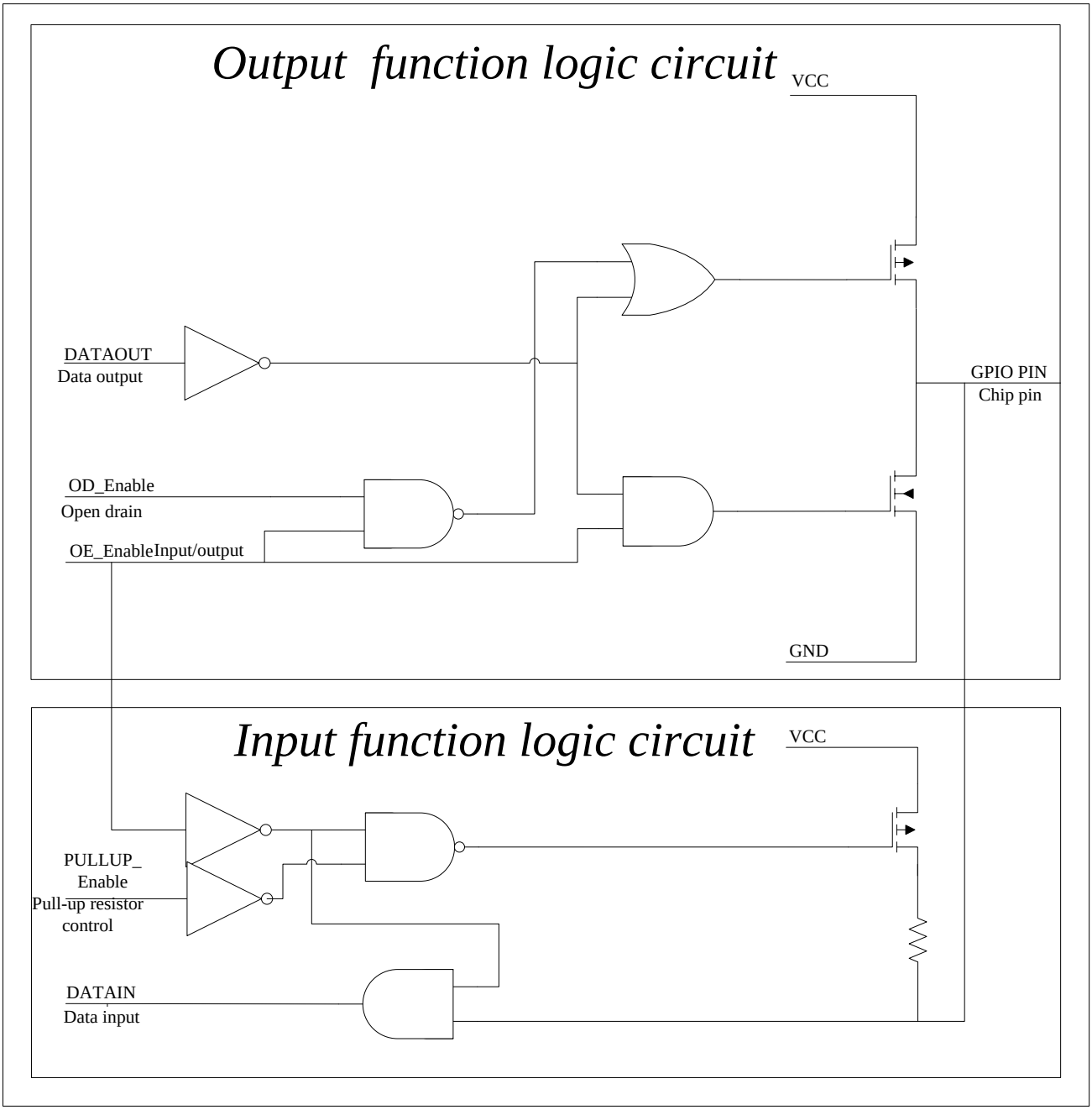



Fig.Chip pins structure introduction

## 7.2 Chip pin description

Refer to 1.6

.

# 7.3I/O port base address list I/O

| GPIO module register base address:<br>0x40011000(PA port);<br>0x40011200(PC port);<br>0x40011300(PD port);<br>0x40011400(PE port); | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Read/write** | **Reset value** | **Function description** |
| 0x00 | IOCFG | R/W | 0x0000 | Port function configuration register1 |
| 0x04 | AFCFG | R/W | 0x0000 | Port function configuration register2 |
| 0x08 | PTDIR | R/W | 0x0000 | Port direction configuration register |
| 0x0C | PTUP | R/W | 0x0000 | Port pull-up configuration register |
| 0x10 | PTDAT | R/W | 0x0000 | Port data configuration register |
| 0x14 | PTSET | W | 0x0000 | Port configuration register (write-only) |
| 0x18 | PTCLR | W | 0x0000 | Port reset register (write-only) |
| 0x1C | PTTOG | W | 0x0000 | Port flip register (write-only) |
| 0x20 | PTOD | R/W | 0xFFFF | Port 0D function configuration register |

| GPIO module register base address:<br>0x40011100(PB port); | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Read/write** | **Reset value** | **Function description** |
| 0x00 | IOCFG | R/W | 0xA000 | Port function configuration register1 |
| 0x04 | AFCFG | R/W | 0xA000 | Port function configuration register2 |
| 0x08 | PTDIR | R/W | 0x0000 | Port direction configuration register |
| 0x0C | PTUP | R/W | 0x0000 | Port pull-up configuration register |
| 0x10 | PTDAT | R/W | 0x0000 | Port data configuration register |
| 0x14 | PTSET | W | 0x0000 | Port configuration register (write-only) |
| 0x18 | PTCLR | W | 0x0000 | Port reset register (write-only) |
| 0x1C | PTTOG | W | 0x0000 | Port flip register (write-only) |
| 0x20 | PTOD | R/W | **0xFFFF** | Port 0D function configuration register |

Note:

PB15/SEG15/SWCLK default function is SWCLK

PB13/SEG13/SWIO default function isSWIO

That is, HT_GPIOB->IOCFGdefault function is0xA000

HT_GPIOB->AFCFGdefault function is0xA000

| GPIO module register base address:<br>0x40011600(PF port); | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Read/write** | **Reset value** | **Function description** |
| 0x00 | IOCFG | R/W | 0x0007 | Port function configuration register1 |

| | | | | |
|---|---|---|---|---|
| 0x04 | AFCFG | R/W | 0x0000 | Port function configuration register2 |
| 0x08 | PTDIR | R/W | 0x0000 | Port direction configuration register |
| 0x0C | PTUP | R/W | 0x0000 | Port pull-up configuration register |
| 0x10 | PTDAT | R/W | 0x0000 | Port data configuration register |
| 0x14 | PTSET | W | 0x0000 | Port configuration register (write-only) |
| 0x18 | PTCLR | W | 0x0000 | Port reset register (write-only) |
| 0x1C | PTTOG | W | 0x0000 | Port flip register (write-only) |
| 0x20 | PTOD | R/W | **0xFFFF** | Port 0D function configuration register |

Note:

After reset, PF.0/PFOUT/TOUT3 default function is PFOUT.

After reset, PF.1/QFOUT default function is QFOUT.

After reset, PF.2/SFOUT/SEG48 default function is SFOUT

That is, HT_GPIOF->IOCFG default value is 0x0007.

HT_GPIOF->AFCFG default value is 0x0000.

| **GPIO module register base address： 0x40011500(large current port)；** | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Read/write** | **Reset value** | **Function description** |
| 0x00 | HDPORT | R/W | 0x0000 | Large current port configuration register |

## 7.4Special function register introduction

| **IOCFG（port function configuration registe1）** | | | **Base address： 0x40011000--0x40011400**<br>**Offset address： 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port IO function configure bits**<br>0：port as GPIO<br>1： port as PIN |

| **AFCFG（port function configuration** | **Base address： 0x40011000--0x40011400**<br>**Offset address： 04H** |
|---|---|

| registe2） | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port multiplex function configuration bits（valid only if corresponding port is configured as PIN）**<br>0：multiplex function1<br>1：multiplex function2 |

| **PTDIR（port direction configuration register）** | | | **Base address：0x40011000--0x40011400**<br>**Offset address：08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **port direction configuration bits（valid only if corresponding port is configured as GPIO）**<br>0：input<br>1：output |

| **PTUP（port pull-up configuration register）** | | | **Base address：0x40011000--0x40011400**<br>**Offset address：0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | PT[15:8] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | PT[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port pull-up configuration bits（valid only if corresponding port is configured as digital input）**<br>0：enable pull-up<br>1：disable pull-up(high-impedance-state) |

Note:

1. Chip pin select GPIO function:

If the direction register is configured as output, the pull-up control is invalid;

If the direction register is configured as input, the pull-up control is valid;

2. Chip pin selection multiplexing function:

The multiplexed output function of the digital output pin is invalid.

The multiplexed digital input pins (except the RST/TEST/JTAGWDTEN, the three pins, and the internal constant pull-up) can be provided with pull functions;

If it is configured as analog input, the pull-up control is invalid;

3. Multiplexing functions for analog functions: LVDINx, LCD_SEG\COM, POWIN, ADCINx, VBAT, in addition to other multiplexing functions for digital functions.

| **PTDAT（port data register）** | | | **Base address：0x40011000--0x40011400**<br>**Offset address：10H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | PT[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | PT[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port data bits（valid only if corresponding port is configured as GPIO）**<br>**Read IO state if configured as input**<br>0：low level<br>1：high level<br>**If configured as output**<br>0：output low level |

| | |
|---|---|
| | 1：output high level |

Note:

1. Chip pin selects the GPIO function or multiplexed digital functions

If the direction register is configured as output, the PTDAT read value is a register setting value, and does not change with the change of the external PIN foot level;

If the direction register is configured as input, the PTDAT read value is pad state value, which reflects the change of the external PIN foot level;

2. Chip pin selection multiplexing simulation function, PTDAT corresponding bit bit value, fixed to 0

| **PTSET（port configuration register）** | | | **Base address：0x40011000--0x40011400 Offset address：14H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:only can be written for this register**

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port set bits（valid only if corresponding port is configured as GPIO and output）**<br>0：invalid for writing 0<br>1：write 1 to output high level for corresponding port(update corresponding value of PxDA at mean time) |

| **PTCLR（port reset register）** | | | **Base address：0x40011000--0x40011400 Offset address：18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:only can be written for this register**

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port reset bits（valid only if corresponding port is configurated as GPIO and output）** |

| | |
|---|---|
| | 0：invalid for writing 0<br>1：write 1 to output low level for corresponding port(update corresponding value of PxDA at mean time) |

| **PTTOG（port flip register）** | | | **Base address：0x40011000--0x40011400<br>Offset address：1CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:only can be written for this register**

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port flip bits（valid only if corresponding port is configured as GPIO and output）**<br>0：invalid for writing 0<br>1：write 1 to corresponding port output level will be flipped(update corresponding value of PxDA at mean time) |

| **PTOD（port open-drain configuration register）** | | | **Base address：0x40011000--0x40011400<br>Offset address：20H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | PT[15:8] | | | | | | | |
| **Reset:** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | PT[7:0] | | | | | | | |
| **Reset:** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| **Bits** | **Function description** |
|---|---|
| PT[15:0] | **Port open-drain configuration bits（valid only if corresponding port is configured as digital output）**<br>0：enable open-drain function（disable PMOS）<br>1：disable open-drain function（enable PMOS）<br>Note: some aspects should be paid attention. Take ICC for example, if users configured IO port as IIC function, open-drain function should be enabled. |

Note:

1. Chip pin select GPIO function:

   If the directional register is configured for output, the open drain OD function is controlled effectively;

   If the direction register is configured as input, the open drain OD function is invalid;

2. Chip pin selection multiplexing function:

   The multiplexed digital output pins are available with Open Drain.

   The multiplexed function of the digital input pin opens the drain function invalid;

   If configured as analog input, open drain OD control is invalid;

3. Multiplexing functions for analog functions: LVDINx, LCD_SEG\COM, POWIN, ADCINx, VBAT, in addition to other multiplexing functions for digital functions.

| **HDPORT（lager current port configuration register）** | | | **Base address：0x40011500**<br>**Offset address：00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | PA8HD | PA7HD | PA6HD | PC0HD |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note:this register has independent beginning address**

| **bits** | **Function description** |
|---|---|
| PA8HD,PA7HD, PA6HD,PC0HD | **Port lager current derive configuration(for PA8,PA7,PA6 and PC0 port respective）**<br>0：normal drive capacity<br>1：extraordinary current derive capacity（20mA-30mA） |

# 8 Interrupt module

## 8.1 Interrupt vector introduction

| System interrupt | Interrupt num | Interrupt enable | Interrupt flag | Function description |
|---|---|---|---|---|
| NMI | -14 | | | Hardware switch low frequency RC interrupt forcibly |
| HardFault | -13 | | | Fault/exception trigger interrupt |
| SVCall | -5 | | | Software trigger interrupt |
| PendSV | -2 | | | Software trigger interrupt |
| SysTick | -1 | | | System timer period interrupt |
| Above interrupts comes with kernel | | | | |
| PMU | 0 | PMUIE.POWIE | PMUIF.POWIF | POW detection interrupt |
| | | PMUIE.LVDIE | PMUIF.LVDIF | LVD detection interrupt |
| | | PMUIE.BORIE | PMUIF.BORIF | BOR detection interrupt |
| | | PMUIE.VCCIE | PMUIF.VCCIF | VCC detection interrupt |
| AES | 1 | AESGHASEIE.AESIE | AESGHASEIF.AESIF | AES interrupt |
| | | AESGHASEIE.GHASHIE | AESGHASEIF.GHASHIF | GHASH interrupt |
| | | AESGHASHIE.RANDIE | AESGHASHIF.RANDIF | RAND interrupt |
| | | ECCCON.ECCIE | ECCSTA.ECCFLG | ECC interrupt |
| EXTI0-6 | 2-8 | EXTIE.RIE[0:6] | EXTIF.RIF[0:6] | External input pins rising edge interrupt |
| | | EXTIE.FIE[0:6] | EXTIF.FIF[0:6] | External input pins falling edge interrupt |
| UART0-5（UART 3 and UART 4 functions has 7816） | 9-14 | UARTCFG.RXIE | UARTSTA.RXIF | UART receive interrupt |
| | | UARTCFG.TXIE | UARTSTA.TXIF | UART send interrupt |
| | | 7816CFG.OVERIE | 7816STA.OVERIF | 7816 overflow interrupt（SCI3,SCI4） |
| | | 7816CFG.RXIE | 7816STA.RXIF | 7816 receive interrupt SCI3,SCI4） |
| | | 7816CFG.TXIE | 7816STA.TXIF | 7816 send interrupt（SCI3,SCI4） |
| TMR0-3 | 15-18 | TMRIE.CMPIE | TMRIF.CMPIF | compare interrupt |
| | | TMRIE.CAPIE | TMRIF.CAPIF | Capture interrupt |
| | | TMRIE.OVERIE | TMRIF.OVERIF | Periodically overflow interrupt |

| | | | | |
|---|---|---|---|---|
| TBS | 19 | TBSIE.ADC1CMPIE | TBSIF.ADC0CMPIF | ADC1 compare interrupt |
| | | TBSIE.ADC0CMPIE | TBSIF.ADC0CMPIF | ADC0 compare interrupt |
| | | TBSIE.VBATCMPIE | TBSIF.VBATCMPIF | Battery voltage compare interrupt |
| | | TBSIE.ADC2IE | TBSIF.ADC2IF | ADC channel2 measure interrupt |
| | | TBSIE.ADC1IE | TBSIF.ADC1IF | ADC channel1 measure interrupt |
| | | TBSIE.ADC0IE | TBSIF.ADC0IF | ADC channel0 measure interrupt |
| | | TBSIE.VCCIE | TBSIF.VCCIF | Power source voltage measure interrupt |
| | | TBSIE.VBATIE | TBSIF.VBATIF | Battery voltage measure interrupt |
| | | TBSIE.TMPIE | TBSIF.TMPIF | Temperature measure interrupt |
| RTC | 20 | RTCIE.ALMIE | RTCIF.ALMIF | Alarm interrupt |
| | | RTCIE.RTC2IE | RTCIF.RTC2IF | RTC timer2 interrupt |
| | | RTCIE.RTC1IE | RTCIF.RTC1IF | RTC timer1 interrupt |
| | | RTCIE.MTHIE | RTCIF.MTHIF | RTC month interrupt |
| | | RTCIE.DAYIE | RTCIF.DAYIF | RTC day interrupt |
| | | RTCIE.HRIE | RTCIF.HRIF | RTC hour interrupt |
| | | RTCIE.MINIE | RTCIF.MINIF | RTC minute interrupt |
| | | RTCIE.SECIE | RTCIF.SECIF | RTC second interrupt |
| I2C | 21 | | I2CCON.SI | |
| SPI | 22 | | SPIF | |
| | | | MODF | |
| EMU | 25 | | See EMU introduction for ub-interrupt flag | EMU interrupt |
| DMA | 26 | DMAIE.TCIE | DMAIF.TCIF | DMA transmission completed interrupt |
| | | DMAIE.BCIE | DMAIF.BCIF | DMA block transmission completed interrupt |
| | | DMAIE.TEIE | DMAIF.TEIF | DMA transmission error interrupt |
| KEY | 27 | | KEYIF.KEYIF | KEY interrupt |
| EXTI7-9 | 28-30 | EXTIE2.RIE[2:0] | EXTIF2.RIF[2:0] | External input pins rising edge interrupt |
| | | EXTIE2.FIE[2:0] | EXTIF2.FIF[2:0] | External input pins falling edge interrupt |

Note:

1. Only UART3 and UART4 can function as 7816,so as to both two ports can be 7816 overflow interrupted,7816 receive interrupted and 7816 send interrupted.

2. Every interrupt whose interrupt numb is larger than 0 has a corresponding interrupt enable signal, see 21.4 CMSIS function declaration for detailed configuration.

3. NMI/HardFault/SVCall/PendSV/SysTick comes with kernel and has no special interrupt enable control bits.

## 8.2 Interrupt enable and disable

Interrupt enable and disable use NVIV module of Cortexm0 kernel and functions comes with CortexM0 software system:

void NVIC_EnableIRQ(IRQn_Type IRQn)

void NVIC_DisableIRQ(IRQn_Type IRQn)

void NVIC_SetPendingIRQ(IRQn_Type IRQn)

void NVIC_ClearPending IRQ(IRQn_Type IRQn)

unint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn)

void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)

uint32_t NVIC_GetPrioriy(IRQn_Type IRQn)

### 8.2.1 Interrupt enable and disable related registers list

| Register address | name | Waite/read | Reset value | Function description |
|---|---|---|---|---|
| 0xE000E100 | ISER | R/W | 0x00000000 | Interrupt enable configuration register |
| 0xE000E180 | ICER | R/W | 0x00000000 | Interrupt disable configuration register |
| 0xE000E200 | ISPR | R/W | 0x00000000 | Interrupt flag set register |
| 0xE000E280 | ICPR | R/W | 0x00000000 | Interrupt flag clear register |
| 0xE000E400-0xE000E41C | IPR0-IPR7 | R/W | 0x00000000 | Interrupt priority register |

### 8.2.2 Interrupt enable and disable related register introduction

| ISER（interrupt enable configuration register） | | | Register address：0xE000E100 | | | | |
|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | |
| Read: | SETENA[31:0] | | | | | | |
| Write: | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Interrupt enable register, all 32 control bits correspond to 32 interrupts, see interrupt vector introduction for bits correspondence, for example:

SETNA[0] corresponds to PMU

SETNA[1] corresponds to 3DES

SETNA[2] corresponds to EXTI0
SETNA[3] corresponds to EXTI1
Recommend to use NVIC_EnableIRQ of CortexM0 function lib to enable interrupt.

| ICER（interrupt disable configurate register） | | | Register address：0xE000E180 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | CLRENA[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Interrupt disable register, all 32 control bits correspond to 32 interrupts, see interrupt vector introduction for bits correspondence.
Recommend to use NVIC_DisableIRQ of CortexM0 function lib to disable interrupt

| ISPR（interrupt flag set resister） | | | Register address：0xE000E200 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | SETPEND[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Interrupt flag set register, all 32 control bits correspond to 32 interrupts, see interrupt vector introduction for bits correspondence.
Recommend to use NVIC_SetPendingIRQ of CortexM0 function lib to set interrupt flag

| ICPR（interrupt flag clear register） | | | Register address：0xE000E280 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | CLRPEND[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Interrupt flag clear register, all 32 control bits correspond to 32 interrupts, see interrupt vector introduction for bits correspondence.
Recommend to use NVIC_ClearPendingIRQ of CortexM0 function lib to clear interrupt flag

| IPR0 --- IPR7（interrupt priority register） | Register address：0xE000E400--0xE000E417 |
|---|---|
| | Bit31…Bit0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | Priority [31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Recommend to use NVIC_SetPriorityIRQ of CortexM0 function lib to set interrupt priority

## 8.3 Special function register list

| Base address：0x40011800 | | | | |
|---|---|---|---|---|
| Offset address | name | Write/read | Reset value | Function description |
| 0x00 | EXTIE | R/W | 0x0000 | External interrupt input edge control register |
| 0x04 | EXTIF | R/W | 0x0000 | External interrupt input filter set |
| 0x08 | PINFLT | R/W | 0x0000 | Pins digital filter enable |
| 0x10 | EXTIE2 | R/W | 0x0000 | External interrupt input edge control register2 |
| 0x14 | EXTIF2 | R/W | 0x0000 | External interrupt input filter set2 |
| 0x18 | PINFLT2 | R/W | 0x0007 | Pins digital filter enable2 |

## 8.4 Special function register introduction

| EXTIE<br>(External interrupt edge configuration register) | | | Base address： 0x40011800<br>Offset address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | | RIE[6:0] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | | FIE[6:0] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| RIE[6:0] | INT external input pin rising edge interrupt enable<br>0：disable<br>1：enable |
| FIE[6:0] | INT external input pin falling edge interrupt enable<br>0：disable<br>1：enable |

| EXTIF （external interrupt flag register） | | | Base address： 0x40011800 Offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | | RIF[6:0] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | | FIF[6:0] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| RIF[6:0] | INT external pin input rising edge interrupt flag<br>0：no interrupt is generated<br>1：interrupt is generated |
| FIF[6:0] | INT external input pin falling edge interrupt flag<br>0：no interrupt is generated<br>1：interrupt is generated |

| PINFLT （external pin filter configuration register） | | | base address： 0x40011800 offset address： 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | | | RXFLT[5:0] | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | | INTFLT[6:0] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| RXFLT[5:0] | RX external input pin digital filter function enable<br>0：disable<br>1：enable |
| INTFLT[6:0] | INT external input pin digital filter function enable<br>0：disable<br>1：enable |

Note:

8

1.RX0~5 and INT0~6, in addition to the digital filter function, has integrated analog filter circuit, analog filter (2us) only on the falling edge, the rising edge is invalid; digital filtering is effective to rise along the river down along the;

2, Users must enable corresponding pin digital filter function of PINFLT while using INT external interrupt pin function.

| **EXTIE2** （External interrupt edge configuration register **2**） | | | **Base Address：** **0x40011800** **Offset Address：** **10H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | RIE[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | FIE[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| RIE[2:0] | **INT7-9external input pin rising edge interrupt enable control bit**<br>0：disable<br>1：enable |
| FIE[2:0] | **INT7-9 external input pin falling edge interrupt enable control bit**<br>0：disable<br>1：enable |

| **EXTIF2** **(**external interrupt flag register **2)** | | | **Base Address：** **0x40011800** **Offset Address：** **14H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | RIF[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | FIF[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| RIF[2:0] | **INT7-9 external pin input rising edge interrupt flag**<br>0：no interrupt is generated<br>1：interrupt is generated |

| | |
|---|---|
| FIF[2:0] | **INT7-9 external input pin falling edge interrupt flag**<br>0：no interrupt is generated<br>1：interrupt is generated |

| **PINFLT2**（external pin filter configuration register **2**） | | | **Base Address：** **0x40011800**<br>**Offset Address：** **18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | INTFLT[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| **Bit** | **Function Description** |
|---|---|
| INTFLT[2:0] | **INT7-9 external input pin enables digital filtering functions**<br>0：disable<br>1：enable |

Note:

1. INT7~9, in addition to the digital filter function, the internal analog filter circuit (2us) is also integrated. The analog filter is only valid for the falling edge and the rising edge is invalid. The digital filter is valid for rising along the falling edge of the river;

2. When the user uses the INT external interrupt pin function, it is necessary to open the corresponding pin digital filtering function in the PINFLT/PINFLT2 register.

# 9 Reset module

## 9.1 Reset priority

There are 8 kinds of reset method which can be dividing into 4 reset priority levels.
RTC module is independent, withan individual RTC POR, MCU differs from POR modules and POR modules of other peripheral.

| Numb | Reset source | Reset level | Registers cannot be reset |
|---|---|---|---|
| 1 | Power-on reset（POR） | Level one | 1，reset status register RSTSR<br>2，registers of RTC modules |
| 2 | Low voltage detect reset（LBOR） | | |
| 3 | External pin/RST reset | Level two | 1，reset status register RSTSR<br>2，registers of RTC<br>3，registers of PMU<br>4，wake-up flag register WAKEIF |
| 4 | Software POR/LBOR reset | | |
| 5 | Power-off reset（BOR） | | |
| 6 | Watchdog reset（WatchDog） | Level three | 1，reset status register RSTSR<br>2，registers of RTC<br>3，registers of PMU<br>4，wake-up flag register WAKEIF<br>5，registers of GPIO：IOCFG，AFCFG，PTDIR，PTUP，PTDAT，PTOD<br>6，LCD related registers：LCDCLK，LCDCR，LCD_BUF[i]<br>7，CMU related registers：CLKCTRL0，CLKCTRL1，LRCADJ，HRCADJ |
| 7 | Debug reset（Debug Reset） | Level four | 1，reset status register RSTSR<br>2，registers of RTC<br>3，registers of PMU<br>4，wake-up flag register WAKEIF<br>5，registers of GPIO：IOCFG，AFCFG，PTDIR，PTUP，PTDAT，PTOD<br>6，LCD related registers：LCDCLK，LCDCR，LCD_BUF[i]<br>7，CMU related registers：CLKCTRL0，CLKCTRL1，LRCADJ，HRCADJ<br>8，register of EXT：EXTIE, EXTIF<br>9，register of TBS（except for DAT）<br>10，register of KEY |
| 8 | Wake-up reset（WakeUp Reset） | | |

Note
8

1:LRCADJ can be reset by Watchdog and Debug Reset

2:POR reset flag and LBOR reset flag of RSTSR can reset each other

3:wake-up reset cannot reset all interrupt flag register

4: All resets will reset the CLKCTRL0 HRC_EN signal. After reset, the system will run HRC (SYSCLK_SEL[2:0]=010).

# 9.2 Reset introduction

CPU process pointer be back to 000H and most register back to default value after reset is generated.

1) Internal reset signal IRST1 will keep valid for 1088 Fosc if BOR and LBOR reset.

2) Internal reset signal IRST2 will keep valid for 1088 Fosc if BOR and external RST reset.

3) Internal reset signal IRST3 will keep valid for 64 Fosc if WDT Reset reset.

4) Internal reset signal IRST4 will keep valid for 64 Fosc if SoftRese and Debug Reset reset.

5)RTC will reset if BOR reset for first time.

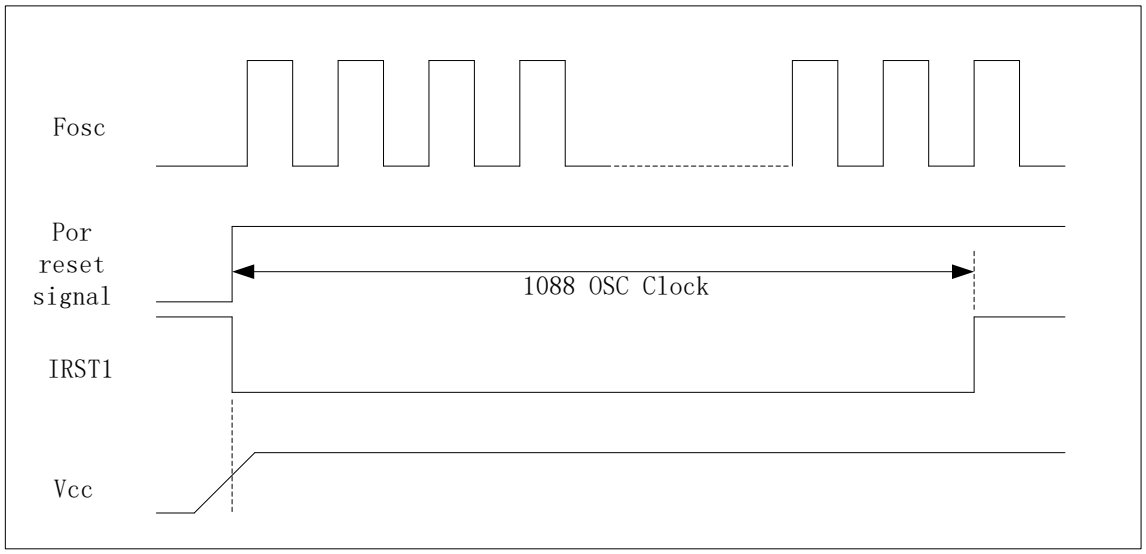
The clock count reset preheating source selection (Fosc/Flrc), by Fosc stop vibration detection module (LF_DET) of the vibration stop sign LF_FLAG control, LF_FLAG=0, LF_FLAG=1, Fosc clock; Flrc clock; if Fosc stop vibration detecting closed, LF_FLAG is fixed at 0, when Zhong Qiang was reset count Fosc, Fosc then stop oscillation effect of preheating will reset the counting function, the normal operation of the system proposed to open the Fosc oscillation stop detection, enhance the reliability of the reset.

## 9.2.1 Power-on reset

If power supply on chip for the first time and power-on reset circuit detects source voltage Vcc rise upon 0.3V,POR output high level and power-on will be indicated.IRST1 will turn to be high level if internal reset signals keep low voltage for 1088 Fosc.

Following event will occur if power-on reset is generated:

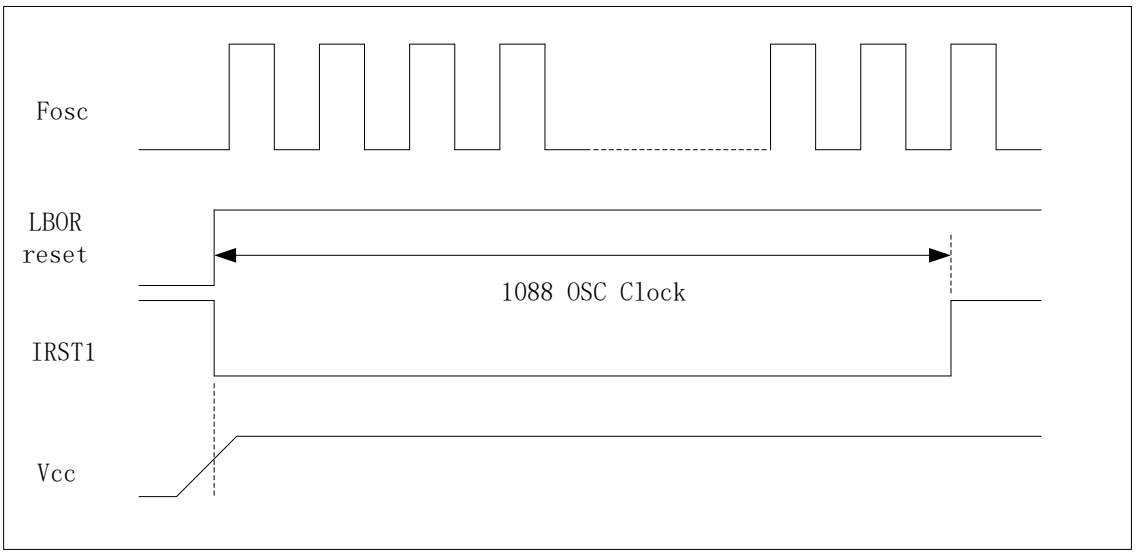
- POR output high level
- RTC will reset if POR reset for the for first time(RTC reset can only be triggered by VRTC power-on POR reset)
- Internal reset signal will turn to be valid
- Count 1088 Fosc
- Power-on reset flag BOR of RSTSTA will be set to 1 and other RSTSTA will be cleared
- CPU start program from 000H




power-on reset POR introduction

### 9.2.2 Low voltage detect reset

Low voltage detect reset (LBOR) has the same power-on reset process with POR after power off

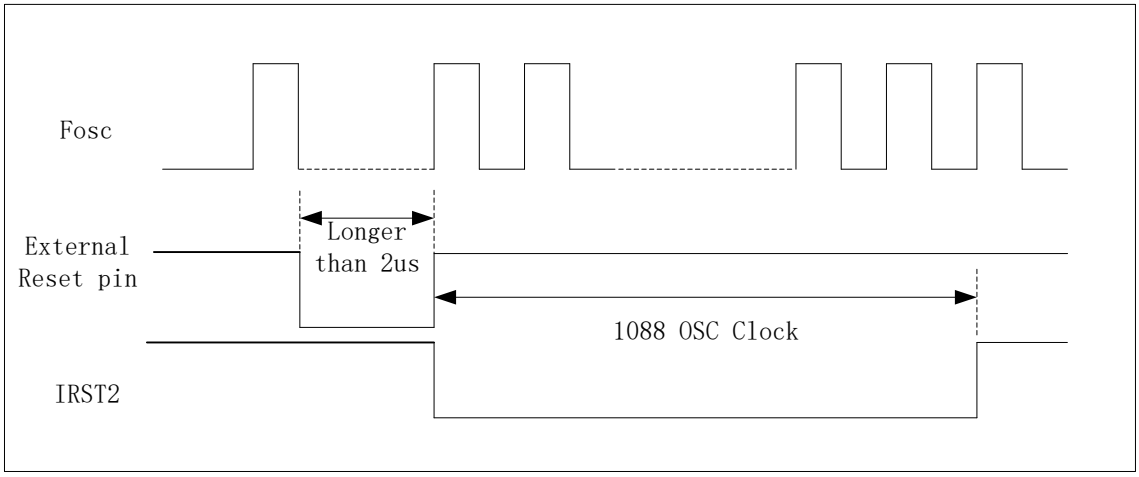



reset introduction

### 9.2.3 External pin reset

Internal reset signal will be valid and reset flag RST of reset status register will be set to 1 if external reset pin/RST output low level for 2us.it will be valid if internal reset signal IRST2 last for 1088 Fosc

If /RST lower level last shorter than 2us,system will not reset.



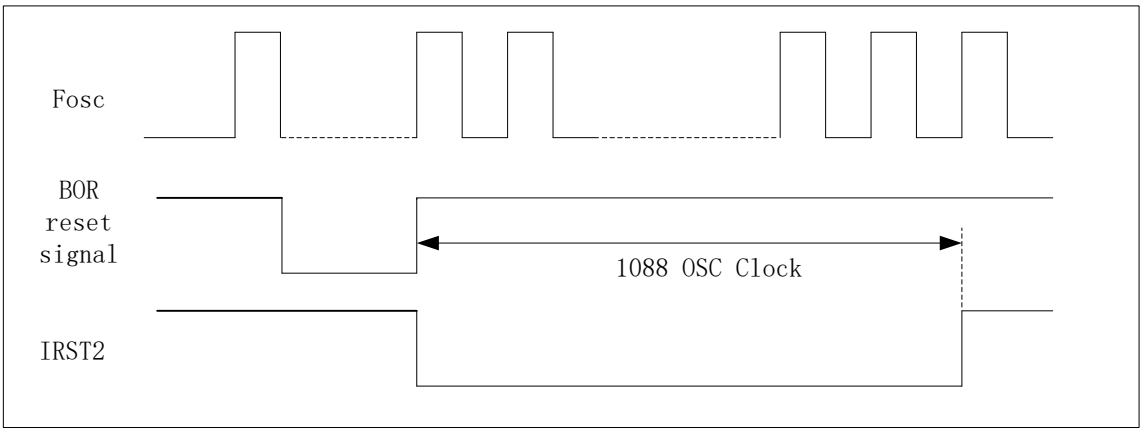

External pin reset introduction

### 9.2.4 Power-off reset

If power-off reset circuit detect source voltage lower than Vbor,BOR output low level and internal reset signal IRST2 will turn to be low level and BOR flag of reset status register RSTSTA set to 1.IRST1 will turn to be high level and IRST2 will turn to be high level after 1088 Fosc if power-off detect circuit detect that source voltage is higher than Vbor.

Following event will occur if power-off reset is generated:

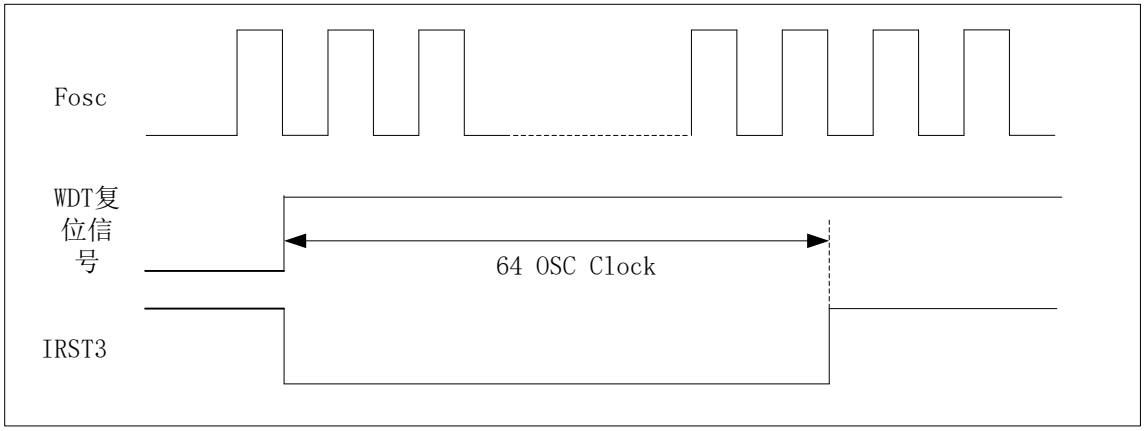
- Generate a BOR pulse
- Internal reset signal IRST2 will be valid
- Count 1088 Fosc
- Power-off reset flag BOR of RSTSTA will be set to 1 and other RSTSTA will be cleared
- CPU start program from 000H




reset introduction

### 9.2.5 Watchdog reset

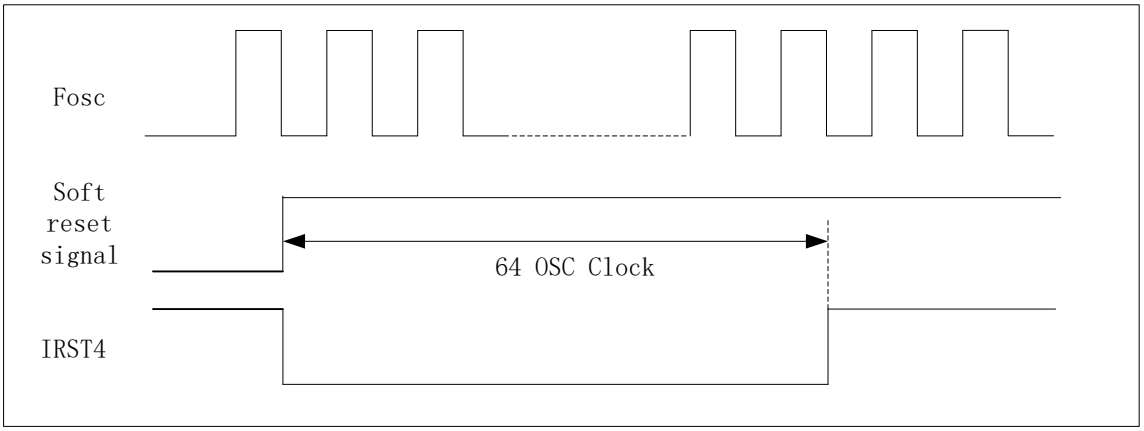
Internal reset IRST3 will be valid and reset flag WDT of reset status register set to 1 if WatchDog Timer overflows.WDT reset pulse last for 1088 Fosc.




WDT reset

### 9.2.6 Soft reset

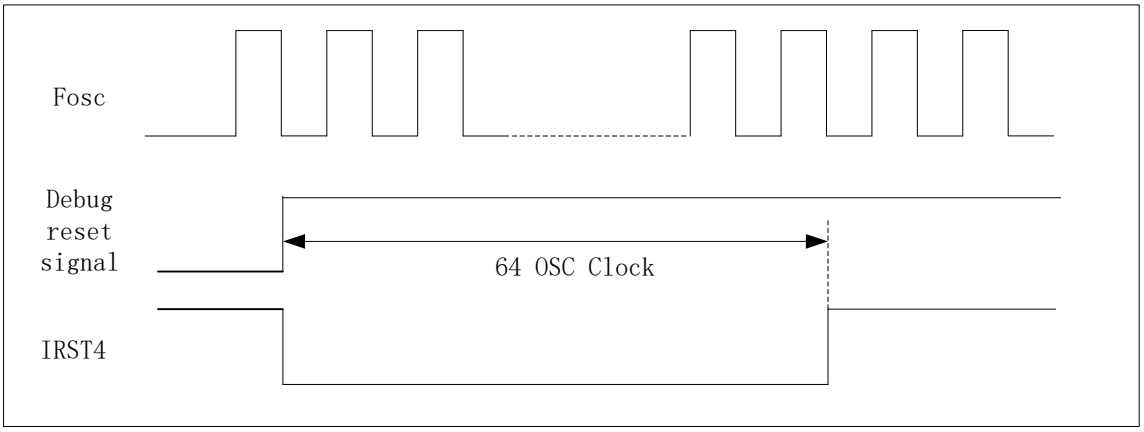
A soft reset will be generated if write 1 to AIRCR bit2 application interrupt and control status register of system control block.(function comes with Cortex-M0)




Soft reset

### 9.2.7 Debug reset

It is generated only in JTAG debug status.

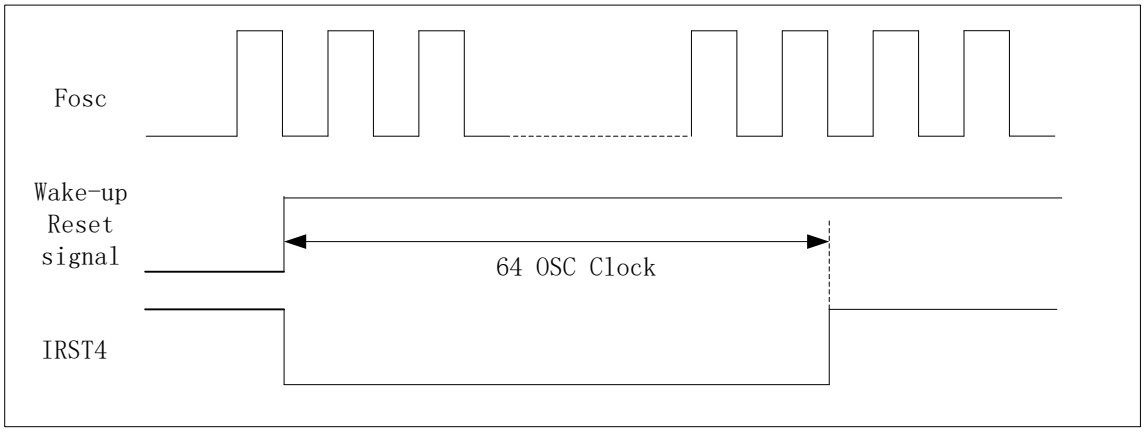



Debug reset

### 9.2.8 Wake-up reset

Following events will be executed in order if sleep wake-up event occurs:

- Internal reset signal IRST4 will be valid
- Power-off flag WKR of Reset status register RSTSTA will be set to 1
- Release internal reset signal IRST4 after counting 64 Fosc




Wake-up reset

## 9.3 Special function register list

| Base address：0x4000F400（same with PMU） | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x30 | RSTSTA | R/*W | ----- | Reset flag register |

**Note: it will be explained in PMU since there is only one register in this chapter**

| Base address：0x4000F400（same with PMU） | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x18 | WAKEIF | R/W | 0x0000 | Wake-up source flag register |

| Base address：0xE000ED00 | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x0C | AIRCR | R/W | ----- | Application interrupt and control status register |

**Note: this is a kernel register**

## 9.4 Special function register introduction

| **RSTSTA（reset flag register ）** | | | **Base address： 0x4000F400 Offset address： 30H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | Hold_Flag | Sleep_Flag | X | X | X | X | X | BORRST |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | DebugRST | SoftRST | ExtRST | FlashWR_LBOR | WakeupRST | WDTRST | LBORRST | PORRST |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| Hold_flag | **Interrupt wake-up flag in Hold mode**<br>0：no interrupt wake-up occurs in Hold mode<br>1：interrupt wake-up occurs in Hold mode<br>Cleared to 0 by writing 0,cannot be written 1 |
| Sleep_flag | **Interrupt wake-up flag in Sleep mode**<br>0：no wake-up occurs in Sleep mode<br>1：wake-up occurs in Sleep mode<br>Cleared to 0 by writing 0 |
| BORRST | **BOR reset flag**<br>0：no BOR reset occurred<br>1：BOR reset occurred<br>Cleared to 0 by writing 0,cannot be written 1 |
| DebugRST | **Debug-reset reset flag**<br>0：no Debug Reset occurred |

| | |
|---|---|
| | 1：Debug Reset occurred<br>Cleared to 0 by writing 0,cannot be written 1 |
| SoftRST | **Soft Reset reset flag**<br>0：no Soft Reset occurred<br>1：Soft Reset occurred<br>Cleared to 0 by writing 0 |
| ExtRST | **External RST reset flag**<br>0：no RST reset occurred<br>1：RST reset<br>Cleared to 0 by writing 0,cannot be written 1 |
| | |
| WakeupRST | **Wake-up Reset reset flag**<br>0：no Wake-up Reset occurred<br>1：Wake-up Reset occurred<br>Cleared to 0 by writing 0,cannot be written 1 |
| WDTRST | **Watch Dog reset flag**<br>0：no WDT reset occurred<br>1：WDT reset<br>Cleared to 0 by writing 0,cannot be written 1 |
| LBORRST | **LBOR reset flag**<br>0：no LBOR reset occurred<br>1：LBOR reset<br>Cleared to 0 by writing 0,cannot be written 1 |
| PORRST | **POR reset flag**<br>0：no POR reset occurred<br>1：POR reset<br>Cleared to 0 by writing 0,cannot be written1 |

Note: when the LBOR and POR reset occurs, all other reset flags will be reset;
In addition to the two reset of the LBOR and the POR, the existing reset flag of the register is not cleared.

| **WAKEIF**<br>**（wake-up flag register）** | | | **Base address：** **0x4000F400**<br>**Offset address：** **18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31** | **30** | **29** | **28** | **27** | **26** | **25** | **Bit24** |
| **Read:**<br>**Write:** | NMIWKIF | INT9WKIF | INT8WKIF | INT7WKIF | KEYWKIF | DMAWKIF | EMUWKIF | RSRV |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:**<br>**Write:** | X | SPIWKIF | I2CWKIF | RTCWKIF | TBSWKIF | TMR3WKIF | TMR2WKIF | TMR1WKIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | TMR0WKI | RX5WKIF | RX4WKIF | RX3WKIF | RX2WKIF | RX1WKIF | RX0WKIF | INT6WKIF |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | INT5WKIF | INT4WKIF | INT3WKIF | INT2WKIF | INT1WKIF | INT0WKIF | ARGEWKIF | PMUWKIF |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note: Sleep wake-up and Hold wake-up share this flag bit**

| Bits | Function description |
|---|---|
| NMIWKIF | **NMI wake-up flag**<br>When system wakes up, an optional interrupt NMI is set, the flag bit is set to 1. |
| INT9WKIF | **INT9 wake-up flag**<br>Under SLEEP/HOLD mode, when INT9 wakes up, the flag bit is set to 1. |
| INT8WKIF | **INT8 wake-up flag**<br>Under SLEEP/HOLD mode, when INT8 wakes up, the flag bit is set to 1. |
| INT7WKIF | **INT7 wake-up flag**<br>Under SLEEP/HOLD mode, when INT7 wakes up, the flag bit is set to 1. |
| KEYWKIF | **Key scan wake-up flag**<br>Under SLEEP/HOLD mode, when KEY wakes up, the flag bit is set to 1. |
| DMAWKIF | **DMA wake-up flag**<br>Under HOLD mode, when DMA wakes up, the flag bit is set to 1. |
| EMUWKIF | **EMU wake-up flag**<br>Under HOLD mode, when EMU wakes up, the flag is set to 1. |
| RSRV | This bit constant 0 |
| SPIWKIF | **SPI wake-up flag**<br>Under HOLD mode, when SPI wakes up, the flag is set to 1. |
| I2CWKIF | **I2C wake-up flag**<br>Under HOLD mode, when I2C wakes up, the flag is set to 1. |
| WDTWKIF | **WDT interrupt wake up**<br>This flag will be set to 1 if WDT interrupt wake-up occurs. |
| RTCWKIF | **RTC wake-up flag**<br>RTC wake up will occur and this bit will be set to 1 if RTC interrupt is generated in SLEEP mode.(particular RTC wake-up source indicated by RTCIF register) |
| TBSWKIF | **TBS wake-up flag**<br>TBS wake up will occur and this bit will be set to 1 if TBS interrupt is generated in SLEEP mode.(particular TBS wake-up source indicated by TBSIF register) |
| TMR3WKIF | **TMR3 wake-up flag**<br>Under HOLD mode, when TMR3 wakes up, the flag is set to 1. |
| TMR2WKIF | **TMR2 wake-up flag**<br>Under HOLD mode, when TMR2 wakes up, the flag is set to 1. |
| TMR1WKIF | **TMR1 wake-up flag**<br>Under HOLD mode, when TMR1 wakes up, the flag is set to 1. |

| | |
|---|---|
| TMR0WKIF | **TMR0 wake-up flag**<br>Under HOLD mode, when TMR0 wakes up, the flag is set to 1. |
| RX5WKIF | **RX5 wake-up flag**<br>If RX5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX4WKIF | **RX4 wake-up flag**<br>If RX4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX3WKIF | **RX3 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX2WKIF | **RX2 wake-up flag**<br>If RX3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX1WKIF | **RX1 wake-up flag**<br>If RX1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| RX0WKIF | **RX0 wake-up flag**<br>If RX0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI6WKIF | **INT6 wake-up flag**<br>If INT6 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI5WKIF | **INT5 wake-up flag**<br>If INT5 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI4WKIF | **INT4 wake-up flag**<br>If INT4 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI3WKIF | **INT3 wake-up flag**<br>If INT3 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI2WKIF | **INT2 wake-up flag**<br>If INT2 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI1WKIF | **INT1 wake-up flag**<br>If INT1 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| EXTI0WKIF | **INT0 wake-up flag**<br>If INT0 wake-up occurs, this flag will be set to 1,otherwise previous flag will be cleared if it happens after hardware is in sleep mode. |
| ARGEWKIF | **AES/RAND/GHASH/ECC wake-up flag** |

| | |
|---|---|
| | Under HOLD mode when AES/RAND/GHASH/ECC wakes up, the flag is set to 1. |
| PMUWKIF | **PMU wake-up flag**<br>If PMU occurs in SLEEP mode, PMU wake-up will occur and this flag will be set to 1, otherwise previous flag will be cleared if it happens after hardware is in sleep mode. (particular PMU wake-up source indicated by PMUIF register) |

Note:

1. Sleep wakeup and Hold wakeup share this flag bit
2. DMAWKIF/EMUWKIF/SPIWKIF/I2CWKIF/TMR3WKIF/TMR2WKIF/TMR1WKIF/TMR0WKIF /ARGEWKIF can only wake up in Hold, sleep mode cannot wake up.
3. registers are read-only register, it will always keep on a chip lead to wake source, when a new wake-up event generated by the hardware, generate new wake-up source symbol, while the 0 mark before awakening. This register can only be reset by POR.

| **AIRCR（application interrupt and control status register）** | | | **Base address：0xE000ED00**<br>**Offset address：0CH** | | | | |
|---|---|---|---|---|---|---|---|
| | **Bit31** | **30** | **29** | **28…11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | AIRCR [31:8] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | | | | | | SYSRESETREQ | RESERVED | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| SYSRESETREQ | =1：chip soft reset occurs<br>=0：invalid |

# 10 UART/7816 Communication Module

## 10.1 Function introduction

Asynchronous communication with external device is implemented by UART serial communication module .

Features:

- 6-ways UART in total
- UART3 and UART4 multiplex two 7816 interface respectively, set by MODESEL register
- Baud rate can be set to 1Mbps at most by software
- Full-duplex communication interface, canbe configured as infrared modulate output whose modulation polarity is configurable
- Sending support 1 stop bit or 2 stop bits
- Support date of 7 or 8 bits date-width
- Odd-even check is automatically done by hardware, remind odd-even check error and set flag when data is received.
- Independent Receive/send interrupt enable

Serial port provide flexible full-duplex asynchronous communication receiver and transmitter, serial port can operate in different mode by configuring UARTCFG. For example:

- Method 1:send through TXD or receive 7 bits data through RXD,no odd-even check,baud rate can be modified.
- Method 2: send through TXD or receive 7 bits data through RXD,1 odd-even check bit,baud rate can be modified.
- Method 3: send through TXD or receive 8 bits data through RXD,no odd-even check,baud rate can be modified.
- Method 4: send through TXD or receive 8 bits data through RXD,1 odd-even check bit,baud rate can be modified..

Output of 6 ways UART,TX0~TX5 can all be modulated into 32K infrared signal. Duty cycle of Modulated signals is adjustable(2400bps at most).

Output of 6 ways UART,TX0~TX5 can all be configured as output of external interrupt which make it easy for the implementation of external communication wake-up in SLEEP and HOLD mode.

Chip provided 2 ways ISO7816 interface at most and support 2 external 7816 device.

Main features of 7816:

- two 7816 interface multiplex UART3 and UART4 respectively, set by MODESEL register
- configuration of baud rate and UART baud rate is same,often use baud rate to cover it(115200 at most)
- response bits support 1,1.5 or 2 bits,odd-even check support odd,even and fixed check
- support to check data transmit status,given by hardware, and flag will be set
- independent receive/send interrupt enable,support error resend/re-receive function and times of resend setting

## 10.2 Baud rate calculation

Serial port baud rate is determined by baud rate creator:

$$baud\ rate = \frac{Fsys}{2 \times (SREL + 1)}$$

Where SREL is a 16 bits unsigned num;Fsys is system clock, baud rate is 1Mbps at most.

## 10.3 Serial port communication mode introduction

### 10.3.1 Method 1

Method1: a standard asynchronous communication method,frame contains 9 or 10 bits of data information:1 bit for start bit(0),7 bits for data bits(low bits at first),1 or 2 bits for end bits(1).TXD pin is data send terminal.RXD pin is the data receive terminal in this method,the waveform is shown as below:

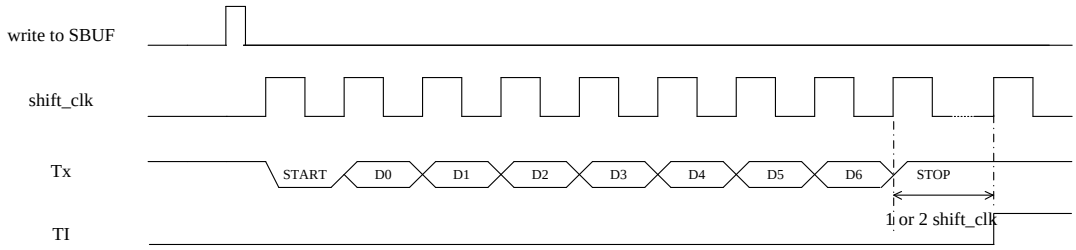



Fig:Serial data transmission sequence of method 1

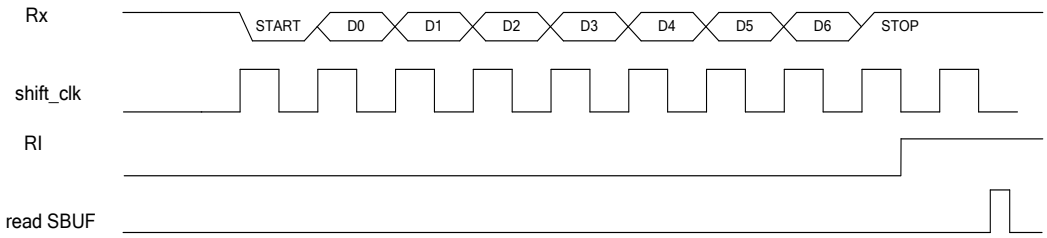



Fig: serial data receiving sequence of method 1

Send interrupt flag TI will be set to 1 when the last bit of data is transmitted in method 1 and the receive interrupt flag will be set to 1 when the last bit of data is received

## 10.3.2Method 2

Method 2: every frame contains 10 or 11 bits of data information:1 bit for start bit(0),7 bits for data bits(low bits at first),1 bit for odd-even data check,1 or 2 bits for end bits(1).TXD pin is data send terminal.RXD pin is the data receive terminal in this method,the waveform is shown as below:

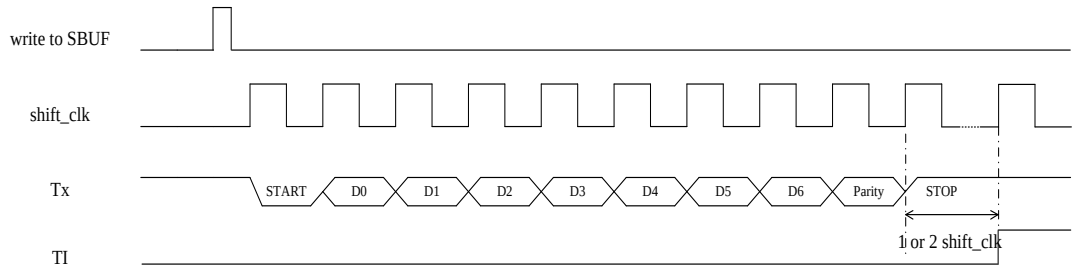



Fig: Serial data transmission sequence of method 2

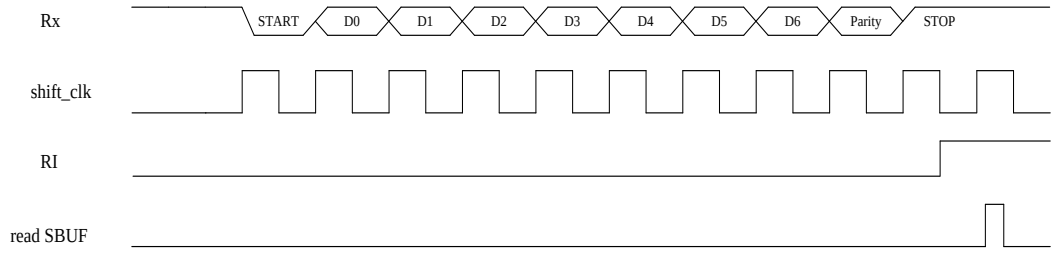



Fig: serial data receiving sequence of method 2

## 10.3.3 Method 3

Method1: a standard asynchronous communication method,frame contains 10 or 11 bits of data information:1 bit for start bit(0),8 bits for data bits(low bits at first),1 or 2 bits for end bits(1).TXD pin is data send terminal.RXD pin is the data receive terminal in this method,the waveform is shown as below:

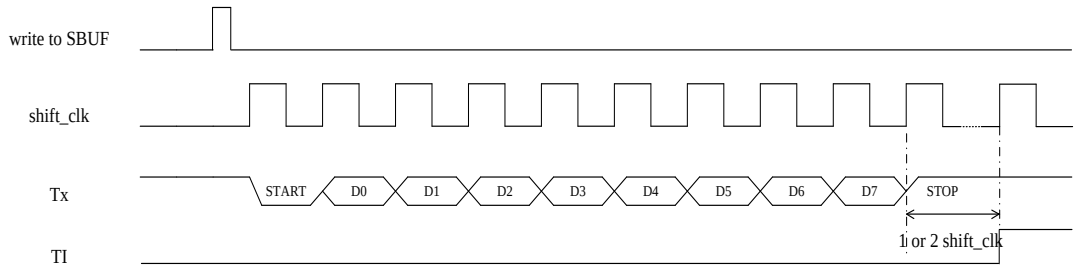



Fig: Serial data transmission sequence of method 3

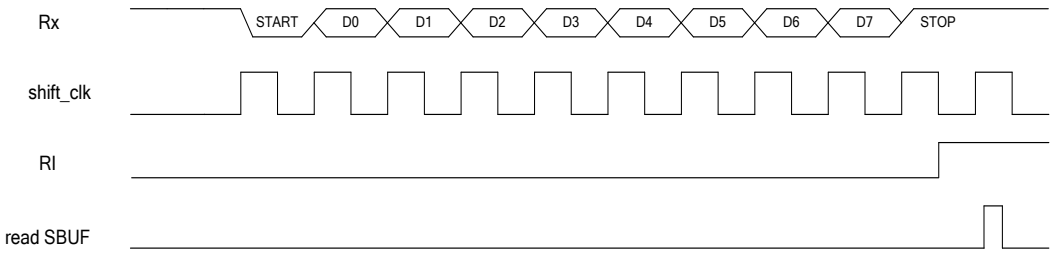



Fig: serial data receiving sequence of method 3

## 10.3.4 Method 4

Method1: a communication method that use 9th bit,frame contains 11 or 12 bits of data information:1 bit for start bit(0),8 bits for data bits(low bits at first),1 or 2 bits for end bits(1).TXD pin is data send terminal.RXD pin is the data receive terminal in this method,the waveform is shown as below:

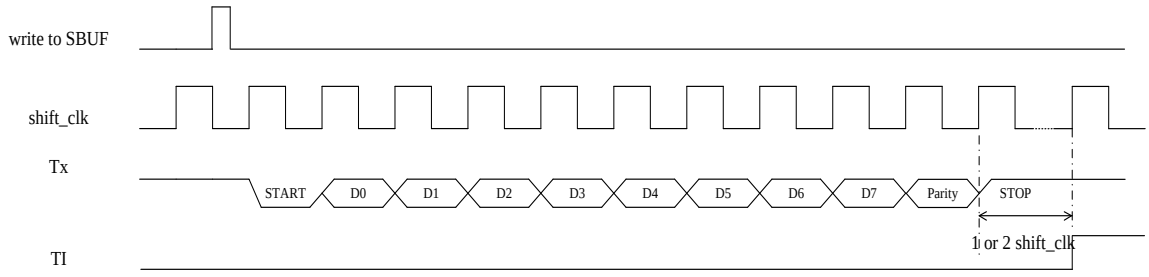



Fig: Serial data transmission sequence of method 4

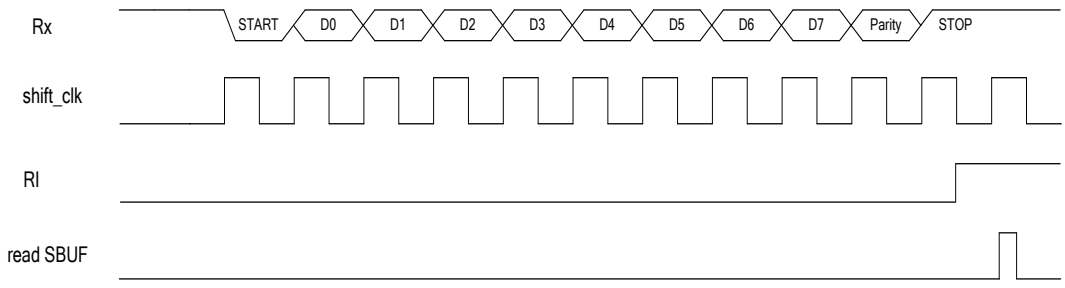



Fig: serial data receiving sequence of method 4

## 10.4 7816 receive and transmit

### 10.4.1 7816 data sending

Write to data buffer register SBUF3/4 to start a data sending,following steps included:

1. Send start bit(0); **(1st ETU)**
2. Send 8 bits of data; **(2nd to 9th ETU)**
3. Send 1 bit of parity bit; **(10th ETU)**
4. Check the received CKACK signal.if CKACK=0,set TX_PAR to 1,if CKACK=1, set TX_PAR to 0; **(11th ETU)**
5. Wait for 2 ETU(in send waiting status)
6. a data frame is transmitted by now,SDIF="1",if SDIE=1, this state will end and a send interrupt will be generated.

   If CKACK=1 or auto resend is disabled(AUTOSD=0), UART port will be back to IDLE state.

   If CKACK=0 and auto resend is enabled(AUTOSD=1),UART port will enter resend wait state.

7.Restart sending last frame.

### 10.4.2 7816 data receiving 7816

A data-receive process will be started if a falling edge is detected at receive port(IO) in IDLE mode.Following steps are included and for each of them it will take few ETUs:

1. Receive start bit(0); **(1st ETU)**
2. Receive 8 bits of data; **(2nd to 9th ETU)**
3. Receive 1 bit of parity bit; **(10th ETU)**
4. Send CKACK to send port.if data is right or auto re-receive is disabled(AUTORC=0),send 1 otherwise send 0.(**width of CKACK can be configured through ACKLEN)**

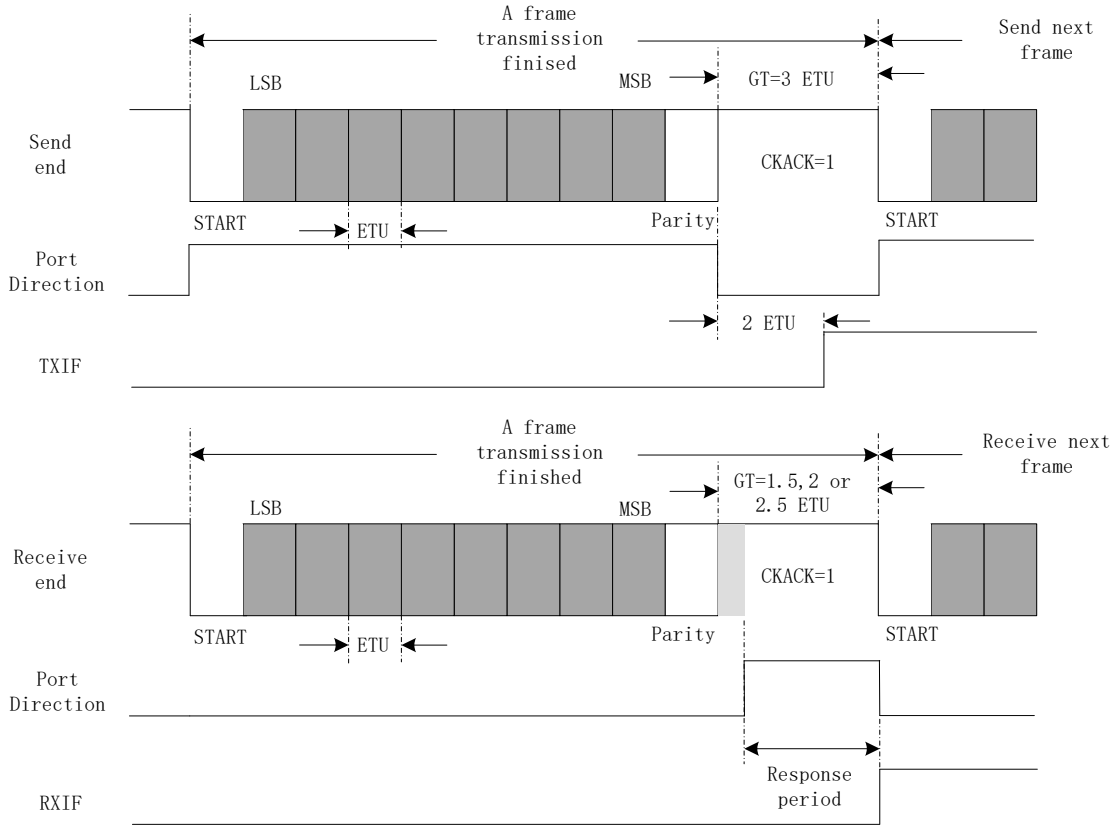
| Parity bit | AUTORC | IO1 |
|---|---|---|
| Correct | "0"disable auto re-receive | "1" |
| Correct | "1"enable auto re-receive | "1" |
| Wrong | "0" | "0" |
| wrong | "1" | "0" |

5. it will be back in IDLE mode and generate a receive interrupt if this state is finished.it will read data in SBUF if parity bits are right.(write operation to SBUF3/4 is invalid while receiving a data and it waits.)

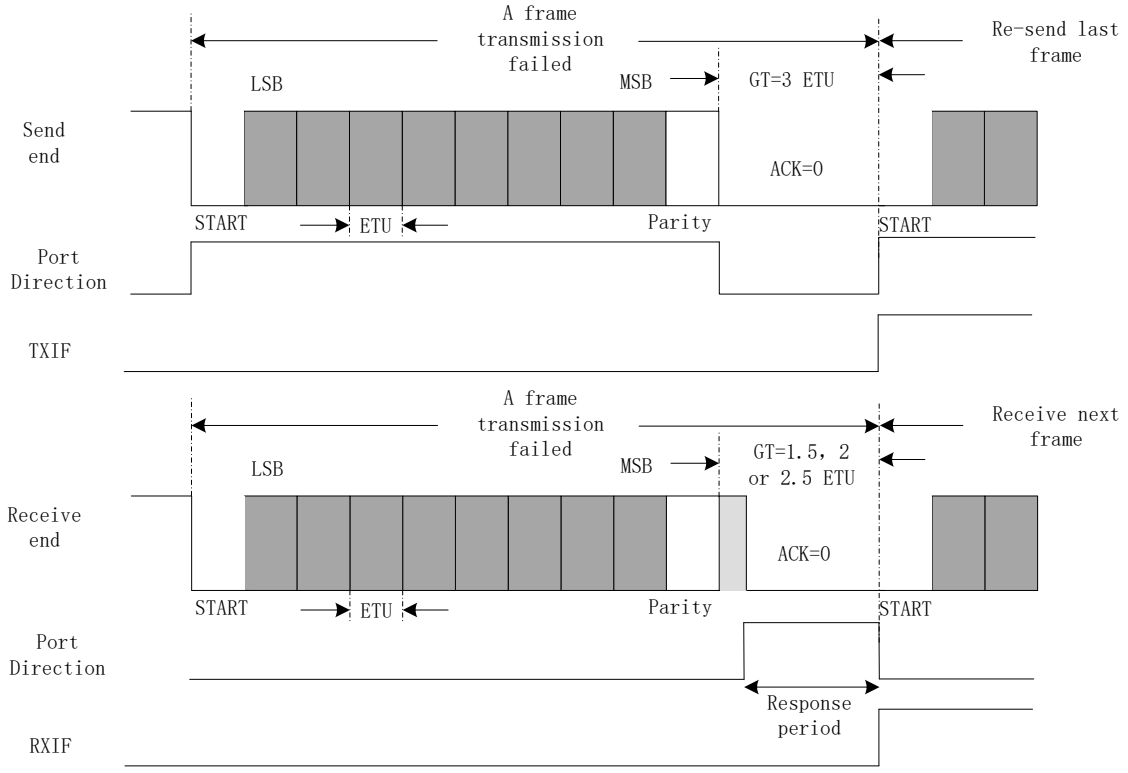
## 10.4.3 7816 communication pictogram


Data transmission pictogram

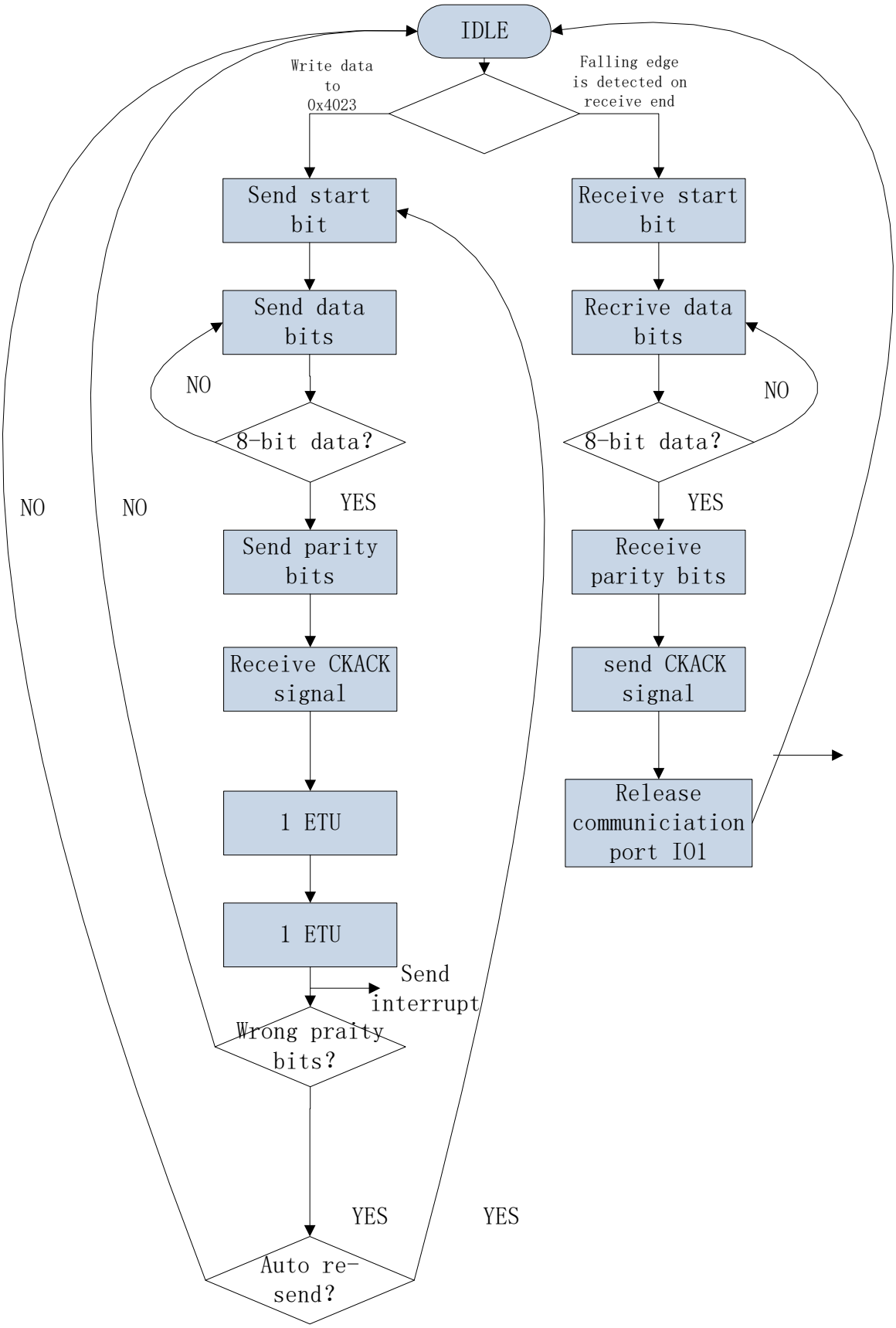
The period it takes to send/receive a bit is defined as Elementary Time Unit(ETU).After sending a data frame, the receiver needs some time to check the received data,the sending end needs to wait for the result of the response, and then send the next data frame or resend the previous frame according to the result of the verification, namely,the waiting time interval between sending frames continuously is defined as Guarding Time(GT),normally, 1 GT=3 ETU.



Successful data transmission sequence




Unsuccessful data transmission sequence



## 10.5 Special function register list

| **UART modules register base address：<br>0x40005000(UART0 port)；<br>0x40006000(UART1 port)；<br>0x40007000(UART2 port)；<br>0x40008000(UART3 port)；<br>0x40009000(UART4 port)；<br>0x40000000(UART5 port)；** | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x00 | MODESEL | R/W | 0x0000 | Serial port function select register |
| 0x04 | UARTCON | R/W | 0x0000 | UART function configuration register |
| 0x08 | ISO7816CON | R/W | 0x0000 | 7816 function configuration register |
| 0x0C | SREL | R/W | 0x0000 | Serial port baud rate generation register |
| 0x10 | SBUF | R/W | 0x0000 | Serial port data buffer register |
| 0x14 | UARTSTA | R/W | 0x0000 | UART state register |
| 0x18 | ISO7816STA | R/W | 0x0000 | 7816 state register |
| 0x30 | IRCON | R/W | 0x0000 | Infrared modulate control register |
| 0x34 | IRDUTY | R/W | 0x0000 | Infrared modulate pulse width adjust register |

**Note:7816 related register is only legal for base address of UART3 and UART4**

## 10.6 Special function register introduction

| **MODESEL (serial port function select register)** | | | **Base address: 0x40005000—0x40009000; 0x40000000 Offset address: 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | X | X | Mode |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| Mode | **Serial** port **function select control bit:**<br>0: UART function<br>1: 7816 function |

| **UARTCON (UART function configuration register)** | | | **Base address: 0x40005000—0x40009000; 0x40000000 Offset address: 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | NEGFIX | UNEG | STOPSEL |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | LENSEL | PARITYSEL[1:0] | | PARITYEN | RXIE | TXIE | RXEN | TXEN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| NEGFIX | **UART logical-lock in communication:**<br>0: user software decide the positive logic or negative logic of UART communication through UNEG control bit<br>1: Hardware self-configured UART communication as positive logic or negative logic. UNEG control bit is invalid. |
| UNEG | **UART** positive **logic or negative logic in communication**<br>0: positive logic(default)<br>1: negative |

| | |
|---|---|
| STOPSEL | **UART communication stop bit width selection bit**<br>1：2bit<br>0：1bit |
| LENSEL | **UART communication data width selection bit**<br>1：7bit<br>0：8bit |
| PARITYSEL | **UART parity selection bit**<br>11：fixed 1<br>00：fixed 0<br>01：odd check<br>10：even check |
| PARITYEN | **UART parity enable bit**<br>1：enable<br>0：disable |
| RXIE | **UART receive interrupt enable bit**<br>1：enable<br>0：disable |
| TXIE | **UART send interrupt enable bit**<br>1：enable<br>0：disable |
| RXEN | **UART receive enable**<br>1：enable<br>0：disable |
| TXEN | **UART send enable bit**<br>1：enable<br>0：disable |

| **ISO7816CON（7816 function configuration register）** | | | **Base address：0x40008000—0x40009000**<br>**Offset address：08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | 7816PARITY | AUTORXEN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | AUTOTXEN | REPTR1 | REPTR0 | ACKLEN1 | ACKLEN0 | PRDIE | RXIE | TXIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| 7816PARITY | **Parity select bit** |

| | |
|---|---|
| | 0：even check<br>1：odd check |
| AUTORXEN | **Auto re-receive enable bit**<br>1：enable<br>0：disable |
| AUTOTXEN | **Auto re-send enable bit**<br>1：enable<br>0：disable |
| REPTR[1:0] | **Auto re-receive/re-send times limit**<br>00：0 time<br>01：1 time<br>10：2 times<br>11：3 times |
| ACKLEN[1:0] | **Response bits width**<br>00：1bit<br>01：1.5bit<br>10/11：2bit |
| PRDIE | **Overspill interrupt enable bit**<br>1：enable<br>0：disable |
| RXIE | **Receive interrupt enable bit**<br>1：enable<br>0：disable |
| TXIE | **Send interrupt enable bit**<br>1：enable<br>0：disable |

| **SREL（serial port baud rate generate register）** | | | **Base address: 0x40005000—0x40009000；0x40000000<br>Offset address： 0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | SREL[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | SREL[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Serial port/7816 baud rate generate register is a baud rate frequency division coefficient of 16 bits, it is between 0~65535, the highest baud rate is 1Mbps.the calculation formula:

$$baud\ \ rate = \frac{Fsys}{2 \times (SREL + 1)}$$

| SBUF（serial port data buffer register） | | | Base address： 0x40005000—0x40009000；0x40000000<br>Offset address： 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | SBUF[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Low 8 bits is valid. Write to SBUF register to start the buffered date transmission of serial port; read the SBUF register to receive buffered data of serial port.

| UARTSTA（UART state register） | | | Base address： 0x40005000—0x40009000；0x40000000<br>Offset address： 14H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | PARITY | RXIF | TXIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| PARITY | **Parity bits received**<br>1：wrong<br>0：right<br>Cleared by writing 0,invalid to write 1. |
| RXIF | **Receive interrupt flag**<br>1：data is received and can be read from SBUF<br>0：data receiving is not completed<br>Cleared by writing 0,invalid to write 1. |
| TXIF | **Send interrupt flag**<br>1：sending completed<br>0：sending is not completed<br>Cleared by writing 0,invalid to write 1. |

| ISO7816STA （7816 state register） | | | Base address： 0x40008000—0x40009000 Offset address： 18H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | RXERRSTAT | TXERRSTAT | PRDIF | RXIF | TXIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| RXERRSTAT | **State of data received**<br>1：wrong<br>0：right<br>Cleared by writing 0,invalid to write 1. |
| TXERRSTAT | **State of data send**<br>1：wrong<br>0：right<br>Cleared by writing 0,invalid to write 1. |
| PRDIF | **Receive overspill interrupt flag**<br>1：overspilled；<br>0： not overspill；<br>Cleared by writing 0,invalid to write 1. |
| RXIF | **Receive interrupt flag**<br>1：data is received and can be read from SBUF<br>0：data receiving is not completed<br>Cleared by writing 0,invalid to write 1. |
| TXIF | **Send interrupt flag**<br>1：sending completed<br>0：sending is not completed<br>Cleared by writing 0,invalid to write 1. |

| IRCON (Infrard modulate control register) | | | Base address： 0x40005000—0x40009000；0x40000000 Offset address: 30H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **Read:** | X | X | X | X | X | X | IRLVL | IRTX |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| IRLVL | **Infrared modulate output polarity select**<br>1：negative<br>0：positive |
| IRTX | **Infrared modulate function enable control**<br>1：enable infrared modulate function output by TX<br>0：disable infrared modulate function output by TX |

Note:operation of writing IRTX=1 to enable infrared modulate function is valid only if PLL_EN=1 and PLL is open.(Otherwise it is invalid because of no clock source)

| **IRDUTY (Infrared modulate pulse width register)** | | | **Base address：** **0x40005000—0x40009000；0x40000000**<br>**Offset address:** **34H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | X | IRDUTY[1:0] | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| IRDUTY[1:0] | **Modulate wave duty cycle configurate**<br>00：50%<br>01：25%<br>10：12.5%<br>11：6.25% |

# 11. LCD

## 11.1. Introduction

HT5023(100pin) supports50 segment *8 Common output at most. HT5025(80pin) supports 39 Segment *8Common output at most.HT5027(64pin) supports27 Segment*8Common output at most. Main feature shown as follow:

- Software programmable drive method.(decide by GPIO pin)
  50 Seg * 8 Com (100pin) | 39 Seg * 8 Com (80pin) | 27 Seg * 8 Com（64pin）
  52 Seg * 6 Com (100pin) | 41 Seg * 6 Com (80pin) | 29 Seg * 6 Com（64pin）
  54 Seg * 4 Com (100pin) | 43 Seg * 4 Com (80pin) | 31 Seg * 4 Com（64pin）
- Software control contrast adjustable。
- LCD drive voltage is optional
- 1/3 Bias，1/4 Bias is optional
- 1/4 ，1/6，1/8 Duty is optional

## 11.2. LCD and GPIO certify pin multiple list

| LCD pin definition(100PIN) | Function pin definition | Chip pin definition |
|---|---|---|
| COM0-COM3 | PD.8—PD.11 | COM0/PD.8 – COM3/PD.11 |
| COM4/SEG24—COM5/SEG25 | PD.12–PD.13 | SEG24/COM4/PD.12 - SEG25/COM5/PD.13 |
| COM6/SEG26–COM7/SEG27 | PD.14-PD.15 | SEG26/COM6/PD.14 – SEG27/COM7/PD.15 |
| SEG0-SEG15 | PB.0-PB.15 | SEG0/PB.0 – SEG15/PB.15 |
| SEG16-SEG17 | PD.0-PD.1 | SEG16/PD.0 – SEG17/PD.1 |
| SEG18-SEG21 | PD.2-PD.5 | SCANOUT0/SEG18/PD.2-SCANOUT3/SEG21/PD.5 |
| SEG22-SEG23 | PD.6-PD.7 | SEG22/PD.6-SEG23/PD.7 |
| SEG28-SEG32 | PA.0-PA.4 | SEG28/PA.0-SEG32/PA.4 |
| SEG33-SEG34 | PA.7-PA.8 | SEG33/INT2/PA.7-SEG34/INT3/PA.8 |
| SEG35-SEG36 | PA.12-PA.13 | SEG35/ADCIN0/PA12-SEG36/ADCIN1/PA13 |
| SEG37 | PE.8 | SEG37/ADCIN2/PE.8 |
| SEG38-SEG42 | PF.3-PF.7 | SEG38/PF.3-SEG42/PF.7 |
| SEG43-SEG46 | PC.0-PC.3 | SEG43/TX1/PC.0, SEG44/RX1/PC.1<br>SEG45/RX0/PC.2, SEG46/TX0/PC.3 |
| SEG47 | PE7 | PE.7/LVDIN0/SEG47 |
| SEG48 | PF2 | PF.2/SFOUT/SEG48 |
| SEG49 | PE5 | PE.5/TX3/SEG49 |
| SEG50 | PA.11 | PA.11/INT6/SEG50 |

| | | |
|---|---|---|
| SEG51 | PC.6 | PC.6/SPI_CLK/SEG51 |
| SEG52 | PC.7 | PC.7/SPI_CS/SEG52 |
| SEG53 | PE.3 | PE.3/CLKOUT/SEG53 |

80PIN package remove pin: SEG23，SEG28-30，SEG38-42,SEG51-52

64PIN package remove pin: SEG8-11，SEG16-17，SEG22，SEG32，SEG36-37,SEG43-44,SEG50

## 11.3. LCD Diagram

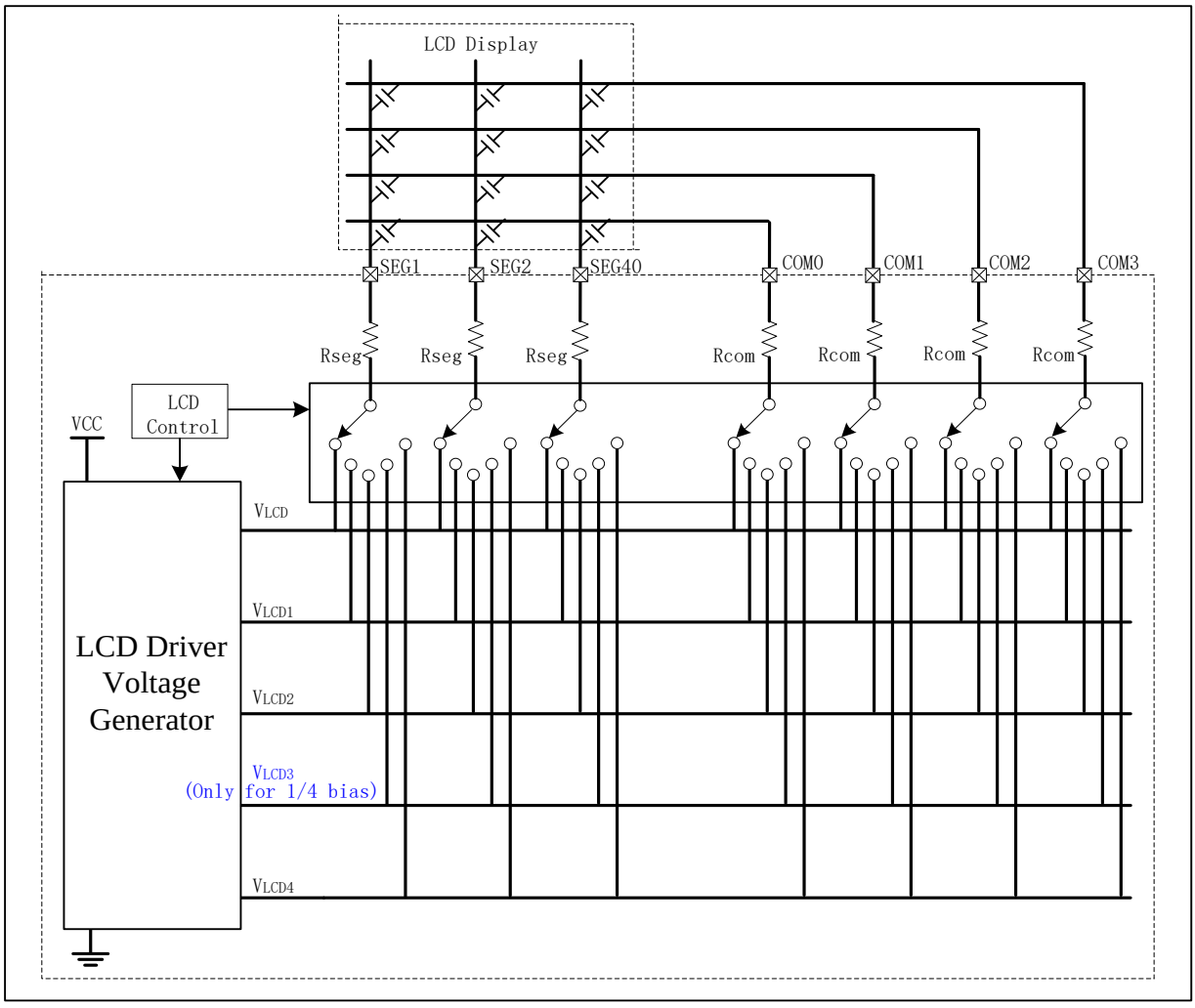
The LCD drive unit employs 2 modes of operation: 1/3, Bias, and 1/4 Bias, controlled by the BIASCTL bit. LCD power supply by the VLCD, LCD drive voltage VLCD1, VLCD2, VLCD3, VLCD4 generated by the internal drive voltage generation module, VLCD3 only 1/4 Bias effective.

DUTY[2:0] is used to control the LCD waveform Duty, refer to chapter 11.5 output waveform section.

The clock source of the LCD unit is Fosc/Flrc, and the frequency of the LCD waveform is scanned by the stop vibration detection after frequency division, and the frequency Flcd and Flcd can be configured by register LCLK[2:0]. LCD frame scan frequency Ffrm=Flcd*duty. For example: 1/4 duty, the frame scanning frequency is Ffrm=Flcd/4. Details refer to the LCDCLK section of the special function register.

The display data of LCD is put into LCD BUF, which is used to control the switch of LCD segment. When a segment of the corresponding SEG and COM control bits (LCD_BUF corresponding bits) is 1, the segment will be lit; otherwise, it will not be lit.

When the LCD display is not required, the LCD_EN (CLKCTRL0.1) is written as 0, and the LCD unit can be turned off. After the LCD is turned off, all the SEG and COM are output high, the internal drive voltage generation module, and the analog circuitry are turned off, and the LCD unit's clock is turned off.




LCD modulate function diagram

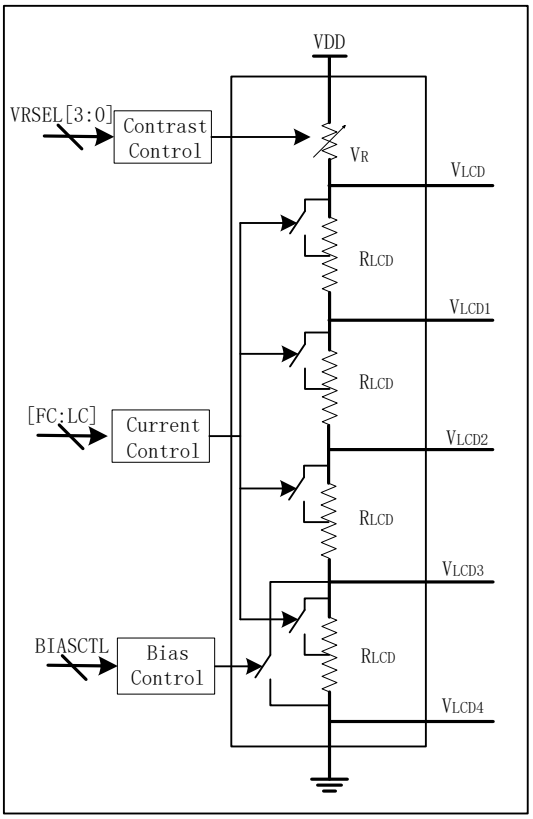
## 11.4. Interval Resistive partial pressure drive

LCD drive mode control bit, optional internal resistor divider drive mode.

In this mode, VLCD0, VLCD1, VLCD2, VLCD3 and VLCD4 by an internal resistor divider network, the most high-end VLCD0 less than VDD.

VRSEL[3:0] is used to control the LCD contrast, referring to the special function register LCDCON, the associated LCD contrast section, and the VRSEL configuration.

The FC and LC are used to control the LCD charging mode.




LCD internal resistance divider drive

## 11.5. Output waveform

Duty of Wave LCD output depends on the COMMON.three kinds of Duty are provided:

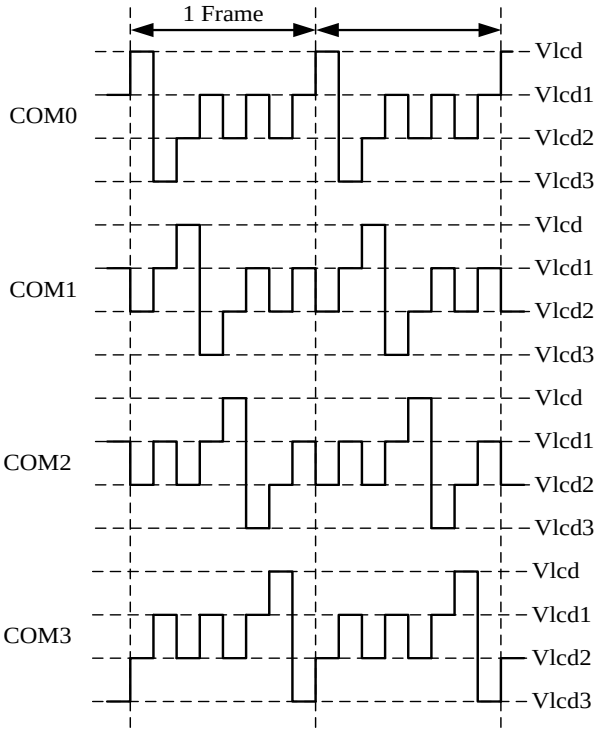
- DUTY[1:0]=00：1/4 duty——COM0、COM1、COM2、COM3 are used
- DUTY[1:0]=01：1/6 duty——COM0 to COM5 are used
- DUTY[1:0]=1x：1/8 duty——COM0 to COM7 are used

LCD drives voltage:

LCD power comes from VLCD,VLCD≤VCC.VLCD1、VLCD2 、VLCD3 and VLCD4 are internal bias of waveform that LCD output.

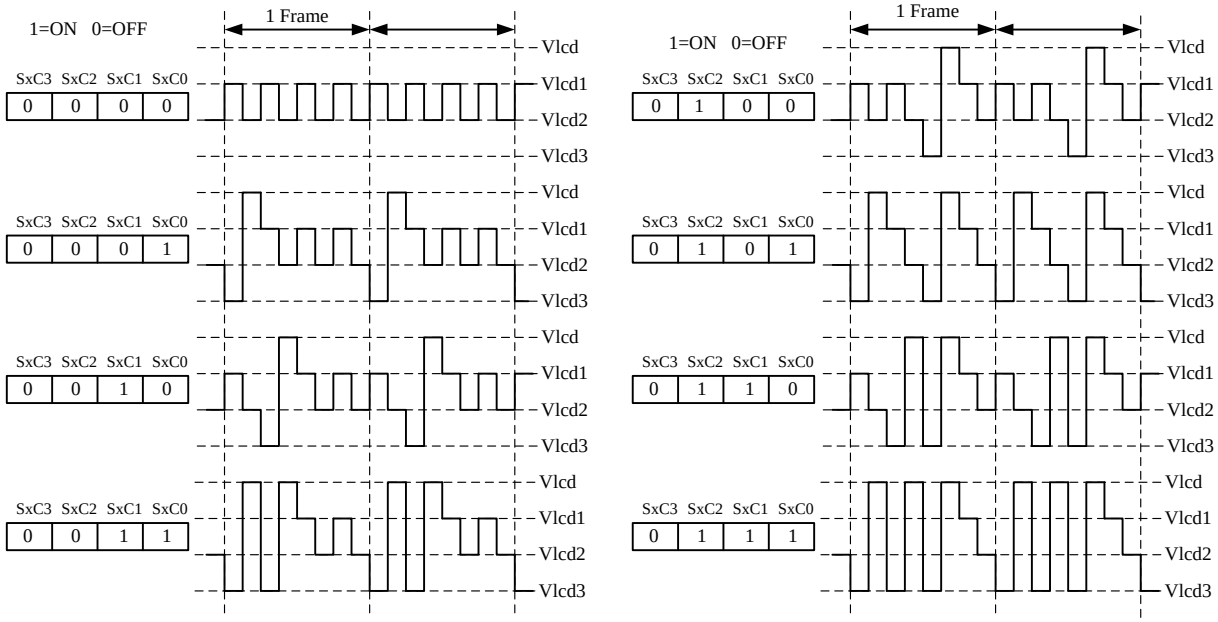
COM output waveform (1/4 Duty):

COM0/1/2/3 is occupied.1 frame equals to 4 clock periods of LCD wave.

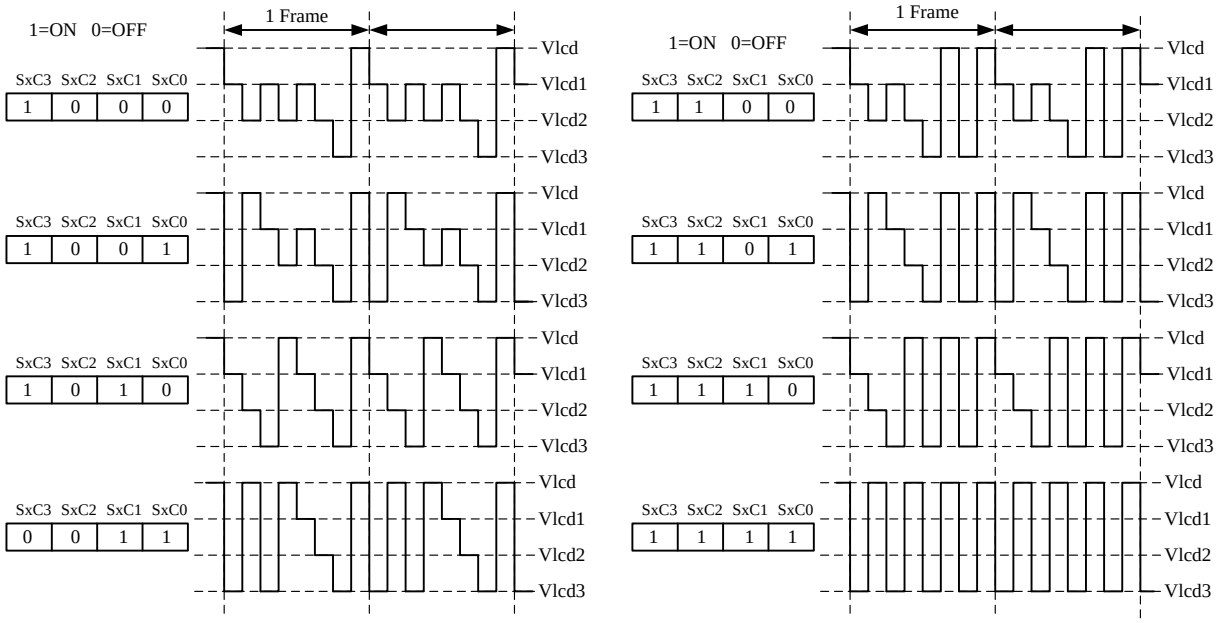



**Duty=1/4 COM Output**

SEGMENT output waveform(1/4 Duty)

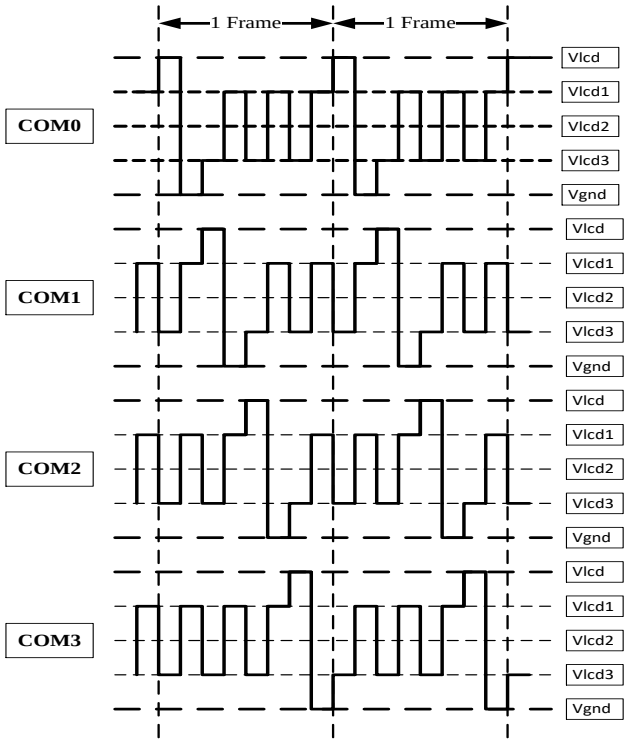



**Duty=1/4 SEGx Output**

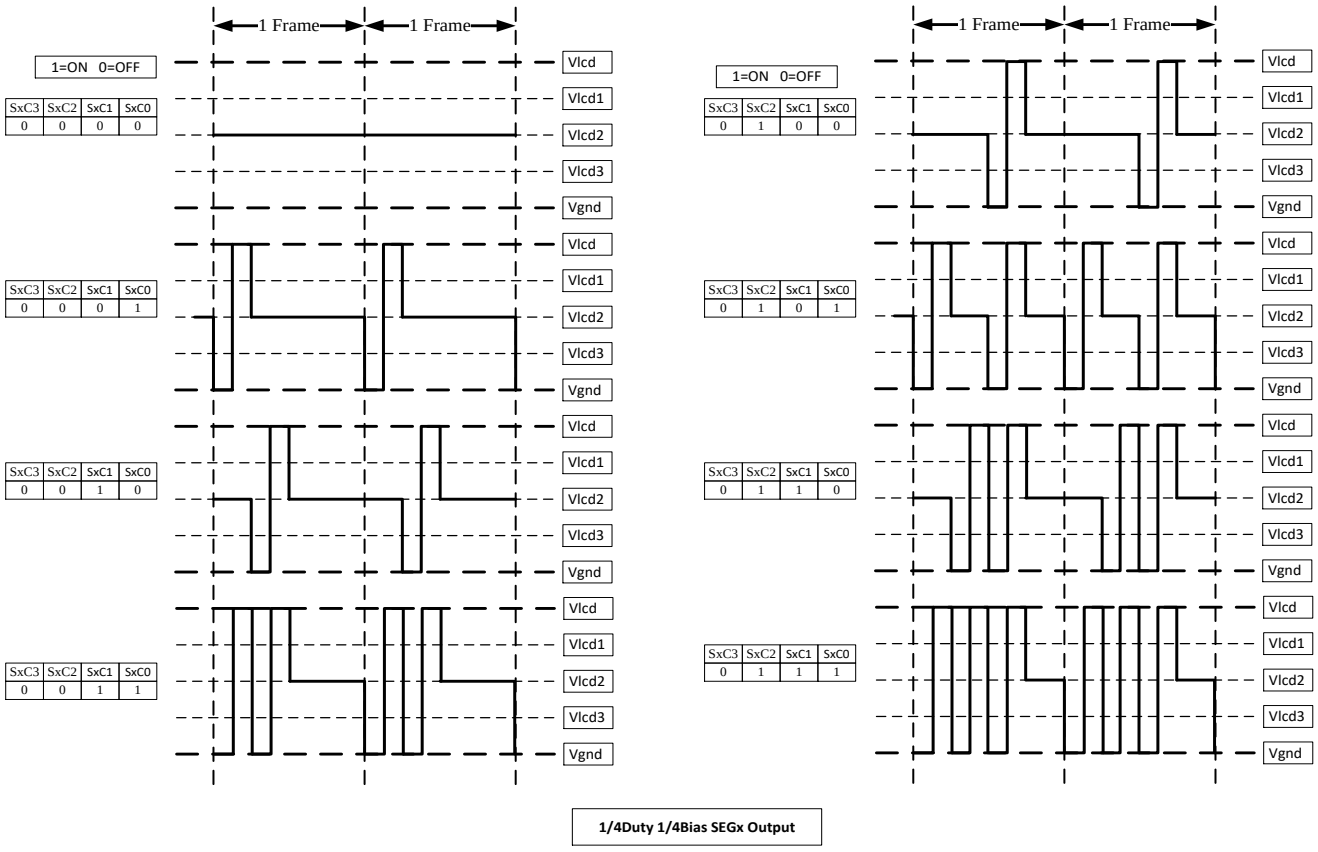


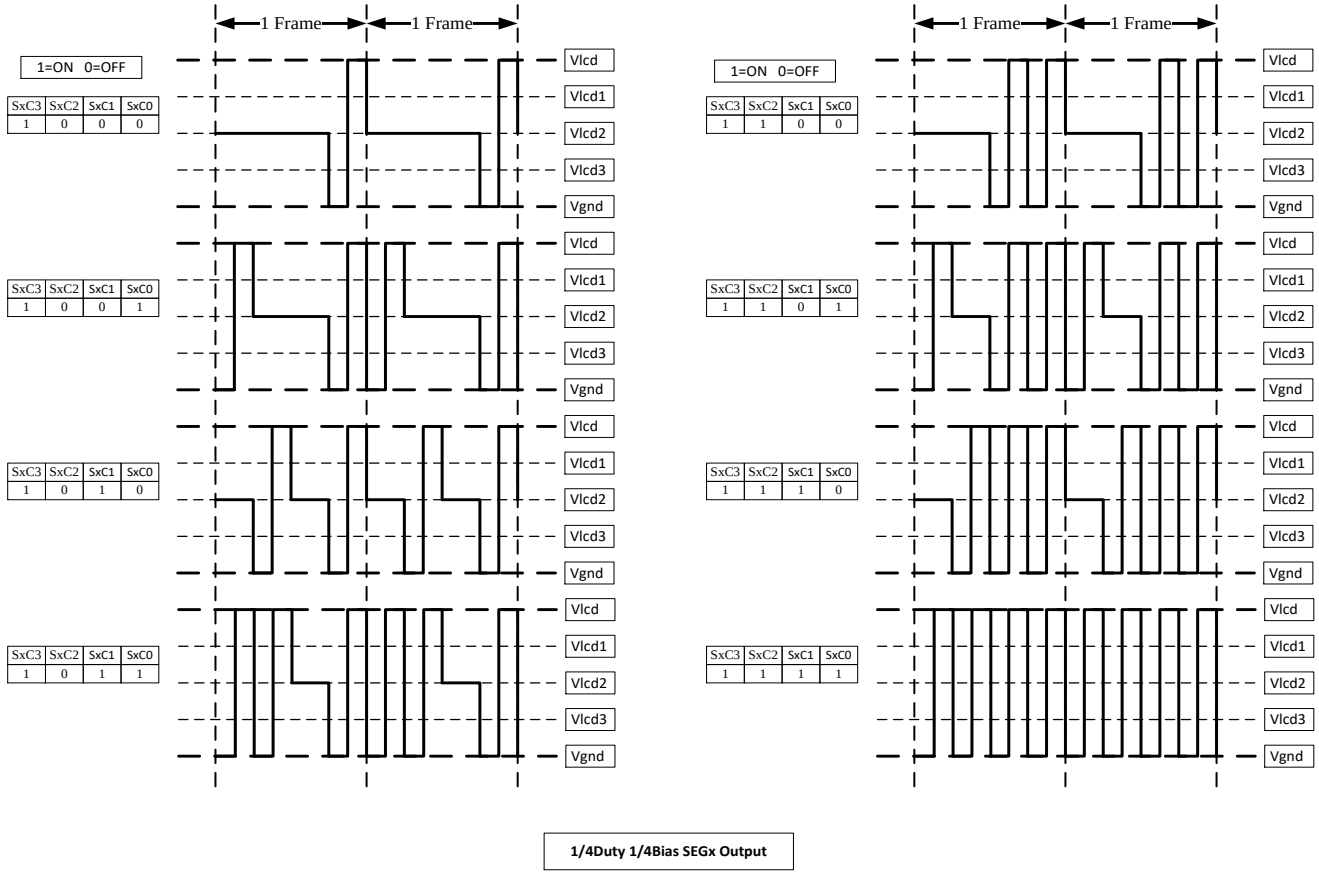

**Duty=1/4 SEGx Output**

## COM wave form（1/4Bias，1/4Duty）



## SEGMENT wave form（1/4Bias，1/4Duty）






## 11.6. LCD display operation

LCD data display writes register function:

Write the corresponding data SEG output to LCD BUFF to output and display LCD segment code.

Register of 40 Bytes function as LCD BUFF,the operation on it is same with normal register and it has the follow relationship:

Address LCD_BUF[0] corresponds to SEG0 （COM7—COM0）

Address LCD_BUF[1] corresponds to SEG1 （COM7—COM0）

Address LCD_BUF[2] corresponds to SEG2 （COM7—COM0）

……

Address LCD_BUF[41] corresponds to SEG41（COM7—COM0）

Address LCD_BUF[42] corresponds to SEG42（COM7—COM0）

Address LCD_BUF[43] corresponds to SEG43（COM7—COM0）

Address LCD_BUF[44] corresponds to SEG44（COM7—COM0）

Address LCD_BUF[45] corresponds to SEG45（COM7—COM0）

Address LCD_BUF[46] corresponds to SEG46（COM7—COM0）

……

Address LCD_BUF[53] corresponds to SEG46（COM7—COM0）

## 11.7. Special function register list

| Offset address | name | Reset value | Function description |
|---|---|---|---|
| **Base ADDR：0x4000D000** | | | |
| 0x00 | LCDCLK | 0x0080 | LCD clock frequency select register |
| 0x04 | LCDCON | 0x0089 | LCD drive control register |
| 0x0C | LCDOUT | 0x0001 | LCD output control register |
| 0x10+i×1.（i=0~42） | LCD_BUF[i] | 0x0000 | LCD data display register |

## 11.8. Special function register introduction

| **LCD Clock Register (LCDCLK LCD clock frequency select register)** | | | **Base address： 0x4000D000**<br>**Offset address： 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | BIASCTL | X | X | DUTY1 | DUTY0 | LCLK2 | LCLK1 | LCLK0 |
| **Reset:** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| BIASCTL | **bias drive select bit**<br>1：choose 1/3 bias drive<br>0：choose 1/4 bias drive<br>To improve the display,it is better to choose 1/4 bias drive for 1/6 and 1/8 Duty while setting DUTY[1:0]=01. |
| DUTY[2:0] | **Duty control select bits of LCD** |

| DUTY1 | DUTY0 | COMMON selection | LCD waveform Duty |
|---|---|---|---|
| 0 | 0 | Com0~Com3 | 1/4 duty |
| 0 | 1 | Com0~Com5 | 1/6 duty |
| 1 | 0 | Com0~Com7 | 1/8 duty |

| | |
|---|---|
| LCLK[2:0] | **LCD csanning frequency select bits:**<br>Clock of LCD unit comes from low frequency clock F(lf), after frequency division F(lf) function as LCD wave scanning frequency flc which can be configurated through register LCLK[2:0](details as following table shows).LCD frame scan frequency ffrm=flcd*duty.for example,if duty=1/4 duty,frame scan frequency F(frm)=flcd/4. |

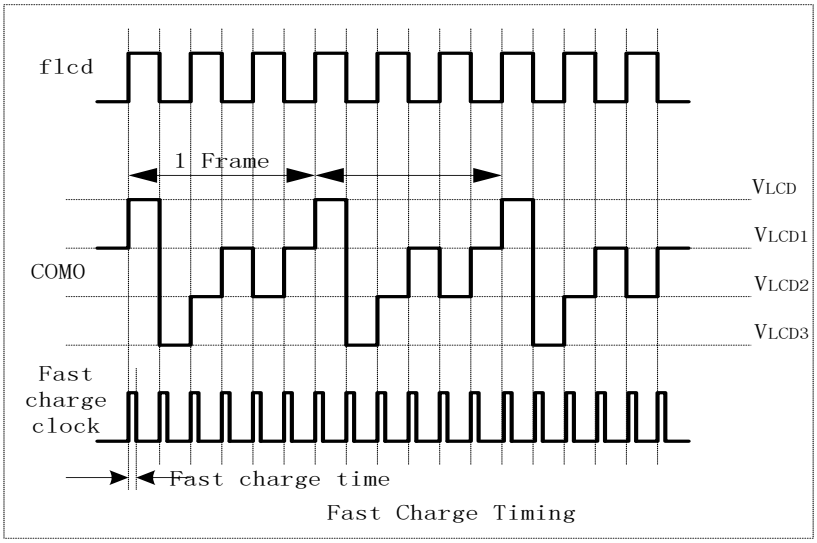
| **LCLK[2:0]** | | | **Divide** | **Flcd** | **frame rate(Hz)** | | |
|---|---|---|---|---|---|---|---|
| LCLK2 | LCLK1 | LCLK0 | **Ratio** | **(Hz)** | 1/4 | 1/6 | 1/8 |
| 1 | 0 | 0 | 64 | 512 | 128 | 85.3 | 64 |
| 0 | 0 | 0 | 128 | 256 | 64 | 42.7 | 32 |
| 0 | 0 | 1 | 256 | 128 | 32 | 21.3 | 16 |
| 0 | 1 | 0 | 512 | 64 | 16 | 10.7 | 8 |
| 0 | 1 | 1 | 1024 | 32 | 8 | 5.3 | 4 |

Note:configuration of BASCTL,DUTY[2:0] reference LCD specification! LCLK[2:0] should be set to 64~100Hz.

| **LCD Control Register (LCDCON LCD drive control register)** | | | **Base address: 0x4000D000**<br>**Offset address: 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | FCSET2 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | FCSET1 | FCSET0 | FC | LC | VRSEL 3 | VRSEL 2 | VRSEL 1 | VRSEL 0 |
| **Reset:** | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| **Bits** | **Function description** |
|---|---|
| FCSET2 | **Resistor divider structure control bit**<br>0: maintain the original resistance divider (default)<br>1: to improve the post resistance divider structure (larger current mode, less current in small current mode) |
| FCSET[1:0] | **Fast charge time select bits**<br>**FCSET1** / **FCSET0** / **Fast charge time**<br>0 / 0 / 1/8 flcd period<br>0 / 1 / 1/16 flcd period<br>1 / 0 / 1/32 flcd period<br>1 / 1 / 1/64 flcd period<br>Details as following figure 5-4-5 fast charge sequence shows |

<table>
<tr><td>FC</td><td>Fast charge mode select bits<br>Used in concert with LC to decide LCD charge mode.<br>Resistor of LCD potential-divider network is 23KΩ(VLCD=3v) by default. And RLCD=138KΩ can be chosen according to specific LCD and corresponding current flow through resistor network will decrease.<br>If RLCD=138KΩ and FC bit is set to 1, fast charge mode is selected, which means, choose RLCD=23KΩ to fast charge and switch to RLCD=138KΩ mode every time LCD output changes. Fast charge time can be selected FCSET[1:0].</td></tr>
<tr><td>LC</td><td>Slow charge mode select bits<br>Used in concert with FC to decide LCD charge mode.
<table>
<tr><th>FC</th><th>LC</th><th>LCD charge mode</th></tr>
<tr><td>X</td><td>0</td><td>RLCD=37KΩ，large current charge mode</td></tr>
<tr><td>0</td><td>1</td><td>RLCD=146KΩ，small current charge mode</td></tr>
<tr><td>1</td><td>1</td><td>Large current for short time,fast charge mode</td></tr>
</table></td></tr>
<tr><td>VRSEL[3:0]</td><td>LCD display contrast set bits
<table>
<tr><th>VRSEL3</th><th>VRSEL2</th><th>VRSEL1</th><th>VRSEL0</th><th>1/3 bias contrast (% of VLCD)</th><th>1/4 bias contrast (% of VLCD)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>97.4</td><td>99.1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>92.4</td><td>94.0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>85.8</td><td>88.8</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>80.0</td><td>84.2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>75.0</td><td>80.0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>70.6</td><td>76.2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>66.6</td><td>72.8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>63.2</td><td>69.6</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>60.0</td><td>66.6</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>57.2</td><td>64.0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>54.6</td><td>60.0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>52.2</td><td>59.2</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>50.0</td><td>57.2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>48.0</td><td>55.2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>46.2</td><td>53.4</td></tr>
</table></td></tr>
</table>




Fast charge sequence

| LCDOUT (LCD output control register) | | | Base address: 0x4000D000 Offset address: 0CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| **Read:** / **Write:** | X | X | X | X | X | X | X | OUTEN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Function Description |
|---|---|
| OUTEN | **LCD output enable control bit**<br>0：LCD output close<br>1：LCD output open（default） |

# 12. WDT module

## 12.1. General introduction

Watchdog only has reset function, no interrupt function;working state of watchdog is controlled by two signals.

(1) External pin JTAG_WDTEN,if this pin is low level,watchdog will be shut down,and if it is high level,state of watchdog depends on low frequency RC

(2) Work state of low frequency RC is controlled by LRC_CTRL bit of ControlByFlash. And it is not relevant to Sleep and Hold mode. Write 0 to LRC_CTRL to shut down low frequency RC and it takes more complex logic. Details as LRC_CTRL control shows.

Watchdog timer is a special timer and internal low frequency functions as its clock. A overflow pulse and WDT reset signal or WDT interrupt signal will be generated if timer counts to pre-set value. If Watchdog Timer is cleared before overflow pulse is generated,no WDT reset signal will be generated. Features:

- employ hardware dog design
- can use external pin TEST and JTAG_WDTEN to control

## 12.2. Operation mode

WDT will be enabled when JTAG_WDTEN is pulled up; WDT counter will be disabled and WDT_CNT will be cleared if pulled down.

| JTAG_WDTEN | WDT module |
|---|---|
| 0 | Disabled |
| 1 | Whether to open depends on low frequency RC(LRC) |

Note:if WDT is shut down because of RC,it is necessary to clear WDT Counter to ensure that it counts from 0 when opened again.

## 12.3. Special function register list

| WDT register base address：0x40010000 | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x04 | WDTCLR | R/W | 0x0040 | Watchdog feed and time configuration register |
| 0x08 | WDTCNT | R | 0x0000 | Watchdog counter register(read-only) |

## 12.4. Special function register introduction

| **WDTCLR (WDT feed and time configuration register)** | | | **Base address：** **0x40010000** **Offset address：** **04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | CLR[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | SET[7:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| CLR[7:0] | **WDT feed control bits：**<br>Write 0xAA to them to clear dog,cleat WDT internal counter WDTCNT,invalid to write other number.<br>Write only for high 8bits,and the read value is always 0 |
| SET[7:0] | **WDT overflow time set：**<br>WDT overflow time =64ms * (1+SET[7:0])<br>SET[7:0] is 8bits unsigned num,get from formula above.it is 64ms at least and 16384ms at most,4160ms by default. |

| **WDTCNT （WDT count register）** | | | **Base address：** **0x40010000** **Offset address：** **08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | CNT[15:0] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | CNT[7:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CNT[15:0] | **WDT count register：**<br>Indicate current internal count value of WDT |

# 13. Timer Module

## 13.1. Timer unit general introduction

There are 4way timer,and clock source of timer0、1、2、3 are system clock Fsys and can be configured through SYSCLK_SEL[2:0] to be:Internal frequency RC clock(Flrc),internal high frequency RC clock(Fhrc),internal low frequency clock(Fosc) and internal high frequency(Fpll).

It is necessary to enable corresponding timer module in CLKCTRL1 before configurations timer unit related register.

Timer has following functions:

1. Timer

2.PWM

3.Capture function

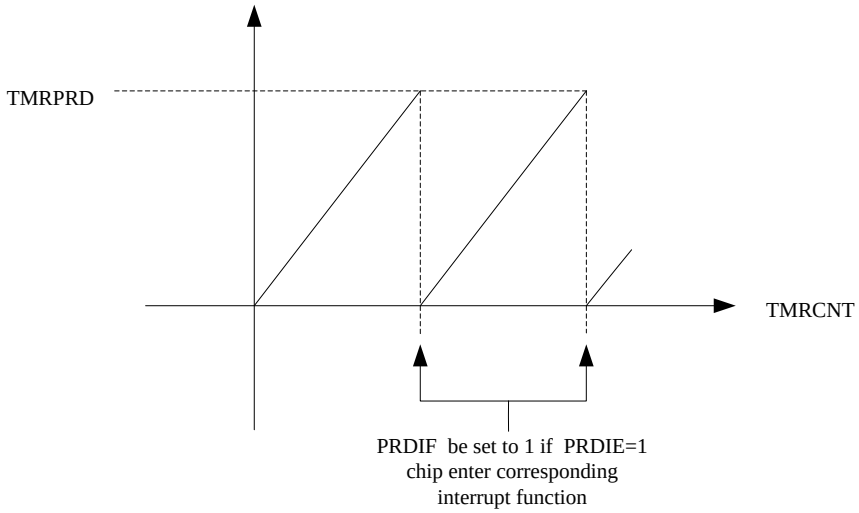
## 13.2. Period time function

General timer contains a 16bits counter and a period register. The counter clock is obtained by dividing the system clock (Fsys) through the prescaler (TMRDIV) in the timer unit. When the enable counter (CNTEN=1) after the timer counter starts counting from 0, when the count register (TMRCNT) value of the cycle register set (TMRPRD+1) will set a cycle timer interrupt flag (PRDIF=1), if the timer interrupt cycle (PRDIE=1), will trigger the timer interrupt cycle, enter the corresponding the periodic interrupt.

When the cycle time interrupt flag is set (PRDIF=1), the value of TMRCNT is automatically cleared, and then the count is restarted.

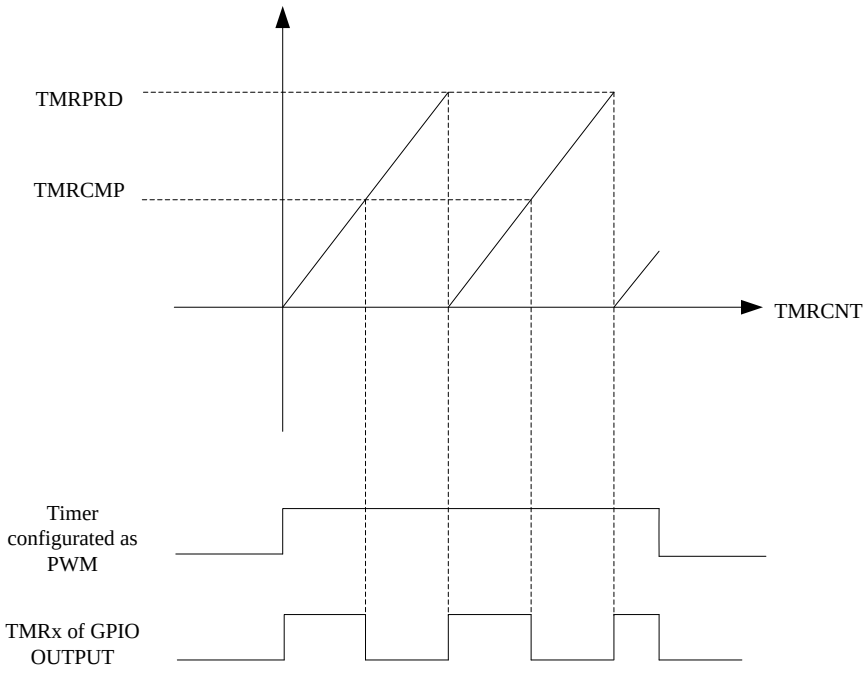
TMRPRD, if modified, will be effective next time after this timing has been completed.

Main function related register: TMRCNT， TMRPRD.

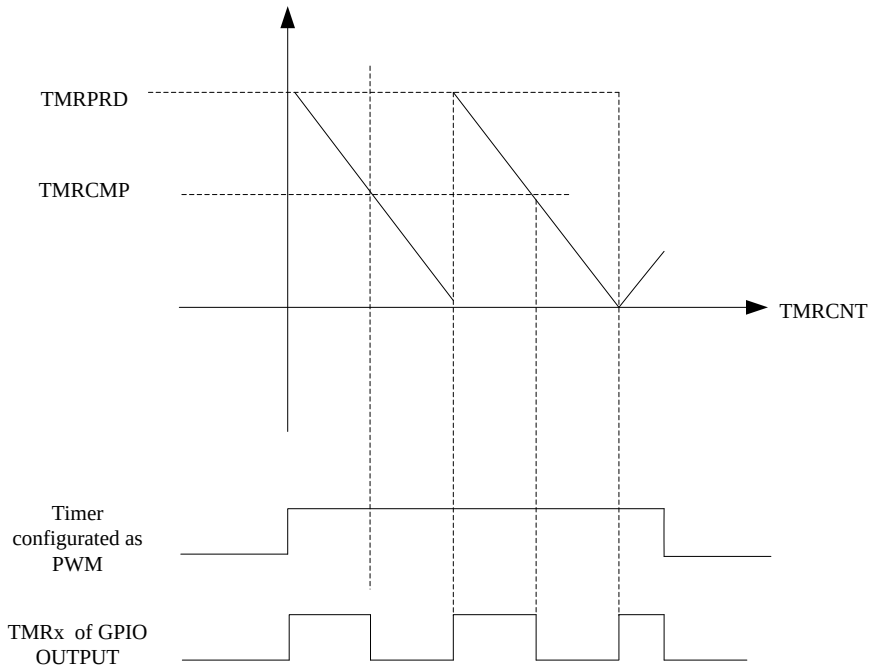


## 13.3. PWM function

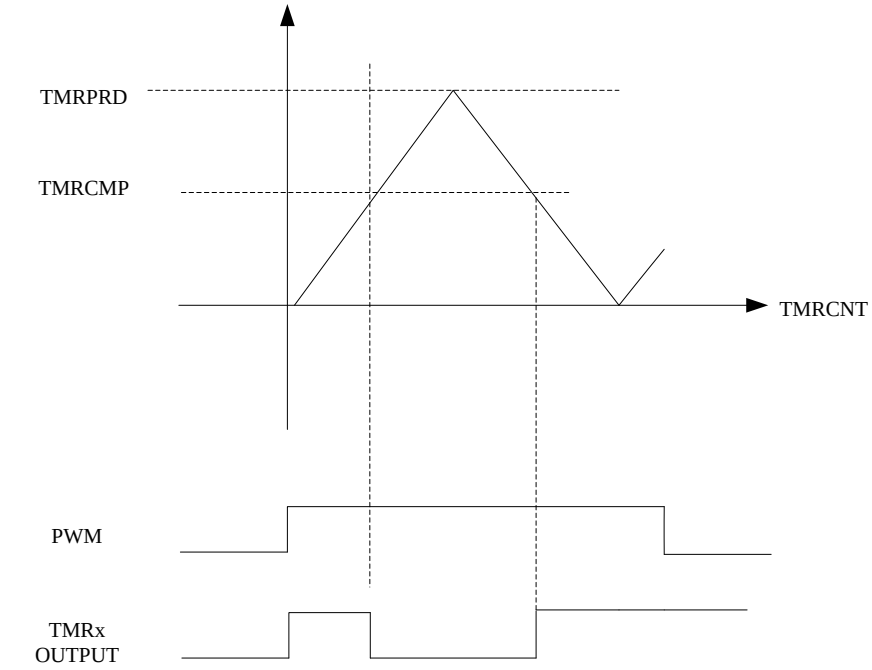
PWM can be configured through register TMRCOM.MODE[1..0],and corresponding GPIO needs to be set as TMR. And correspond GPIO multiplex TMR pin will output wave if the configuration is right. Period and duty of PWM can be configured through TMRPRD,TMRCMP. Counter start counting and output of PWM turnover,compare interrupt flag will be set when value of counter TMRCNT equals to that of TMRCMP if counter is enabled. Output of PWM turnover again and period interrupt flag will be set when value of counter TNRCNT equals to TMRPRD. Waveform of PWM:




There are three counting ways of PWM: count up, count down, center-aligned. Fig above is for count up method.




PWM configured as center—aligned counting. If it count down from TMRCNT and the count value equals to TMRCMP, output of PWM will turnover. If it continues to count down from TMRCNT, output of PWM will turn-over again if count value equals to 0. Waveform of PWM:




Initial level of PWM is high if 19.66MHz generated by PLL is chosen to be system clock and pre frequency divider TMRDIV of timer is 0 by default,timer0 is PWM of counting up. If need to set PWM duty to be 30%,TMRCMP/TMRPRD=30%. Determine value of TMRPRD according to demands. The maximum period of TMRPRD is (0Xffff+1)/(19.66M/(TMRDIV+1))≈218.46s.

It will take effect after this round of counting if TMRCMP,TMRPRD are modified.

Main function related register:TMRCNT,TMRCMPM,and TMRPRD.

## 13.4. Capture function

If rising edge is detected on Timer0~Timer3 in input capture mode if it is set to detected rising edge,current counter value will be locked to capture register. When capture event occurs, corresponding capture interrupt flag TMRIF.1 will be set to 1 and interrupt generated if it is enabled.

Period interrupt will generate when TMRCNT equals to TMRPRD if no edge is detected during capturing. And CPU will enter interrupt service routine and TMRCNT count from 0 later if period is enabled.

Rising edge and falling edge can be detected by Capture detection.

Main function related register:TMRCAP

## 13.5. Event count function

In the input event count mode, if the falling edge is detected, when the falling edge is detected on the Timer0~Timer3 pin, the current value of the count register (TMRCNT) is added to 1. When the value of the count register (TMRCNT) is equal to the set of comparison registers (TMRCMP+1), the event count interrupt flag (ACIF=1) is set, and the count register (TMRCNT) is automatically reset and counted again. If enabled, the event count interrupt (ACIE=1) generates an event count interrupt and enters the corresponding event count interrupt service routine.

The event counting process, in the case of TMRCMP>TMRPRD, when the count register (TMRCNT) value of the cycle register set (TMRPRD+1) will set a cycle timer interrupt flag (PRDIF=1), count register (TMRCNT) to count until the comparison set equal to send register (TMRCMP+1), if the timer interrupt cycle (PRDIE=1) will enter the periodic interrupt.

The number of single maximum count is 0xFFFF, which can be combined with any number of cycles. Event count detection can select the rising edge and the falling edge mode, which can be used for MCU pulse.

Compare register (TMRCMP) as being modified, if the modified value is less than the current count register (TMRCNT) value, immediately triggered the event count interrupt, and count register (TMRCNT) cleared, start counting; if the modified value is greater than the current count register (TMRCNT) value, continue the count.

Main function related registers: TMRCNT, TMRCMP, TMRPRD.

## 13.6. Interrupt function

### 13.6.1. Timer period interrupt

A period interrupt will be generated(under any circumstance) if period interrupt is enabled(TMRIE.PRDIE=1)

when TMRCNT equals to TMRPRD.

### 13.6.2. Capture interrupt

A capture interrupt will be generated if capture interrupt is enabled(TMRIE.CMPIE=1) when corresponding edge of external input signal is detected. Value of TMRCNT is lock to TMRCAP.

### 13.6.3. Compare interrupt

Compare interrupt will be generated if PWM compare interrupt is enabled(TMRIE.CMPIE=1) when value of TMRCNT equals to TMRCMP.

Period interrupt will be generated if period interrupt is enabled(TMRIE.PRDIE=1) when value of TMRCNT equals to TMRPRD.

## 13.6.4. Event counter interrupt

In the input event count mode, if the falling edge is detected, when the falling edge is detected on the Timer0~Timer3 pin, the current value of the count register (TMRCNT) is added to 1. When the value of the count register (TMRCNT) is equal to the set of comparison registers (TMRCMP+1), the event count interrupt flag (ACIF=1) is set, and the count register (TMRCNT) is automatically reset and counted again. If enabled, the event count interrupt (ACIE=1) generates an event count interrupt and enters the corresponding event count interrupt service routine.

The event counting process, in the case of TMRCMP>TMRPRD, when the count register (TMRCNT) value of the cycle register set (TMRPRD+1) will set a cycle timer interrupt flag (PRDIF=1), count register (TMRCNT) to count until the comparison set equal to send register (TMRCMP+1), if the timer interrupt cycle (PRDIE=1) will enter the periodic interrupt.

The number of single maximum count is 0xFFFF, which can be combined with any number of cycles. Event count detection can select the rising edge and the falling edge mode, which can be used for MCU pulse.

Compare register (TMRCMP) as being modified, if the modified value is less than the current count register (TMRCNT) value, immediately triggered the event count interrupt, and count register (TMRCNT) cleared, start counting; if the modified value is greater than the current count register (TMRCNT) value, continue the count.

Main function related registers: TMRCNT, TMRCMP, TMRPRD.

## 13.7. Special function register list

| TMR module register base address：<br>0x40001000(TMR0);<br>0x40002000(TMR1);<br>0x40003000(TMR2);<br>0x40004000(TMR3); | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 0x00 | TMRCON | R/W | 0x0000 | Control register |
| 0x04 | TMRDIV | R/W | 0x0000 | Pre-frequency-dividion register |
| 0x08 | TMRPRD | R/W | 0x0000 | Period register |
| 0x0C | TMRCAP | R/*W | 0x0000 | Data capture register |
| 0x10 | TMRCNT | R/*W | 0x0000 | Counter register |
| 0x14 | TMRCMP | R/W | 0x0000 | Compare register |
| 0x18 | TMRIE | R/W | 0x0000 | Interrupt enable register |
| 0x1C | TMRIF | R/W | 0x0000 | Interrupt flag register |

## 13.8. Special function register introduction

| **TMRCON（timer control register）** | | | **Base address：0x40001000--0x40004000<br>Offset address：00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:<br>Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:<br>Write:** | RESERVED | PWMHL | PWMC[1:0] | | CCMODE | MODE[1:0] | | CNTEN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **bits** | **Function description** |
|---|---|
| RESERVED | **Fixed 0** |
| PWMHL | **PWM initial level selection：**<br>0：high level<br>1：low level |
| PWMC[1:0] | **PWM operation mode select：（PWM counting method）**<br>00：count up<br>01：count down |

| | |
|---|---|
| | 1X：center-aligned |
| CCMODE | **Capture edge select：**<br>0：rising edge<br>1：falling edge |
| MODE[1:0] | **Timer function select：**<br>00：shut down<br>01：PWM function<br>10：capture function<br>11：period timer function<br>Need to configure GPIO as TMRx |
| CNTEN | **Counter enable bit：**<br>0：disable<br>1：enable |

| **TMRDIV (pre-frequency-division register)** | | | **Base address：** **0x40001000--0x40004000**<br>**Offset address：** **04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | TMRDIV[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | TMRDIV[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| TMRDIV[15:0] | Pre-frequency-division range:0-65535<br>Frequency after pre-frequency-divided is 1/（TMRDIV[15:0] +1） input frequency |

| **TMRPRD （period register）** | | | **Base address：** **0x40001000--0x40004000**<br>**Offset address：** **08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | TMRPRD[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | TMRPRD[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| TMRPRD[15:0] | a 16-bit register<br>period register of counter and PWM are both this register<br>need to configure this register before use any function of any mode |

| TMRCAP（capture data register） | | | Base address: 0x40001000--0x40004000<br>Offset address: 0CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | TMRCAP[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | TMRCAP[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| TMRCAP[15:0] | Current value of counter is stored to this register when capture events occurs |

| TMRCNT（count register） | | | Base address: 0x40001000--0x40004000<br>Offset address: 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | TMRCNT[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | TMRCNT[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| TMRCNT[15:0] | Current value of counter |

| TMRCMP（compare register） | | | Base address: 0x40001000--0x40004000<br>Offset address: 14H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | TMRCMP[15:8] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | TMRCMP[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| TMRCMP[15:0] | Output of PWM turnover when value of counter equals to set value of TMRCMP if timer function as PWM. |

| **TMRIE (timer interrupt enable register)** | | | **Base address: 0x40001000--0x40004000 Offset address: 18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | X | X | X | X | X | CMPIE | CAPIE | PRDIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| CMPIE | **Compare interrupt enable**<br>0：disable<br>1：enable |
| CAPIE | **Capture interrupt enable**<br>0：disable<br>1：enable |
| PRDIE | **Periodically overspill interrupt enable**<br>0：disable<br>1：enable |

| **TMRIF (Timer interrupt flag)** | | | **Base address: 0x40001000--0x40004000 Offset address: 1CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | CMPIF | CAPIF | PRDIF |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CMPIF | **Compare interrupt flag**<br>0：no interrupt<br>1：interrupt（cleared by writing 0） |
| CAPIF | **Capture interrupt flag**<br>0：no interrupt<br>1：interrupts（cleared by writing 0） |
| PRDIF | **Periodically overspill interrupt flag**<br>0：no interrupt<br>1：interrupts（cleared by writing 0） |

# 14. SPI module

## 14.1. General introduction

Pins of SPI chip are SPI_CS/PC.7、SPI_CLK/PC.6、SPI_MOSI/RX5/PC.4 和 SPI_MISO/TX5/PC.5.

MCU can full-duplex asynchronous communicate with external peripherals,external MCU included,(contains SPI module) by engaging SPI module. Features:

- Full-duplex mode
- Three-way synchronous transmission
- Master mode and slave mode
- 7 kinds of baud rate for master
- Maximum clock frequency for slave:fcpu/4
- Programmable polarity and phase for serial clock
- Write-conflict dealing mechanism
- 8-bit transmission data,high bit first,low bit last
- 8-bit slave select interface,control external slave
- Dedicated function register interface for master CPU
- Unambiguous port,standard SPI

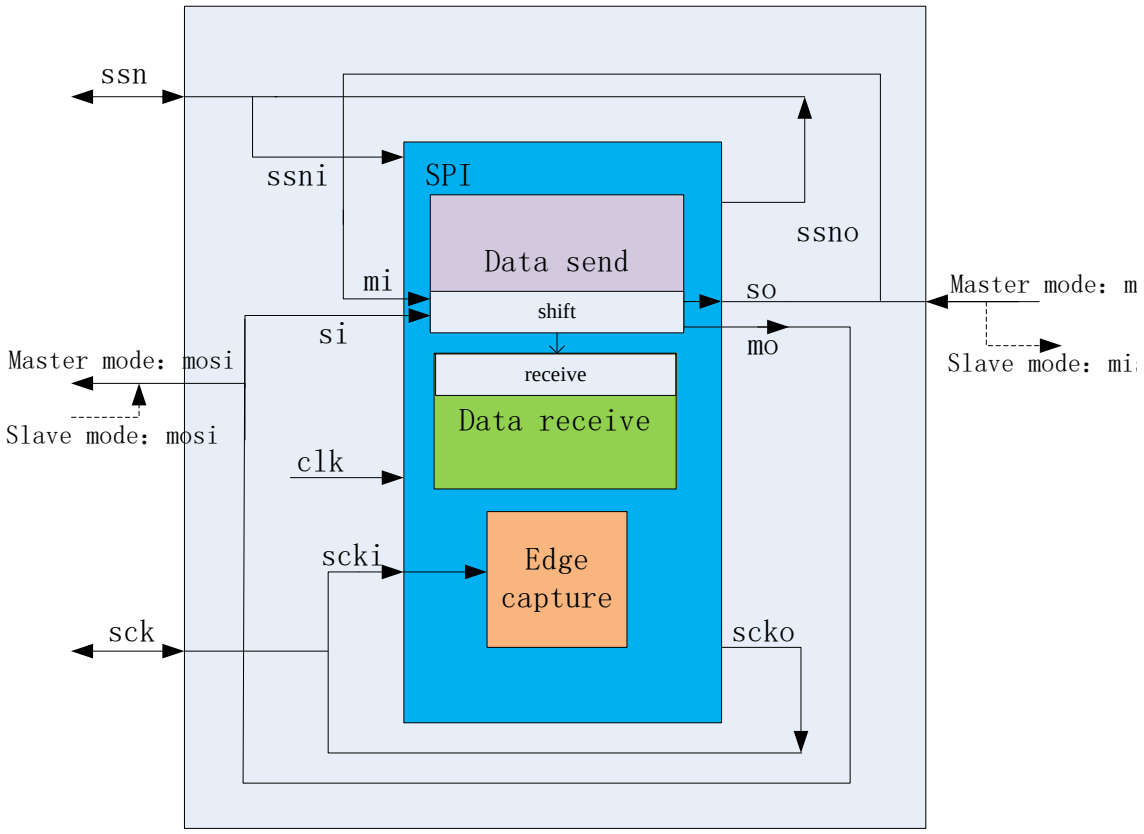
## 14.2. Detailed function introduction

Serial peripheral interface(SPI) support the communication ways with other device of half/full-duplex,synchronous,serial way.it can be configured as master mode and provide communication clock for slave device.

### 14.2.1. SPI main features

- Three-line full-duplex synchronous transmission
- Master mode or slave mode
- 7 kinds of frequency for master mode(fcpu/2, fcpu /4, fcpu /8, fcpu /16,fcpu /32,fcpu /64,fcpu /128)
- Detect Level and falling edge on input pin SPI_CS
- Programmable clock polarity and phase
- Trigger dedicated send and receive flag for interrupt

### 14.2.2. SPI




### 14.2.3. SPI interface transmission format

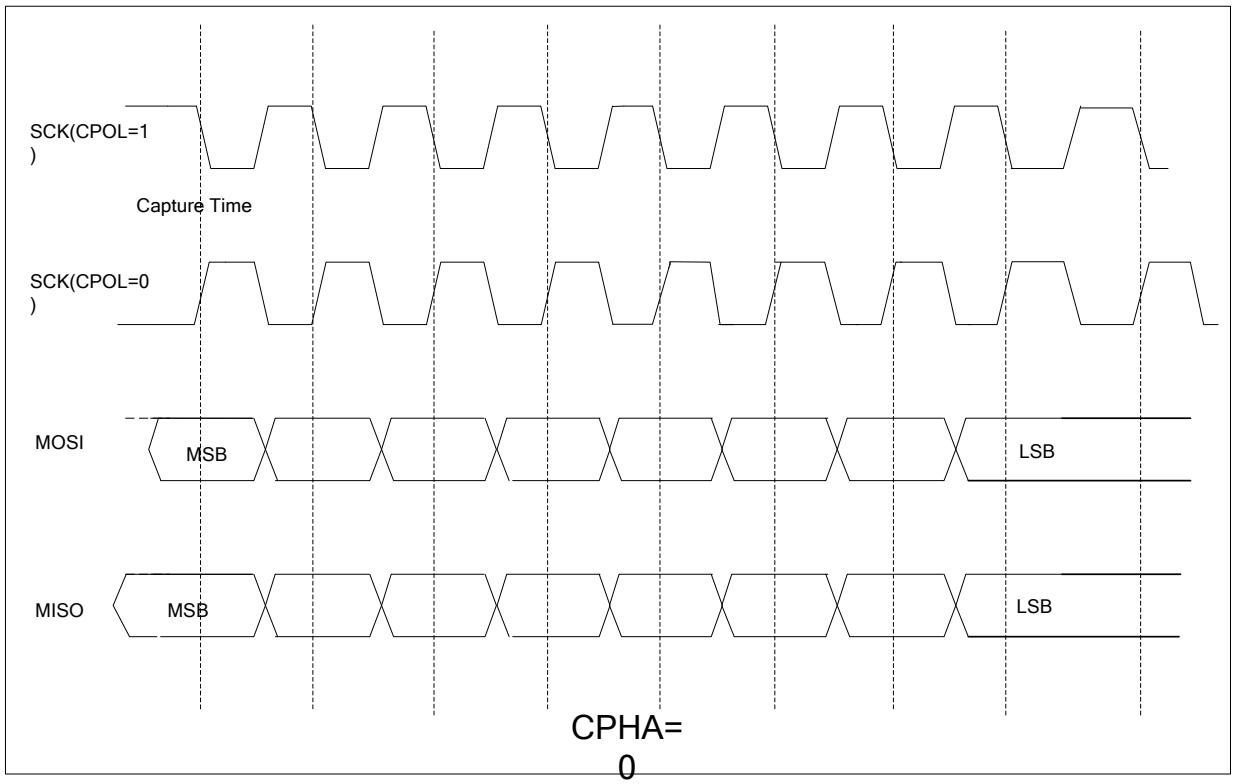
Fig blow shows the main format of data transmission. Every bit of data will be transmitted on the rising edge or falling edge of SCK according to SPI settings.And data is received on falling edge (CPOL=0) or rising edge (CPOL=1).it goes for master mode or transmitter/receiver of slave mode on the condition that SCK is the main clock during transmission. If CPHA is set,first bit(MSB) will be sent on the first dynamic edge by MOSI/MISO. If CPHA is cleared, first bit(MSB) will be sent half period before the first dynamic edge.

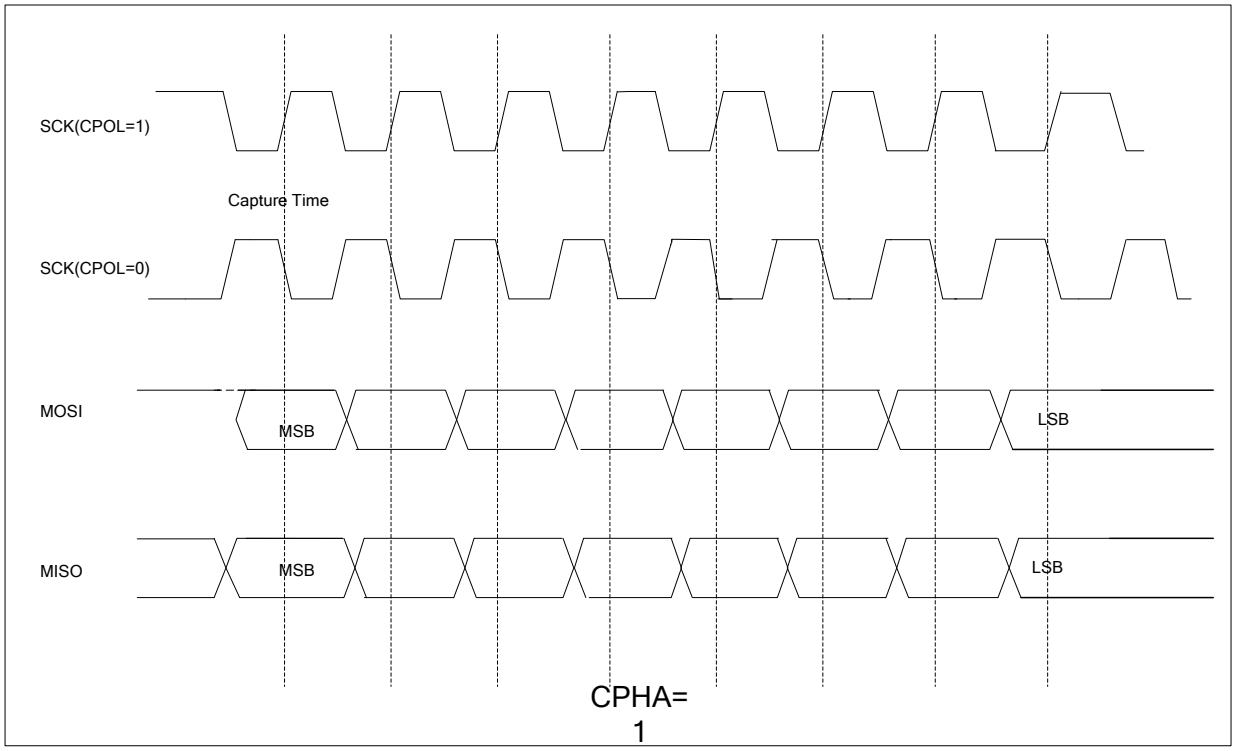
Besides,input data will be sampled when every bit is being transmitted and data will be shifted to output signal MOSI on reversed level of this period.

### 14.2.4. Master mode transmission format

SPI is in master mode by default.

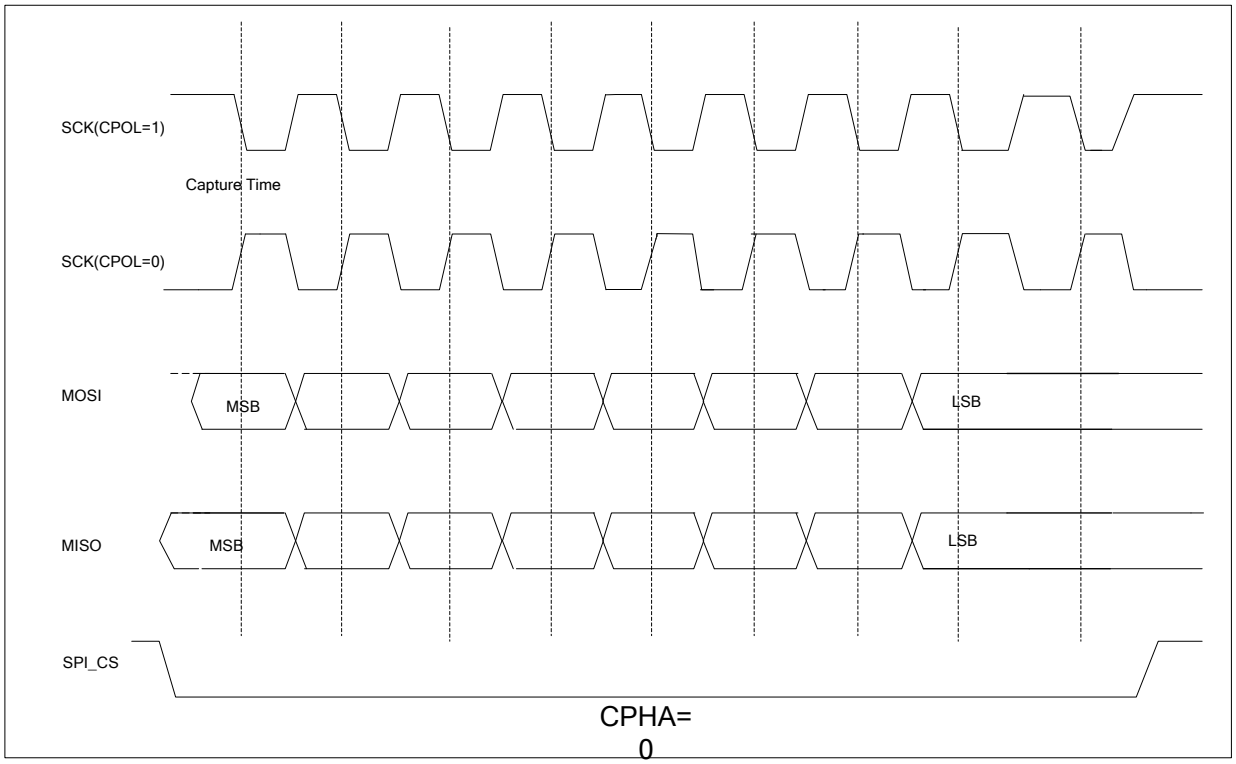
SPI wait the program writing data to register SPDAT in master mode. If write operation to SPDAT is completed, transmission starts. Data is shifted to MOSI pin on SCK sending edge. And other byte data from slave is shifted to input pin MISO of master.

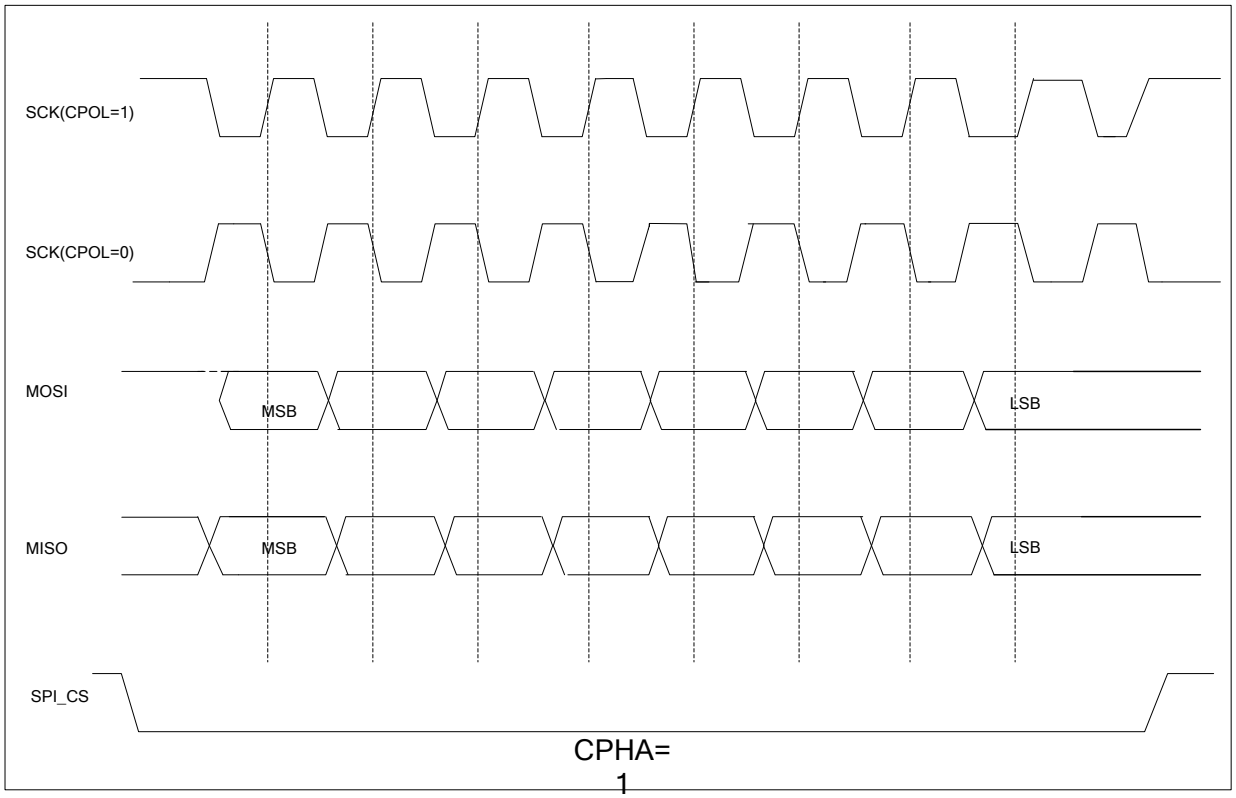


## 14.2.5. Slave mode transmission format

First,need to write to MSTR=0,to configure SPI as slave mode, and set SPI_EN to enable SPI.



data transmission format in slave mode

SPI wait low level of input signal SPI_CS. When SPI_CS falling edge is captured,transmission start and SPI_CS keeps low level until transmission complete in slave mode. CPHA of SPCON decide the start position of transmission ,if CPHA is cleared,slave must start transmitting before first falling edge of SCK; if CPHA is set,slave will start transmitting on first falling edge of SCK

### 14.2.6. Interrupt function

| Name | SPI interrupt flag description |
|---|---|
| SPIF | This flag is set when transmission completed |
| MODF | When state of SPI_CS is in conflict with master-slave mode |

## 14.3. Special function register list

| **SPI module register base address：0x4000B000** | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | Write/read | **Reset value** | **Function description** |
| 0x00 | SPICON | R/W | 0x0000 | SPI control register |
| 0x04 | SPISTA | R/W | 0x0000 | SPI state register |
| 0x08 | SPIDAT | R/W | 0x0000 | SPI data register |
| 0x0C | SPISSN | R/W | 0x00FF | SPI slave select register |

## 14.4. Special function register introduction

| **SPICON （control register）** | | | **Base address： 0x4000B000 Offset address： 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | SSDIS | SPR[2:0] | | | CPHA | CPOL | MSTR | SPI_EN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|

| | |
|---|---|
| SSDIS | **SS control bit**<br>0：enable SPI_CS input in master/slave mode<br>1：disable SPI_CS input in master/slave mode,no MODF interrupt request will generate in this case;if CPHA=0, this bit is invalid in slave mode |
| SPR[2:0] | **SPI clock rate control bits(in master mode)** |

| SPR[2:0] | | | **SPI clock rate** |
|---|---|---|---|
| 0 | 0 | 0 | Fsys/2 |
| 0 | 0 | 1 | Fsys/4 |
| 0 | 1 | 0 | Fsys/8 |
| 0 | 1 | 1 | Fsys/16 |
| 1 | 0 | 0 | Fsys/32 |
| 1 | 0 | 1 | Fsys/64 |
| 1 | 1 | 0 | Fsys/128 |
| 1 | 1 | 1 | Generate no main clock |

| | |
|---|---|
| CPHA | **Clock phase**<br>0：high bits(MSB) will be transmitted half period before first dynamic edge of SCK<br>1：high bits(MSB) will be transmitted by MOSI/MISO on first dynamic edge of SCK |
| CPOL | **Clock priority**<br>0：“sck” is set to low level when it is free<br>1：“sck” is set to high level when it is free |
| MSTR | **SPI mode select bit**<br>0：slave mode<br>1：master mode |
| SPI_EN | **SPI enable bit**<br>0：disable SPI module<br>1：enable SPI module |

| **SPISTA（state register）** | | | **Base address：** **0x4000B000**<br>**Offset address：** **04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | SPIF | WCOL | SSERR | MODF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| SPIF | **Data transmission complete flag**<br>Set by hardware when transmission complete;reset by hardware while transmission or read register “spsta”SPDAT to reset; |

| | |
|---|---|
| WCOL | **Write conflict flag**<br>Set by hardware if it is in conflict when write to SPDAT;reset by hardware if no conflict generate while transmitting or access register "spsta"SPDAT to reset it. |
| SSERR | **Synchronous slave error flag**<br>Set by hardware if SPI_CS is valid before receive completed;disable SPI to clear this bit(set spen=0) |
| MODF | **Mode error flag**<br>Set by hardware if state of SPI_CS pin is in conflict with mode set;reset by hardware if SPI_CS pin recover to proper level or read "spsta" register by software to reset |

| **SPIDAT（data register）** | | | **Base address: 0x4000B000**<br>**Offset address: 08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | SPDAT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| SPIDAT[7:0] | Register SPIDAT is write/read buffer of "data receive" register.it write to shift register(without transmission buffer) when write data to SPIDAT;it returns data from receive buffer but not shift register when read data of SPIDAT |

| **SPISSN（slave select register）** | | | **Base address: 0x4000B000**<br>**Offset address: 0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | SSN0 |

Note: SPSSN is a read-write register,every bit of it can be used for selection of independent external SPI slave device.

| **Bits** | **Function description** |
|---|---|
| SSN0 | It uses SSN0 control bit to control level of external SPI_CS pin if chip is used for master of SPI communication.<br>Write 1 to pull up CS and write 0 to pull down it when SPI is enabled |

# 15. I2C module I2C

## 15.1. General introduction

Pins of I2C are SCL/PF0 and SDA/PF1.

I2C provide a serial interface conform to Philips I2C bus regulations,two lines for data transmission between device and bus,and state register I2CSTA reflect the real-time state of I2C bus controller.

## 15.2. Diagram

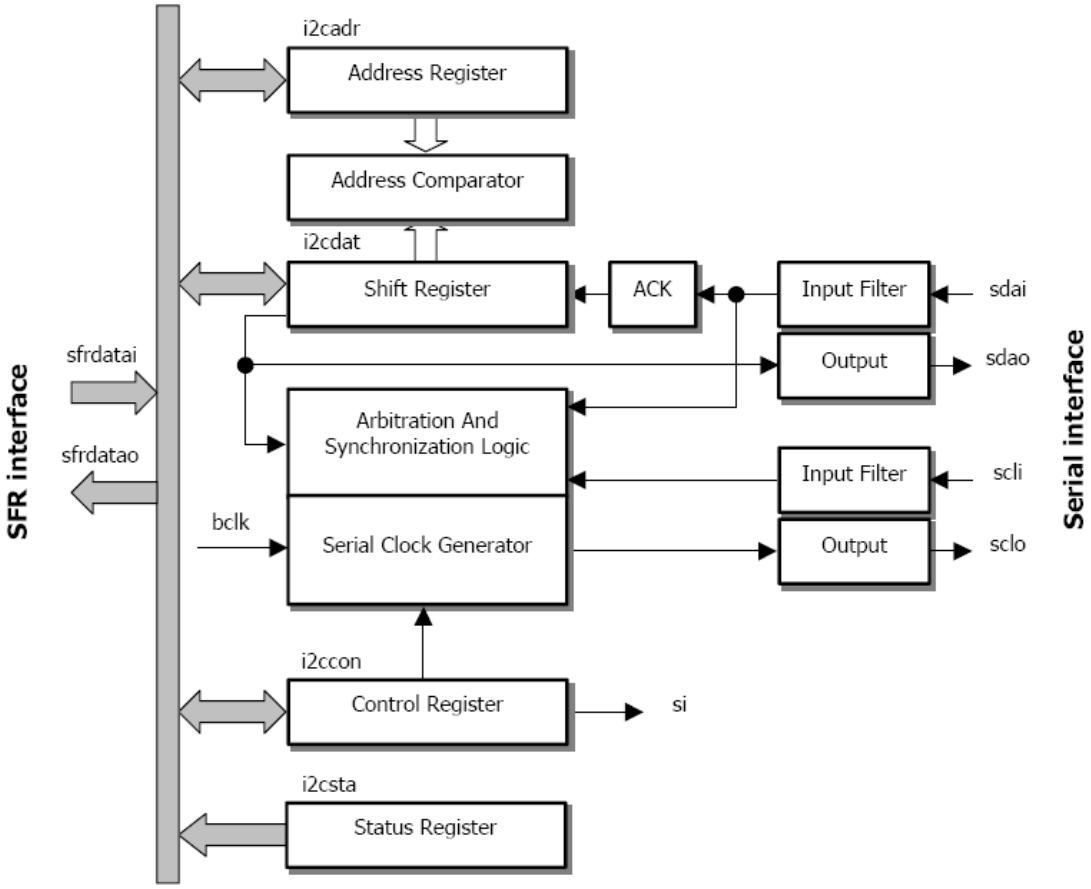



Fig1 module function diagram

## 15.3. Function description

Two lines of I2C for data transmission between device and bus:serial clock SCL and serial data SDA. Every device connected to bus occupies a unique address. I2C is a real multiple-computer bus,it has conflict detect and

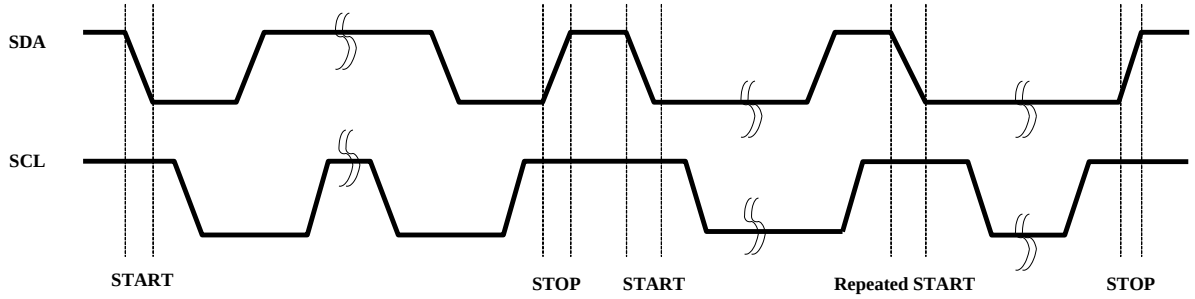
arbitrate mechanism in case of data loss when multiple computers start transmitting data.

### 15.3.1. Start Condition and End Condition

The I2C protocol defines two special waveforms: the starting and ending conditions.

When the clock line is high, the falling edge of the data line is defined as the starting condition; when the clock line is high, the rising edge of the data line is defined as the termination condition. The starting conditions and termination conditions are issued by the host.

A host can initiate and terminate a transmission. When the host sends a start condition, it starts the transmission and ends the transmission when a termination condition is sent. Between the start and end conditions, the bus is defined as "busy" state. Other hosts should not attempt to initiate transmissions. In the "busy" state, if the host sends the starting condition again, it is defined as the "repeated start condition", which indicates that the host wants to start a new transmission without giving up the bus. After sending a duplicate start condition, the bus is still in busy state until the bus terminates. In view of the consistent nature of the repetition start conditions and the starting conditions, starting conditions shall be used in place of this unless otherwise stated.



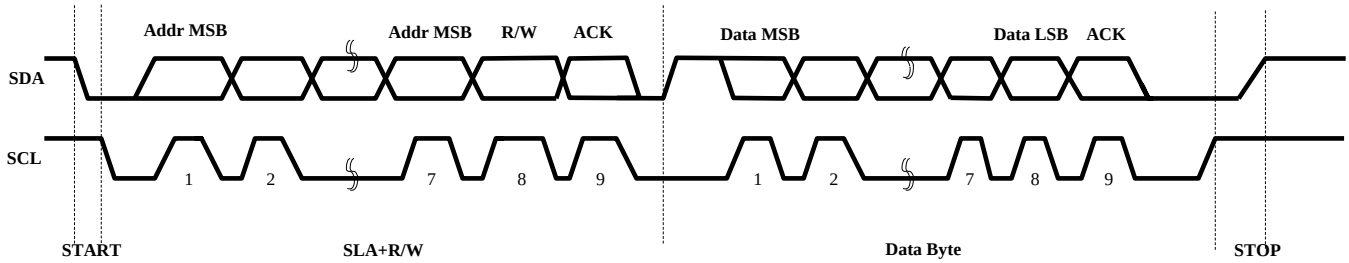

### 15.3.2. Data Transfer Format

In data transmission, each bit of the data line needs a pulse on the clock line. The data line should be stable at high voltage. This rule is not required for the start and termination conditions of the sending.

I2C transmits from high to low bit by bit, and all data packets (including address packets) consist of 9 bits, including 1 bytes and one response bit. The host is responsible for issuing the clock and the starting and ending conditions, and the receiver is responsible for giving the reply signal.

The receiver sends a "ACK" signal by pulling the data line down at ninth clock pulses; or maintains a high level at ninth pulses indicating an "NACK" signal. When the receiver receives the last byte, or, for some reason, cannot continue to receive the data, it should respond to the "NACK" signal.

A transmission usually includes a start condition, address + read / write, one or more data packets, and a termination condition. Data formats that contain only starting and terminating conditions are incompatible with the rules of communication.



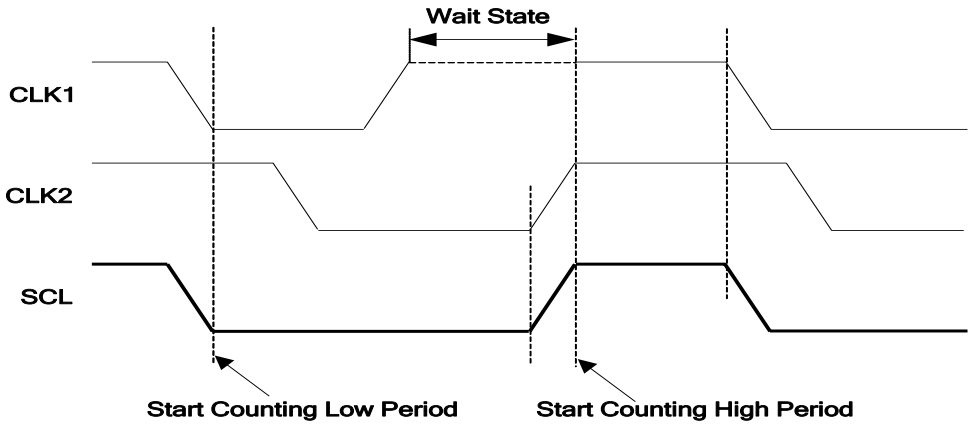

## 15.3.3. Clock Synchronization and Data Arbitration

When multiple hosts want to control the bus at the same time, the bus will determine the clock line level according to the line and principle.

**Clock Synchronization:**

The clock line jumps from high to low, causing all devices that participate in the transmission to start clocking low. Every time to release their own device clock line low level requirements, the clock line waiting period for high level before entering the high level; when all devices are full of low level cycle when the clock line is changed to high levels. After that, all devices start clocking high, and the first device, which is full of high level cycles, pulls the clock line into the next clock cycle.

In this way, the low level cycle of the synchronous SCL clock is determined by the longest device in the low-level clock cycle, while the high level cycle is determined by the shortest device in the high clock period.




It is important to note that the "wire" and "structure" provide convenience for handshaking between the host and slave. When the host is relatively fast or the slave needs to process other transactions, the slave can extend the low level of the clock line by pulling down the clock line, thus reducing the communication frequency. The slave can elongate the low level of the clock line, but does not affect the period of the clock line high level.

**Data Arbitration:**

The host can only start the transmission once the bus is in the "idle" state. Two or more hosts may simultaneously start a start condition at the minimum hold time (T, HD: STA), so that only one starting condition is visible on the bus.

Because the host of the starting condition of the transmission cannot know whether any other host is in the contention bus, the arbitration of the data line can only be determined by the clock at which the host takes the bus. When a host transmits low power, the host that transmits the high level will lose arbitration and disconnect its data output stage.

The host that loses the arbitration will continue sending the clock until the current transport byte is sent. When two hosts access one slave at the same time, they may pass through the address phase smoothly, and the arbitration continues to compare the data bits (if the host sender) or compare the response bits (if the host receiver). Therefore, the address and data information of the I2C bus is determined by the host who wins the arbitration and will not lose the information during the arbitration

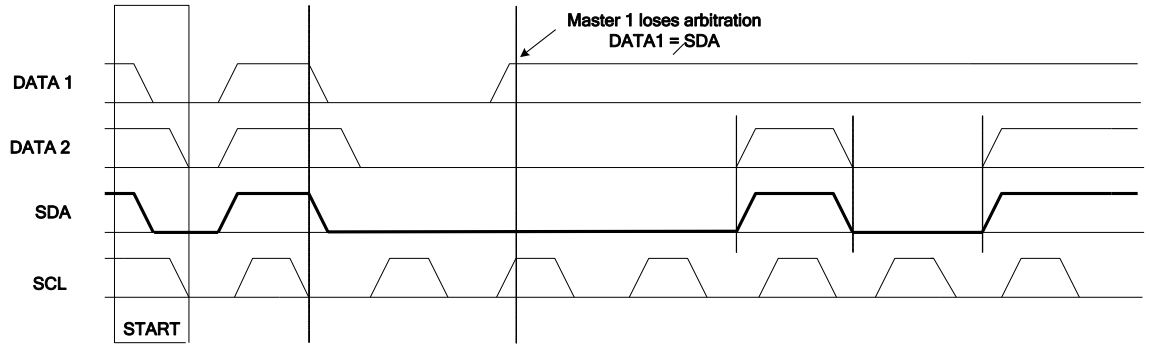
If the host was started at the same time from the mode, loss of arbitration in the sending address stage should detect whether the address line match with their own; if it is for their own access, should immediately switch to slave mode, receiving information.

In serial transmission, the arbitration process is still ongoing when the repeating start condition or stop condition is sent to the I2C bus.

**Arbitration shall not be conducted between the following circumstances:**

1. Duplicate start conditions and data
2. Data and termination conditions
3. Termination conditions and repeated starting conditions

The slave machine, not to participate in arbitration.



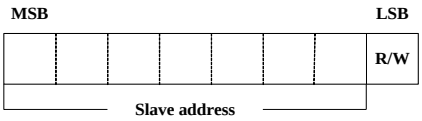

## 15.3.4. Bit address format

The addressing process of the I2C bus is usually the first byte after the start condition, which determines which slave is selected by the host. The address matching unit from the machine checks whether the received address matches the 7 bit address in the register I2CADR. If the broadcast call address enable bit I2CADR[0] is set, it also detects whether it matches the broadcast call address 00H. When the address matches, the control unit generates the appropriate action and the corresponding status code.

The first 7 bits of the first byte form the slave address, and the lowest bit LSB is the eighth bit, which determines the direction of the message. The lowest byte of the first byte is "0", which indicates that the host will write information to the selected slave, and "1" indicates that the host will read the information from the slave.

When sending an address, each device in the system in the initial conditions after the comparison, the first 7 with its own address if the same, the device will think that it is the host addressing, as is from the receiver or

transmitter from the machine by R/W decision machine.




The first byte after Start condition

### 15.3.5. Serial Clock Generate

When the I2C is in host mode, the programmable clock generator provides the SCL clock; when the I2C is in slave mode, the clock generator is turned off and the clock from the host is received. The output frequency of the clock generator can be controlled by the bit CR[2:0] in the register I2CCON. Which contains I2CCON[0... 1], I2CCON[7... 14].

### 15.3.6. Interruption

Enable ENS1, start the I2C module, I2C module real-time monitoring I2C bus state, and according to the user settings, the corresponding operation and response to the bus. When the application requirements of the bus are detected, the flag bit SI in the register I2CCON is set, and the current application state is written to the status register I2CSTA. If the I2C interrupt is enabled, a I2C interrupt is generated.

When the interrupt flag SI is set up, the SCL of the clock line is pulled down, the communication is paused, and the user needs to clear the sign to continue the communication.

### 15.3.7. Operation mode

I2C bidirectional transmit 8-bit data, and transmit rate can reach 100kbit/s in normal mode and 400kbit/s in fast mode.4 operation modes can be shown as follow:

- Master sending mode:SDA output serial data, SCL output serial clock
- Master receiving mode:SDA input serial data, SCL output serial clock
- Slave receiving mode:SDA input serial data, SCL input serial clock
- Slave sending mode:SDA output serial data, SCL input serial clock

The following four main modes of I2C communication are described, and all possible status codes are described. The following figures are abbreviated below:

S : Start condition
Rs : Re-start condition
R : Read control bit
W : Write control bit
A : Answer Bit
Ā : No answer Bit
DATA : 8 bit data

P : Pause condition

SLA : Slave address

A circle is used to indicate an interrupt; a flag has been set. The numbers represent the status codes in the current status register I2CSTA which are masked to three bits low. Before SI is cleared, the I2C communication is paused, and the application must decide whether to continue or terminate the current transmission. For each status code, the required software actions and subsequent transport details are described.

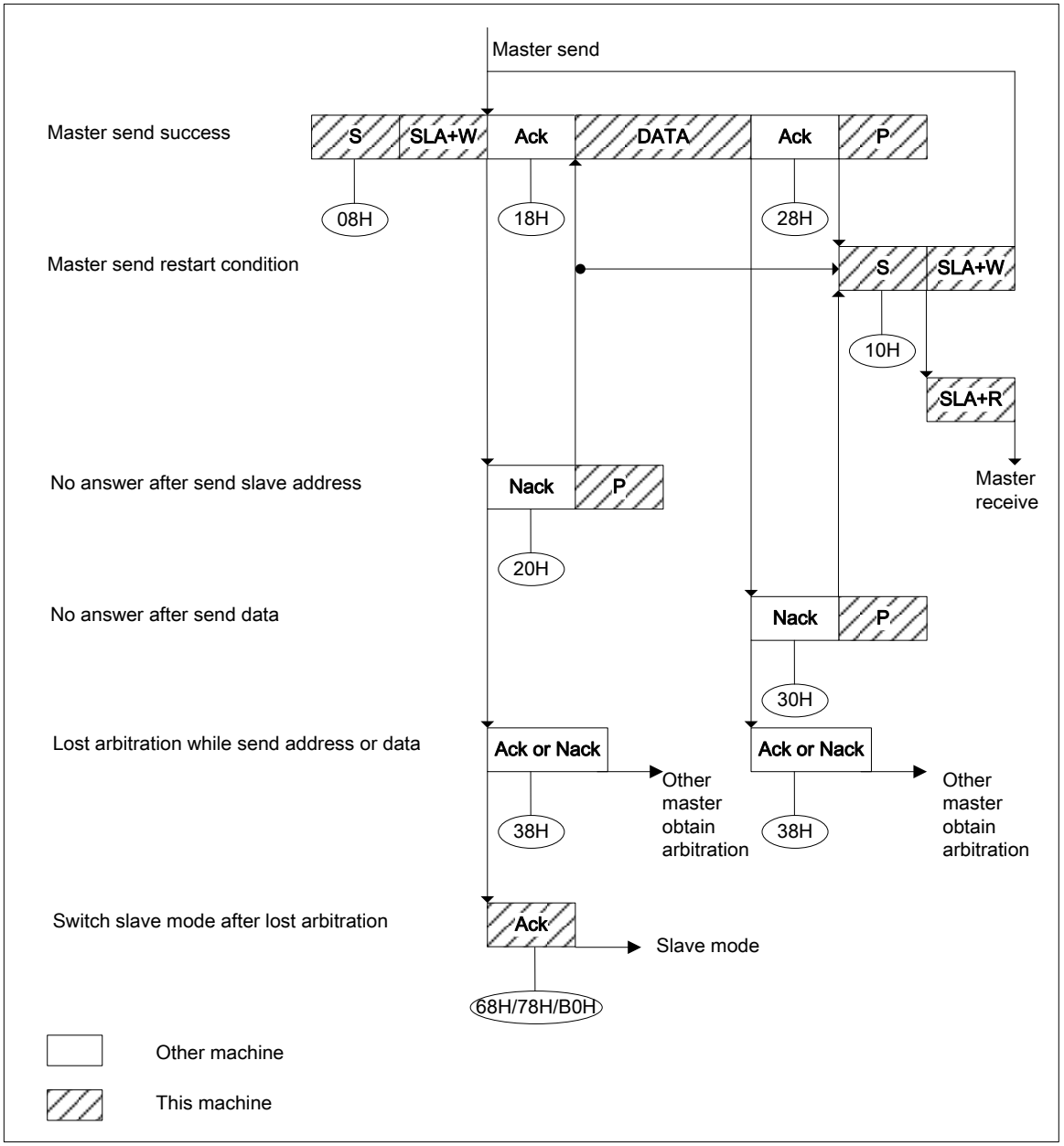
**I2C master send mode:**

In master sand mode, master machine send data to slav, one start condition(S), follow a slave address(SLA)+ write control Bit(W), indicates the mode of access to master sand mode.

**I2C Status In Master Transmitter Mode**

| Status code | Status of the I2C | Application software response to/from I2CDAT | to I2CCON sta | sto | si | aa | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| 08H | START condition has been transmitted | Load SLA+W | X | 0 | 0 | X | SLA+W will be transmitted<br>ACK will be received |
| 10H | Repeated START condition has been transmitted | Load SLA+W<br>or | X | 0 | 0 | X | As above |
| | | Load SLA+R | X | 0 | 0 | X | SLA+R will be transmitted<br>I2C will be switched to "master receiver" mode |
| 18H | SLA+W has been transmitted;<br>ACK has been received | Load data byte | 0 | 0 | 0 | X | Data byte will be transmitted; ACK will be received |
| | | or no action | 1 | 0 | 0 | X | Repeated START will be transmitted; |
| | | or no action | 0 | 1 | 0 | X | STOP condition will be transmitted; the "sto" flag will be reset |
| | | or no action | 1 | 1 | 0 | X | STOP condition followed by a START condition will be transmitted; the "sto" flag will be reset |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 20H | SLA+W has been transmitted; "not ACK" has been received | Load data byte | 0 | 0 | 0 | X | Data byte will be transmitted; ACK will be received |
| | | or no action | 1 | 0 | 0 | X | Repeated START will be transmitted |
| | | or no action | 0 | 1 | 0 | X | STOP condition will be transmitted; the "sto" flag will be reset |
| | | or no action | 1 | 1 | 0 | X | STOP condition followed by a START condition will be transmitted; the "sto" flag will be reset |
| 28H | Data byte in i2cdat has been transmitted; ACK has been received | Load data byte | 0 | 0 | 0 | X | Data byte will be transmitted; ACK bit will be received |
| | | or no action | 1 | 0 | 0 | X | Repeated START will be transmitted |
| | | or no action | 0 | 1 | 0 | X | STOP condition will be transmitted; the "sto" flag will be reset |
| | | or no action | 1 | 1 | 0 | X | STOP condition followed by a START condition will be transmitted; sto flag will be reset |
| 30H | Data byte in i2cdat has been transmitted | Data byte | 0 | 0 | 0 | X | Data byte will be transmitted; ACK will be received |
| | | or no action | 1 | 0 | 0 | X | Repeated START will be transmitted; |
| | | or no action | 0 | 1 | 0 | X | STOP condition will be transmitted; sto flag will be reset |
| | | or no action | 1 | 1 | 0 | X | STOP condition followed by a START condition will be transmitted; sto flag will be reset |
| 38H | Arbitration lost in SLA+R/W or data bytes | No action | 0 | 0 | 0 | X | I2C bus will be released; the "not addressed slave" state will be entered |
| | | or no action | 1 | 0 | 0 | X | A START condition will be transmitted when the bus becomes free |

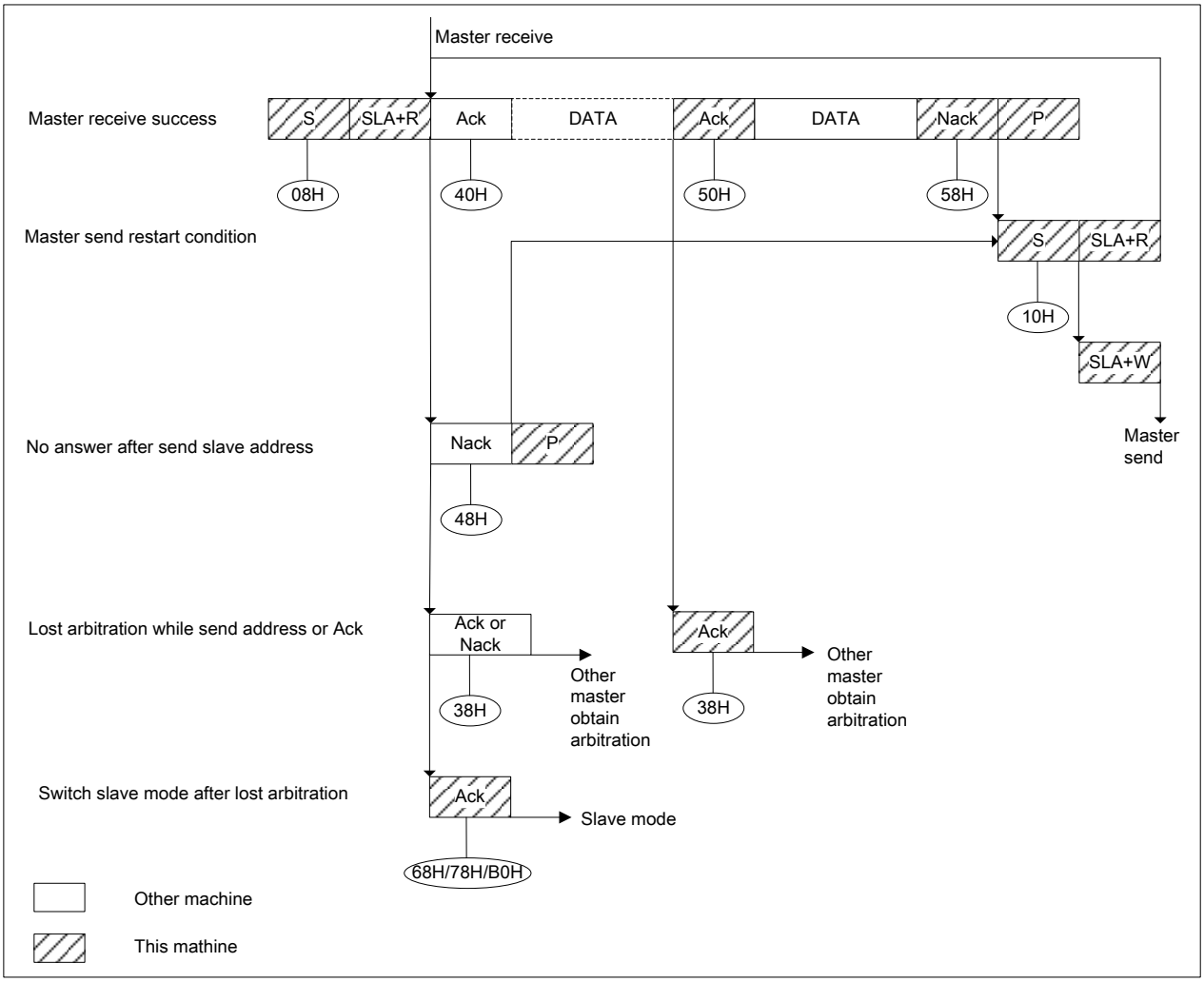


**I2C master receive mode:**

In master receive mode, master receive data from slave: start condition(S) ,follow a slave address(SLA)+ read control Bit(R), indicate the mode of access to master receive mode.

**I2C Status In Master Receiver Mode**

| Status code | Status of the I2C | Application software response to/from I2CDAT | to i2CCON sta | sto | si | aa | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| 08H | START condition has been transmitted | Load SLA+R | X | 0 | 0 | X | SLA+R will be transmitted; ACK will be received |
| 10H | Repeated START condition has been transmitted | Load SLA+R | X | 0 | 0 | X | As above |
| | | or | | | | | |
| | | Load SLA+W | X | 0 | 0 | X | SLA+W will be transmitted; I2C will be switched to "master transmitter" mode |
| 38H | Arbitration lost in "not ACK" bit | No action | 0 | 0 | 0 | X | I2C bus will be released; I2C will enter a "slave" mode |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | X | A start condition will be transmitted when the bus becomes free |
| 40H | SLA+R has been transmitted; ACK has been received | No action | 0 | 0 | 0 | 0 | Data byte will be received; not ACK will be returned |
| | | or | | | | | |
| | | no action | 0 | 0 | 0 | 1 | Data byte will be received; ACK will be returned |
| 48H | SLA+R has been transmitted; "not ACK" has been received | No action | 1 | 0 | 0 | X | Repeated START condition will be transmitted |
| | | or no action | 0 | 1 | 0 | X | STOP condition will be transmitted; the "sto" flag will be reset |
| | | or no action | 1 | 1 | 0 | X | STOP condition followed by START condition will be transmitted; the "sto" flag will be reset |
| 50H | Data byte has been received; ACK has been returned | Read data byte | 0 | 0 | 0 | 0 | Data byte will be received; "not ACK" will be returned |
| | | or | | | | | |
| | | read data byte | 0 | 0 | 0 | 1 | Data byte will be received; ACK will be returned |
| 58H | Data byte has been received; "not ACK" has been returned | Read data byte | 1 | 0 | 0 | X | Repeated START condition will be transmitted |
| | | or | | | | | |
| | | read data byte | 0 | 1 | 0 | X | STOP condition will be transmitted; the "sto" flag will be reset |
| | | or | | | | | |
| | | read data byte | 1 | 1 | 0 | X | STOP condition followed by START condition will be transmitted; the "sto" flag will be reset |




**I2C slave receive mode:**

In slave mode, a series of data is received from the host from the computer.

Before entering the slave mode, you need to set the slave address, where I2CADR[7..1]Bit is the slave address, and I2CADR is the slave address. If I2CADR[0] sets the Bit, the slave will also respond to the broadcast call address (00H); otherwise, the broadcast call address will not be answered.

In slave mode, the I2C module waits for the address of the bus to the local address or the broadcast call address (if the I2CADR[0] is set Bit). If the read and write data Bit is' write ', the I2C enters the slave mode of reception, otherwise it will enter the slave mode.

After the address and read-write data Bit is accepted, the interrupt flag (SI) is set to Bit, and the status register I2CSTA is written to the current state.

I2C Status In Slave Receiver Mode

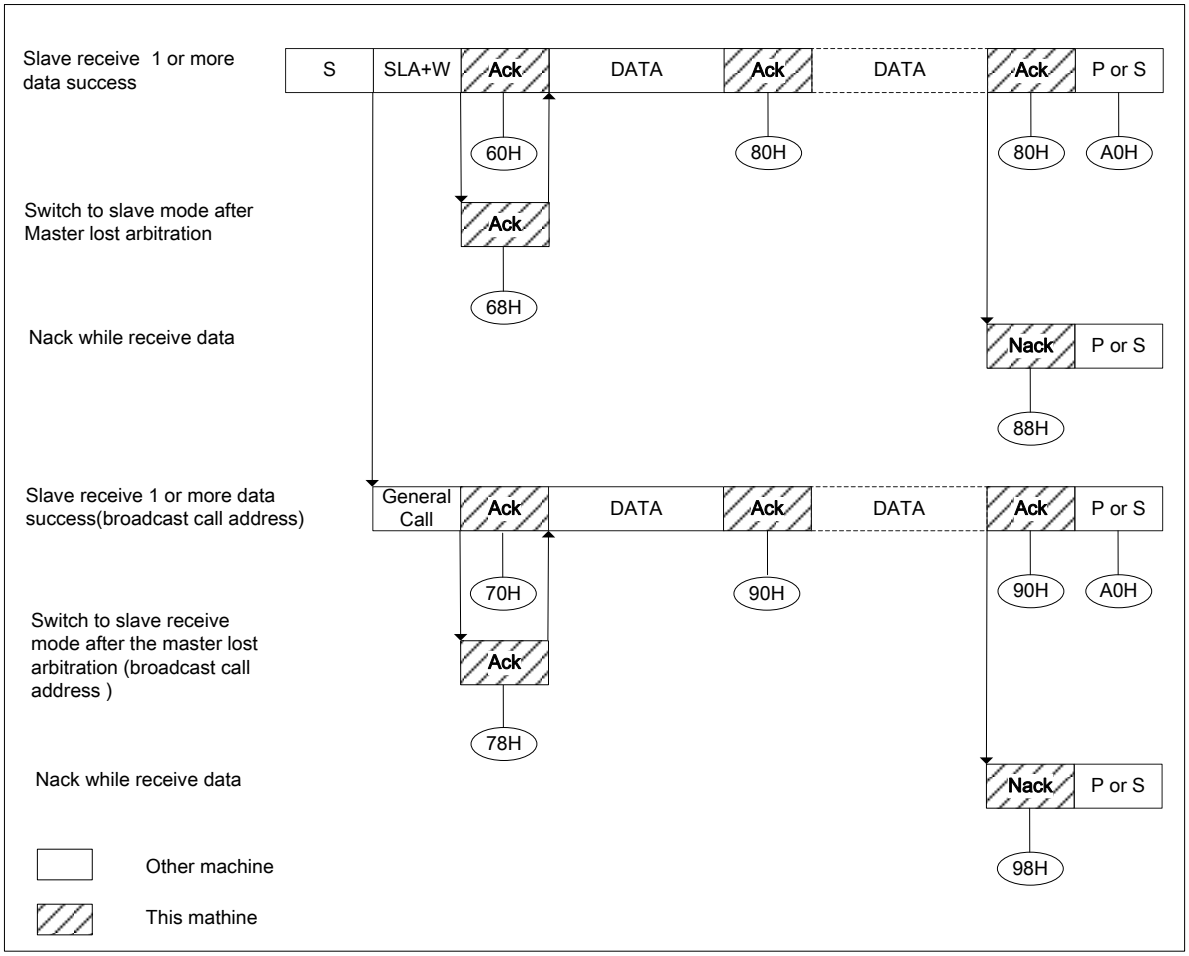
| Status code | Status of the I2C | Application software response to/from I2CDAT | to i2CCON | | | | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| | | | sta | sto | si | aa | |
| 60H | Own SLA+W has been received; ACK has been returned | No action | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | or no action | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |
| 68H | Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned | No action | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | or no action | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |
| 70H | General call address (00H) has been received; ACK has been returned | No action | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | or no action | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |
| 78H | Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned | No action | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | or no action | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |
| 80H | Previously addressed with own SLV address; DATA has been received; ACK returned | Read data byte or read data byte | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |

I2C Status In Slave Receiver Mode

| Status code | Status of the I2C | Application software response to/from I2CDAT | Application software response to i2CCON sta | sto | si | aa | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| 88H | Previously addressed with own SLA; DATA byte has been received; "not ACK" returned | Read data byte | 0 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address |
| | | or | | | | | |
| | | read data byte | 0 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized |
| | | or | | | | | |
| | | read data byte | 1 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free |
| | | or | | | | | |
| | | read data byte | 1 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free |
| 90H | Previously addressed with general call address; DATA has been received; ACK returned | Read data byte | X | 0 | 0 | 0 | Data byte will be received and "not ACK" will be returned |
| | | or | | | | | |
| | | read data byte | X | 0 | 0 | 1 | Data byte will be received and ACK will be returned |

I2C Status In Slave Receiver Mode

| Status code | Status of the I2C | Application software response | | | | | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| | | to/from I2CDAT | to i2CCON | | | | |
| | | | sta | sto | si | aa | |
| 98H | Previously addressed with general call address; DATA has been received; ACK returned | Read data byte | 0 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address |
| | | or | | | | | |
| | | read data byte | 0 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized |
| | | or | | | | | |
| | | read data byte | 1 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free |
| | | or | | | | | |
| | | read data byte | 1 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free |
| A0H | STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX | No action | 0 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address |
| | | or | | | | | |
| | | no action | 0 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free |




**I2C slave send mode:**

In slave mode, a series of data is sent from the machine to the host.

Before entering the slave mode, you need to set the slave address, where I2CADR[7..1]Bit is the slave address, and I2CADR is the slave address. If I2CADR[0] sets the Bit, the slave will also respond to the broadcast call address (00H); otherwise, the broadcast call address will not be answered.

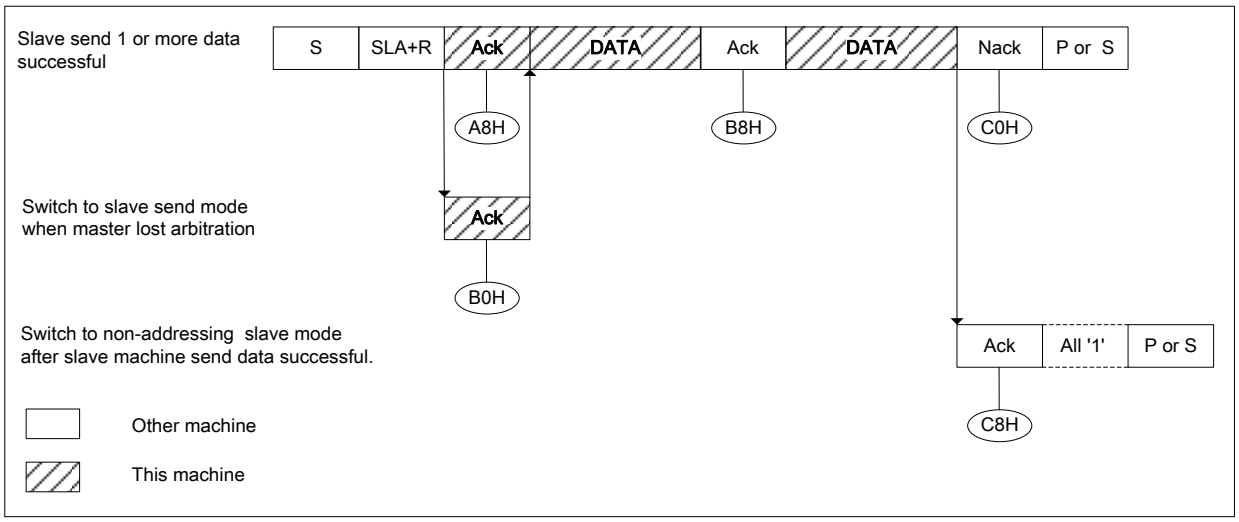
In slave mode, the I2C module waits for the address of the bus to the local address or the broadcast call address (if the I2CADR[0] is set Bit). If the read and write data Bit is' write ', the I2C enters the slave mode of reception, otherwise it will enter the slave mode.

After the address and read-write data Bit is accepted, the interrupt flag (SI) is set to Bit, and the status register I2CSTA is written to the current state.

I2C Status In Slave Transmitter Mode

| Status code | Status of the I2C | Application software response to/from I2CDAT | to i2CCON sta | sto | si | aa | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| A8H | Own SLA+R has been received; ACK has been returned | Load data byte | X | 0 | 0 | 0 | Last data byte will be transmitted and ACK will be received |
| | | or | | | | | |
| | | load data byte | X | 0 | 0 | 1 | Data byte will be transmitted; ACK will be received |
| 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 | Last data byte will be transmitted and ACK will be received |
| | | or | | | | | |
| | | load data byte | X | 0 | 0 | 1 | Data byte will be transmitted; ACK will be received |
| B8H | Data byte has been transmitted; ACK has been received | Load data byte | X | 0 | 0 | 0 | Last data byte will be transmitted and ACK will be received |
| | | or | | | | | |
| | | load data byte | X | 0 | 0 | 1 | Data byte will be transmitted; ACK will be received |
| C0H | Data byte has been transmitted; not ACK has been received | No action | 0 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address |
| | | or | | | | | |
| | | no action | 0 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free |

| Status code | Status of the I2C | Application software response | | | | | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| | | to/from I2CDAT | to i2CCON | | | | |
| | | | sta | sto | si | aa | |
| C8H | Last data byte has been transmitted; ACK has been received | No action | 0 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address |
| | | or | | | | | |
| | | no action | 0 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 0 | Switched to "not addressed slave" mode; no recognition of own slave address or general call address; START condition will be transmitted when the bus becomes free |
| | | or | | | | | |
| | | no action | 1 | 0 | 0 | 1 | Switched to "not addressed slave" mode; own slave address or general call address will be recognized; START condition will be transmitted when the bus becomes free |



**I2C composite state：**

I2C Status - miscellaneous states

| Status code | Status of the I2C | Application software response to/from I2CDAT | to i2CCON | | | | Next action taken by the I2C hardware |
|---|---|---|---|---|---|---|---|
| | | | sta | sto | si | aa | |
| F8H | No relevant state information available; si=0 | No action | No action | | | | Wait or proceed current transfer |
| 00H | Bus error during MST or selected slave modes | No action | 0 | 1 | 0 | X | Only the internal hardware is affected in the "master" or "addressed slave" modes. In all cases, the bus is released and I2C is switched to the "not addressed slave" mode. The "sto" flag is reset. |

# 15.4. Special function register list.

Micro-control is in conjunction with interface of I2C component by following 4 special function register:

| **I2C module register base address：0x4000A000** | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/name** | **Reset value** | **Function description** |
| 00H | I2CDAT | R/W | 0000H | I2C data register |
| 04H | I2CADR | R/W | 0000H | I2C address register |
| 08H | I2CCON | R/W | 4000H | I2C control register |
| 0CH | I2CSTA | R/W | 00F8H | I2C state register |

## 15.5. Special function register introduction

| I2CDAT （I2C data register） | | | Base address： 0x4000A000 Offset address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | I2CDAT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Register I2CDAT is data to send to bus or data received from bus.there is no shadow register or double buffer for I2CDAT there for MCU need to read data from it in time when I2C interrupt generated in case of data loss.

| I2CADR （address register） | | | Base address： 0x4000A000 Offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | I2CADR[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| I2CADR[7:1] | **I2C slave address(7 bits)** |
| I2CADR[0] | **Call address confirm bit** Confirm address can be recognized if this bit is 1 otherwise not |

| I2CCON （control register） | | | Base address： 0x4000A000 Offset address： 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | CR[9:3] | | | | | | |
| **Reset:** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | CR2 | ENS1 | STA | STO | SI | AA | CR[1:0] | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CR[9:0] | **I2C clock frequency control bits**<br>I2C CLOCK=fsys/( CR[9:0]+1)/4 |
| ENS1 | **I2C enable bit**<br>1：enable IIC；<br>0：disable IIC； |
| STA | **Start flag**<br>1：check the state of IIC bus,if it is free and start signals will be generated;<br>0： generate no start signal; |
| STO | **Stop flag**<br>1：send stop signal to bus when in master mode<br>0： send no stop signal to bus； |
| SI | **Interrupt flag bit**<br>SI will be set by hardware when enter one of 25 IIC states except for “F8H”;cleared by write 0,invalid to write 1 |
| AA | **Generate answer flag**<br>1: return answer in following situations: receive address when functioning as slave; receive address call if gc is set; a byte is received in master receive mode; a byte is received in slave receive mode<br>0: return non-answer in following situations: a byte is received in master receive mode;a byte is received in slave receive mode |

| **I2CSTA（state register）** | | | **Base address：** **0x4000A000**<br>**Offset address：** **0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | I2CSTA[4:0] | | | | | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| I2CSTA[4:0] | **I2C state code** |

Register “i2xsta” indicate the real-time state of I2C. three low bits of this register are always 0.and there are 26 possible states in total.And interrupt will be generated if entering any state of them except for F8h.

‘SLA” means slave address,”R” means that bit send with slave address is write bit and ‘W’ means write bit.

# 16. RTC module

## 16.1. General introduction

RTC unit provide function of real-time clock、calender and can auto-adjust according to leap year,support alarm clock and periodic interrupt.

RTC will not be shut down in any mode and can work properly in low-power mode.

RTC output register and clock adjust register will not be reset to ensure the accuracy.

The minimum operating voltage can reach 1V for uncompensated RTC,minimum operating voltage of compensated output second clock of RTC is the set threshold of BOR(in register **VDETCFG**)

## 16.2. Function introduction

- Provide function of clock and calendar:output register contains second,minute,hour,day,month,year,week
- Auto-adjust function according to leap year and leap month
- 1 alarm clock interrupt
- 2 timer periodic interrupt
- 5 time interrupt (second,minute,hour,day,month)
- Can output square wave of 1/2/8/16/32/64/128/524288Hz
- Can output compensated calibration impulse of every second
- Add read-only register to restore times of adjusting time and time the last time to adjust

## 16.3. Clock adjust

RTC read the output temperature of TPS and calculate real-time frequency deviation according to OSC temperature features and send it to frequency division module for clock adjusting.there is a internal polynomial compensate curve whose coefficient is modifiable.Calculation formula of DFi:

DFi=(DFA+DFB*(TMPDAT- Toff)+DFC*(TMPDAT- Toff)^2+ DFD*(TMPDAT- Toff)^3+DFE*(TMPDAT- Toff)^4）>>2

Where DFA/ DFB/ DFC/ DFD/ DFE is compensation coefficient of 0 to 4 power term,TMPDAT is output of temperature sensor,Toff is temperature sensor bias correction coefficient.

Every LSB of DFi stands for 0.06ppm(1/512Hz for OSC;1/32Hz for Mems)

## 16.4. Correspondence between RTC compensation coefficient register and Info Flash

RTC compensation coefficient register will be loaded automatically after power on,given reliability of system,users can read corresponding address of Information Block through software,then write it to corresponding RTC register.

| Offset address of register | Register name | Information Block corresponding offset address （Information Block base address：0x00040000） |
|---|---|---|
| 0x50 | DFAH | 0x104 |
| 0x54 | DFAL | 0x108 |
| 0x58 | DFBH | 0x10C |
| 0x5C | DFBL | 0x110 |
| 0x60 | DFCH | 0x114 |
| 0x64 | DFCL | 0x118 |
| 0x68 | DFDH | 0x11C |
| 0x6C | DFDL | 0x120 |
| 0x70 | DFEH | 0x124 |
| 0x74 | DFEL | 0x128 |
| 0x78 | Toff | 0x12C |
| 0x7C | MCON01 | 0x130 |
| 0x80 | MCON23 | 0x134 |
| 0x84 | MCON45 | 0x138 |
| ---- | ---- | 0x13c |

Note:

1.DataInfo-Flash will be loaded to corresponding register only when power on if bit2 of FC0 address of Flash is 1.

2.0x13c store the sum of unsigned num of 0x104 to 0x138, data in Info is valid if the sum is correct, otherwise not.

3.MCONxx register is an internal control register and need to load after power on

## 16.5. Time and perpetual calendar

RTC provide second、minute、hour、day、month、year and week output register.

Leap-year auto-adjust Perpetual calendar function can be obtained from RC output register,from 2001.1.1 to 2099.12.31

# 16.6. Interrupt function

RTC provide 8 kinds of interrupt source, share IRQ-RTC interrupt vector 12 of MUC.configurate RTCIE(AAH) to enable 8 kinds of interrupt source of RTC.

Shown below for detailed interrupt generating condition and steps of clearing interrupt:

**ALMF**：Alarm clock interrupt flag

If hour and minute matches the set value,alarm clock interrupt will be generated and set ALMF to 1

Clear flag by write 0 to this bit

**RTC1F**：RTC timer1 interrupt flag

This flag is set to 1 (X+1)*1S after RTC1E count is enabled if set RTC1CNT=X.

Clear flag by write 0 to this bit

**RTC2F**：RTC timer2 interrupt flag

This flag is set to 1 (X+1)*0.0625S after RTC2E count is enabled if set RTC2CNT=X.

Clear flag by write 0 to this bit

**MTHF**：Month interrupt

A month interrupt is generated and MTHF is set to 1 if month counter MTHR is bumped up by 1

Clear flag by write 0 to this bit

**DAYF**：Day interrupt

A day interrupt is generated and DAYF is set to 1 if day counter DAYR is bumped up by 1

Clear flag by write 0 to this bit

**HRF**：Hour interrupt

An hour interrupt is generated and HRF is set to 1 if hour counter HRR is bumped up by 1

Clear flag by write 0 to this bit

**MINF**：Minute interrupt

A minute interrupt is generated and MINF is set to 1 if month counter MINR is bumped up by 1

Clear flag by write 0 to this bit

**SECF**：Second interrupt

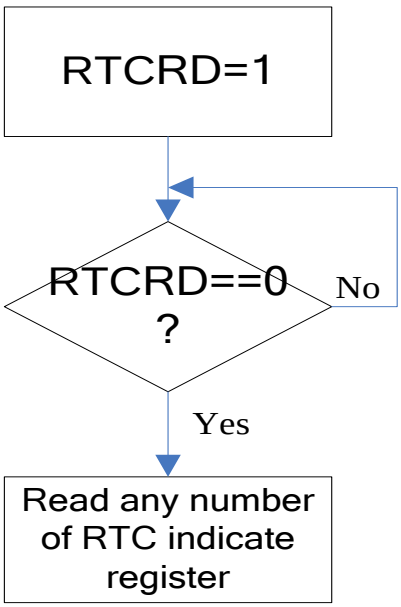
A second interrupt is generated and SECF is set to 1 if month counter SECR is bumped up by 1

Clear flag by write 0 to this bit

# 16.7. RTC indicate register read-write process

## 16.7.1. Process of read RTC indicate register

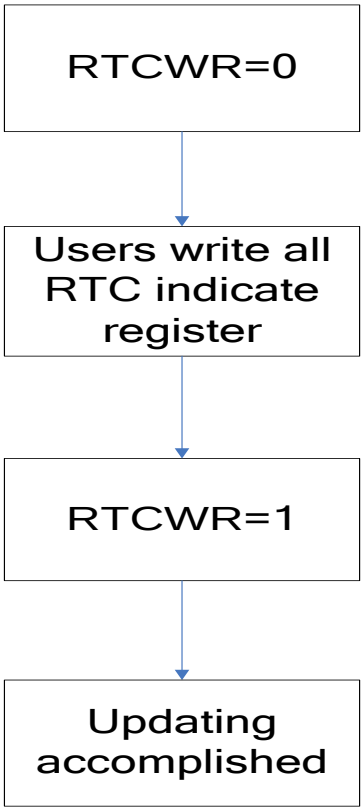
Process as below of reading RTC indicate register （SECR，MINR，HRR，DAYR，MTHR，YRR，DOWR） should be followed when necessary.




Note: process of reading auxiliary RTC is same with above except that read control register is RTCRD2

## 16.7.2. Process of write RTC indicate register

Process as below of updating RTC indicate register（SECR，MINR，HRR，DAYR，MTHR，YRR，DOWR）should be followed when necessary and in the order of year、month、day、hour、minute、second、week：




RTC indicated register write flow

Note: process of writing auxiliary RTC is same with above except that read control register is RTCWR2

## 16.8. Record of adjusting clock

Adjusting clock times register is Incremented by 1 when write to any register of YRR、MTHR、DAYR、HRR、MINR、SECR、DOWR

Every time UpdatFlag is set to 1 even if many registers are updated at one time while UpdatFlag is set to 1 for only one time,adjust times RTCCOUNT is incremented by 1.

Set UpdatFlag to 1 and RTCCNT will be incremented by 1 if writing operation failed because the writing value is out of range

## 16.9. Auxiliary RTC

There is an internal auxiliary RTC module,driven by LRC, for system to offer second set of year/month/day/hour/minute/second/week register.system can switch to internal low frequency RC to get timekeeping time if external OSC oscillator stops. Process of writing/reading auxiliary RTC is same with first set of RTC except that it has independent RTCWR2 and RTCRD2

## 16.10. Special function register list

| **RTC module register base address：0x4000C000** | | | | |
|---|---|---|---|---|
| **Offset address** | **Name** | **Write/read** | **Reset value** | **Function description** |
| 0x00 | RTCCON | R/W | 0x0000 | RTC control register |
| 0x04 | RTCIE | R/W | 0x0000 | RTC interrupt enable register |
| 0x08 | RTCIF | R/W | 0x0000 | RTC interrupt flag register |
| 0x0C | ALMR | R/W | 0x0000 | Alarm clock register |
| 0x10 | RTCTMR1 | R/W | 0x0000 | RTC timer1 count configuration |
| 0x14 | RTCTMR2 | R/W | 0x0000 | RTC timer2 count configuration |
| 0x18 | SECR | R/W | 0x0000 | Second register(write protect) |
| 0x1C | MINR | R/W | 0x0000 | Minute register(write protect) |
| 0x20 | HOURR | R/W | 0x0000 | Hour register(write protect)） |
| 0x24 | DAYR | R/W | **0x0001** | Day register(write protect) |
| 0x28 | MONTHR | R/W | **0x0001** | Month register(write protect) |
| 0x2C | YEARR | R/W | **0x0000** | Year register(write protect) |
| 0x30 | WEEKR | R/W | **0x0001** | Week register(write protect) |
| 0x34 | RTCCNTH | R/W | 0x0000 | high 16-bit of Adjust clock times register |

| | | | | |
|---|---|---|---|---|
| 0x38 | RTCCNTL | R/W | 0x0000 | Low 16-bit of Adjust clock times register |
| 0x3C | RTCRd | R/W | 0x0000 | RTC read control register |
| 0x40 | RTCWr | R/W | 0x0000 | RTC write control register |
| 0x50 | DFAH | R/W | 0x0000 | Low bits of constant term coefficient of RTC hardware compensate |
| 0x54 | DFAL | R/W | 0x0000 | high bits of constant term coefficient of RTC hardware compensate |
| 0x58 | DFBH | R/W | 0x0000 | high bits of monomial coefficient of RTC hardware compensate |
| 0x5C | DFBL | R/W | 0x0000 | low bits of monomial coefficient of RTC hardware compensate |
| 0x60 | DFCH | R/W | 0x0000 | high bits of quadratic term coefficient of RTC hardware compensate |
| 0x64 | DFCL | R/W | 0x0000 | Low bits of quadratic term coefficient of RTC hardware compensate |
| 0x68 | DFDH | R/W | 0x0000 | High bits of cubic term coefficient of RTC hardware compensate |
| 0x6C | DFDL | R/W | 0x0000 | Low bits of cubic term coefficient of RTC hardware compensate |
| 0x70 | DFEH | R/W | 0x0000 | High bits of quartic term coefficient of RTC hardware compensate RTC |
| 0x74 | DFEL | R/W | 0x0000 | Low bits of quartic term coefficient of RTC hardware compensate |
| 0x78 | Toff | R/W | 0x0000 | Temperature bias register |
| 0x7C | MCON01 | R/W | **0x2010** | Mems control register |
| 0x80 | MCON23 | R/W | **0x2088** | Mems control register |
| 0x84 | MCON45 | R/W | **0x0490** | Mems control register |
| 0x88 | DFiH | R/W | 0x0000 | High 5-bit of frequency error register |
| 0x8C | DFiL | R/W | 0x0000 | Low 16-bit of frequency error register |
| 0x90 | RTCRSTFLG | R/W | ------ | RTC module reset flag register |
| 0x100 | CTRLByFlash | R/*W | ------ | |
| 0x104 | LRCCOMAND | W | 0 | LRC write shut-down instruct register |

| **RTC module register base address：0x4000C000** | | | | |
|---|---|---|---|---|
| **Offset address** | **Name** | **Write/read** | **Reset value** | **Function description** |
| 0x200 | SECR2 | R/W | 0x0000 | Auxiliary second register,write protect |
| 0x204 | MINR2 | R/W | 0x0000 | Auxiliary minute register,write protect |
| 0x208 | HOURR2 | R/W | 0x0000 | Auxiliary hour register,write protect |
| 0x20C | DAYR2 | R/W | 0x0001 | Auxiliary day register,write protect |
| 0x210 | MONTHR2 | R/W | 0x0001 | Auxiliary month register,write protect |

| | | | | |
|---|---|---|---|---|
| 0x214 | YEARR2 | R/W | 0x0000 | Auxiliary year register,write protect |
| 0x218 | WEEKR2 | R/W | 0x0001 | Auxiliary week register,write protect |
| 0x21C | RTC2CAL | R/W | 0x0000 | Auxiliary RTC2 correct register |
| 0x220 | RTCRD2 | R/W | 0x0000 | Auxiliary RTC read control register |
| 0x224 | RTCWR2 | R/W | 0x7FFF | Auxiliary RTC write control register |
| 0x228 | FRE_LRC | R/W | 0x0000 | Auxiliary LRC frequency measurement register |

## 16.11. Special function register introduction

| **Correspondence between auto-load RTC register and InfoFlash address** | | |
|---|---|---|
| **Offset address** | **name** | **Function description** |
| 0x50 | DFAH | **0x104** |
| 0x54 | DFAL | **0x108** |
| 0x58 | DFBH | **0x10C** |
| 0x5C | DFBL | **0x110** |
| 0x60 | DFCH | **0x114** |
| 0x64 | DFCL | **0x118** |
| 0x68 | DFDH | **0x11C** |
| 0x6C | DFDL | **0x120** |
| 0x70 | DFEH | **0x124** |
| 0x74 | DFEL | **0x128** |
| 0x78 | Toff | **0x12C** |
| 0x7C | MCON01 | **0x130** |
| 0x80 | MCON23 | **0x134** |
| 0x84 | MCON45 | **0x138** |

Note: auto-load is accomplished before digital circuit reset.

Note: it auto load 32 bits pre time

## 16.12. Special function register introduction

| **RTCCON**<br>**（RTC control register）** | | | **Base address:** **0x4000C000**<br>**Offset address:** **00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |

| Read: / Write: | X | RTC2EN | RTC1EN | X | TOUT[2:0] | | | AutoC |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function |
|---|---|
| RTC2EN | **RTC timer2 enable bit**<br>RTC2EN=0：disable RTC timer2<br>RTC2EN=1：enable RTC timer2,RTC2IF will be set if overflow |
| RTC1EN | **RTC timer1 enable bit**<br>RTC1EN=0：disable RTC timer1<br>RTC1EN=1：enable RTC timer1,RTC1IF will be set if overflow |
| TOUT[3:0] | **TOUT output frequency introduction shown as blow** |
| AutoC | **Auto compensation control bit**<br>0：enable auto compensation,invalid to write to DFiH/DFiL<br>1：manual compensation,compensation value(frequency error) is the value write to DFiH/DFiL by users |

| TOUT[2:0] | | | TOUT | TOUT（PLL enable） |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 524288/32768Hz | 524288/32768Hz |
| 0 | 1 | 1 | Internal RTC 1Hz | Obtained from compensating 128Hz by high frequency |
| 1 | 0 | 0 | Internal RTC 2Hz | Obtained from compensating 128Hz by high frequency |
| 1 | 0 | 1 | Internal RTC 4Hz | Obtained from compensating 128Hz by high frequency |
| 1 | 1 | 0 | Internal RTC 8Hz | Obtained from compensating 128Hz by high frequency |
| 1 | 1 | 1 | Internal RTC 1Hz | Internal RTC 1Hz |

| **RTCIE (RTC interrupt enable register)** | | | **Base address： 0x4000C000 Offset address： 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | ALMIE | RTC2IE | RTC1IE | MTHIE | DAYIE | HRIE | MINIE | SECIE |

| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|

| Bits | Function description |
|---|---|
| ALMIE | **alarm clock interrupt enable bit**<br>0：disable<br>1：enable |
| RTC2IE | **RTC timer2 interrupt enable bit**<br>0：disable<br>1：enable |
| RTC1IE | **RTC timer1 interrupt enable bit**<br>0：disable<br>1：enable |
| MTHIE | **RTC month interrupt enable bit**<br>0：disable<br>1：enable |
| DAYIE | **RTC day interrupt enable bit**<br>0：disable<br>1：enable |
| HRIE | **RTC hour interrupt enable bit**<br>0：disable<br>1：enable |
| MINIE | **RTC minute interrupt enable bit**<br>0：disable<br>1：enable |
| SECIE | **RTC second interrupt enable bit**<br>0：disable<br>1：enable |

| **RTCIF**<br>**（RTC interrupt flag register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | ALMIF | RTC2IF | RTC1IF | MTHIF | DAYIF | HRIF | MINIF | SECIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| ALMIF | **Alarm clock interrupt flag** |

| | |
|---|---|
| | 0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| RTC2IF | **RTC timer2 interrupt flag**<br>0：no interrupt generated<br>interrupt generated.cleared by writing 0 |
| RTC1IF | **RTC timer1 interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| MTHIF | **RTC month interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| DAYIF | **RTC day interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| HRIF | **RTC hour interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| MINIF | **RTC minute interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |
| SECIF | **RTC second interrupt flag**<br>0：no interrupt generated<br>1：interrupt generated.cleared by writing 0 |

| **ALMR**<br>**（alarm clock register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | ALMH[5:0] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | ALMM[5:0] | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **bits** | **Function description** |
|---|---|
| ALMH[4:0] | Alarm clock interrupt hour set,number beyond 0~23 is allowed to write but no alarm clock will generate |
| ALMM[5:0] | Alarm clock interrupt minute set,number beyond 0~59 is allowed to write but no alarm clock will generate<br>Note:interrupt generate only if hour and minute register matches with the set value in |

| | |
|---|---|
| | register(and second register is 0) |

| RTCTMR1 （RTC timer1 register） | | | Base address： 0x4000C000 Offset address： 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | CNT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | CNT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CNT[15:0] | Minimum resolution is 1s,namely it can generate interrupt 1 times every second,and 65536s at most.RTC1IF is set if counter overflows<br>CNT[15：0] stands for a unsigned 16-bit binary,if set SCT[15:0]=00H,RTC1IF will be set once every (00H+1)*1S =1*1S=1S counting period of RTC internal second function interrupt<br>Note:timer start counting from 0 if timer overflows and users do not shut down it. |

| RTCTMR2 （RTC timer2 register） | | | Base address： 0x4000C000 Offset address： 14H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | CNT[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | CNT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| CNT[15:0] | Minimum resolution is 0.0625s,namely it can generate interrupt one times every 0.0625s,and 4096s at most.RTC2IF is set if counter overflows<br>CNT[15：0] stands for a unsigned 16-bit binary,if set SCT[15:0]=00H,RTC2IF will be set once every (00H+1)*0.0625S =1*0.0625S=0.0625S counting period of RTC internal second function interrupt<br>Note:timer start counting from 0 if timer overflows and users do not shut down it |

| SECR（RTC second register） | | | Base address： 0x4000C000<br>Offset address： 18H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | SEC[5:0] | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| SEC[5:0] | **Second counter：**<br>Valid range:0-59.it is invalid to write any num beyond 0-59. |

| MINR（RTC minute register） | | | Base address： 0x4000C000<br>Offset address： 1CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | MIN[5:0] | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| MIN[5:0] | **Minute counter：**<br>Valid range:0-59.it is invalid to write any num beyond 0-59. |

| HOURR（RTC hour register） | | | Base address： 0x4000C000<br>Offset address： 20H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | HOUR[4:0] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| HOUR[4:0] | **Hour counter：**<br>Valid range:0-23.it is invalid to write any num beyond 0-23. |

| **DAYR**<br>**（RTC day register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **24H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | DAY[4:0] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| DAY[4:0] | **Day counter：**<br>Valid range：1-28/29/30/31.any value mismatches year and month is invalid |

| **MONTHR**<br>**（RTC month register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **28H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | MONTH[3:0] | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| MONTH[3:0] | **Month counter：**<br>Valid range: 1-12. It is invalid to write any num beyond 1-12 |

| **YEARR**<br>**（RTC year register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **2CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | X | YEAR[6:0] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| YEAR[6:0] | **Year counter:**<br>Valid range: 0-99. It is invalid to write any num beyond 0-99 |

| **WEEKR**<br>**（RTC week register）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **30H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | X | X | X | X | X | WEEK[2:0] | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| WEEK[2:0] | **Week counter：**<br>Valid range:1-7. It is invalid to write any num beyond 1-7 |

| **RTCCNTH**<br>**（high 16-bit of RTC correct clock times register）** | | | | **Base address：** **0x4000C000**<br>**Offset address：** **34H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** / **Write:** | RTCCNTH[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **RTCCNTL**<br>**（low 16-bit of RTC correct clock times register）** | | | | **Base address：** **0x4000C000**<br>**Offset address：** **38H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** / **Write:** | RTCCNTL[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| RTCCNTH[15:0]<br>RTCCNTL[15:0] | RTCCNTH and RTCCNTL make the 32-bit register for recording correct clock times.read only |

| RTCRD （RTC rad control register） | | | Base address： 0x4000C000<br>Offset address： 3CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | READFLAG |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| READFLAG | RTC indicate register read control bit,see 16.7.1 read/write process of RTC indicate register for detailed direction |

| RTCWR （RTC write control register） | | | Base address: 0x4000C000<br>Offset address： 40H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | UPDATE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| UPDATE | RTC indicate register write control bit,see 16.7.2 read/write process of RTC indicate register for detailed direction |

| DFAH （high 7-bit of constant term） | | | Base address： 0x4000C000<br>Offset address： 50H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Read: | X | DFx[22:16] | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| DFAL（low 16-bit of constant term） | | | Base address： 0x4000C000<br>Offset address： 54H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | DFx[15:8] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | DFx[7:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DFx[22:0] | DFAH/DFAL make the 23-bit signed number,it is constant term for frequency error calculation |

| DFBH（high 7-bit of monomial term） | | | Base address： 0x4000C000<br>Offset address： 58H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | X | X | X | X | X | X | X | X |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | X | DFx[22:16] | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| DFBL（low 16-bit of monomial term） | | | Base address： 0x4000C000<br>Offset address： 5CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | DFx[15:8] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | DFx[7:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DFx[22:0] | DFAH/DFAL make the 23-bit signed number,it is monomial term for frequency error calculation |

| DFCH （high 7-bit ofquadratic term） | | | Base address： 0x4000C000 Offset address： 60H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | X | DFx[22:16] | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| DFCL （low 16-bit ofquadratic term） | | | Base address： 0x4000C000 Offset address： 64H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | DFx[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | DFx[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DFx[22:0] | DFAH/DFALmake the 23-bit signed number,it is quadratic term for frequency error calculation |

| DFDH （high 7-bit of cubic term） | | | Base address： 0x4000C000 Offset address： 68H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | X | DFx[22:16] | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **DFDL**<br>**（low 16-bit of cubic term）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **6CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | DFx[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | DFx[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| DFx[22:0] | DFDH/DFDL make the 23-bit signed number,it is cubic term for frequency error calculation |

| **DFEH**<br>**（high 7-bit of quartic term）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **70H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | DFx[22:16] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **DFEL**<br>**（low 16-bit of quartic term）** | | | **Base address：** **0x4000C000**<br>**Offset address：** **74H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | DFx[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | DFx[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DFx[22:0] | DFEH/DFEL make the 23-bit signed number,it is quartic term for frequency error calculation |

| Toff （temperature bias register） | | | Base address： 0x4000C000 Offset address： 78H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | DFx[15:8] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | DFx[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DFx[15:0] | 16-bit signed num |

| MCON01 （Mems control register） | | | Base address： 0x4000C000 Offset address： 7CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | X | X | REFBITS[5:0] | | | | | |
| Reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | PROBEBITS[2:0] | | COUPLEBITS[4:0] | | | | | |
| Reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

| MCON23 （Mems control register） | | | Base address： 0x4000C000 Offset address： 80H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | X | X | REFBITS[5:0] | | | | | |
| Reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | PROBEBITS[2:0] | | COUPLEBITS[4:0] | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| **MCON45 （Mems control register）** | | | **Base address: 0x4000C000 Offset address: 84H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | C_LDO_LP[2:0]（added） | | | IDIGBITS[2:0]（added） | | | ADCI_CTRL | BIASFLOAT |
| **Reset:** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | VREGBITS[2:0](1 bit more than 1202) | | | AMPBITS[4:0] | | | | |
| **Reset:** | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| **DFiH （high 7-bit of constant term）** | | | **Base address: 0x4000C000 Offset address: 88H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | DFi[20:16] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **DFiL （low 7-bit of constant term）** | | | **Base address: 0x4000C000 Offset address: 8CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | DFi[15:8] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | DFi[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| DFx[20:0] | 21-bit register,highest bit is sign bit<br>It is invalid to write to DFiH/ DFiL if AutoC is 0<br>Compensating value is write to DFiH/ DFiL register by users if AutoC is 1, manual |

| | |
|---|---|
| | compensation method.users must write high bits first and then write low bits(follow the order DFiH-DFiL). |

| **RTCRSTFlag（RTC module reset flag register）** | | | **Base address：0x4000C000**<br>**Offset address：90H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | PORRST | SoftRST |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| PORRST | PORRST reset flag：<br>0：no POR reset<br>1：POR reset（cleared by writing 0） |
| SoftRST | SoftRST reset flag：<br>0：no software reset<br>1：software reset（cleared by writing 0） |

| **CTRLByFlash** | | | **Base address：0x4000C000**<br>**Offset address：100H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | FLASH[7:0] | | | | | | | |
| **Reset:** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | x | x | x | x | Reversed | AUTORELOAD | LRC_CTRL | x |
| **Reset:** | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

| **Bits** | **Function description** |
|---|---|
| FLASH[7:0] | **Code space encryption bit**<br>Flash is not encrypted if Flash[7:0]=0xFF<br>otherwise：Flash is encrypted,and this register is read-only |
| AUTORELOAD | **Auto load enable bit**<br>=1，enable auto load<br>=0，disable auto load |

| | |
|---|---|
| LRC_CTRL | **Low frequency RC control bit**<br>=1：enable low frequency RC<br>=0：disable low frequency RC<br>**Note:users need to write 0x5555 and then 0xAAAA to LRCCOMAND register to shut down LRC if LRC_CTRL is 0** |

| **LRCCOMMAND（LRC write disable instruct register）** | | | **Base address：0x4000C000**<br>**Offset address：104H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| Bit[15:0] | users need to write 0x5555 and then 0xAAAA to LRCCOMAND register to shut down LRC if LRC_CTRL is 0 |

**Register below is auxiliary RTC clock register.the clock source of this set of clock is LRC.**

| **SECR2（ auxiliary RTC second register）** | | | **Base address：0x4000C000**<br>**Offset address：200H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | SEC[5:0] | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **bits** | **Function description** |
|---|---|
| SEC[5:0] | **Auxiliary second counter（LRC work as its clock reference）:**<br>Valid range：0-59.it is invalid to write any number beyond the range 0-59. |

| MINR2（auxiliary RTC minute register） | | | Base address: 0x4000C000<br>Offset address: 204H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | MIN[5:0] | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| MIN[5:0] | **Auxiliary minute counter:**<br>Valid range: 0-59.it is invalid to write any number beyond the range 0-59. |

| HRR2（auxiliary RTC hour register） | | | Base address: 0x4000C000<br>Offset address: 208H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | X | X | HOUR[4:0] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| HOUR[4:0] | **Auxiliary hour counter（LRC work as its clock reference）:**<br>Valid range: 0-23.it is invalid to write any number beyond the range 0-23. |

| DAYR2（auxiliary RTC day register） | | | Base address: 0x4000C000<br>Offset address: 20CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** **Write:** | X | X | X | DAY[4:0] | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAY[4:0] | **Auxiliary day counter（LRC work as its clock reference）:**<br>Valid range：1-28/29/30/31.any number mismatch year and month is invalid. |

| **MOUTHR2（ auxiliary RTC month register）** | | | **Base address： 0x4000C000**<br>**Offset address： 210H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | MONTH[3:0] | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| MONTH[3:0] | **Auxiliary month counter （LRC work as its clock reference）:**<br>Valid range：0-12.it is invalid to write any number beyond the range 0-12. |

| **YEARR2（auxiliary RTC year register）** | | | **Base address： 0x4000C000**<br>**Offset address： 214H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | YEAR[6:0] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| YEAR[6:0] | **Auxiliary year counter （LRC work as its clock reference）:**<br>Valid range：0-99.it is invalid to write any number beyond the range 0-99. |

| **WEEKR2（auxiliary RTC week register）** | **Base address： 0x4000C000**<br>**Offset address： 218H** |
|---|---|

| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
|---|---|---|---|---|---|---|---|---|
| Read: | X | X | X | X | X | X | X | X |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | X | X | X | X | X | WEEK[2:0] | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| WEEK[2:0] | **Auxiliary week counter（LRC work as its clock reference）:**<br>Valid range：0-7.it is invalid to write any number beyond the range 0-7. |

| RTC2CAL（auxiliary RTC correct register） | Base address： 0x4000C000<br>Offset address： 21CH | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit23 | 22 | 21 | 20 | 19 | 18 | 17 | Bit16 |
| Read: | X | X | X | X | X | X | X | CAL16 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | CAL15 | CAL14 | CAL13 | CAL12 | CAL11 | CAL10 | CAL9 | CAL8 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| CAL[16…0] | Auxiliary RTC correct register, to configurate clock reference of second pulse. Initial value of LRC clock is uncertain,configurate this register to determine the number of counting LRC in 1s<br>For example:<br>It will generate a second pulse every time it counts 2000 LRC if this register is set to 2000 |

| RTCRD2（auxiliary RTC read control | Base address： 0x4000C000<br>Offset address： 220H |
|---|---|

| register） | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| **Read:** **Write:** | X | X | X | X | X | X | X | READFLAG |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| READFLAG | RTC indicate register read control bit,see read/write process of RCT for detailed direction |

| RTCWR2 （auxiliary RTC write control register） | | | Base address： 0x4000C000 Offset address： 224H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | X | X | UPDATE |
| **Reset:** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bits | Function description |
|---|---|
| UPDATE | RTC indicate register write control bit,see write/read process of RTC for detailed direction |

| FRE_LRC （RTC2 correct register auto test ） | | | Base address： 0x4000C000 Offset address： 228H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:** **Write:** | X | X | X | X | X | X | X | LRC16 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | LRC15 | LRC14 | LRC13 | LRC12 | LRC11 | LRC10 | LRC9 | LRC8 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | LRC7 | LRC6 | LRC5 | LRC4 | LRC3 | LRC2 | LRC1 | LRC0 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| LRC[16…0] | LRC frequency register:<br>This register is used in concert with RTC2CAL.after enabling LRC frequency measuring function by CLKCTRL1.14,chip will employ LRC clock AS clock reference for measuring second pulse of first set of RTC and write the measuring value,the LRC clock frequency, to FRE_LRC register.it is recommended that users write it to RTC2CAL,auxiliary RTC register can synchronize with first set of RTC as much as possible in this way. |

# 17. TBS module

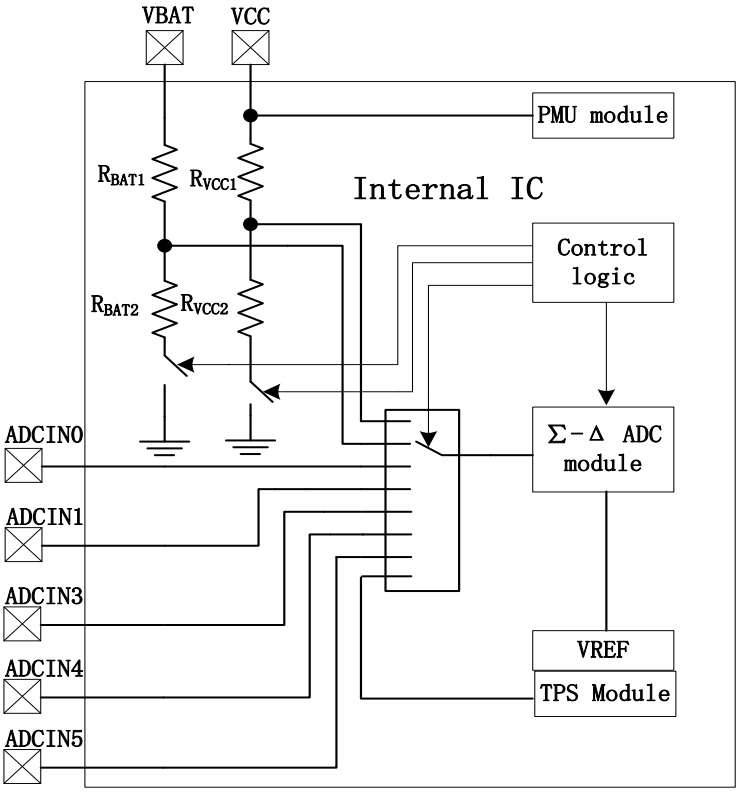
## 17.1. General introduction

Chip can quantitatively measure temperature and 5 ADC（VSYS,VBAT,ADCIN0,ADCIN1,ADCIN2） and keep the result in corresponding register.

## 17.2. Function description

TBS contains two parts of function mainly:

- Measure temperature of IC base
- Measure 5 ADC voltage（VSYS,VBAT,ADCIN0,ADCIN1,ADCIN2）




The internal resistance of VBAT was 30K (RBAT1 = 24K, RBAT2 was 6K), and the internal resistance of VDD was 42K (RVDD1 = 36K, RVDD2 = 6K). ADCIN0, ADCIN1 and ADCIN2 have no internal resistance.

## 17.3. Special function register list

| TBS module register base address：0x4000E000 | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 00H | TBSCON | R/W | 0x0101 | TBS set register |
| 04H | TBSPRD | R/W | 0x0000 | TBS enable period set register |
| 08H | TBSIE | R/W | 0x0000 | TBS interrupt enable register |
| 0CH | TBSIF | R/W | 0x0000 | TBS interrupt flag register |
| 10H | TMPDAT | R/W | 0x0000 | Temperature measure output |
| 14H | VBATDAT | R/W | 0x0000 | Battery voltage measure output |
| 18H | VCCDAT | R/W | 0x0000 | Source voltage measure output |
| 1CH | ADC0DAT | R/W | 0x0000 | ADC measuring output of channel0 |
| 20H | ADC1DAT | R/W | 0x0000 | ADC measuring output of channel1 |
| 24H | ADC2DAT | R/W | 0x0000 | ADC measuring output of channel2 |
| 28H | VBATCMP | R/W | 0x0000 | battery voltage compare register |
| 34H | ADC0CMP | R/W | 0x0000 | ADC0 compare value,16-bit signed num |
| 38H | ADC1CMP | R/W | 0x0000 | ADC1 compare value,16-bit signed num |

## 17.4. Special function register introduction

| **TBSCON**<br>**（TBS set register）** | | | **Base address：** **0x4000E000**<br>**Offset address：** **00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | x | x | x | ADC1CMPEn | ADC0CMPEn | x | Filter1 | Filter0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | x | VbatCMPEn | ADC2En | ADC1En | ADC0En | VccEn | VbatEn | TPSEn |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bits | Function description |
|---|---|
| ADC1CMPEn | This bit is valid on the condition that ADC1EN is enabled<br>ADC1CMPEn =0：disable comparison between ADC1DAT and ADC1CMP<br>ADC1CMPEn =1：enable comparison between ADC1DAT and ADC1CMP |
| ADC0CMPEn | This bit is valid on the condition that ADC0EN is enabled<br>ADC0CMPEn =0：disable comparison between ADC0DAT and ADC0CMP |

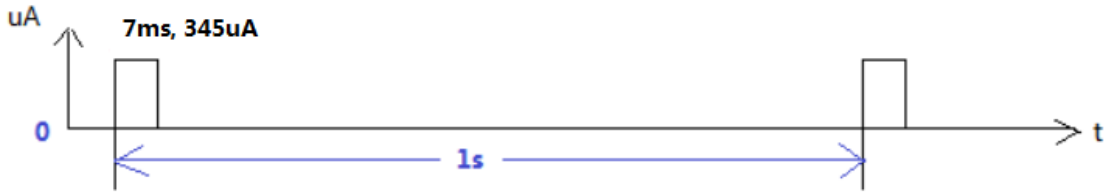
| | |
|---|---|
| | ADC0CMPEn =1：enable comparison between ADC0DAT and ADC0CMP |
| Filter[1..0] | Only measure temperature：<br>Filter [1..0] =00b：output outcome of ADC directly<br>Filter [1..0] =01b：calculate the average of quadratic term of ADC outcome<br>Filter [1..0] =10b：calculate the average of quartic term of ADC outcome<br>Filter [1..0] =11b：calculate the average of 8-order term of ADC outcome |
| VbatCMPEn | This bit is valid on the condition that VbatEN is enabled<br>VbatCMPEn=0：disable comparison between VBATDAT and VBATCMP<br>VbatCMPEn=1：enable comparison between VBATDAT and VBATMP |
| ADC2En | ADC2En =0：disable ADC channel2<br>ADC2En =1：enable ADC channel2<br>Note:it is necessary to configurate Pin of GPIO as input Pin of ADC if it is employed |
| ADC1En | ADC1En =0：disable ADC channel1<br>ADC1En =1：enable ADC channel1<br>Note:it is necessary to configurate Pin of GPIO as input Pin of ADC if it is employed |
| ADC0En | ADC0En =0：disable ADC channel0<br>ADC0En =1：enable ADC channel0<br>Note:it is necessary to configurate Pin of GPIO as input Pin of ADC if it is employed |
| VccEn | VccEn =0：disable Vcc measuring<br>VccEn =1：enable Vcc measuring |
| VbatEn | VbatEn =0：disable battery voltage measuring<br>VbatEn =1：enable battery voltage measuring |
| TPSEn | TDCEn =0：disable temperature measuring<br>TDCEn =1：enable temperature measuring |

| **TBSPRD（TBS enable period set register）** | | | **Base address：0x4000E000**<br>**Offset address：04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | ADC2PRD1 | ADC2PRD0 | ADC1PRD1 | ADC1PRD0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | ADC0PRD1 | ADC0PRD0 | VccPRD1 | VccPRD0 | VbatPRD1 | VbatPRD0 | TPSPRD1 | TPSPRD0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **bits** | **Function description** |
|---|---|
| ADC2PRD[1..0] | ADC2PRD[1..0]=00b open ADC channel2 once every 1s<br>ADC2PRD[1..0]=01b open ADC channel2 once every 4s<br>ADC2PRD[1..0]=10b open ADC channel2 once every 8s |

| | |
|---|---|
| | ADC2PRD[1..0]=11b open ADC channel2 once every 32s |
| ADC1PRD[1..0] | ADC1PRD[1..0]=00b open ADC channel1 once every 1s<br>ADC1PRD[1..0]=01b open ADC channel1 once every 4s<br>ADC1PRD[1..0]=10b open ADC channel1 once every 8s<br>ADC1PRD[1..0]=11b open ADC channel1 once every 32s |
| ADC0PRD[1..0] | ADC0PRD[1..0]=00b open ADC channel0 once every 1s<br>ADC0PRD[1..0]=01b open ADC channel0 once every 4s<br>ADC0PRD[1..0]=10b open ADC channel0 once every 8s<br>ADC0PRD[1..0]=11b open ADC channel0 once every 32s |
| VccPRD[1..0] | VccPRD [1..0]=00b open Vcc detecting once every 1s<br>VccPRD [1..0]=01b open Vcc detecting once every 4s<br>VccPRD [1..0]=10b open Vcc detecting once every 8s<br>VccPRD [1..0]=11b open Vcc detecting once every 32s |
| VbatPRD[1..0] | VbatPRD [1..0]=00b open Vbat detecting once every 1s<br>VbatPRD [1..0]=01b open Vbat detecting once every 4s<br>VbatPRD [1..0]=10b open Vbat detecting once every 8s<br>VbatPRD [1..0]=11b open Vbat detecting once every 32s |
| TPSPRD[1..0] | TPSPRD [1..0]=00b open TPS detecting once every 1s<br>TPSPRD [1..0]=01b open TPS detecting once every 8s<br>TPSPRD [1..0]=10b open TPS detecting once every 32s<br>TPSPRD [1..0]=11b open TPS detecting once every 1/8s（opening frequency is 8Hz，open 8 times every 1s） |

In normal mode, you can set TMPPRD's opening frequency at the fastest configure, in the low power mode,suggest the opening frequency 8s or 32s. For an example,the detect time and consumption as below if you configure open 1 time every 1s.




| **TBSIE (TBS interrupt enable register)** | | | **Base address: 0x4000E000**<br>**Offset address: 08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | ADC1CMPIE | ADC0CMPIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | VbatCMPIE | ADC2IE | ADC1IE | ADC0IE | VccIE | VbatIE | TPSIE |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bits | Function description |
|---|---|
| ADC1CMPIE | ADC1CMPIE =0：disable ADC1 measuring comparison interrupt<br>ADC1CMPIE =1：enable ADC1 measuring comparison interrupt |
| ADC0CMPIE | ADC0CMPIE =0：disable ADC0 measuring comparison interrupt<br>ADC0CMPIE =1：enable ADC0 measuring comparison interrupt |
| VbatCMPIE | VbatCMPIE=0：disable batter voltage comparison interrupt<br>VbatCMPIE=1：enable batter voltage comparison interrupt |
| ADC2IE | ADC2IE=0：disable ADC2 detecting interrupt<br>ADC2IE=1：enable ADC2 detecting interrupt |
| ADC1IE | ADC1IE=0：disable ADC1 detecting interrupt<br>ADC1IE=1：enable ADC1 detecting interrupt |
| ADC0IE | ADC0IE=0：disable ADC0 detecting interrupt<br>ADC0IE=1：enable ADC0 detecting interrupt |
| VccIE | VccIE=0：disable Vcc measuring interrupt<br>VccIE=1：enable Vcc measuring interrupt |
| VbatIE | VbatIE=0：disable battery voltage measuring interrupt<br>VbatIE=1：enable battery voltage measuring interrupt |
| TPSIE | TPSIE =0：disable temperature measuring interrupt<br>TPSIE =1：enable temperature measuring interrupt |

| **TBSIF**<br>**(TBS interrupt enable register)** | | | **Base address：** **0x4000E000**<br>**Offset address：** **0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | X | X | X | X | ADC1CMPIF | ADC0CMPIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | X | VbatCMPIF | ADC2IF | ADC1IF | ADC0IF | VccIF | VbatIF | TPSIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| ADC1CMPIF | ADC1CMPIF =0：no ADC1 measuring comparison interrupt generated<br>ADC1CMPIF =1：ADC1 measuring comparison interrupt generated<br>Cleared by writing 0 |
| ADC0CMPIF | ADC0CMPIF =0：no ADC0 measuring comparison interrupt generated<br>ADC0CMPIF =1：ADC0 measuring comparison interrupt generated<br>Cleared by writing 0 |
| VbatCMPIF | VbatCMPIF=0：no battery voltage comparison interrupt generated |

| | |
|---|---|
| | VbatCMPIF=1：battery voltage comparison interrupt generated<br>Cleared by writing 0 |
| ADC2IF | ADC2IF=0：no ADC channel2 detecting interrupt generated<br>ADC2IF=1：ADC channel2 detecting interrupt generated<br>Cleared by writing 0 |
| ADC1IF | ADC1IF=0：no ADC channel1 detecting interrupt generated<br>ADC1IF=1：ADC channel1 detecting interrupt generated<br>Cleared by writing 0 |
| ADC0IF | ADC0IF=0：no ADC channel0 detecting interrupt generated<br>ADC0IF=1：ADC channel0 detecting interrupt generated<br>Cleared by writing 0 |
| VccIF | VccIF=0：no Vcc measuring interrupt generated<br>VccIF=1：Vcc measuring interrupt generated<br>Cleared by writing 0 |
| VbatIF | VbatIF=0：no battery voltage measuring interrupt generated<br>VbatIF=1：battery voltage measuring interrupt generated<br>Cleared by writing 0 |
| TPSIF | TPSIF =0：no temperature measuring interrupt generated<br>TPSIF =1：temperature measuring interrupt generated<br>Cleared by writing 0 |

| **TMPDAT (temperature measuring output register)** | | | **Base address：** **0x4000E000**<br>**Offset address：** **10H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| DAT[15:0] | Temperature measuring output register,16-bit signed num |

Temperature calculating formula:

Temperature Tr =12.9852 - TMPDAT *0.0028

Where:Tr is the real temperature（℃）

| **VBATDAT (battery voltage measuring output register)** | | | **Base address：** **0x4000E000**<br>**Offset address：** **14H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |

| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|

| Bits | Function description |
|---|---|
| DAT[15:0] | Battery voltage measuring output register,16-bit signed num |

Battery voltage calculating formula:

VBAT = 0.0897*VBATDAT+2946.2662；

Where: Vbat is real battery voltage(mV)

| **VCCDAT (power source voltage measuring output register)** | | | **Base address：** **0x4000E000** **Offset address：** **18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[15:0] | Power source voltage measuring output register,16bit signed num |

Power source voltage calculating formula:

VDD = 0.0897*VDDDAT+ 2998.1189

Where: Vcc is real power source voltage(mV)

| **ADC0DAT (ADC0 measuring output register)** | | | **Base address：** **0x4000E000** **Offset address：** **1CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[15:0] | ADC channel0 measuring output register,16-bit signed num |

ADC channel0 measuring voltage calculating formula:

VADCIN0 = 0.0128*ADC0DAT+425.5623；

Where: Vadc0 is real ADC measuring voltage(mV)

| **ADC1DAT (ADC1 measuring output register)** | **Base address：** **0x4000E000** **Offset address：** **20H** |
|---|---|

| | Bit15…Bit0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[15:0] | ADC channel1 measuring output register,16-bit signed num |

ADC channel1 measuring voltage calculating formula:

VADCIN1 = 0.0128*ADC0DAT+425.5623；

Where: Vadc1 is real ADC measuring voltage(mV)

| **ADC2DAT (ADC2 measuring output register)** | **Base address： 0x4000E000**<br>**Offset address： 24H** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[15:0] | ADC channel2 measuring output register,16-bit signed num |

ADC channel2 measuring voltage calculating formula:

VADCIN2 = 0.0128*ADC0DAT+425.5623；

Where: Vadc2 is real ADC measuring voltage(mV)

| **VBATCMP (battery voltage compare register)** | **Base address： 0x4000E000**<br>**Offset address： 28H** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[15:0] | Battery voltage measuring comparison register,16-bit signed num |

Interrupt will be generated when VBATDAT below VSRCMP if enabled battery voltage measuring(VBATEn) and battery voltage comparison interrupt(VbatCMPIE)

| **ADC0CMP** | **Base address： 0x4000E000** |
|---|---|

| (ADC0 measuring comparation register) | | | Offset address： 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | DFx15 | DFx14 | DFx13 | DFx12 | DFx11 | DFx10 | DFx9 | DFx8 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | DFx7 | DFx6 | DFx5 | DFx4 | DFx3 | DFx2 | DFx1 | DFx0 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| **DFx[15..0]** | ADC0 measuring comparison register,16-bit signed num |

Interrupt will be generated when ADC0 measuring value is greater than value of ADC0CMP register

| **ADC1CMP** **(ADC1** measuring comparison register) | | | **Base address： 0x4000E000** **Offset address： 38H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | DFx15 | DFx14 | DFx13 | DFx12 | DFx11 | DFx10 | DFx9 | DFx8 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | DFx7 | DFx6 | DFx5 | DFx4 | DFx3 | DFx2 | DFx1 | DFx0 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| **DFx[15..0]** | ADC1 measuring comparison register,16-bit signed num |

Interrupt will be generated when ADC1 measuring value is greater than value of ADC1CMP register

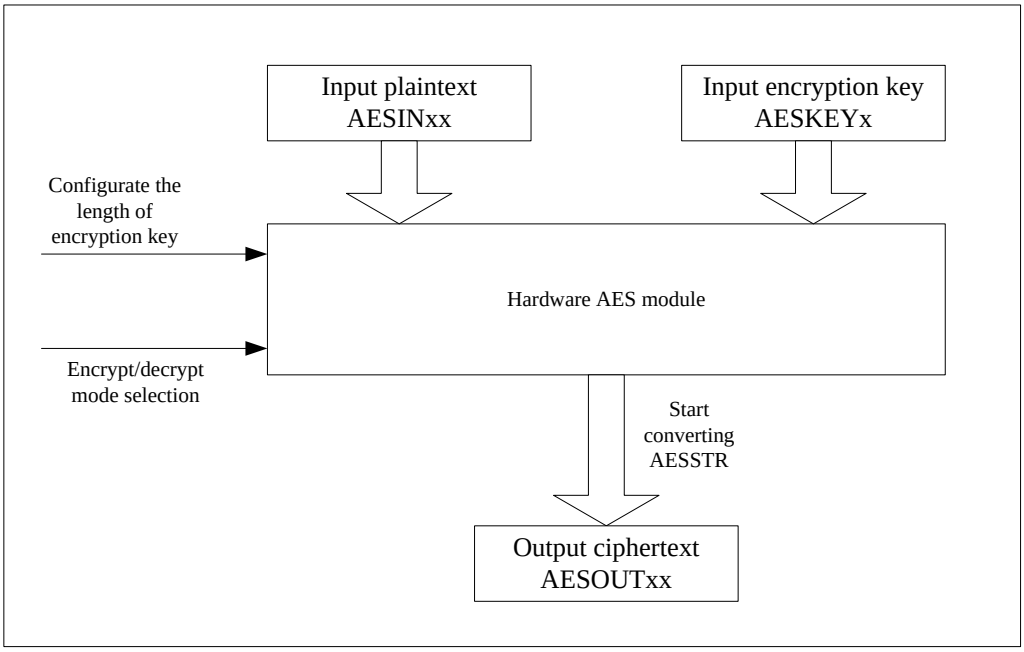
# 18. AES&Ghash&RAND module

## 18.1. General introduction

There are internal integrated AES hardware encrypt/decrypt module in HT502X which conform with FIPS197 standard,configurated trough corresponding software,support EBC/CBC/CTR/GCM/GMAC modules etc.the length of encryption key of AES can be 128bit,192bit,256bit.different encrypt/decrypt mode of AES algorithm implemented by users hardware is reduced significantly by employing hardware module and Ghash conductor module.

HT502X also integrates the random number generator module, which generates 32 bits random numbers at one time, and can be used to generate random numbers in ECC encryption.

## 18.2. Block diagram




## 18.3. Special function register list

| AES module register base address：0x40012000 | | | | |
|---|---|---|---|---|
| **Offset address** | **name** | **Write/read** | **Reset value** | **Function description** |
| 00H | AESCON | R/W | 0000H | 3DES configurate register |
| 04H | AESSTR | W | 0000H | 3DES start order register |
| 08H | AESFLG | R/W | 0000H | 3DES flag register |
| 0CH | AESINLL | R/W | 00000000H | AES input data to be encrypted/decrypted |

| | | | | |
|---|---|---|---|---|
| 10H | AESINML | R/W | 00000000H | AES input data to be encrypted/decrypted |
| 14H | AESINHM | R/W | 00000000H | AES input data to be encrypted/decrypted |
| 18H | AESINHH | R/W | 00000000H | AES input data to be encrypted/decrypted |
| 1CH | AESOUTLL | R/W | 00000000H | AES output data encrypted/decrypted |
| 20H | AESOUTML | R/W | 00000000H | AES output data encrypted/decrypted |
| 24H | AESOUTHM | R/W | 00000000H | AES output data encrypted/decrypted |
| 28H | AESOUTHH | R/W | 00000000H | AES output data encrypted/decrypted |
| 2CH | AESKEY0 | R/W | 00000000H | AES encryption key register |
| 30H | AESKEY1 | R/W | 00000000H | AES encryption key register |
| 34H | AESKEY2 | R/W | 00000000H | AES encryption key register |
| 38H | AESKEY3 | R/W | 00000000H | AES encryption key register |
| 3CH | AESKEY4 | R/W | 00000000H | AES encryption key register |
| 40H | AESKEY5 | R/W | 00000000H | AES encryption key register |
| 44H | AESKEY6 | R/W | 00000000H | AES encryption key register |
| 48H | AESKEY7 | R/W | 00000000H | AES encryption key register |

| **RAND module register base address：0x40012080** | | | | |
|---|---|---|---|---|
| **Offset address** | **Name** | **Write/read** | **Reset value** | **Function description** |
| 00H | RANDSTR | R/W | 0x0000 | Random start instruction register |
| 04H | RANDDAT | R/W | 0x00000000 | Random data register |

| **GHASH module register base address：0x40012100** | | | | |
|---|---|---|---|---|
| **Offset address** | **Name** | **Write/read** | **Reset value** | **Function description** |
| 00H | GHASHSTR | W | 0000H | GHASH start order register |
| 04H | GHASHFLG | R/W | 0000H | GHASH flag register |
| 08H | INPUT1LL | R/W | 00000000H | GHASH input data1 |
| 0CH | INPUT1ML | R/W | 00000000H | GHASH input data1 |
| 10H | INPUT1HM | R/W | 00000000H | GHASH input data1 |
| 14H | INPUT1HH | R/W | 00000000H | GHASH input data1 |
| 18H | INPUT2LL | R/W | 00000000H | GHASH input data2 |
| 1CH | INPUT2ML | R/W | 00000000H | GHASH input data2 |
| 20H | INPUT2HM | R/W | 00000000H | GHASH input data2 |
| 24H | INPUT2HH | R/W | 00000000H | GHASH input data2 |
| 28H | OUTPUTLL | R/W | 00000000H | GHASH output data |
| 2CH | OUTPUTML | R/W | 00000000H | GHASH output data |
| 30H | OUTPUTHM | R/W | 00000000H | GHASH output data |
| 34H | OUTPUTHH | R/W | 00000000H | GHASH output data |

| | | | | |
|---|---|---|---|---|
| 38H | AESGHASHIE | R/W | 00H | AES,Ghash interrupt enable bit |
| 3CH | AESGHASHIF | R/W | 00H | AES,Ghash interrupt flag bit |

## 18.4. Special function register introduction

| **AESCON （AES configuration register）** | | | **Base address： 0x40012000 Offset address： 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | KEY_MODE[1] | KEY_MODE[0] | MODE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| KEY_MODE | **KEY_MODE[1。。。0]:** =B00: AES-128，encryption key is the high bits of 128-bit(KEY0—KEY3)KEY0,KEY3 is the low bits =B01：AES-192，encryption key is the high bits of 192bits(KEY0—KEY5)KEY0,KEY5 is the low bits =B10：AES-256，encryption key is the high bits of 256bits(KEY0—KEY7)KEY0, KEY7 is the low bits =B11：AES-256，encryption key is the high bits of 256bits(KEY0—KEY7)KEY0, KEY7 is low bits |
| MODE | **Encode/decode mode select control：** 0：encrypt 1：decrypt |

| **AESSTR （AES start register）** | | | **Base address： 0x40012000 offset address： 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| DAT[15:0] | Write 0x8329 to start AES encrypt/decrypt |

| | |
|---|---|
| | It is read-only,the read value is always 0.<br>Write 0x8581 to the register and Key remains unchanged to start encoding plaintext and it can inprove AES calculating efficiency.And it is unnecessary to repeating calculating KEY part to be converted in AES algorithm if the KEY user configurated is not modified. |

| **AESFLG<br>（AES flag register）** | | | **Base address：0x40012000<br>offset address：08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:<br>Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | X | X | X | X | X | X | BUSY |
| **Write:** | | | | | | | | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| BUSY | 1：AES encrypt/decrypt is not accomplished<br>0：AES encrypt/decrypt is accomplished<br>It is read-only<br>Corresponding interrupt flag will be generated and entering corresponding interrupt if encrypt/decrypt is accomplished. The AES related interrupt vector number is 1 |

| **AESINLL<br>（AES middle and low bits of input data）** | | | **Base address：0x40012000<br>offset address：0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:<br>Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit AES input data to be encrypted/decrypted |

| **AESINML<br>（AES middle and low bits of input data）** | **Base address：0x40012000<br>offset address：18H** |
|---|---|
| | **Bit31…Bit0** |

| Read: | DAT[31:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit AES input data to be encrypted/decrypted |

| AESINHM （AES middle and high bits of input data） | | | | Base address： 0x40012000 offset address： 10H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit AES input data to be encrypted/decrypted |

| AESINHH （AES high bits of input data） | | | | Base address： 0x40012000 offset address： 14H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit AES input data to be encrypted/decrypted |

| AESOUTLL （AES low bit of output data） | | | | Base address： 0x40012000 offset address： 1CH | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit encrypted/decrypted data AES output |

| AESOUTML（AES middle and low bits of output data） | | | Base address： 0x40012000 offset address： 20H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit encrypted/decrypted data AES output |

| AESOUTHM (AES middle and high bits of output data) | | | Base address： 0x40012000 offset address： 24H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit encrypted/decrypted data AES output |

| AESOUTHH (AES high bits of output data) | | | Base address： 0x40012000 offset address： 28H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit encrypted/decrypted data AES output |

| AESKEY0 (AES encryption key0) | | | Base address： 0x40012000 offset address： 2CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit AES encryption key,readable and writable register<br>bit160-bit191 of 192-bit AES encryption key,readable and writable register<br>bit224-bit256 of 256-bit AES encryption key,readable and writable register |

| AESKEY1<br>(AES encryption key1) | | | Base address: 0x40012000<br>offset address: 30H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit AES encryption key,readable and writable register<br>bit128-bit159 of 192-bit AES encryption key,readable and writable register<br>bit192-bit223 of 256-bit AES encryption key,readable and writable register |

| AESKEY2<br>(AES encryption key2) | | | Base address: 0x40012000<br>offset address: 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit AES encryption key,readable and writable register<br>bit96-bit127 of 192-bit AES encryption key,readable and writable register<br>bit160-bit191 of 256-bit AES encryption key,readable and writable register |

| AESKEY3<br>(AES encryption key3) | | | Base address: 0x40012000<br>offset address: 38H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit AES encryption key,readable and writable register<br>bit64-bit95 of 192-bit AES encryption key,readable and writable register |

| | |
|---|---|
| | bit128-bit159 of 256-bit AES encryption key,readable and writable register |

| AESKEY4 （AES encryption key4） | | | Base address： 0x40012000 offset address： 3CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 192-bit AES encryption key,readable and writable register<br>bit96-bit127 of 256-bit AES encryption key,readable and writable register |

| AESKEY5 （AES encryption key5） | | | Base address： 0x40012000 offset address： 40H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 192-bit AES encryption key,readable and writable register<br>bit64-bit95 of 256-bit AES encryption key,readable and writable register |

| AESKEY6 （AES encryption key6） | | | Base address： 0x40012000 offset address： 44H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 256-bit AES encryption key,readable and writable register |

| AESKEY7 （AES encryption key7） | | Base address： 0x40012000 offset address： 48H |
|---|---|---|
| | Bit31…Bit0 | |
| Read: | DAT[31:0] | |

| Write: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 256-bit AES encryption key,readable and writable register |

| RANDSTR （Random number start control register） | | | Base Address： 0x40012080 Offset Address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | RANDEN | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | X | X | X | X | RESERVED | Start | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function description |
|---|---|
| RANDEN | **Random module enable control bit**<br>0：disable<br>1：enable<br>After enable RANDEN module, write 1 to Start is valid. |
| RESERVED | **Reserved bit , invalid operation, and read value is meaningless.** |
| Start | **Start control bit**<br>When the register bit is written 1, the true random number generator is started. The control bit is cleared by the hardware after the random number is generated. |

| RANDDAT （random data register） | | | Base address： 0x40012080 Offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function description |
|---|---|
| DAT[31:0] | Random data register, total 32 bits. |

| GHASHSTR（GHASH start register） | | | Base address： 0x40012100 offset address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: | X | | | | | | | |
| Write: | DAT[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[15:0] | Write 0x0501 to the register to start GHASH conduct calculating<br>This control register is read only,the read value of this register is always 0. |

| GHASHFLG（GHASH flag register） | | | Base address： 0x40012100 offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: | X | X | X | X | X | X | X | X |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: | X | X | X | X | X | X | X | BUSY |
| Write: | | | | | | | | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| BUSY | 1：GHASH conducting calculation is not completed<br>0：GHASH conducting calculation is completed<br>This bit is read only<br>Corresponding interrupt flag will be generated and enter corresponding interrupt if interrupt enabled when conduct operation completed.the AESGHASH related interrupt vector num is 1. |

| INPUT1LL（low bits of input data1） | | | Base address： 0x40012100 offset address： 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit Ghash input calculating data1 |

| INPUT1ML（middle and low bits of input data1） | | | | Base address： 0x40012100<br>offset address： 0CH | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit Ghash input calculating data1 |

| INPUT1MH（middle and high bits of input data1） | | | | Base address： 0x40012100<br>offset address： 10H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit Ghash input calculating data1 |

| INPUT1HH（high bits of input data1） | | | | Base address： 0x40012100<br>offset address： 14H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read:<br>Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit Ghash input calculating data1 |

| INPUT2LL（low bits of input data2） | | Base address： 0x40012100<br>offset address： 18H |
|---|---|---|
| | Bit31…Bit0 | |
| Read:<br>Write: | DAT[31:0] | |

| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|

| Bits | function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit Ghash input calculating data1 |

| INPUT2ML （middle and low bits of input data2） | | | | Base address： 0x40012100 offset address： 1CH | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit Ghash input calculating data1 |

| INPUT2MH （middle and high bits of input data2） | | | | Base address： 0x40012100 offset address： 20H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit Ghash input calculating data1 |

| INPUT2HH （high bits of input data2） | | | | Base address： 0x40012100 offset address： 24H | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit Ghash input calculating data1 |

| OUTPUTLL | Base address： 0x40012100 |
|---|---|

| （low bits of output data） | | | offset address： 28H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[31:0] | bit0-bit31 of 128-bit Ghash input calculating data1 |

| OUTPUTML （middle and low bits of output data） | | | Base address： 0x40012100 offset address： 2CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[31:0] | bit32-bit63 of 128-bit Ghash input calculating data1 |

| OUTPUTMH （ middle and high bits of output data） | | | Base address： 0x40012100 offset address： 30H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|
| DAT[31:0] | bit64-bit95 of 128-bit Ghash input calculating data1 |

| OUTPUTHH （high bits of output data） | | | Base address： 0x40012100 offset address： 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Function description |
|---|---|

| | |
|---|---|
| DAT[31:0] | bit96-bit127 of 128-bit Ghash input calculating data1 |

| **AESGHASHIE（ AESGHASH interrupt enable）** | | | **Base address： 0x40012100 offset address： 38H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | X | GHASHIE | AESIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| GHASHIE | GHASH calculating interrupt enable |
| AESIE | AES calculating interrupt enable |

| **AESGHASHIF（AESGHASH interrupt flag）** | | | **Base address： 0x40012100 offset address： 3CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | X | X | X | X | X | X | GHASHIF | AESIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **Function description** |
|---|---|
| GHASHIF | Interrupt flag |
| AESIF | AES calculating interrupt flag |

# 19. ECC Module

## 19.1. General

The ECC256 module of HT502X is an acceleration module of elliptic curve (EC, Elliptic, Curve) operation, which can significantly improve the efficiency of the encryption protocol based on elliptic curve. The cryptographic protocols or algorithms supported by this module include ECDSA (EC, Digital, Signature, Algorithm), ECDH (EC, Deffie-Hellman) and related mutation protocols.

Function Description

- Support elliptic curves on all primary fields GF (P) based on the simplified Weierstrass equation (y2=x3+ax+b, mod, P), which are defined as elliptic curves in NIST, SEC2, Brainpool and other protocols

- Supports ECDSA signature / authentication operations

- Supports whether public keys verify operations on elliptic curves

- Support elliptic curve point operations

  Point multiplication (ECSM, EC Scalar Multiplication)
  Point addition operation (ECA, EC Addition)
  Doubling operation (ECD, EC Doubling)

- Support large number modulo operation

  Modular addition operation (MA, Modular Addition)
  Modulo subtraction operation (MS, Modular Subtraction)
  Modular multiplication (MM, Modular Multiplication）
  Modulo division operation (MD, Modular Division)
  Modulo inverse operation (MI, Modular Inversion）

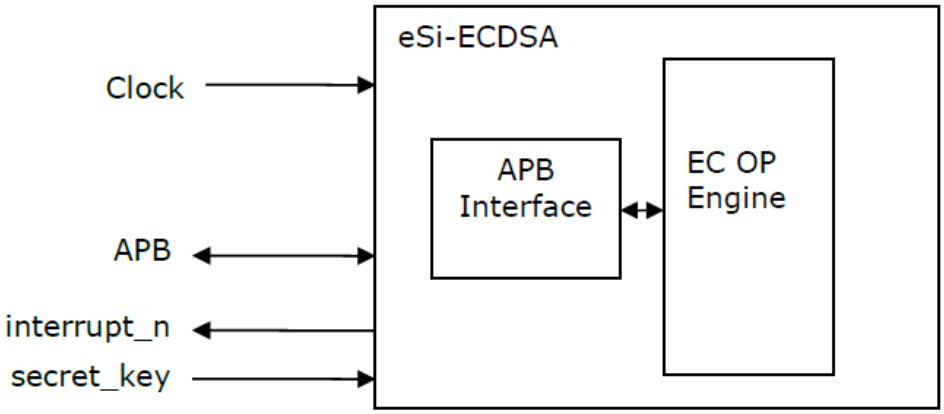
- Support ECC256，Downward compatible ECC224 and ECC192

- Can block STA attack (both Simple and Statistical Timing side channel Attacks) and SPA attack (Simple Power Analysis attacks).

AES_EN in Clkctrl0 is AES/RAND, shared enable control Bit, Clkctrl1 new control BitECC_EN, independent control ECC module, Bit set bit 17 (Clkctrl1.<17>), application ECC module before, can make this control Bit. Each time the shutdown is enabled, the ECC module is restored to the Bit ECC_EN.

The Secret key is connected to the internal 256-bits register, where the user stores the private key in the flash 0x0007FC00~0007FC1F; the user can configure the ECCCON.6<SE> and select Secret, key, or KEYREG as the private key. Each self-loading, the 0x0007FC00~0007FC1F in the data is loaded into the 256-bits register, low Bit stored in 0x0007FC00, high Bit stored in 0007FC1F; the 256-bits register is not open, the user cannot operate.

## 19.2. ECC Operational Acceleration Module




ECC Configuration of each arithmetic register:

| **Acceleration module input / output** | | | |
|---|---|---|---|
| | Calculate mode | Input parameters register | Output result register |
| ECDSA calculate | **Signature operation** | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>PXREG:X coordinate of base point of Elliptic curve<br>PYREG: Y coordinate of base point of Elliptic curve<br>SXREG:Order parameter of base point<br>KEYREG: private key **k**<br>SYREG: Random large integer<br>MREG: Message digest after HASH | RXREG: Signature results **R** parameter<br>RYREG: Signature results **S** parameter<br>ECCSTA: ECC status register<br>[3:3]ECDSA_S = 0(if the value is 1, Retrieves a random large integer, re – operation)<br>[0:0]ECCFLG = 1 |
| | **Authenticatio n operation** | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>PXREG: X coordinate of base point of Elliptic curve<br>PYREG: Y coordinate of base point | ECCSTA: ECC status register<br>[0:0]ECCFLG = 1<br>[2:2]ECDSA_V =<br>1(Authentication pass)<br>0(Authenticaion failed) |

| | | | |
|---|---|---|---|
| | | of Elliptic curve<br>SXREG:Order parameter of base point<br>KEYREG：Public key X coordinate<br>SYREG: Public key Y coordinate<br>MREG: Message digest after HASH<br>RXREG:Signature results **R** parameter<br>RYREG: Signature results **S** parameter | |
| The point is verified on the curve | **Public key authentication** | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>SYREG: Elliptic curve parameter **b**<br>PXREG: Public key X coordinate<br>PYREG: Public key Y coordinate | ECCSTA: ECC status register<br>[0:0]ECCFLG = 1<br>[4:4]PKV =<br>1(Authentication pass)<br>0(Authenticaion failed) |
| ECC Point operation | Point addition operation | **P(X1,Y1) + S(X2,Y2) = R(X3,Y3)** | |
| | | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>PXREG: The X coordinates of first points on an elliptic curve<br>PYREG: The Y coordinates of first points on an elliptic curve<br>SXREG: The X coordinates of second points on an elliptic curve<br>SYREG: The Y coordinates of second points on an elliptic curve | RXREG: Point plus result X coordinate<br>RYREG: Point plus result Y coordinate<br>(the result point on the elliptic ccurve)<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Doubling addition operation | **2 *P(X1,Y1) = R(X3,Y3)** | |
| | | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>PXREG: The X coordinates of first points on an elliptic curve<br>PYREG: The Y coordinates of first points on an elliptic curve | RXREG: Doubling plus result X coordinate<br>RYREG: Doubling plus result Y coordinate<br>(the result point on the elliptic ccurve)<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Point multiplication | **k*P(X1,Y1) = R(X3,Y3)** | |
| | | PREG: prime numbers **p**<br>AREG: Elliptic curve parameter**a**<br>PXREG: The X coordinates of point on an elliptic curve<br>PYREG: The Y coordinates of point | RXREG: Point multiplication result X coordinate<br>RYREG: Point multiplication result Y coordinate<br>(the result point on the elliptic ccurve) |

| | | | |
|---|---|---|---|
| | | on an elliptic curve<br>KEYREG：Scalar coefficient **k** | ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| Scalar modulo operation | Modular addition operation | **PX + PY ( mod n ) = RX** | |
| | | PREG: Modulo arithmetic parameter **n**<br>PXREG: the first scalar<br>PYREG: the second scalar | RXREG: scalar of modular addition result<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Modulo subtraction operation | **PX - PY ( mod n ) = RX** | |
| | | PREG: Modulo arithmetic parameter **n**<br>PXREG: the first scalar<br>PYREG: the second scalar | RXREG: scalar of modulo subtraction result<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Modular multiplication | **PX * PY ( mod n ) = RX** | |
| | | PREG: Modulo arithmetic parameter **n**<br>PXREG: the first scalar<br>PYREG: the second scalar | RXREG: scalar of modular multiplication result<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Modulo division operation | **PY / PX ( mod n ) = RX** | |
| | | PREG: Modulo arithmetic parameter **n**<br>PXREG: the first scalar<br>PYREG: the second scalar | RXREG: scalar of modulo division result<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |
| | Modulo inverse operation | **$PX^{-1}$ ( mod n ) = RX** | |
| | | PREG: Modulo arithmetic parameter **n**<br>PXREG: the first scalar | RXREG: scalar of modulo inverse result<br>ECCSTA: ECC status register<br>[0:0]ECCFLG = 1 |

## 19.3. Special Function registers list

| **ECCmodule registerBase address：0x4001B000** | | | | |
|---|---|---|---|---|
| **offset address** | **Name** | **Read/Write** | **Reset value** | **Function Description** |
| 0x00 | ECCCON | R/W | 0x0000 | ECC module control register |
| 0x04 | ECCSTA | R/W | 0x0000 | ECC module status register |
| 0x08 | PXREG | R | 0x0000 | Base point X coordinate register |
| 0x0C | PYREG | R | 0x0000 | Base point Y coordinate register |
| 0x10 | KEYREG | R | 0x0000 | Prive key register |
| 0x14 | AREG | R | 0x0000 | Elliptic curve parameter **a** |
| 0x18 | PREG | R | 0x0000 | prime numbers **p** |

| | | | | |
|---|---|---|---|---|
| 0x1C | RXREG | R/W | 0x0000 | ECC Arithmetic output point X coordinate |
| 0x20 | RYREG | R/W | 0x0000 | ECC Arithmetic output point Y coordinate |
| 0x24 | SXREG | R | 0x0000 | ECC Second operand X coordinate |
| 0x28 | SYREG | R | 0x0000 | ECC Second operand Y coordinate |
| 0x2C | MREG | R/W | 0x0000 | Hash Algorithm message digest(for ECDSA) |

## 19.4. Special Function registers description

| **ECCCON** (ECC module controlregister) | | | **Base address: 0x4001B000** **offset address: 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | X | SE | ECCIE | OP_SEL 3 | OP_SEL 2 | OP_SEL 1 | OP_SEL 0 | OP_STR |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| SE | **Private key select control Bit**<br>1：Force the Secret Key as the private key instead of the value in the KEYREG register<br>0：KEYREG register as private key |
| ECCIE | **ECC interrupt enable control(Interrupt vector sharing with AES)**<br>1：enable ECC interrupt<br>0：disable ECC interrupt |
| OP_SEL[3:0] | **Arithmetic mode selection Bit**<br>0000: Point multiplication<br>0001：Point addition operation<br>0010：Doubling addition operation<br>0011：Modular addition operation<br>0100：Modulo subtraction operation<br>0101：Modular multiplication<br>0110：Modulo division operation<br>0111：Modulo inverse operation<br>1000：ESDA Signature operation<br>1001：ESDA Authentication operation<br>1010：Verification on public key curves<br>Others: reserved |
| OP_STR | **Start authenticate**<br>1：Start the set of operations, the end of the calculation, the hardware cleared<br>0：Operation complete |

| ECCSTA (ECC module status register) | | | Base address: 0x4001B000 offset address: 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | X | X | X | X | X | X | X | X |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | X | X | X | PKV | ECDSA_S | ECDSA_V | BUSY | ECCFLG |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| PKV | **Validation mark on public key curve**<br>0：Validation failure<br>1：Verify through<br>This Bit need ECCFLG=1, read it is valid. |
| ECDSA_S | **ECDSA Signature retry flag**<br>0：signature completely<br>1：Satisfy retry condition<br>This Bit need ECCFLG=1, read it is valid. |
| ECDSA_V | **ECDSA verify flag**<br>0：Validation failure<br>1：Verify through<br>This Bit need ECCFLG=1, read it is valid. |
| BUSY | **Calculate module status Bit**<br>0：module idle<br>1：module busy |
| ECCFLG | **ECCArithmetic completion flag Bit**<br>0：unfinished<br>1：finished<br>Note：Write 0 to clear this flag |

| PXREG (Base point X coordinate register) | | | Base address: 0x4001B000 offset address: 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: Write: | DAT[31:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|

| | |
|---|---|
| DAT[31:0] | **Entrance data of Base point X coordinate(4 bytes subsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right).<br>If the actual Bit width is less than 256bit, the read data is 0. |

| **PYREG**<br>**(**Base point Y coordinate register**)** | | | **Base address:** **0x4001B000**<br>**offset address:** **0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| DAT[31:0] | **Entrance data of Base point Y coordinate(4 bytes subsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right).<br>If the actual Bit width is less than 256bit, the read data is 0. |

| **KEYREG**<br>**(**Private key register**)** | | | **Base address:** **0x4001B000**<br>**offset address:** **10H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| DAT[31:0] | **Entrance data of Private key register(4 bytes subinsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid |

| | |
|---|---|
| | bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **AREG**<br>**(**Elliptic curve parameter **a)** | | | | **Base address:** **0x4001B000**<br>**offset address:** **14H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| DAT[31:0] | **Entrance data of Elliptic curve parameter, a(4 bytes subinsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **PREG**<br>**(**prime numbers **p)** | | | | **Base address:** **0x4001B000**<br>**offset address:** **18H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bit** | **Function Description** |
|---|---|
| DAT[31:0] | **Entrance data of prime numbers ,p (4 bytes subinsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **RXREG**<br>**(**ECC Arithmetic output point | **Base address:** **0x4001B000**<br>**offset address:** **1CH** |
|---|---|

| X coordinate) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| DAT[31:0] | **Entrance data of ECCArithmetic output point X coordinate (4 bytes subinsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **RYREG** (ECC Arithmetic output point Y coordinate) | **Base address:** 0x4001B000<br>**offset address:** 20H | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| DAT[31:0] | **Entrance data of ECCArithmetic output point Y coordinate (4 bytes subinsection)**<br>**Write in:**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **SXREG** (ECC Second operand X coordinate) | **Base address:** 0x4001B000<br>**offset address:** 24H | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| Read: | DAT[31:0] | | | | | | | |
| Write: | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| DAT[31:0] | **Entrance data of ECCSecond operand X coordinate (4 bytes subinsection)**<br>**Write in：**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out：**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **SYREG**<br>**(**ECC Second operand Y coordinate**)** | | | | **Base address：** **0x4001B000**<br>**offset address：** **28H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| DAT[31:0] | **Entrance data of ECCSecond operand X coordinate (4 bytes subinsection)**<br>**Write in：**<br>Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out：**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

| **MREG**<br>**(**Hash Algorithm message digest**)** | | | | **Base address：** **0x4001B000**<br>**offset address：** **2CH** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function Description |
|---|---|
| DAT[31:0] | **Hash Algorithm message digest（4 bytes subsection）**<br>**Write in:** |

| | |
|---|---|
| | Bit width of 256bit data, write 8 registers, from the minimum 4 valid bytes start (right to left write), if the actual Bit width is less than 256bit, high Bit zero padding.<br>**Read out:**<br>4bytes segmentation, read 8 registers, to obtain data, first read the maximum 4 valid bytes (from left to right)<br>If the actual Bit width is less than 256bit, the read data is 0 |

Note: after the transfer of information through the HASH algorithm, the summary of information is obtained at most, 256bit can participate in the ECDSA operation, and the information digest is inputted into the register as the parameters involved in the ECDSA operation.

# 20. DMA function

## 20.1. General introduction

DMA module provides high speed data transmission between prepheral and RAM or RAMs.it is not necessary for CPU to involve the DMA data transmission but system bus is.

## 20.2. Function description

（1）there are 3 independent configurable channel for DMA
（2）data can be transferred between peripherals ,memory and peripherals,memories for DMA
（3）DMA and CPU take turns to occupy and employ bus
（4）DMA support 32 external require source at most and a software require source for each channel
（5）priority of three channels can be set and the transmission cannot be interrupt once it begins.The channel of highest priority will be answered if three channels. The channel of lowest priority will be answered after data transmission of the high priority channel is completed.
（6）the channel of high priority will be answered if two channels choose the same external require source at the same time.
（7）DMA support two transmission method:Transfer a data every one require,transfer all data at a require
（8）DMA support three interrupts:Transmission completed interrupt, transmission error interrupt,half-transmission interrupt
（9）DMA close the corresponding channel automatically after transmission completed in non-cyclical pattern.Users can configurate cycle times CHNxBULKNUM[8:15] in cyclical pattern and corresponding channel will be closed automatically when the cycle times is reached.

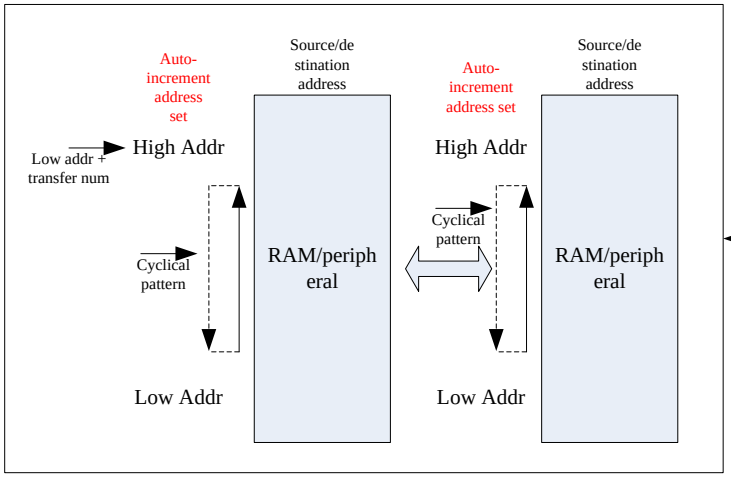
## 20.3. DMA channel require list

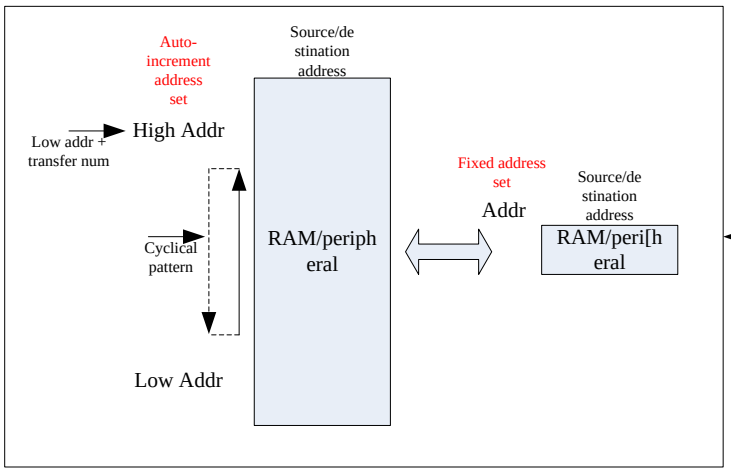
| DMA_CTL[8…12] | introduction |
|---|---|
| 0 | Soft request |
| 1 | UART0 send |
| 2 | UART0 receive |
| 3 | UART1 send |
| 4 | UART1 receive |
| 5 | UART2 send |
| 6 | UART2 receive |
| 7 | UART3 send |
| 8 | UART3 receive |
| 9 | UART4 send |

| | |
|---|---|
| 10 | UART4 receive |
| 11 | UART5 send |
| 12 | UART5 receive |
| 13 | ISO7816_0 send |
| 14 | ISO7816_0 receive |
| 15 | ISO7816_1 send |
| 16 | ISO7816_1 receive |
| 17 | SPI send |
| 18 | SPI receive |
| 19 | I2C send |
| 20 | I2C receive |
| 21 | EMU voltage waveform data |
| 22 | EMU current channel1 waveform data |
| 23 | EMU current channel2 waveform data |
| 24 | Timer0 |
| 25 | Timer1 |
| 26 | Timer2 |
| 27 | Timer3 |
| 28 | reverse |
| 29 | reverse |
| 30 | reverse |
| 31 | reverse |
| 32 | reverse |

Note:UART corresponding TXIF/RXIF will be set when DMA send/receive serial data(UART) is selected

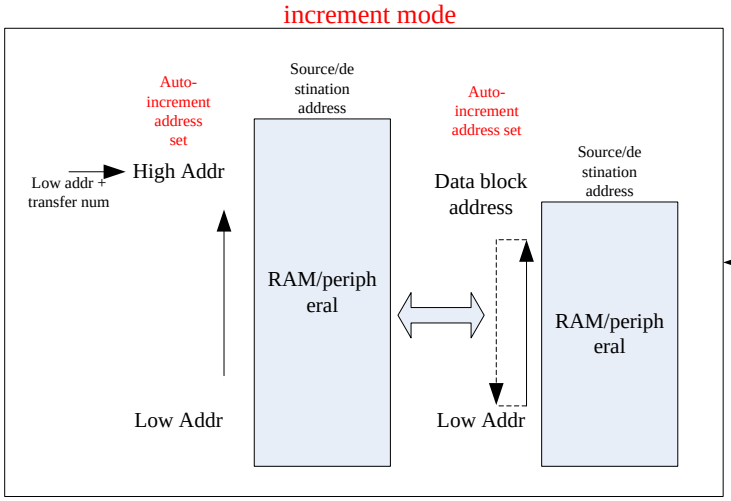
# 20.4. DMA data transmission introduction




Users can choose block transmission for these two transmission methods. It differs from single transmission on the aspect that it will not be interrupt during a block transmission

Data blcok auto-increment mode




Data block auto increment mode can be used for data transmission between two unknown-size data block

## 20.5. Special function register list

| DMA module register base address：0x40016000 | | | | |
|---|---|---|---|---|
| **offset address** | **Name** | **Write/read** | **Reset value** | **function description** |
| 00H | DMAIE | R/W | 0000H | DMA interrupt enable register |
| 04H | DMAIF | R/W | 0000H | DMA interrupt flag register |
| 08H | CHNSTA | R/W | 0000H | DMA state register |
| **DMA channel configuration register base address：<br>0x4001600C（Channel0）<br>0x40016024（Channel1）<br>0x4001603C（Channel2）** | | | | |
| **offset address** | **name** | **Write/read** | **Reset value** | **function description** |
| 00H | CHNCTL | R/W | 0000H | Channel control register |
| 04H | CHNSRC | R/W | 0000H | Channel source address register |
| 08H | CHNTAR | R/W | 0000H | Channel destination address register |
| 0CH | CHNCNT | R/W | 0000H | Channel transfer num set register |
| 10H | CHNTCCNT | R | 0000H | Num of data completed channel of transmission |
| 14H | CHNBULKNUM | R/W | 0000H | Channel block transmission num set |
| 0CH | CHN0CTL | R/W | 0000H | Channel0 control register |
| 10H | CHN0SRC | R/W | 0000H | Channel0 source address register |
| 14H | CHN0TAR | R/W | 0000H | Channel0 destination address register |
| 18H | CHN0CNT | R/W | 0000H | Channel0 transmission num set reigster |
| 1CH | CHN0TCCNT | R | 0000H | Num of data completed transmission of Channel0 |
| 20H | CHN0BULKNUM | R/W | 0000H | Channel0 block transmission num set |
| 24H | CHN1CTL | R/W | 0000H | Channel1control register |
| 28H | CHN1SRC | R/W | 0000H | Channel1 source address register |
| 2CH | CHN1TAR | R/W | 0000H | Channel1 destination address register |
| 30H | CHN1CNT | R/W | 0000H | Channel1 transmission num set register |
| 34H | CHN1TCCNT | R | 0000H | Num of data completed transmission of Channe1 |
| 38H | CHN1BULKNUM | R/W | 0000H | Channel1 block transmission num set |
| 3CH | CHN2CTL | R/W | 0000H | Channel2 control register |
| 40H | CHN2SRC | R/W | 0000H | Channel2 source address register |
| 44H | CHN2TAR | R/W | 0000H | Channe2destination address register |

| | | | | |
|---|---|---|---|---|
| 48H | CHN2CNT | R/W | 0000H | Channel2 transmission num set reigster |
| 4CH | CHN2TCCNT | R | 0000H | Num of data completed transmission of Channel2 |
| 50H | CHN2BULKNUM | R/W | 0000H | Channel2 block transmission num set |

## 20.6. Special function register introduction

| **DMAIE (DMA interrupt enable register)** | | | **Base address: 0x40016000 offset address: 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | | | | | | TEIE2 | TEIE1 | TEIE0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | | BCIE2 | BCIE1 | BCIE0 | | TCIE2 | TCIE1 | TCIE0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| TCIE[2…0] | channel0/1/2 transmission accomplished interrupt enable<br>0：disable<br>1：enable |
| BCIE[2…0] | channel0/1/2 block transmission interrupt enable<br>0：disable<br>1：enable |
| TEIE[2…0] | Channel0/1/2 transmission error interrupt enable<br>0：disable<br>1：enable |

| **DMAIF (DMA interrupt flag register)** | | | **Base address: 0x40016000 offset address: 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | | | | | | TEIF2 | TEIF1 | TEIF0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | | BCIF2 | BCIF1 | BCIF0 | | TCIF2 | TCIF1 | TCIF0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| TCIF[2…0] | Channel0/1/2 transmission accomplished interrupt flag<br>0：no interrupt generated<br>1：interrupt generated |
| BCIF[2…0] | Channel0/1/2 block transmission completed interrupt flag<br>0：no interrupt generated<br>1：interrupt generated |
| TEIF[2…0] | Channel0/1/2 transmission error interrupt flag<br>0：no interrupt generated<br>1：interrupt generated |

Note:

1.BCIE[2…0] and BCIF[2…0] is valid only in block transmission mode(TMOD=1)(invalid in single mode);

2.BCIF[2…0] will be set every time block transmission completed and TCIF[2…0] and BCIF[2…0] will be set at the same time when all data blocks transmission completed(set CHNCNT as transmission num) in block transmission mode.

| **CHNSTA**<br>**(DMA state register)** | | | **Base address：** **0x40016000**<br>**offset address：** **08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | | | | | | BUSY2 | BUSY1 | BUSY0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| bit | function description |
|---|---|
| BUSY[2…0] | channel0/1/2 transmission BUSY flag<br>0：free<br>1：busy |

| **CHNCTL**<br>**(DMA channel control register)** | | | **Base address：** **0x4001600C，0x40016024，0x4001603C**<br>**offset address：** **00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | | | Channel4 | Channel3 | Channel2 | Channel1 | Channel0 | DESTIN_INC1 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **Read:** | DESTIN | SOURC_ | SOURC_ | CYCLE | MODE | SIZE1 | SIZE0 | DMA_C |
| **Write:** | _INC0 | INC1 | INC0 | | | | | HN0EN |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: where x is 0,1,2

| bits | function description |
|---|---|
| Channel[4…0] | Trigger channel selection,decided by above DMA channel require list |
| DESTIN_INC[1…0] | Destination address auto-increment mode<br>00：not increment<br>01：increment<br>10：data block internal cycle increment<br>11：data block internal cycle increment |
| SOURC_INC[1…0] | Source address auto increment mode<br>01：increment<br>10：data block internal cycle increment<br>11：data block internal cycle increment |
| TMOD | Transmission mode<br>0：single transmission mode<br>1：block transmission mode（it will not be interrupted during 1 block data transmission |
| CYCLE | 0：non-cyclical pattern<br>1：cyclical pattern |
| PSIZE[1…0] | Transmission length of peripheral：<br>00：8 bits<br>01：16 bits<br>10：32 bits<br>11：32 bits |
| DMA_CHN0EN | DMA channel0 enable<br>0：disable<br>1：enable |

| **CHNSRC（DMA channel source address register）** | | | **Base address： 0x4001600C，0x40016024，0x4001603C<br>offset address： 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | ADDR15 | ADDR14 | ADDR13 | ADDR12 | ADDR11 | ADDR10 | ADDR9 | ADDR8 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | ADDR7 | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: where x is 0,1,2

| Bits | function description |
|---|---|
| ADDR[15…0] | Data transmission source address register |

| CHNTAR （DMA channel destination address register） | | | Base address： 0x4001600C，0x40016024，0x4001603C offset address： 08H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | ADDR15 | ADDR14 | ADDR13 | ADDR12 | ADDR11 | ADDR10 | ADDR9 | ADDR8 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | ADDR7 | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: where x is 0/1/2

| Bits | function description |
|---|---|
| ADDR[15…0] | Data transmission destination address register |

| CHNCNT （DMA channel transmission num register） | | | Base address： 0x4001600C，0x40016024，0x4001603C offset address： 0CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read: Write: | Num15 | Num14 | Num13 | Num12 | Num11 | Num10 | Num9 | Num8 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read: Write: | Num7 | Num6 | Num5 | Num4 | Num3 | Num2 | Num1 | Num0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: where x is 0/1/2

| Bits | function description |
|---|---|
| Num[15…0] | DMA data transmission num set register<br>Set to 65535 transmission data at most<br>This register indicate the num of data blocks to transmit in block transmission mode |

| CHNTCCNT (DMA channel num of transmitted data) | Base address： 0x4001600C，0x40016024，0x4001603C offset address： 10H |
|---|---|

| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
|---|---|---|---|---|---|---|---|---|
| **Read:** | Num15 | Num14 | Num13 | Num12 | Num11 | Num10 | Num9 | Num8 |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | Num7 | Num6 | Num5 | Num4 | Num3 | Num2 | Num1 | Num0 |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: where x is 0/1/2

| Bits | function description |
|---|---|
| Num[15…0] | Indicate the num of DMA transmitted data<br>This register indicate the num of data blocks DMA transmitted in block transmission mode |

| **CHNBULKNUM (DMA channel block transmission set register)** | | | **Base address: 0x4001600C，0x40016024，0x4001603C<br>offset address: 14H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | X | X | X | X |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | Num7 | Num6 | Num5 | Num4 | Num3 | Num2 | Num1 | Num0 |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| NUM[7…0] | Num selection of data Block to transmit,it indicate the num of data the data block contains：<br>0 — 255<br>This register indicate the num of data the data block contains in block transmission mode |

# 21. Key Scan module

## 21.1. General introduction

KEYSCAN module built-in automatic 4*4 keyboard scan function contains 4 line scan line input port SCANIN0~3 pins and 4 line scanning output port SCANOUT0~4 pin. If the user needs to use the button scan function, the corresponding pin must be configured as SCANINx or SCANOUTx multiplexing function.

## 21.2. Function Description

- Support maximum 4*4 button
- Support SLEEP\hold mode button wakeup
- Only supports falling edge triggering
- Only support single key detection

If you want to configure the 4*1 keyboard, while (4SCANIN + 1 SCANOUT) or (1 SCANIN + 4 SCANOUT) of the pin configuration for the corresponding multiplexing function can, if the configuration of the 4*2 keyboard is (4 + 2 SCANOUT (SCANIN) or 2 SCANIN + 4 SCANOUT) of the pin configuration for the corresponding function can be reused.

Configure the CLKCTRL0 KEY_EN=1 configuration, and corresponding I/O port for the SCANIN/SCANOUT function, the user can choose according to the number of SCANOUT, up to 4, were selected for SCANIN mode pin falling edge and keep the most low time 20 FOSC, then the key scan interrupt flags, check the KEYSTA status register, can be key position the specific location, if you open the interrupt enable (NVIC_EnableIRQ (KEY_IRQn)), while button interrupt, key scan interrupt number is 27.

By using the SCANIN pin, the pullup resistor must be enabled; the SCANOUT pin is used; the open drain function must be switched off.

## 21.3. Special function register list

| **KEYSCAN module register base address：0x40014000** | | | | |
|---|---|---|---|---|
| **offset address** | **name** | **Write/read** | **Reset value** | **function description** |
| 00H | KEYSTA | R/W | 0000H | KEYSCAN state register |
| 04H | KEYIF | R/W | 0000H | Key interrupt flag register |

# 21.4. Special function register introduction

| KEYSTA （KEYSCAN state register） | | | Base address： 0x40014000 offset address： 00H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | STA15 | STA14 | STA13 | STA12 | STA11 | STA10 | STA9 | STA8 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | STA7 | STA6 | STA5 | STA4 | STA3 | STA2 | STA1 | STA0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| STA[0…15] | Indicate the position of pressed KEY and support 16 KEYs at most,namely 16 states of KEY.<br>It is 16-bit register,which stands for the 16 states of KEY,if the corresponding KEY is pressed,the corresponding state bit will be set to 1,it two KEYs are pressed at the same time,the corresponding two bits will be set to 1. |

| | SCANIN0 | SCANIN1 | SCANIN2 | SCANIN3 |
|---|---|---|---|---|
| SCANOUT0 | STA0 | STA4 | STA8 | STA12 |
| SCANOUT1 | STA1 | STA5 | STA9 | STA13 |
| SCANOUT2 | STA2 | STA6 | STA10 | STA14 |
| SCANOUT3 | STA3 | STA7 | STA11 | STA15 |

| KEYIF （KEY flag register） | | | Base address： 0x40014000 offset address： 04H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** **Write:** | | | | | | | | KEYIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| KEYIF | Key interrupt flag register<br>Cleared by writing 0 |

# 22. Cortex-M0 kernel brief introduction

## 22.1. General introduction

Cortex-M0 is a high-integrated and low-power 32-bit processor kernel with 3-level pipeline Von Neumann architecture.Cortex-M0 can achieve high efficiency by adopting simple and powerful instruction set and fully optimized design(providing an advanced process hardware with single period multiplier)

Cortex-M0 processor adopt ARMv6-M structure which is based on 16-bit Thumb introduction set and Thumb-2 is supported.And a good performance of modern 32-bit structure is achieved,besides,the code density is higher than other 8 and 16 bit micro-controller.

## 22.2. System timer

CMSIS interface function SysTick_Config(uint32_t ticks) is recommended for system timer configuration.

For example:if Fcpu is 19.660800MHz,the following steps can be followed to configurate SysTick period interrupt as 100ms:

SysTick_Config (19660800/10);

Note:ticks cannot exceed the 0xFFFFFF because SysTick counter is 24-bit.

## 22.3. Interrupt priority introduction

A configurable nested vector interrupt processor is integrated into Cortex-M0 processor which makes the excellent interrupt performance possible. Following function of NVIC is provided:

- A non-masking interrupt(NMI) with fixed priority level,-2 which is highest level.
- A exception handle interrupt(HardFault) with fixed priority level,-1 which is second only to NMI
- Other interrupt with 4 priority levels to configurate which below above two interrupts

4 priority levels for other interrupt is 0x0, 0x1, 0x2, 0x3 among them 0x0 is the highest level priority and 0x3 the lowest. For example, to cofigurate DMA to be the priority of level 0(highest level), following ways of calling functions can be followed:

NVIC_SetPriority (DMA_IRQn, 0x0);

Note:DMA_IRQn is the interrupt num of DMA.see 8.1 interrupt vector instruction for more interrupt num.

## 22.4. CMSIS function introduction

Following table list parts of CMSIS functions, for the convenience of users. The unitive CMISIS functions are recommended to configurate kernel register instead of operating on kernel register directly.

| CMSIS function | Function introduction |
|---|---|
| __enable_irq() | Enable global interrupt |
| __disable_irq() | Disable global interrupt<br>（note：NMI and HardFault will not be masked） |
| void NVIC_EnableIRQ(IRQn_Type IRQn) | Enable the interrupt whose interrupt num is IRQn<br>（note:IRQn should >=0,the operation is invalid if the IRQn<=0） |
| void NVIC_DisableIRQ(IRQn_Type IRQn) | Mask the interrupt whose interrupt num is IRQn<br>（note:IRQn should >=0,the operation is invalid if the IRQn<=0） |
| uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn) | Get the suspended state of interrupt whose interrupt num is IRQn |
| void NVIC_SetPendingIRQ(IRQn_Type IRQn) | Set the suspended state of interrupt whose interrupt num is IRQn<br>（note:if the corresponding interrupt is enabled already,the system will enter corresponding interrupt process program when calling this function） |
| void NVIC_ClearPendingIRQ(IRQn_Type IRQn) | Clear the suspended state of interrupt whose interrupt num is IRQn<br>（note:interrupt suspended state will be cleared automatically after entering corresponding interrupt process program） |
| void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority) | Set the priority level of interrupt whose interrupt num is IRQn<br>（note:operation of set priority level id invalid for NMI and HardFault,the priority level range:0x0-0x3） |
| uint32_t NVIC_GetPriority(IRQn_Type IRQn) | Get the priority level of interrupt whose num is IRQn |
| void NVIC_SystemReset(void) | Reset system |
| uint32_t SysTick_Config(uint32_t ticks) | System timer configuration<br>（note:ticks cannot exceed 0xFFFFFF） |

Note:see 8.1 interrupt vector introduction for detailed information about interrupt IRQn

# 23. EMU module

## 23.1. General introduction

Provide all function necessary for single-phase energy measurement,including measurement of active power and active energy, reactive power and reactive energy, apparent power and apparent energy, rms voltage, rms current and frequency,support flexible power stealing prevention scheme and correcting table scheme.

- Active power error is less than 0.1% within the dynamic range of 5000:1
- The precision of reactive power is far higher than the national standard 2
- Active power and active energy;reactive power and reactive energy; apparent power and apparent energy
- Provide two channels low speed and high speed, of active power, reactive power and apparent power at the same time
- Support RMS voltage and RMS current of low speed and high speed at the same time
- Voltage frequency measurement
- Three ADC synchronous sampling waveform data; active power , reactive power, apparent power waveform data
- Flexible auto power stealing prevention scheme the power steal prevention threshold can be modified through register
- Flexible creeping and start scheme
- Many modes of energy calculation
- Open fast pulse countyregister, reduce the energy loss when power on and power off
- Optional pulse output PF/QF/SF width
- Support single-wire three phase form
- Zero-crossing interrupt detection signal
- Reactive power phase shift compensation
- RMS value bias correction
- Active power P, reactive power Q bias correction
- Ultra-low power support flexible null line break off stealing energy measurement
- Q channel energy pulses can be multiplexed into P2 energy pulses
- Random channel ADC sampling waveform buffer BUFFER
- Provide 3 ADC auto/manual temperature compensation function

## 23.2. Function description

Energy measurement unit contains three independent Σ-ΔADC and digital signal process part. Two of Three ADCs are used for current signal sampling and one for voltage signal sampling, digital signal process part for the calculation of active power and active energy, reactive power and reactive energy, apparent power and apparent energy, rms voltage, rms current and frequency etc.

Table correction parameter configuration and measurement parameter reading of digital signal process part can

be implemented though SFR register and interrupt; the measuring result is output by PF/QF/SF pin and table correction pulse output, which can compared to standard table directly for the error.

### 23.2.1. Analog-digital converter

HT502X contains 3 independent 2-order Σ-ΔADC; every ADC contains an analog gain amplifier, which contains a internal 1.2V reference voltage of high stability. Every ADC can work independently and configured by register EMUCTRL.

Analog gain amplifier enlarge the amplitude of input differential signal and then the ADC sample it and the linearity of measurement can be guaranteed if the input signal is the minimum. The analog amplification factor of three ADC can be configured independently through ADCCON (68H).

Digital gain of sampled signals of three ADC can be set through ADCCON(68H), the amplification factor can be 1,2,4,8. The significant digits of small signals can be added and higher accuracy can be achieved by employing digital gain on the condition that the large signal does not overflow.

Three ADC is closed by default and configure the correction table parameter register EMUCTRL to open or close the ADCs individually.

The second current channel ADC provide gain correct register I2GAIN(30H) which can adjust the magnitude of second current channel ADC for the energy stealing auto-prevention.

### 23.2.2. ADC sampling output and power waveform output

22bit waveform data, SPLI1 (00H)SPLI2(04H) andSPLU(08H), that Three ADC output of HT502X is accessed by users. And power waveform data of accumulated energy SPLP0CH) and SPLQ(10H) is accessed to users and the data sampled can be configured whether is filtered by high pass filter.

System clock frequency of EMU can be configured as Femu=819.2KHz and 409.6KHz through EMUCTRL and waveform data updating frequency can be 12.8KHz ,6.4KHz, 3.2KHz, 1.6KHz, 0.8KHz by configuring ADCCFG(50H SPL[2:0])

Note:since the highest waveform data updating frequency can be 12.8 KHz, the ADC_IRQ interrupt require can be 12.8 KHz at most. Users need to pay attention that whether there is plenty time for CPU to answer the interrupt.

### 23.2.3. Active power, reactive power and apparent power

HT502X can output two measurement channels of active power and reactive power at the same time and provide two independent table correction parameter and apparent power output register.

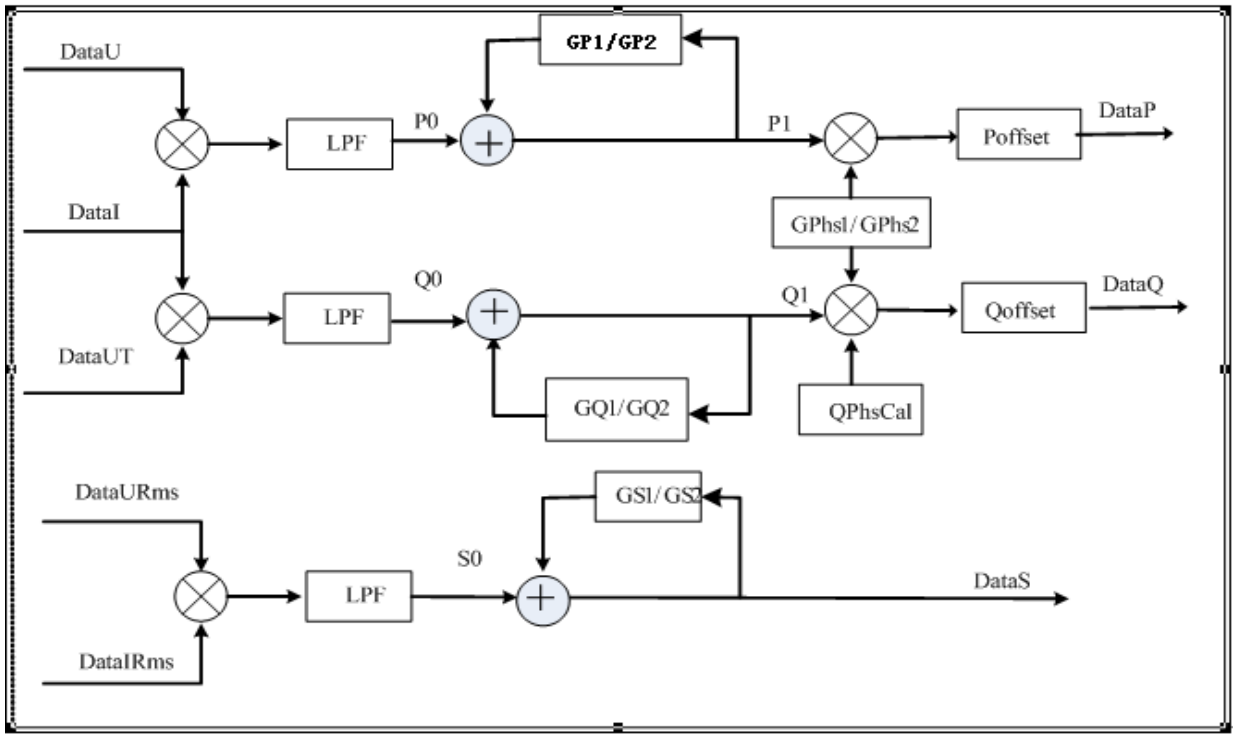



Fig22-2-1 power calculation and compensation

Active power is the product of voltage and current filtered by low pass filter.

Shift phase of voltage by 90de, current times it,the product which is filtered by low pass filter is reactive power.

Apparent power is the product of rms voltage and rms current.

Gain correction and phase correction is available for Power of channel1 and channel2 respective and bias correction is available for small signal to eliminate the disturbance and improve accuracy.

The performance of filter that shift the phase of voltage by 90de is related to input signal frequency and ADC sampling frequency, and the shift phase can be compensated by employing the reactive power phase compensation register QphsCal(2CH). QphsCal is 0xFF00 by default; Femu is 819.2 KHz which can achieve the 90de phase shifting of 50Hz input signal accurately without correction.

## 23.2.4. Root mean square

HT502X output rms of two channels of current and one channel of voltage.

The precision can be 0.5% at the dynamic range of 3000:1 by employing Rms.

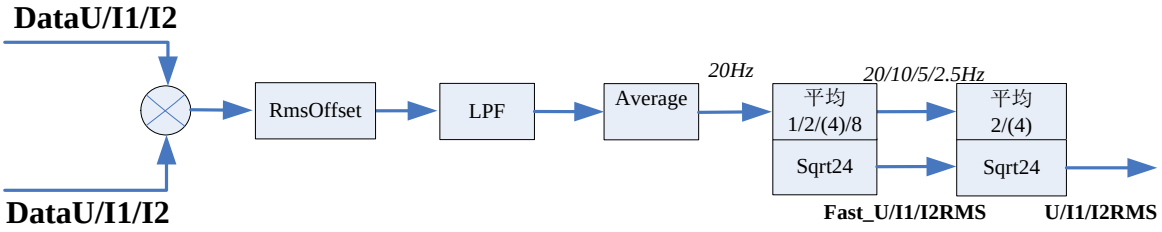


图 22-2-2

Rms register output is shown as above and user can select fast or slow rms as needed in normal mode(working clock of EMU is 819.2KHz/408.6kHz）.

When EMU is in low power measurement mode(working clock of EMU is 204.8KHz/32kHz）,Chip only provide fast rms and the updating time is 0.5s/1s/2s/4s which can be set through LpCtrl[1:0] of table correction register 94H ROSICTRL and the read value of slow rms register is always 0.

### 23.2.5. Voltage line frequency

By counting the zero crossing point of voltage sampling signal,the voltage frequency can be obtained.
The frequency accuracy can reach 0.01Hz conservatively.

### 23.2.6. Start/Creep

HT502X provides a flexible and creeping start function, by setting the P/Q/S Start calibration register (40H/44H/48H), the user can complete the function of start and creeping prevention. Users at the same time through the EMUCTRL.StartSel register (74H.Bit13) selection of creeping start mode.

**StartSel is not enabled (Default), according to the PQ / start creeping judgment alone:**
It does not output active pulse PF if |P| is smaller than PStart.
It does not output reactive pulse QF if |Q| is smaller than QStart.
It does not output apparent pulse SF if |S| is smaller than SStart.

**Enable StartSel, start / creeping according to P|Q joint judgment:**
1) When the |P| is greater than PStart or |Q| is greater than QStart two, the condition satisfies at least one, and the active pulses PF and QF all output pulses. The corresponding NOQLD1 and NOPLD1 flags are 0, or the NOQLD2 and NOPLD2 flags are 0.
2) When |P| is less than PStart and |Q| is less than QStart, PF and QF do not output pulses, the NOQLD1 and NOPLD1 flags are 1, or the NOQLD2 and NOPLD2 flags are 1.

There are two sets of power register,fast(24H~38H) and slow(48H~5CH),for HT502X internal measurement parameter register. Users can select the fast or slow power as judgement source by setting the control bit PQSStarScouceSel[1.0] of table correction register ADCCFG(50H). And choose the fast power as start power source, the dynamic load experiment will benefit from it, choose the slow power,it will improve the reliability of creeping.
Besides, NOQLD1 NOPLD1 NOQLD2 NOPLD2 of state register EMUSR(00H) can indicate whether two channels of energy is started which is convenient for users to select the threshold.

### 23.2.7. Power reverse indicate

Check the REVQ、REVP of state register EMUSR(00H) to find out whether the reactive power and active power is reversed.

REVQ、REVP will be updated on the PF and QF start edge synchronously.

### 23.2.8. Current bias correction

open or close the high pass filter of voltage and current by setting the HPFONU、HPFONI2、HPFONI1 of CHNLCR(54H). set these bits to 1 to open the high pass filter.

Close these high pass filter and dc signal will continue the subsequent calculation, sampled data will be dc bias corrected by set the bias correction register I1Off(34H)、I2Off(38H)、UOff(3CH).

Input channel should be short connected during dc bias correction. Write the average value of SPL_I1(00H)、SPL_I2(04H)、SPL_U(08H) of ADC sampling output waveform register to bias correction register I1Off(34H)、I2Off(38H)、UOff(3CH) to finish the dc bias correction.

### 23.2.9. Energy pulse output

HT502X offers active energy register ENERGY_P(60H), reactive energy register ENERGY_Q(64H) and apparent energy register ENERGY_S(68H), besides the corresponding pulse output pin PF, QF, SF is offered for correction.

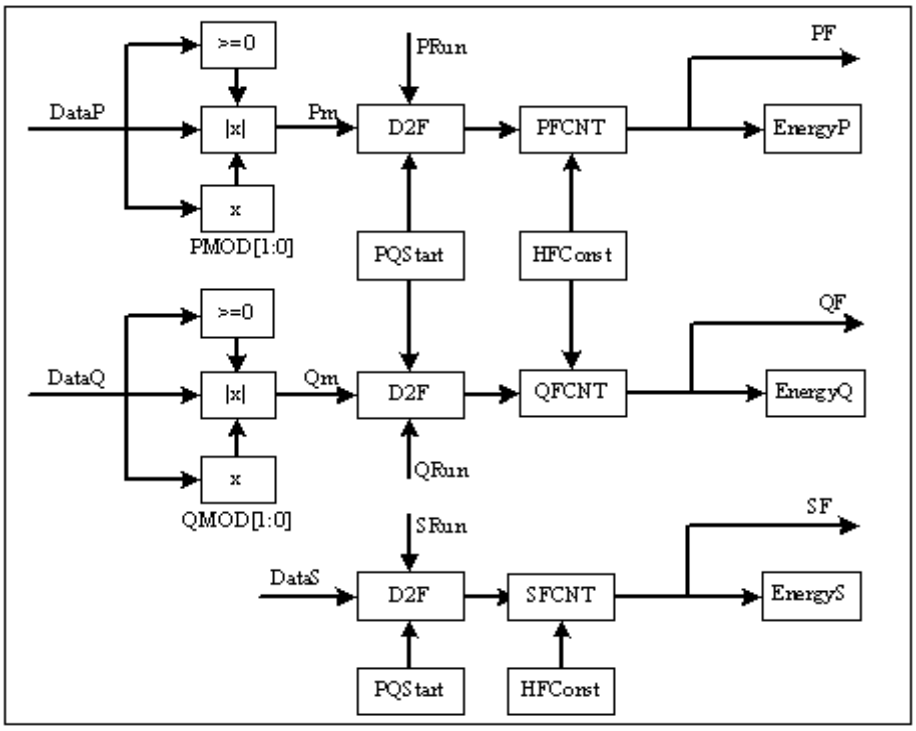



Fig 22-2-3 energy calculation and pulse output

Internal energy accumulation method can be configurated as positive calculation, absolute value calculation and algebraic addition calculation by setting QMOD、PMOD of EMCON(58H) before the pulse output stage.

Internal power value register accumulates the power and generate a overflow pulse for fast pulse register PFCNT(5CH)QFCNT(60H) and SFCNT(64H) every time the register overflows. Fast pulse register count the overflow times.when the absolute value of it is greater or equals to the set value of output pulse frequency set register HFConst(4CH), a CF pulse is generated and the corresponding energy register value increment by 1.

Fast pulse counter of HT502X, PFCNT/QFCNT/SFCNT, is accessible, the energy value which is not enough for a pulse in this register can be read and restored when power off and rewrite it ot the fast pulse counter register to prevent missing counting these energy.

Pulse output pins PF/QF/SF and energy register is control by the PRun/QRun/SRun of EMCON(58H) and PQStart(40H).

The active level can be set by the PF/QF/SF of POS（52H.6）. if the POS is 0, the active level is high;if it is 1 the active pulse level is low.

PF/QF/SF output the active power, reactive power and apparent power respectively by default and can be set by configurating the CFxCFG of EMCON(58H). and the output active level of PF/QF/SF can be configurated through control bit POS(54H.6).

The sequence of output of PF/QF/SF:

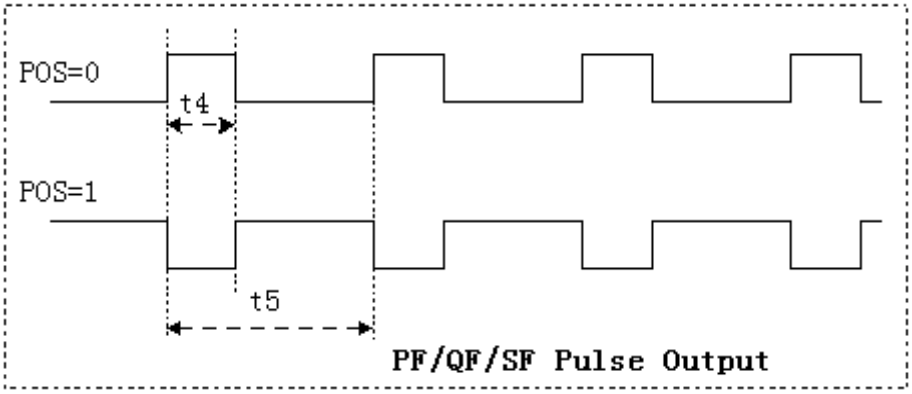



Fig 22-2-4 pulse output waveform

Sequence feature:

| parameter | index | unit | Test condition and note |
|---|---|---|---|
| t4 | 80 | ms | Pulse width of high level PF/QF/SF output |
| t5 | — | s | The period of the signal PF/QF/SF output |

Note:if the period of signal they output is less than 180ms,the duty of pulse they output will be 0.5.

The pulse width HT502X output can be modified by setting the CFP[1:0]( 68H.[7:6]).

## 23.2.10. Power Stealing detection

The two channel of current or power can be compared by the power stealing prevention module and the larger current or power will be selected for calculation.

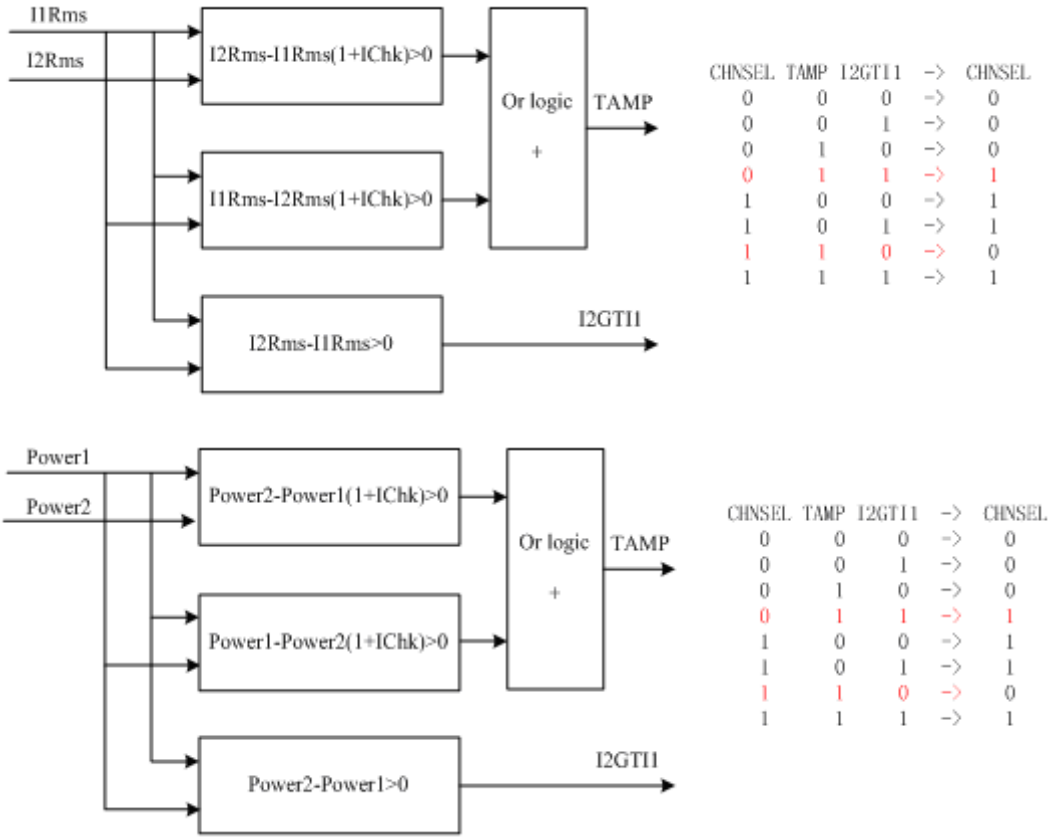



Fig 22-2-5 power stealing auto-prevention scheme

Configure the FLTON(50H.5) to decide whether to open the power stealing auto-prevention. User can choose the channel according to CHNSEL(50H.4) if FLTON=0; power stealing prevention unit automatically choose the channel for calculation according to the user setting if FLTON=1.

User can set the rate of power stealing prevention through ICHK(70H), for example. If it is set to be 0x10,the power stealing is thought to be experienced if the rms values differ by more than 6.25%.

If the rms current(or power) of two channels are both smaller than the IPRAMP(6CH), add register bit CHNFix(50H.8) can be configutated as :

（1）Select the channel1 as the calculation unit by default

（2）Not switch,select the previous calculation channel

0 of bit I2GT1(00H.6) means that I1 is greater than I2 while 1 means that I2 is greater than I1

1 of bit TAMP(00H.7) means that the the power stealing is be committed,namely two channels of current differs by more than the threshold.

**Power stealing auto-prevention setting steps: (take the current stealing as example)**

a) Open the current channel of 2ADC through EMUCTRL(74H.5)

b) Correct the output of current channel2 through I2GAIN (30H) to make sure that output rms values are the same when the input currents are the same.

c) Set ICHK(70H) as the need of power stealing prevention threshold.

d) Check the minimum current or power of power stealing prevention as needed and Set IPTAMP(6CH)

e) Set the FLRON(0x50H.5) to 1 to start the power stealing auto prevention function .

After start the power stealing auto prevention, CHNSEL and CIADD is read-only, and the channel is selected according to the result of power stealing prevention and the power stealing prevention state is indicated by the register bit CHNSEL/TAMP/I2GTI1.

## 23.2.11. Interrupt source

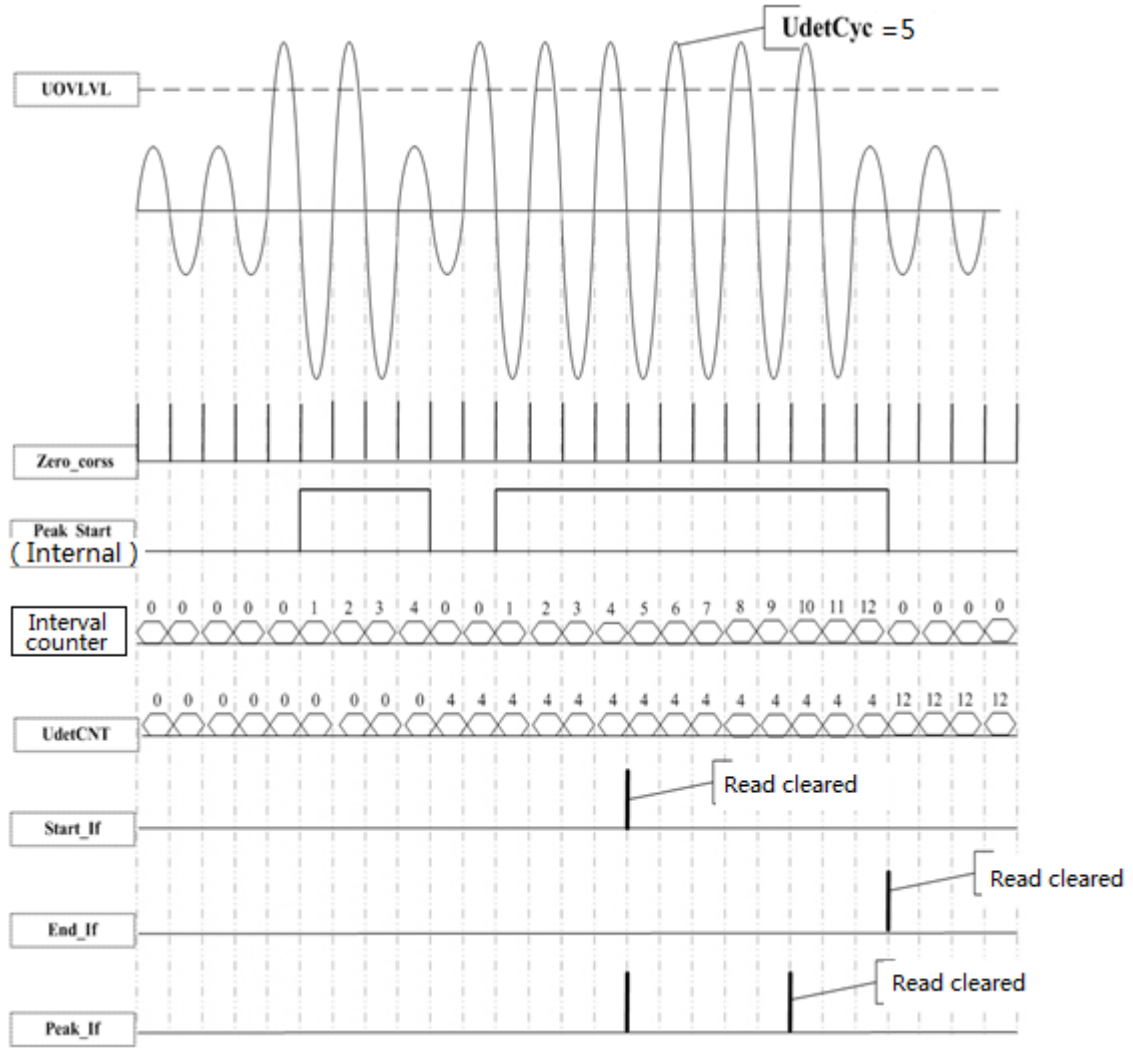
EMU of HT502X offers 17 interrupt sources and are defined in the EMU interrupt enable register (EMUIE,04H):ZXlost_IRQ, Buffer_Full_IRQ, Tamp_IRQ, DCUpdate_IRQ, SlowRmsUpdate_IRQ, FastRmsUpdate_IRQ, ZXI2_IRQ, ZXI1_IRQ, I2OV_IRQ, PF_IRQ, QF_IRQ, SF_IRQ, SPL_IRQ, ZXU_IRQ, I1OV_IRQ, UOV_IRQ and SAG_IRQ which stands for Zero voltage loss, interrupt request, synchronous cache data, full interrupt request, power stealing interrupt request, dc rms update interrupt request, slow rms update interrupt request, fast rms update interrupt request, I2 zero-crossing interrupt request, I1 zero-crossing interrupt request, I2 overflow interrupt request, active power pulse interrupt request, reactive power pulse interrupt request, apparent power pulse interrupt request, ADC original data update interrupt request, voltage zero-crossing interrupt request, I1 overflow interrupt request, over-voltage interrupt request and voltage sags interrupt request. The zero-crossing ways can be configurated to be positive and negative and bidirectional zero crossing method through ZXD1 and ZXD0.

The 17 interrupts of EMS share the one interrupt vector IRQ_EMU. Users manage the interrupt by setting the EMU interrupt enable register EMUIE and EMU interrupt flag register EMUIF.

The flag can be cleared after EMUIF is readed.

## 23.2.12. PEAK/SAG function

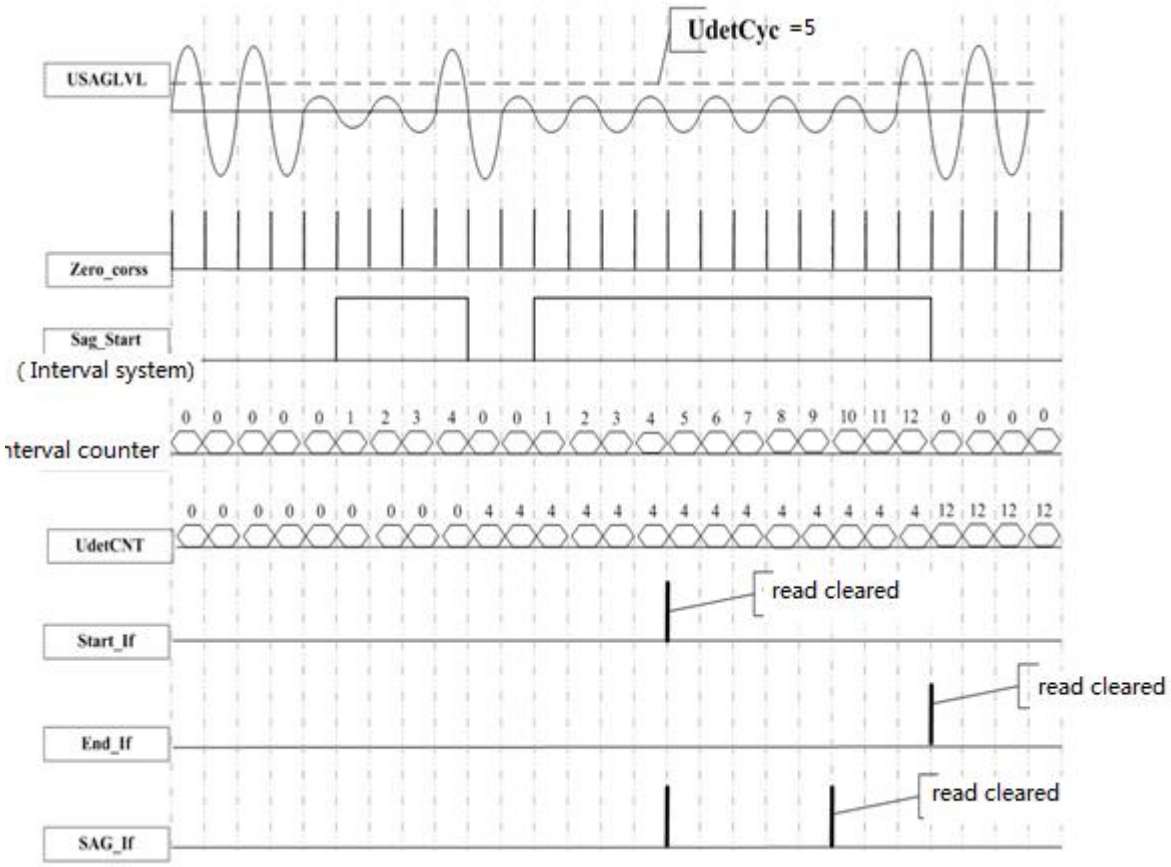
**The PEAK event is defined as:** the system detects the peak value of the voltage channel after the high pass, and detects the peak value of the half cycle by positive and negatives zero crossing, while updating the peak to the Upd_Half register (0x88). When a Upd_Half is greater than the absolute value of the UOVLVL register (0xAC) PEAK threshold, we defined for the Peak_Start event, the counter inside the system began to count the Peak_Start event, until a Upd_Half absolute value less than the threshold value, will be updated to UdetCNT, while the internal counter is cleared. If the internal counter is greater than or equal to the number of period OVCyc is set, Start_If mark position 1 (reading, at the same time zero) Peak_If mark position 1 (after reset), until a Upd_Half absolute value less than the threshold, will be updated to UdetCNT internal counter counts, zero, the Start_If flag to 0, while the Peak_If flag 0, End_If is 1 (after reset). The PEAK event sequence diagram is shown below:




Peak event

**The SAG event is defined as:** the system detects the peak value of the voltage channel after the high pass, and detects the peak value of the half cycle by positive and negatives zero crossing, while updating the peak to the Upd_Half register (0x88). When an Upd_Half absolute value is less than the USAGLVL register (0xA4) PEAK threshold, we defined for the Sag_Start event, the counter inside the system began to count the Sag_start event, until a Upd_Half absolute value is greater than the set threshold, the counting value is updated to UdetCNT, while the internal counter is cleared. If the internal counter is greater than or equal to the set number of cycles (OVCyc), Start_If (1 to mark a position, at the same time reading clear) Sag_If mark position 1 (after reset), until a Upd_Half absolute value is greater than the set threshold, the count is updated to UdetCNT, the internal counter is reset, Start_If mark position 0, and the Sag_If logo position 0, End_If is 1 (after reset).

Among them, the voltage channel SAG event and the PEAK event share the UdetCNT, Start_If, and End_If registers. The SAG event sequence diagram is shown below:




SAG Event

## 23.2.13. OVI function description

Over-voltage and over-current detection(OVI) is taken place in the normal calculation, which belonds to th erea of energy quality, the data it employed is filtered by high pass filter, if the high pass filter is closed, the data ADCOFFSET output will replace it.

The system checks the ADC sampling absolute value every half period(2 zero-crossing points) to find the maximum value. The basis for judgement of OVI event: the high 16-bit of peak value of voltage or current waveform sampling absolute value is greater than the set value of voltage or current peak value detection threshold set register UOVLVL(ACH) or IOVLVL(B4H) and it last for the period that equals to the set num of half period in OVCyc width set register OVCyc（B0H). The OVI function is closed when OVCyc=0x0000.

Table correction register:

1）Observe width register OVCyc（B0H）:In unit of half period

2）Threshold set register UOVLVL(ACH) or IOVLVL(B4H): The set compare threshold,aligned to high 16-bit of ADC sampling value

3）OVI interrupt enable bits of UOVIE、I1OVIE、I2OVIE（04H）: Set it to 1 to enable over-voltage and over-current interrupt output.

4) OVI status flag UOVIF、I1OVIF、I2OVIF（08H）: Read 1 indicates overpressure event, overflow event, read

after 0.

5）Voltage waveform point peak value UPeak（88H）、I1Peak（8CH）、I2Peak（90H）:the maximum peak value of waveform point, signed num, 24-bit register, Bit21-Bit23 for sign.

Besides no special enable bit for this function, remains the working state after reset, the voltage and current waveform point filtered by high pass filter is data source.

### 23.2.14. Loss of Voltage calculation mode

Take both calculation accuracy and power consumption into consideration after entering HT502X loss of voltage mode, several methods for loss of null line calculation are offered.

1. The apparent power is the product of rms current and register UCONST instead of rms voltage, apparent energy, the active energy output pin PF output pulse signal.

Following register need to be configurated in loss of voltage mode:

（1） UCONST：the constant replace the rms voltage

（2） HFCONST：for the accuracy correction of apparent energy pulse output at 100%Ib point(coarse turning)

（3） SGAIN：for the accuracy correction of apparent energy pulse output at 100%Ib point(fine turning)

（4） ADCx_EN：select which current channel to open(control the power consumption in loss of voltage mode)

（5） Channel_Sel：choose the current channel for energy calculation

（6） SRUN：open the apparent energy calculation

（7） PSSel：configurate it that active energy pulse pin PF output the apparent energy pulse

2. Low power calculation mode of internal chip,EMU clock can be configurated as 204.8KHz or 32KHz, see application notes for detailed information.

3. Custom constant calculation mode is supported for chips,if user write to constant power register and enable constant calculation,this constant power will be accumulated for pulse generation and energy will be accumulated.

## 23.2.15. ADC waveform cache with following frequency sampling

HT502X built in 128*16bit with the frequency sampling, ADC waveform cache buffer, used to store the ADC synchronous sampling data with frequency changes, for users to do further analysis. By configuring the calibration parameter register BufferStart (10CH) need to specify the channel and gain the cache, and the calibration parameters with BufferCoff (110H) to adjust the sampling rate, will save the corresponding ADC waveform data with frequency sampling Buffer cache, the write pointer is automatically incremented by 1, until the cache is full until buffer. Full storage may generate interrupt flags, and the Buffer_Full IE setting of the EMUIE registers enables full interruption of the Buffer function. Before the user sending a new cache command, the cached data will remain on a data cache; if not full, again sent command command will immediately start caching, execution, and from the cache the address of the buff 0000H to save the data. The following cache function is not used in low power metering mode.

**Cache data manual adjustment coefficient:** the adjustment coefficient of BufferCoff users can register

through the manual calibration (110H) method, the 128*16bit frequency with cache data adjusted to two complete signals Zhou Bo, i.e. 64 per Zhou Bo point. Input signal range 45Hz~65Hz.

**Read data with the following frequency buffer:** after the Buffer data is full, the user can extract data by reading the metering register R_Buffer (ACH). Once read this register, the internal cache address pointer is automatically added 1 until the value of the cache Buffer is fully read out. It is recommended that the user have to wait until the Buffer is full before you start the waveform caching function (full of interrupt flags). If the read operation is carried out in the process of keeping in storing up data to the Buffer, the data read out may go wrong (represented by first errors).

**Cache data format:** 16bit ADC data is the complement format, the actual bit Bit12-Bit0 is data, and Bit15-Bit13 is the 3 symbol bit. Therefore, the signal size is Buffer / 2^13 / PGA, and the number of bits is independent of PGA.

## 23.2.16. Multiplexing of Q energy pulse channels P2

HT502X's reactive power module can be configured for second channel metering with functional pulse output. After the configuration register EMUCTRL (74H.BIT11) P2EN is enabled, the active energy P2 of the second metering channel replaces the QFCNT energy pulse output, thereby realizing the active output of both channels at the same time.

# 23.3. Special function register list

## 23.3.1. Calculation parameter register

| | **EMU module calculation register Base address:<br>Calculation register：0x40013000** | | | | |
|---|---|---|---|---|---|
| **offset address** | **Name** | **Write/ read** | **Significant word length** | **Reset value** | **function description** |
| 00H | SPLI1 | R | 3 | 0x000000 | ADC sampled data of current channel1 |
| 04H | SPLI2 | R | 3 | 0x000000 | ADC sampled data of current channel2 |
| 08H | SPLU | R | 3 | 0x000000 | ADC sampled data of voltage channel |
| 0CH | SPLP | R | 4 | 0x00000000 | Waveform data of active power |
| 10H | SPLQ | R | 4 | 0x00000000 | Waveform data of reactive power |
| 14H | FastRMSI1 | R | 3 | 0x000000 | Rms value of fast current channel1 |
| 18H | FastRMSI2 | R | 3 | 0x000000 | Rms value of fast current channel2 |
| 1CH | FastRMSU | R | 3 | 0x000000 | Rms value of fast voltage channel |
| 20H | FreqU | R | 2 | 0x0000 | Voltage frequency |
| 24H | FastPowerP1 | R | 4 | 0x00000000 | Active power of first fast channel |
| 28H | FastPowerQ1 | R | 4 | 0x00000000 | reactive power of first fast channel |

| | | | | | |
|---|---|---|---|---|---|
| 2CH | FastPowerP2 | R | 4 | 0x00000000 | active power of second fast channel |
| 30H | FastPowerQ2 | R | 4 | 0x00000000 | reactive power of second fast channel |
| 34H | FastPowerS1 | R | 4 | 0x00000000 | apparent power of first fast channel |
| 38H | FastPowerS2 | R | 4 | 0x00000000 | apparent power of second fast channel |
| 3CH | RMSI1 | R | 3 | 0x000000 | Rms value of slow current channel1 |
| 40H | RMSI2 | R | 3 | 0x000000 | Rms value of slow current channel2 |
| 44H | RMSU | R | 3 | 0x000000 | Rms value of slow voltage channel |
| 48H | PowerP1 | R | 4 | 0x00000000 | active power of first slow channel |
| 4CH | PowerQ1 | R | 4 | 0x00000000 | reactive power of first slow channel |
| 50H | PowerP2 | R | 4 | 0x00000000 | active power of second slow channel |
| 54H | PowerQ2 | R | 4 | 0x00000000 | reactive power of second slow channel |
| 58H | PowerS1 | R | 4 | 0x00000000 | apparent power of first slow channel |
| 5CH | PowerS2 | R | 4 | 0x00000000 | apparent power of second slow channel |
| 60H | EnergyP | R | 3 | 0x000000 | Active energy |
| 64H | EnergyQ | R | 3 | 0x000000 | Reactive energy |
| 68H | EnergyS | R | 3 | 0x000000 | Apparent energy |
| 6CH | EnergyPC | R | 3 | 0x000000 | Active energy who will be cleared after readed |
| 70H | EnergyQC | R | 3 | 0x000000 | reactive energy who will be cleared after readed |
| 74H | EnergySC | R | 3 | 0x000000 | apparent energy who will be cleared after readed |
| 78H | DC_UAverage | R | 4 | 0x00000000 | DC voltage average value |
| 7CH | DC_I1Average | R | 4 | 0x00000000 | DC current average value of channel1 |
| 80H | DC_I2Average | R | 4 | 0x00000000 | DC current of average value channel2 |
| 84H | Checksum | R | 3 | 0x000000 | Checksum register, to ensure that the table correction register is not modified. |
| 88H | Upd_Half | R | 3 | 0x000000 | U half wave peak value register, 19bit. |
| 8CH | I1pd_Half | R | 3 | 0x000000 | I1 half wave peak value register, 19bit. |
| 90H | I2pd_Half | R | 3 | 0x000000 | I2 half wave peak value register, 19bit. |
| 94H | PFCNT_Photo | R | 2 | 0x0000 | PFCNT snapshot |
| 98H | QFCNT_Photo | R | 2 | 0x0000 | QFCNT snapshot |
| 9CH | SFCNT_Photo | R | 2 | 0x0000 | SFCNT snapshot |
| A0H | AutoUgain | R | 2 | 0x0000 | Utemperature auto-compensation coefficient of channel |
| A4H | AutoI1gain | R | 2 | 0x0000 | I1 temperature auto-compensation coefficient of channel |
| A8H | AutoI2gain | R | 2 | 0x0000 | I2 temperature auto-compensation coefficient of channel |
| ACH | R_Buffer | R | 4 | 0x000000 | ADC buffer memory read data register |
| B0H | Reserved | R | 4 | 0x000000 | RESERVED |
| B4H | Reserved | R | 4 | 0x000000 | RESERVED |

| | | | | | |
|---|---|---|---|---|---|
| B8H | UdetCNT | R | 3 | 0x000000 | SAG/Peak Condition duration count register |

### 22.3.1.2 Calucation register introduction

| **SPLI1** | | | **Base address: 0x40013000**<br>**offset address: 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **SPLI2** | | | **Base address: 0x40013000**<br>**offset address: 04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **SPLU** | | | **Base address: 0x40013000**<br>**offset address: 08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

The update frequency of waveform register is determined by 3 bits of clock configuration register FreqCFG[2:0].the significant bit of these 3 registers are all 22 bits,bit21 for sign and extended to 24 bits meanwhile which means the bit23-bit21 of read data from thin register are sign bits. ADC waveform register can be set to receive data if it was filtered by high pass filter through ADCCFG(50H.WaveSel) .the data is in binary complement format.

| **SPLP** | | **Base address: 0x40013000**<br>**offset address: 0CH** |
|---|---|---|
| | **Bit31…Bit0** | |
| **Read:** | DAT[31:0] | |
| **Write:** | X | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **SPLQ** | | | **Base address:** 0x40013000<br>**offset address:** 10H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | DAT[31:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

The update frequency of power waveform register is determined by 3 bits of clock configuration register FreqCFG[2:0].the significant bit of these 2 registers are all 32 bits,bit31 for sign. ADC waveform register can be set to receive data if it was filtered by high pass filter through ADCCFG(50H.WaveSel) .the data is in binary complement format.

| **FastRMSI1** | | | **Base address:** 0x40013000<br>**offset address:** 14H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastRMSI2** | | | **Base address:** 0x40013000<br>**offset address:** 18H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastRMSU** | | | **Base address:** 0x40013000<br>**offset address:** 1CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Fast Rms is 24-bit unsigned num,the most significant bit(MSB) is always 0. If the clock frequency of EMU is 819.2 KHz and the parameter update frequency is 5Hz by default, EMUCTRL[1:0] can be configurated to be the maximum frequency 20Hz.

If more accurate rms register value is required while it is small signal, the the rms value should be zero drift corrected through registers I1RMSOFFSET and I2RMSOFFSET.

| **FreqU** | | | **Base address: 0x40013000**<br>**offset address: 20H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Frequency value is a 16-bit unsigned num, the parameter formatting formula is:

$$Frequency = \frac{femu}{(UFREQ \times 2)}$$

Where:

Femu is operation frequency of calculation mudule which is 819.2 KHz by default.

For example:if clock frequency of EMU(femu) is 819.2KHz and register UFREQ=8192, the measured actual frequency is : Frequency=819.2KHz/2/8192=50Hz.

| **FastPowerP1** | | | **Base address: 0x40013000**<br>**offset address: 24H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastPowerQ1** | | | **Base address: 0x40013000**<br>**offset address: 28H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastPowerP2** | | | **Base address: 0x40013000**<br>**offset address: 2CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastPowerQ2** | | | **Base address:** 0x40013000<br>**offset address:** 30H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastPowerS1** | | | **Base address:** 0x40013000<br>**offset address:** 34H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **FastPowerS2** | | | **Base address:** 0x40013000<br>**offset address:** 38H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fast power for 32 bit binary format, the highest bit is the sign bit, fast power register parameters update the default frequency of 5Hz (EMU clock frequency of 819.21KHz, configuration of EMU clock is 409.6KHz or 2.5Hz), can be configured by EMUCTRL[1:0] to a maximum of 20Hz.

The data in the register is FastPowerP1, and the Preg for calculation is:

Preg＝FastPowerP1 ; if FastPowerP1<2^31

Preg=FastPowerP1-2^32 ; if FastPowerP1>=2^31

The conversion ratio coefficients used for display are described in detail in the slow power register,

| **RMSI1** | | | **Base address:** 0x40013000<br>**offset address:** 3CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **RMSI2** | | | **Base address:** 0x40013000<br>**offset address:** 40H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **RMSU** | | | **Base address: 0x40013000**<br>**offset address: 44H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | DAT[23:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Slow rms is a 24-bit unsigned num, the MSB is always 0. If the clock frequency of EMU is 819.2 KHz,the parameter update frequency is 1.25Hz by default. Configurate EMUCTRL[7] to adjust the update frequency and can correspond to 1/2 or 1/4 update frequency of fast rms.

If more accurate rms register value is required while it is small signal, the the rms value should be zero drift corrected through registers I1RMSOFFSET and I2RMSOFFSET.

| **PowerP1** | | | **Base address: 0x40013000**<br>**offset address: 48H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PowerQ1** | | | **Base address: 0x40013000**<br>**offset address: 4CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PowerP2** | | | **Base address: 0x40013000**<br>**offset address: 50H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PowerQ2** | | | **Base address: 0x40013000**<br>**offset address: 54H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PowerS1** | | | **Base address:** 0x40013000<br>**offset address:** 58H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PowerS2** | | | **Base address:** 0x40013000<br>**offset address:** 5CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit31…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Fast/slow power are all in 32-bit binary complement format, MSB for sign, update frequency of fast power register parameter is 5Hz by default(EMU clock frequency is 819.2KHz and it is 2.5Hz when EMU clock frequency is 409.6KHz), it can be configurated to be the maximum frequency 20Hz by configurating EMUCTRL[1:0].

Update frequency of slow power register parameter is 1.25Hz by default(EMU clock frequency is 819.2KHz and it is 0.625Hz when EMU clock frequency is 409.6KHz), Comfigurate EMUCTRL[7] to adjust the update frequency and can correspond to 1/2 or 1/4 update frequency of fast rms.

Power parameter of two channels FastPowerP1/P2、FastPowerQ1/Q2、FastPowerS1/S2、PowerP1/P2、PowerQ1/P2、PowerS1/S2 are in 32-bit binary complement format,and the MSB for sign.

Set the data in register is PowerP1, the Preg for calculation is:

Preg＝PowerP1 ； if PowerP1<2^31

Preg=PowerP1-2^32 ； if PowerP1>=2^31

Set the active power we know is P, the conversion coefficient is Kpqs,and:

P＝Preg×Kpqs

Kpqs is the ratio of rated power to PowerP1 readed value if input reteds active power.

The coefficient is same with active power coefficient Kpqs when reactive power and apparent power is dislayed.

For example:

Input active power is 1000w and the readed average value of PowerP1 is 0x00C9D9(51673),then:

Kpqs＝1000/51673=0.01935

If PowerP1 reading is 0xFFFF4534, the power is：

P=Kpqs*Preg=0.01935*(-47820)＝ -925.3 w

Where Preg＝PowerP1－2^32= -47820

| **EnergyP** | **Base address:** 0x40013000 |
|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | **offset address：　60H** | | | | | |
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[18:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **EnergyQ** | | | **Base address：　0x40013000**<br>**offset address：　64H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[18:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **EnergyS** | | | **Base address：　0x40013000**<br>**offset address：　68H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The EnergyP / EnergyQ / EnergyS energy accumulation register is read not clear. Type 0 energy register, the smallest unit of the register, represents the energy of 1/EC kWh.

Example:

When the pulse constant EC is 3200imp/kWh and the register reading is 0x001000 (4096), the energy represented is:

E = 4096/3200 = 1.28 kWh

| **EnergyPC** | | | **Base address：　0x40013000**<br>**offset address：　6CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **EnergyQC** | | | **Base address：　0x40013000**<br>**offset address：　70H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **EnergySC** | **Base address：　0x40013000** |
|---|---|

| | | | offset address： 74H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit23…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

EnergyP / EnergyQ / EnergySenergy accumulation register will be cleared after read, the minimum unit of this register is 1/EC kWh.

For example:

If pulse constant EC is 3200imp/kWh and the register reading is 0x001000（4096）, the energy it stands for:

E＝4096/3200＝1.28 kWh

| **DC_UAverage** | | | **Base address： 0x40013000**<br>**offset address： 78H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **DC_I1Average** | | | **Base address： 0x40013000**<br>**offset address： 7CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **DC_I2Average** | | | **Base address： 0x40013000**<br>**offset address： 80H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit31…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[31:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

DC average value DC_Average is 32-bit num.the parameter update frequency is 8 Hz by default if the EMU clock frequency is 819.2KHz. configurate the EMUCTRL [9:8]（74H） to adjust the update frequency to 1,2,4,8Hz.

If more accurate DC average value is required while it is small signal, the the DC average value should be zero drift corrected through registers Uoff I1off and I2off.

| **Checksum** | **Base address： 0x40013000**<br>**offset address： 84H** |
|---|---|

| | Bit23…Bit0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Parameter sum check register is the sum of all table correction parameter register, Base address0x40013800，offset address0CH---104H(exclude register 5CH-64H from calculation).

Scheck register calculation ways:

All check register will be added up after converted to 3-byte unsigned num. the high bits of single-byte/two-byte register will be filled with zero.

If table correction register is configurated, this register will be updated immediately after waiting for about 2us which is negligible.

| Upd_Half | | | **Base address：** **0x40013000**<br>**offset address：** **88H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit24…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[24:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| I1pd_Half | | | **Base address：** **0x40013000**<br>**offset address：** **8CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit24…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[24:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| I2pd_Half | | | **Base address：** **0x40013000**<br>**offset address：** **90H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit24…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[24:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Peak value register, related to OVI function, the register will be updated after the set num of ADC half periods to obtain the maximum value(absolute value) of ADC waveform during this time. The data of this register is filtered by high pass filter and to bits align with ADC, this register is 22 bits. Bit21 for sign and it can be extended to 24bit which means bit23-bit21 of the register reading is sign bits.data is in binary complement format.

| **PFCNT_Photo** | **Base address：** **0x40013000**<br>**offset address：** **94H** |
|---|---|

| | Bit15…Bit0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **QFCNT_Photo** | **Base address: 0x40013000**<br>**offset address: 98H** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **SFCNT_Photo** | **Base address: 0x40013000**<br>**offset address: 9CH** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | DAT[15:0] | | | | | | | |
| **Write:** | X | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Fast energy pulse snapshot register, when users read the active energy register(without clearing energy register/clearing energy register), the pfcnt will be write to PFCNT_Photo register in 5 EMU CLK to improve the resolution of energy register.

For example: if the PFCNT_Photo reading is 0x20 and Hfconst register reading is 0x40, the energy PFCNT_Photo stands for is :

0x20 / 0x40）/ EC kWh

QFCNT_Photo work the same with SFCNT_Photo.

| **AutoUgain** | **Base address: 0x40012000**<br>**offset address: A0H** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

If the temperature auto-compensation function is enabled,the value of this register meas the auto compensation value of U channel at different temperature point.

| **AutoI1gain** | **Base address: 0x40012000**<br>**offset address: A4H** |
|---|---|
| | **Bit15…Bit0** |
| **Read:** | X |

| Write: | DAT[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

If tempertature auto-compensation is enabled, the value of this register meas the auto compensation value of I1 channel at different temperature point.

| AutoI2gain | | | Base address: 0x40012000<br>offset address: A8H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: | X | | | | | | | |
| Write: | DAT[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

If temperature auto-compensation is enabled, the value of this register meas the auto compensation value of I2 channel at different temperature point.

| R_Buffer | | | Base address: 0x40012000<br>offset address: ACH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: | X | | | | | | | |
| Write: | DAT[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The synchronous waveform data cache register, whose data is valid only after the cache is started. Cache register startup: the table register BufferStar (0100H) is written to 0xCCCX. ADC synchronous sampling, data storage, cache completion, interrupt flag (Buffer_FullIF setting), the user can read R_Buffer, access to cache buff data.

R_Buffer stores the first ADC waveform synchronized cache data, each read R_Buffer, the internal counter automatically updated to the next ADC waveform data address. When the number of reads reaches 128, or the ADC waveform data cache is restarted, R_Buffer will return to the first address of the cached buff.

| Reserved | | | Base address: 0x40012000<br>offset address: B0H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: | X | | | | | | | |
| Write: | DAT[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Reserved | Base address: 0x40012000<br>offset address: B4H |
|---|---|

| | Bit15…Bit0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | X | | | | | | | |
| **Write:** | DAT[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **UdetCNT** | | | **Base address: 0x40012000**<br>**offset address: B8H** | | | | |
|---|---|---|---|---|---|---|---|
| | Bit23…Bit0 | | | | | | | |
| **Read:** | X | | | | | | | |
| **Write:** | DAT[23:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SAG/PEAK Condition duration count register, if enable the SAG/PEAK function of U channel, and the SAG/PEAK events was detected and started, UdetCnt will counter by half wave, untile the SAG/PEAK event is detected, the UdetCNT register value is updated.

This register records the number of half waves for SAG/PEAK continued.

## 23.3.2. Table correction parameter register

| | **EMU table correction register Base address:**<br>**Table correction register: 0x40013800** | | | | |
|---|---|---|---|---|---|
| **offset address** | **name** | **Write /read** | **Valid length** | **Reset value** | **function description** |
| 00H | EMUSR | R/W | 2 | 0x0000 | EMU state flag register |
| 04H | EMUIE | R/W | 2 | 0x0000 | EMU interrupt enable register |
| 08H | EMUIF | R/W | 2 | 0x0000 | EMU interrupt flag register |
| 0CH | GP1 | R/W | 2 | 0x0000 | Active power correction of channel1 |
| 10H | RSRV | R/W | 2 | 0x0000 | --- |
| 14H | RSRV | R/W | 2 | 0x0000 | --- |
| 18H | GPhs1 | R/W | 2 | 0x0000 | Phase correction of channel1 |
| 1CH | GP2 | R/W | 2 | 0x0000 | Active power correction of channel2 |
| 20H | RSRV | R/W | 2 | 0x0000 | --- |
| 24H | RSRV | R/W | 2 | 0x0000 | --- |
| 28H | GPhs2 | R/W | 2 | 0x0000 | Phase correction of channel2 |
| 2CH | QPhsCal | R/W | 2 | 0xFF00 | Reactive power phase compensation |
| 30H | I2Gain | R/W | 2 | 0x0000 | Gain compensation of current channel2 |
| 34H | I1Off | R/W | 2 | 0x0000 | DC offset correction of current channel1 |
| 38H | I2Off | R/W | 2 | 0x0000 | DC offset correction of current channel2 |

| | | | | | |
|---|---|---|---|---|---|
| 3CH | UOff | R/W | 2 | 0x0000 | DC offset correction of voltage channel |
| 40H | PStart | R/W | 2 | 0x0040 | Start power set,16-bit unsigned num, if it is greater than absolute value of high 24 bits of 32-bit active power register value,accumulate the active power |
| 44H | QStart | R/W | 2 | 0x0080 | Start power set,16-bit unsigned num, if it is greater than absolute value of high 24 bits of 32-bit reactive power register value,accumulate the reactive power |
| 48H | SStart | R/W | 2 | 0x0090 | Start power set,16-bit unsigned num, if it is greater than absolute value of high 24 bits of 32-bit apparent power register value,accumulate the apparent power |
| 4CH | HFConst | R/W | 2 | 0x0080 | Output pulse frequency set |
| 50H | ADCCFG | R/W | 2 | 0x0003 | ADC control register |
| 54H | CHNLCR | R/W | 2 | 0x0607 | Channel control register |
| 58H | EMCON | R/W | 2 | 0x**1870** | Energy calculation control register |
| 5CH | PFCnt | R/W | 2 | 0x0000 | Fast active power pulse counting |
| 60H | QFCnt | R/W | 2 | 0x0000 | Fast reactive power pulse counting |
| 64H | SFCnt | R/W | 2 | 0x0000 | Fast apparent power pulse counting |
| 68H | ADCCON | R/W | 2 | 0x0000 | ADC channel gain selection(including digital and analog gain) |
| 6CH | IPTAMP | R/W | 2 | 0x0020 | Power stealing detection threshold |
| 70H | ICHK | R/W | 2 | 0x0010 | Power stealing detection threshold |
| 74H | EMUCTRL | R/W | 2 | 0x0382 | EMU control register |
| 78H | P1OFFSET | R/W | 2 | 0x0000 | Active power small signal offset correction of channel1 |
| 7CH | P2OFFSET | R/W | 2 | 0x0000 | Active power small signal offset correction of channel2 |
| 80H | Q1OFFSET | R/W | 2 | 0x0000 | Reactive power small signal offset correction of channel1 |
| 84H | Q2OFFSET | R/W | 2 | 0x0000 | Reactive power small signal offset correction of channel2 |
| 88H | I1RMSOFFSET | R/W | 2 | 0x0000 | Current channel1 Rms small signal correction register |
| 8CH | I2RMSOFFSET | R/W | 2 | 0x0000 | Current channel2 rms small signal correction register |
| 90H | URMSOFFSET | R/W | 2 | 0x0000 | Voltage channel rms small signal correction register |
| 94H | ROSICTRL | R/W | 2 | 0x4000 | Rogowski coil enable control bit |
| 98H | ANA_control | R/W | 2 | 0x27A6 | Analog control register |
| 9CH | UCONST | R/W | 2 | 0x0000 | Voltage that involve calculation when voltage |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | loss,open phase to prevent power stealing. |
| A0H | LpIdleTime | R/W | 2 | 0x0000 | In low power calculation mode, time share mode idle time set register |
| A4H | USAGLVL | R/W | 2 | 0x0000 | Voltage sag detection threshold register |
| A8H | IpeakCyc | R/W | 2 | 0x0020 | Current channels of I1 I2 PEAK check period register |
| ACH | UOVLVL | R/W | 2 | 0x0000 | Voltage peak value detection threshold register |
| B0H | OvCyc | R/W | | 0x0001 | Peak value detection ADC half period num set register |
| B4H | IOVLVL | R/W | 2 | 0x0000 | Current peak value threshold register |
| B8H | ZXILVL | R/W | | 0x0000 | Current zero-crossing threshold register |
| BCH | PDataCpH | R/W | 2 | 0x0000 | High 16 bits of constant active power register,the constant calculation will start after writing the high and low 16 bits and writing 0x00BC to LoadDataCp |
| C0H | PDataCpL | R/W | 2 | 0x0000 | Low 16 bits of constant active power register |
| C4H | QDataCpH | R/W | 2 | 0x0000 | High 16 bits of constant reactive power register, the constant calculation will start after writing the high and low 16 bits and writing 0x00BC to LoadDataCp |
| C8H | QDataCpL | R/W | 2 | 0x0000 | Low 16 bits of constant reactive power register |
| CCH | SDataCpH | R/W | 2 | 0x0000 | High 16 bits of constant apparent power register, the constant calculation will start after writing the high and low 16 bits and writing 0x00BC to LoadDataCp |
| D0H | SDataCpL | R/W | 2 | 0x0000 | Low 16 bits of constant apparent power register |
| D4H | FilterCtrl | R/W | 2 | 0x02F7 | Filter coefficient selection control |
| D8H | TUgain | R/W | 2 | 0x0000 | Temperature manual compensation coefficient of voltage channel |
| DCH | TI1gain | R/W | 2 | 0x0000 | Temperature manual compensation coefficient of current I1 channel |
| E0H | TI2gain | R/W | 2 | 0x0000 | Temperature manual compensation coefficient of current I2 channel |
| E4H | UTCcoffA | R/W | 2 | 0x0000 | Temperature auto-compensation quadratic term coefficient of voltage channel Ugain |
| E8H | UTCcoffB | R/W | 2 | 0x0000 | Temperature auto-compensation monomial coefficient of voltage channel Ugain |
| ECH | UTCcoffC | R/W | 2 | 0x0000 | Temperature auto-compensation constant term of voltage channel Ugain |
| F0H | I1TCcoffA | R/W | 2 | 0x0000 | Temperature auto-compensation quadratic term coefficient of current channel1 I1gain |
| F4H | I1TCcoffB | R/W | 2 | 0x0000 | Temperature auto-compensation monomial |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | coefficient of current channel1 I1gain |
| F8H | I1TCcoffC | R/W | 2 | 0x0000 | Temperature auto-compensation constant term of current channel1 I1gain |
| FCH | I2TCcoffA | R/W | 2 | 0x0000 | Temperature auto-compensation quadratic term coefficient of current channel2 I2gain |
| 100H | I2TCcoffB | R/W | 2 | 0x0000 | Temperature auto-compensation monomial coefficient of current channel2 I2gain |
| 104H | I2TCcoffC | R/W | 2 | 0x0000 | Temperature auto-compensation constant term of current channel2 I2gain |
| Above regisers do not participate calculating of Checksum register. | | | | | |
| 108H | LoadData Cp | R/W | 2 | 0x0000 | Write 0x00BC to this register and constant power value P/Q/SDATACP will be load to accumulation source |
| 10CH | Reserved | R/W | 2 | 0x0000 | Reserved register |
| 110H | Reserved | R/W | 2 | 0x0000 | Reserved register |
| 114H | SRSTREG | R/W | 1 | 0x00 | Software reset register (write 0x55 to reset table correction and 0xAA to reset EMU) |
| 118H | PFCntN | R/W | 2 | 0x0000 | Reversed fast active pulse counter register |
| 11CH | QFCntN | R/W | 2 | 0x0000 | Reversed fast reactive pulse counter register |

### 22.3.2.2 Calculation parameter register introduction

| **EMUSR （EMU state register）** | | | **Base address： 0x40013800 offset address： 00H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | X | X | X | X | SFOF | PEOF | QEOF | CHSTS |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | TAMP | I2GTI1 | NoQLd2 | NoPLd2 | NoQLd1 | NoPLd1 | REVQ | REVP |
| **Write:** | | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| SFOF | Apparent energy register overflow flag, cleared by write 0 |
| PEOF | Active energy register overflow flag, cleared by write 0 |
| QEOF | Reactive energy energy register overflow flag, cleared by write 0 |
| CHSTS | **Metering channel state flag.**<br>0: use current channel1 for calculation<br>1: use current channel2 for calculation |

| | |
|---|---|
| TAMP | **Power stealing flag**<br>0：no power stealing,the difference between I1Rms and I2Rms or Power1 and Power2 is less than the set IChk.<br>1：power is stolen<br>If TampSel =0，it means：I1Rms>I2Rms*(1+IChk) or I2Rms>I1Rms*(1+IChk)<br>If TampSel =1，it means：Power1>Power2*(1+IChk) or Power2>Power1*(1+IChk) |
| I2GTI1 | **Flag of channel 2 current or power is greater than channel1 current or power**<br>0：I2Rms<I1Rms or Power2<Power1<br>If TampSel =0,it means：I2Rms<I1Rms<br>1：I2Rms>I1Rms or Power2>Power1<br>If TampSel =0, it means：I2Rms>I1Rms<br>If TampSel =1, it means：Power2>Power1<br>If TampSel =1, it means：Power2<Power1 |
| NoQLd2 | **Channel 2 reactive power defluction flag:**<br>1: Set NoQLd2 to 1 if reactive power2 is less than start power,<br>0: clear NoQLd2 to zero if reactive power is greater or equals to start power |
| NoPLd2 | **Channel 2 active power defluction flag**<br>1: Set NoPLd2 to 1 if active power2 is less than start power,<br>0: clear NoPLd2 to zero if active power is greater or equals to start power |
| NoQLd1 | **Channel 1 reactive power defluction flag**<br>1: Set NoQLd1 to 1 if reactive power1 is less than start power,<br>0: clear NoQLd1 to zero if reactive power is greater or equals to start power |
| NoPLd1 | **Channel 1 active power defluction flag**<br>1: Set NoPLd1 to 1 if active power1 is less than start power,<br>0: clear NoPLd1 to zero if active power is greater or equals to start power |
| REVQ | **Reverse reactive power indicate identify signal**<br>1: if the reactive power is detected to be negative, set this signal to 1,<br>0: if it is detected to be positive, set this signal to be 0.<br>This value is updated when QF outputs pulse. |
| REVP | **Reverse active power indicate identify signal**<br>1: if the active power is detected to be negative, set this signal to 1,<br>0: if it is detected to be positive, set this signal to be 0.<br>This value is updated when PF outputs pulse. |

Note: public identification(REVP、REVQ、SEOF、PEOF、QEOF) indicate the current state of rms calculation channel.

| **EMUIE**<br>**（EMU state register）** | | | **Base address：** **0x40013800**<br>**offset address：** **04H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:** | X | X | X | X | X | | ZXlostIE | Buffer_F |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | | | | | | | | ull IE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | UdetIE | TAMPIE | DCavreageUpdateIE | SlowRmsUpdIE | FastRmsUpdIE | ZXI2IE | ZXI1IE | I2OVIE |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | PFIE | QFIE | SFIE | SPLIE | ZXIE | I1OVIE | UOVIE | SAGIE |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits name** | **Description** |
|---|---|
| ZXlostIE | **Zero crossing loss interrupt enable**<br>0：disable<br>1：enable |
| Buffer_Full IE | **Buffer memory complete interrupt enable**<br>0：disable<br>1：enable |
| UdetIE | **U channel SAG/PEAK events enter/exist interrupt enable**<br>0：disable<br>1：enable |
| TAMPIE | Power stealing interrupt enable(0:disable 1:enable) |
| DCavreageUpdateIE | DC average value update interrupt enable(0:disable 1:enable) |
| SlowRmsUpdateIE | Interrupt while slow current voltage rms updating enable (0:disable 1:enable) |
| FastRmsUpdateIE | Interrupt while fast current voltage rms updating enable (0:disable 1:enable) |
| ZXI2IE | User-specified zero-crossing of current channel2 interrupt enable (0:disable 1:enable) |
| ZXI1IE | User-specified zero-crossing of current channel1 interrupt enable (0:disable 1:enable) |
| I2OVIE | CurrentI2 channel peak value exceed the threshold interrupt enable(0:disable 1:enable) |
| PFIE | Interrupt while PF output pulse enable(0:disable 1:enable) |
| QFIE | Interrupt while QF output pulse enable (0:disable 1:enable) |
| SFIE | Interrupt while SF output pulse enable (0:disable 1:enable) |
| SPLIE | Interrupt enable while updating waveform register(0:disable 1:enable) |
| ZXIE | Interrupt enable while voltage zero crossing in user-specified way(0:disable 1:enable) |
| I1OVIE | Current I1 channel peak value exceed the threshold interrupt |

| | |
|---|---|
| | enable(0:disable 1:enable) |
| UOVIE | Voltage peak value excess the threshold interrupt enable(0:disable 1:enable) |
| SAGIE | Loss of voltage below the threshold and the time exceed the setted num of half period interrupt enable(0:disable 1:enable) |

| **EMUIF（EMU state register）** | | | **Base address： 0x40013800 offset address： 08H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit23** | **22** | **21** | **20** | **19** | **18** | **17** | **Bit16** |
| **Read:** **Write:** | X | X | X | X | X | ZXlostIF | Buffer_FullIF | Uend |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** **Write:** | Ustart | TAMPIF | DCAveUpdIF | SlowRmsUpdIF | FastRmsUpdIF | ZXI2IF | ZXI1IF | I2OVIF |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | PFIF | QFIF | SFIF | SPLIF | ZXIF | I1OVIF | UOVIF | SAGIF |
| **Write:** | X | X | X | X | X | X | X | X |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **Bits name** | **Description** |
|---|---|
| ZXlostIF | **Zero crossing loss interrupt flag**<br>1：happened interrupt<br>0：no interrupt |
| Buffer_FullIF | **Buffer memory complete interrupt flag**<br>1：happened interrupt<br>0：no interrupt |
| Uend | **U channel SAG/PEAK events exist interrupt flag**<br>1：happened interrupt<br>0：no interrupt<br>Read cleared |
| Ustart | **U channel SAG/PEAK events enter interrupt**<br>1：happened interrupt<br>0：no interrupt<br>Read cleared |
| TAMPIF | Interrupt flag while stealing power, cleared after readed |
| DCavreageUpdateIF | Interrupt flag while updating DC average value, cleared after readed |
| SlowRmsUpdateIF | Interrupt flag while updating slow current voltage rms, cleared after readed |
| FastRmsUpdateIF | Interrupt flag while updating fast current voltage rms, cleared after |

| | |
|---|---|
| | readed |
| ZXI2IF | Interrupt flag when zero crossing of current channel2 in user-specified ways, cleared after readed |
| ZXI1IF | Interrupt flag when zero crossing of current channel1 in user-specified ways, cleared after readed |
| I2OVIF | Interrupt flag when peak value of current I2 channel exceed threshold, cleared after readed |
| PFIF | Interrupt flag while PF output pulse, cleared after readed |
| QFIF | Interrupt flag while QF output pulse, cleared after readed |
| SFIF | Interrupt flag while SF output pulse, cleared after readed |
| SPLIF | Interrupt flag while updaring waveform register, cleared after readed |
| ZXIF | Interrupt flag while voltage zero crossing in user-specified way, cleared after readed |
| I1OVIF | Current I1 channel peak value exceed the threshold interrupt flag, cleared after readed |
| UOVIF | Voltage peak value excess the threshold interrupt flag, cleared after readed |
| SAGIF | Loss of voltage below the threshold and the time exceed the setted num of half period interrupt flag, cleared after readed |

| **GP1** | | | **Base address：0x40013800**<br>**offset address：0CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | GP1[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

This register is 16-bit signed num,MSB for sign.

If power factor is 1, the measured error during table correction is : Err%

Pgain = -Err% /（1+Err%）

If Pgain is positive, write Pgain *32768 to GP1

If Pgain is negative, writr 65536+Pgain *32768 to GP1

| **GPHS1** | | | **Base address：0x40013800**<br>**offset address：18H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | GPHS1[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Phase correction calculation formula in PQ method is shown below:

Output error is corrected to nearly 0 through GP register if the signal input is resistive:

Adjust the signal input to 0.5L, and the observed error is Err%

If Err is negative：

Gphs1 = -Err%*32768/1.732

If Err is positive：

Gphs1 = 65536 – Err%*32768/1.732

| **GP2** | | | **Base address：** **0x40013800**<br>**offset address：** **1CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | GP2[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Same calculation formula with GP1

| **GPHS2** | | | **Base address：** **0x40013800**<br>**offset address：** **28H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | GPHS2[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Phase correction calculation formula in PQ method is shown below:

Output error is corrected to nearly 0 through GP register if the signal input is resistive:

Adjust the signal input to 0.5L, and the observed error is Err%

If Err is negative：

Gphs1 = -Err%*32768/1.732

If Err is positive：

Gphs1 = 65536 – Err%*32768/1.732

| **QPHSCal** | | | **Base address：** **0x40013800**<br>**offset address：** **2CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | QPHSCal[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Reactive power phase compensation register is in binary complement format, MSB for sign.

It is 0000H by default.

The reactive phase compensation register also adopts the binary complement form, and the highest bit is the symbol bit.

QphsCal defaults to 0x0000, Femu to 819.2KHz, corresponding to the 50Hz input signal OSR128 (EMCON bit7), select 64, QphsCal is 0x005E, you can achieve accurate 90 degrees phase-shifting. If OSR128 selects 128, QphsCal is 0x0028. Corresponding to the 60Hz input signal, when OSR128 selects 64, the QphsCal is 0x44.

If femu changes or signal is at other frequency, it is corrected with following formula:

Reactive power 0.5L, it is corrected on the condition that angle between U and I is 30de,power Q error is : Err%

Calculation formula of QPhasCal：

Result = Err%*32768/1.732-256

If Result is positive, QphsCal = Result；

If Result is negative, QphsCal = 65536+Result；

Note:

This register corrects the internal phase shift filter, the correction result goes for both two calculation channels.

| **I2Gain** | | | | **Base address：** **0x40013800**<br>**offset address：** **30H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | I2Gain[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The relative gain correction of I2 relative to I1 uses the binary complement form and the highest bit is the symbol bit.

In case of stealing power, the current effective value or power of the two channels need to be compared, so that the input value of the channel 1 and the channel 2 should be equal at the same current input.

Through the channel 2 gain correction register I2GAIN (30H), so that the input same current case, the value of the two registers consistent.

Suppose the input is equally rated current, the current channel 1 is valid, the register reading is I1rms, the current channel 2 is valid, the register reading is I2rms, the current channel is 1, the active power is Power1, and the current channel is 2, the active power is Power2

**If the current is selected, the way to prevent electric larceny is selected:**

Then Gain = I1rms/I2rms - 1

**If power is selected, the way to prevent electricity stealing is selected:**

Then Gain = Power1/Power2 - 1

If Gain>=0, I2Gain=Gain*2^15; if Gain<0, I2Gain = Gain*2^15+2^16

| **I1Off** | | **Base address：** **0x40013800**<br>**offset address：** **34H** |
|---|---|---|
| | **Bit15…Bit0** | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | I1Off[15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:While measuring the DC signal, close high pass filter first, average the consecutive register 00H reading for I1Off register when the input channel signal is 0, users keep this value and write the I1Off value to register before every power-on.

The register is aligned with the bit[19]~bit[4] bits of the ADC output data.

The main purpose of this register: close high pass filter of I1/I2/U if chip internal high pass filter is closed when users mean to measure DC digital to avoid phase error. Users need not configurate this register when test AC signal if users correct ADC zero drift of external input zero signal point through this register.

| **I2Off** | | | **Base address:** 0x40013800<br>**offset address:** 38H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | I2Off[15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

While measuring the DC signal, close high pass filter first, average the consecutive register 04H reading for I2Off register when the input channel signal is 0, users keep this value and write the I2Off value to register before every power-on.

The register is aligned with the bit[19]~bit[4] bits of the ADC output data.

| **UOff** | | | **Base address:** 0x40013800<br>**offset address:** 3CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | UOff[15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:While measuring the DC signal, close high pass filter first, average the consecutive register 08H reading for UOff register when the input channel signal is 0, users keep this value and write the UOff value to register before every power-on.

The register is aligned with the bit[19]~bit[4] bits of the ADC output data. Use DC offset correction only after high pass filter. High pass filter of I1/I2/U should be closed together to avoid phase error.

| **PStart** | | | **Base address:** 0x40013800<br>**offset address:** 40H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | Ptart[15:0] | | | | | | | |

| Write: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Default value: 0x0040

Note:

Pstart is 16-bit unsigned num, compare it with bit8-bit23 of PowerP absolute value to judge the action.

PF do not output pulse if |P| is smaller than PStart. And PEVP reverse flag will be cleared at the same time.

Application:

1. Input Ib, Un after table correction.
2. Read the 32 bits of PowerP and set the high 24 bits to 1.

   If x1 is positive, x2 = x1;

   If x1 is negative, its signed-magnitude is x2;
3. Setwrites Y to PStart, if electricity is able to start with 0.4%Ib, then:

   Y = x 2 *0.2 %

| QStart | | | Base address: 0x40013800<br>offset address: 44H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | QStart[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Default value: 0x0080

Note:

Qstart is 16-bit unsigned num, compare it with bit8-bit23 of PowerQ absolute value to judge the action.

QF do not output pulse if |Q| is smaller than PStart. And PEVQ reverse flag will be cleared at the same time.

Application:

1. Input Ib, Un after table correction.
2. Read the 32 bits of PowerQ and set the high 24 bits to 1.

   If x1 is positive, x2 = x1;

   If x1 is negative, its signed-magnitude is x2;
4. Set write Y to QStart, if electricity is able to start with 0.4%Ib, then:

   Y = x 2 *0.2 %

| SStart | | | Base address: 0x40013800<br>offset address: 48H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | SStart[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Default value: 0x0090

Note:

Sstart is 16-bit unsigned num, compare it with bit8-bit23 of PowerS absolute value to judge the action.

SF do not output pulse if |S| is smaller than PStart.

Application:

1. Input Ib, Un after table correction.
2. Read the 32 bits of PowerS and set the high 24 bits to 1.

   If x1 is positive, x2 = x1;

   If x1 is negative, its signed-magnitude is x2;
3. Set writes Y to SStart, if electricity is able to start with 0.4%Ib, then:

   Y = x 2 *0.2 %

HT502X provides two ways: start creeping alone starting with creeping creeping starting. Refer to section 23.2.6 starting / creep function description.

| **HFConst** | | | **Base address:** 0x40013800<br>**offset address:** 4CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **Bit14…Bit0** | | | | | | |
| **Read:**<br>**Write:** | X | HFConst[14:0] | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Note: default 0x0080

HFConst is 15-bit unsigned num, set is to low 15 bits and compare the value to absolute value of fast pulse counting register 0x6FH~0x71H value. If it is greater than HFConst, then corresponding PF/QF/SF pulse will be output.

| **ADC Config Register (ADCCFG)** | | | **Base address:** 0x40013800<br>**offset address:** 50H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | PQSStart SourceSe l1 | PQSStart SourceSe l0 | Vref_En | Add _CIADD | WaveSel | TampSou rceSel | IPTampS ource_Se l | CHNFix |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | IPTamp_ Sel | TampSel | FLTON | CHNSEL | CIADD | SPL2 | SPL1 | SPL0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| **Bits name** | **description** |
|---|---|
| PQSStartSourceSel[1:0] | **PQS start and creep compare source control bits:**<br>PQSStartSourceSel1 / PQSStartSourceSel0 / PQS start creep source<br>0 / 0 / Fast power<br>0 / 1 / Slow power<br>1 / X / Instantaneous power |

<table>
<tr><td>Vref_En</td><td>Vref enable control bits：<br>与 LDO_EMU_En 、EMU_En ADC_On take effect together.see following table for detailed information
<table>
<tr><td>LDO_EMU_En</td><td>EMU_en</td><td>Vref_En</td><td>Adc_on</td><td>Vref actual output</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td></tr>
<tr><td>X</td><td>0</td><td>X</td><td>X</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>Add_CIADD</td><td>Accumulation mode control bits in Single-phase three-wire mode<br>= 0, accumulate absolute value<br>= 1,accumulate algebraic sum</td></tr>
<tr><td>WaveSel</td><td>ADC waveform sampled data source selection<br>= 0,select the original data which is not filtered by high pass filter as ADC waveform data<br>= 1, select the original data which is filtered by high pass filter as ADC waveform data</td></tr>
<tr><td>TampSourceSel</td><td>Power stealing auto-prevention channel interchannel compare source selection：<br>=0,select the slow Rms or power as compare souce<br>=1,select the fast Rms or power as compare souce</td></tr>
<tr><td>IPTampSource_Sel</td><td>Power stealing auto-prevention threshold judgement source fast/slow selection bit<br>=0, select slow rms or power as threshold source<br>=1, select fast rms or power as threshold source</td></tr>
<tr><td>CHNFix</td><td>Power stealing auto-prevention small signal channel switch configuration bit<br>=0,measuring the first current channel when current of both channels drops below IPTAMP<br>=1, measuring the previous current channel when current of both channels drops below IPTAMP</td></tr>
<tr><td>IPTamp_Sel</td><td>Power stealing auto-prevention threshold judgement basis<br>=0, IPTamp register select RMS as the threshold<br>=1, IPTamp register select active power as the threshold</td></tr>
<tr><td>TampSel</td><td>Power stealing prevention comparisonmethod selection<br>=0, choose current rms for power stealing prevention comparison<br>=1, choose active power for power stealing prevention comparison</td></tr>
<tr><td>FLTON</td><td>Power stealing auto-prevention module enable（0：disable 1：enable）,see the back table for detailed information</td></tr>
<tr><td>CHNSEL</td><td>Select the channel to measure（0：select channel1 1：select channel2）, see the back table for detailed information</td></tr>
</table>

| | |
|---|---|
| CIADD | Single-phase three-wire accumulation mode selection(0：single channel mode 1：current accumulated sum mode） |
| SPL[2:0] | Waveform sample frequency selection control register, see the back table for detailed information |

Note:

Data of Channels themselves are employed in current accumulation mode. The power stealing auto-prevention is opened if FLTON=1, and the write and read operation to CIADD and CHNSEL is valid only when FLTON=0.

| **Input signal** | | | **Output signal** | |
|---|---|---|---|---|
| FLTON | CIADD | CHNSEL | Chanel state | Energy accumulation |
| 1 | X | X | Indicate the result of power stealing auto-prevention channel selection | Decide the channel of power for calculation according to Chanel state |
| 0 | 0 | 0 | 0 | Select channel1 for measurement（default） |
| 0 | 0 | 1 | 1 | Select channel2 for measurement |
| 0 | 1 | x | 0 | Single-phase three-wire mode |

Power stealing auto-prevention judgement basis set explanation

| TampSel | IPTamp_Sel | TampSourceSel | IPTampSource_Sel | Power stealing auto-prevention judgement basis |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | IPTAMP(6CH) base it on whether slow rms register exceed power stealing threshold, ICHK(70H) compare the slow rms difference of two channels with setted percents; |
| 0 | 0 | 0 | 1 | IPTAMP(6CH) base it on whether slow rms register exceed power stealing threshold, ICHK(70H) compare the slow rms difference of two channels with setted percents; |
| 0 | 0 | 1 | 0 | IPTAMP(6CH) base on slow Rms, ICHK(70H) base on fasr rms |
| 0 | 0 | 1 | 1 | IPTAMP(6CH) base on fast Rms, ICHK(70H) base on fast Rms |
| 0 | 1 | 0 | 0 | IPTAMP(6CH) base on slow power, ICHK(70H) base on slow Rms |

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | IPTAMP(6CH) base on fast power, ICHK(70H) base on slow Rms |
| 0 | 1 | 1 | 0 | IPTAMP(6CH) base on slow power, ICHK(70H) base on fast Rms |
| 0 | 1 | 1 | 1 | IPTAMP(6CH) base on fast power, ICHK(70H) base on fast Rms |
| 1 | 0 | 0 | 0 | IPTAMP(6CH) base on slow Rms, ICHK(70H) base on slow power |
| 1 | 0 | 0 | 1 | IPTAMP(6CH) base on fast Rms ICHK(70H) base on slow power |
| 1 | 0 | 1 | 0 | IPTAMP(6CH) base on slow Rms ICHK(70H) base on fast power |
| 1 | 0 | 1 | 1 | IPTAMP(6CH) base on fast Rms, ICHK(70H) base on fast power |
| 1 | 1 | 0 | 0 | IPTAMP(6CH) base on slow power , ICHK(70H) base on slow power |
| 1 | 1 | 0 | 1 | IPTAMP(6CH) base on fast power, ICHK(70H) base on slow power |
| 1 | 1 | 1 | 0 | IPTAMP(6CH) base on slow power, ICHK(70H) base on fast power |
| 1 | 1 | 1 | 1 | IPTAMP(6CH) base on fast power, ICHK(70H) base on fast power |

SPL[2:0]: waveform sample interrupt frequency selection, if Femu=819.2K, the frequency is shown as below:

| SPL2 | SPL1 | SPL0 | Waveform sample frequency |
|---|---|---|---|
| 0 | 0 | 0 | 1.6k Hz (femu/512) |
| 0 | 0 | 1 | 3.2k Hz (femu/256) |
| 0 | 1 | 0 | 6.4k Hz (femu/128) |
| 0 | 1 | 1 | 12.8k Hz (fadc/64) |
| 1 | X | x | 12.8k Hz (fadc/64) |

Note:If femu=409.6KHz,the waveform sample frequency is in proportion to it.

| **Current Channel Control Register (CHNLCR channel control register)** | | | **Base address: 0x40013800 offset address: 54H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | PDatacpEn | QDatacpEn | SDatacpEn | Sag_En | Ovl_En | RmsLpfEn | PQLpfEn | Add_DataCP |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | LPMode | POS | LP FreqSel | Uconst En | PSSel | HPFONI2 | HPFONI1 | HPFONU |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | | 0 | 0 | 1 | 1 | 1 |

| Bits | function description |
|---|---|
| PDatacpEn | **Active power constant calculation enable**<br>0：disable （default）<br>1：enable |
| QDatacpEn | **Reactive power constant calculation enable**<br>0：disable （default）<br>1：enable |
| SDatacpEn | **Apparent power constant calculation enable**<br>0：disable （default）<br>1：enable |
| Sag_En | **Voltage SAG function enable control：**<br>0：disable；<br>1：enable |
| Ovl_En | **Over-voltage over-current enable control：**<br>**0**：disable；<br>1：enable |
| RmsLpfEn | **RmsLpf enable control ：**<br>=0,disable low pass filter of rms channel<br>=1, enable low pass filter of rms channel（default） |
| PQLpfEn | **PQLpf enable control：**<br>=0, disable low pass filter of power channel<br>=1, enable low pass filter of power channel（default） |
| Add_DataCP | **Energy accumulation rate selection bit during constant calculation：**<br>= 0， 32K<br>= 1, BS bit stream rate |
| LPMode | **Low power consumption mode enable control：**<br>=0，EMU is in normal calculation mode（default）<br>=1，switch EMU clock to low frequency end enter low power consumption mode |
| POS | **Pulse significant level selection bit**<br>0： PF/QF/SF active high level<br>1： PF/QF/SF active low level |
| LP FreqSel | **Low power consumption mode clock selection bit**<br>= 0：select 32K clock（OSC clock，PLL Off）<br>= 1：select 204.8K clock（PLL On） |

| | |
|---|---|
| UconstEn | **Low power consumption mode apparent power voltage source selection bit**<br>0：use voltage channel rms and current channel rms for apparent power calculation<br>1：use Uconst register and current channel rms for apparent power calculation |
| PSSel | **Output Pulse type selection bit**<br>0：PF output active power pulse, SF output apparent power pulse<br>1：PF output apparent power pulse, SF output active power pulse<br>This function goes for situation where replace apparent power with active power and output active and apparent power pulse in the same pulse interface during Low power consumption calculation mode |
| HPFONU | Voltage channel high pass filter switches(0：close 1：open） |
| HPFONI2 | Current channel2 high pass filter switches（0：close 1：open） |
| HPFONI1 | Current channel1 high pass filter switches（0：close 1：open） |

| **Energy Measure Control (EMCON energy accumulation control register )** | | | **Base address: 0x40013800**<br>**offset address: 58H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | CF1CFG 1 | CF1CFG 0 | CF2CFG 1 | CF2CFG 0 | CF3CFG 1 | CF3CFG 0 | AverPow erCal | AverPow erSource |
| **Reset:** | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | OSR128 | SRun | QRun | PRun | QMOD1 | QMOD0 | PMOD1 | PMOD0 |
| **Reset:** | **0** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| **Bits** | **function description** |
|---|---|
| CF1CFG[1:0] | **CF1 interface output configuration control bits:**<br>CF1CFG1 \| CF1CFG 0 \| CF1<br>0 \| 0 \| P (default)<br>0 \| 1 \| Q<br>1 \| 0 \| S<br>1 \| 1 \| P |
| CF2CFG[1:0] | **CF2 interface output configuration control bits:**<br>CF2CFG1 \| CF2CFG 0 \| CF2<br>0 \| 0 \| P<br>0 \| 1 \| Q （default）<br>1 \| 0 \| S<br>1 \| 1 \| P |
| CF3CFG[1:0] | **CF3 interface output configuration control bits:**<br>CF3CFG1 \| CF3CFG 0 \| CF3<br>0 \| 0 \| P |

<table>
<tr><td></td><td>
<table>
<tr><td>0</td><td>1</td><td>Q</td></tr>
<tr><td>1</td><td>0</td><td>S（default）</td></tr>
<tr><td>1</td><td>1</td><td>P</td></tr>
</table>
</td></tr>
<tr><td>AverPowerCal</td><td>Energy accumulation source selection bit<br>=0, select instantaneous power as energy accumulation source<br>=1, select average power as energy accumulation source</td></tr>
<tr><td>AverPowerSource</td><td>If average power accumulation is enabled（AverPowerCal =1），average power source selection：<br>=0, select slow average power as average power accumulation source<br>=1, select fast average power as average power accumulation source</td></tr>
<tr><td>OSR128</td><td>OSR selection bit of normal calculation mode：（goes for only normal calculation mode, not for low power consumption mode）<br>=0，OSR =64<br>=1，OSR =128</td></tr>
<tr><td>SRun</td><td>Apparent energy accumulation enable<br>0：stop apparent energy calculation<br>1：allow apparent energy calculation</td></tr>
<tr><td>QRun</td><td>Reactive energy accumulation enable<br>0：stop reactive energy calculation<br>1：alow reactive energy calculation</td></tr>
<tr><td>PRun</td><td>Active energy accumulation enable<br>0：stop active energy calculation<br>1：allow active energy calculation</td></tr>
<tr><td>QMOD[1：0]</td><td>Reactive energy accumulation mode selection bits
<table>
<tr><td>QMOD1</td><td>QMOD0</td><td>Accumulated power Qm</td><td>Reactive energy accumulation method selection</td></tr>
<tr><td>0</td><td>0</td><td>Qm=DataQ</td><td>Pulse energy accumulate reactive power in algebraic addition way</td></tr>
<tr><td>0</td><td>1</td><td>DataQ≧0,Qm=DataQ;<br>DataQ<0,Qm=0</td><td>Pulse energy only accumulate the positive reactive power rather than negative reactive power</td></tr>
<tr><td>1</td><td>0</td><td>Qm=|DataQ|</td><td>Pulse energy accumulate reactive power in absolute value format</td></tr>
<tr><td>1</td><td>1</td><td>Qm=DataQ</td><td>Pulse energy accumulate reactive power in algebraic addition way</td></tr>
</table>
</td></tr>
<tr><td>PMOD[1：0]</td><td>Active energy accumulation mode selection bits
<table>
<tr><td>PMOD1</td><td>PMOD0</td><td>Accumulated power Pm</td><td>Active energy accumulation method selection</td></tr>
<tr><td>0</td><td>0</td><td>Pm=DataP</td><td>Pulse energy accumulate active power in algebraic addition way</td></tr>
<tr><td>0</td><td>1</td><td>DataP≧0,Pm=DataP;<br>DataP<0,Pm=0</td><td>Pulse energy only accumulate the positive active power rather than</td></tr>
</table>
</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | | | | negative active power |
| | 1 | 0 | Pm=\|DataP\| | Pulse energy accumulate active power in absolute value format |
| | 1 | 1 | Pm=DataP | Pulse energy accumulate active power in algebraic addition way |

| **PFCnt** | | | **Base address: 0x40013800**<br>**offset address: 5CH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | PFCNT[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **QFCnt** | | | **Base address: 0x40013800**<br>**offset address: 60H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | QFCNT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **SFCnt** | | | **Base address: 0x40013800**<br>**offset address: 64H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | SFCNT[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:To avoid energy loss when power-on/power-off, MCU will restore the register PFCnt/QFCnt/SFCnt reading when power off and rewrite these values to PFCnt/QFCnt/SFCnt when power on.

Corresponding PF/QF/SF pulse will overflow and energy register value will be incremented by 1 when fast pulse register PFCnt/QFCnt/SFCnt counting value is greater or equals to HFconst.

| **ADC Channel Gain (ADCCON ADC channel gain control register)** | | | **Base address: 0x40013800**<br>**offset address: 68H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | DGI3 | DGI2 | DGI1 | DGI0 | DGU1 | DGU0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |

| Read: / Write: | CFP1 | CFP0 | PGA3 | PGA2 | PGA1 | PGA0 | UPGA1 | UPGA0 |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | function description |
|---|---|
| DGI[3:0] | **Current channel digital gain selection bits**<br>DGI1 / DGI0 / I1 digital gain / DGI3 / DGI2 / I2 digital gain<br>0 / 0 / DG=1 / 0 / 0 / DG=1<br>0 / 1 / DG=2 / 0 / 1 / DG=2<br>1 / 0 / DG=4 / 1 / 0 / DG=4<br>1 / 1 / DG=8 / 1 / 1 / DG=8 |
| DGU[1:0] | **Voltage channel digital gain selection bits**<br>DGU 1 / DGU 0 / Voltage channel digital gain<br>0 / 0 / DG=1<br>0 / 1 / DG=2<br>1 / 0 / DG=4<br>1 / 1 / DG=8 |
| CFP[1:0] | **Pulse width selection bits**<br>Namely t4 parameter of pulse output,see PF/QF/SF sequence feature<br>CFP1 / CFP0 / Pulse width（femu=819.2KHz）<br>0 / 0 / 80ms<br>0 / 1 / 40ms<br>1 / 0 / 20ms<br>1 / 1 / 10ms |
| PGA[3：0] | **Current channel analog gain selection bits**<br>PGA1 / PGA0 / Current channel1 / PGA3 / PGA2 / Current channel2<br>0 / 0 / PGA=1 / 0 / 0 / PGA=1<br>0 / 1 / PGA=2 / 0 / 1 / PGA=2<br>1 / 0 / PGA=8 / 1 / 0 / PGA=8<br>1 / 1 / PGA=16 / 1 / 1 / PGA=16 |
| UPGA[1:0] | **Voltage channel analog gain selection bits**<br>UPGA1 / UPGA0 / Voltage channel gain<br>0 / 0 / PGA=1<br>0 / 1 / PGA=2<br>1 / 0 / PGA=4<br>1 / 1 / PGA=8 |

| IPTAMP | Base address： 0x40013800 |
|---|---|

| | | | offset address： 6CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| **Read:** **Write:** | IPTAMP[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Note: This register is 0x0020 by default

Same format with rms register or power register, IPTAMP[15:0] is 16-bit current rms register or power register. When choose power stealing threshold as power: corresponds to Bit27-Bit12 of power register.

If choose power stealing prevention threshold as rms: corresponds to Bit19-Bit4 of rms register

**If power stealing auto-prevention dealing module is open:**

If choose fast or slow current rms for threshold judgement and fast or slow current rms of channel1 and channel2 is less than IPTAMP, select channel1 as rms input or keep the previous measuring channel, TAMP I2GTI1 are both 0.

If choose absolute value of fast or slow power P for threshold judgement and fast or slow PowerP1 and PowerP2 are less than IPTAMP, select channel1 as rms input or keep the previous measuring channel, TAMP I2GTI1 are both 0.

| **IChk** | | | **Base address： 0x40013800** **offset address： 70H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit7…Bit0 | | | | | | | |
| **Read:** **Write:** | IChk[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Note:

Interchanel power stealing difference percent threshold register is in binay complement format, range(0,+1).

ICHK=ICK7*2^（-1）+ ICK6*2^（-2）+ ICK5*2^（-3）+ …+ ICK1*2^（-7）+ ICK0*2^（-8）

If |I2Rms-I1Rms|/I1Rms or |PowerP2-PowerP1|/PowerP1 is greater than ICHK, set power stealing flag;

If certain Bit of check register is 1,corresponding threshold is showing as following:

| | |
|---|---|
| Bit7 | 0.5 |
| Bit6 | 0.25 |
| Bit5 | 0.125 |
| Bit4 | 0.0625 |
| Bit3 | 0.03125 |
| Bit2 | 0.015625 |
| Bit1 | 0.007813 |
| Bit0 | 0.003906 |

For example:

If Check Register=0x1A,power stealing threshold is 0.0625+0.03125+0.007813=10.1563%

Default:0.0625 namely 6.25%.

If select fast or slow current rms as power stealing comparison source after power stealing auto-prevention is enabled, select the greater current to involve power calculation automatically and TAMP=1 when difference percent between slow or fast current 1 and channel2(|I2Rms-I1Rms|/I1Rms) exceed the setted value. If current2 is greater than current1, set flag I2GTI1 to 1,other wise set I2GTI1 to 0.

If select fast or slow power as power stealing comparison source and the difference between PowerP1 and PowerP2(|PowerP2-PowerP1|/PowerP1) exceed the set value, select the greater power to involve power calculation automatically and TAMP=1.

| **EMU and adc control (EMUCTRL EMU control register)** | | | **Base address: 0x40013800**<br>**offset address: 74H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | X | X | StartSel | Vref_LP_En | P2EN | EMU_ClkCtrl | DC PRFCFG1 | DC PRFCFG0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | SlowPRMS | Adc_i1on | Adc_i2on | Adc_uon | Zxd1 | Zxd0 | PRFCFG1 | PRFCFG0 |
| **Reset:** | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| **Bits** | **function description** |
|---|---|
| StartSel | **Anti creep mode selection bit**<br>1：P/Q joint judgment mode, that is at least meat \|P\|>PStar or \|Q\|>QStar, P/Q will start to metering.<br>0：\|P\| according PStart value to judge creep, \|Q\| according QStart value to judge creep.(default) |
| Vref_LP_En | **In low power mode, Lvref involve calculation control bit**<br>if LvrefEn<br>= 0，use Vref as calculation reference<br>= 1，use Lvref as calculation reference<br>Configure this bit and LDO_EMU_En 、EMU_En Vref_LP_En, the results are above: |
| P2EN | **P2 replace Q1 metering channel enable select**<br>1：P2<br>0：Q1 |
| EMU_ClkCtrl | **EMU clock control bit：** |

| EMU_En | LDO_EMU_En | Vref_LP_En | Vref_LP output |
|---|---|---|---|
| 0 | X | X | 0 |
| X | 0 | X | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

<table>
<tr><td></td><td>= 0，EMUclk is 819.2K<br>= 1，EMUclk is 409.6K, need to concern pulse width</td></tr>
<tr><td>DC PRFCFG[1:0]</td><td>DC average update frequency control register
<table>
<tr><th>DC PRFCFG1</th><th>DC PRFCFG0</th><th>DC average value update frequency（EMU = 81</th></tr>
<tr><td>0</td><td>0</td><td>25Hz</td></tr>
<tr><td>0</td><td>1</td><td>12.5 Hz</td></tr>
<tr><td>1</td><td>0</td><td>6.25Hz</td></tr>
<tr><td>1</td><td>1</td><td>3.125Hz（default）</td></tr>
</table></td></tr>
<tr><td>SlowPRMS</td><td>Slow Rms power update frequency control bit<br>=0,drops to 1/2 that of fast<br>=1, drops to 1/4 that of fast (Default)</td></tr>
<tr><td>Adc_i1on</td><td>I1_ADC enable bit<br>0：disable first channel of current ADC(Default)<br>1：enable first channel of current ADC</td></tr>
<tr><td>Adc_i2on</td><td>I2_ADC enable bit<br>0：disable second channel of current ADC（Default）<br>1：enable second channel of current ADC</td></tr>
<tr><td>Adc_uon</td><td>U_ADC enable bit<br>0：disable voltage ADC（default）<br>1：enable voltage ADC</td></tr>
<tr><td>Zxd[1:0]</td><td>Voltage current zero crossing interrupt method select control bit
<table>
<tr><td>ZXD1</td><td>ZXD0</td><td>Voltage current zero crossing interrupt method</td></tr>
<tr><td>0</td><td>0</td><td>Positive zero crossing interrupt</td></tr>
<tr><td>0</td><td>1</td><td>Negative zero crossing interrupt</td></tr>
<tr><td>1</td><td>x</td><td>Bidirectional zero crossing interrupt</td></tr>
</table></td></tr>
<tr><td>PRFCFG[1:0]</td><td>Fast Rms power update frequency control bit
<table>
<tr><th>PRFCFG1</th><th>PRFCFG0</th><th>Rms update frequency（EMU = 819.2kHz）</th></tr>
<tr><td>0</td><td>0</td><td>20Hz</td></tr>
<tr><td>0</td><td>1</td><td>10Hz</td></tr>
<tr><td>1</td><td>0</td><td>5Hz（default）</td></tr>
<tr><td>1</td><td>1</td><td>2.5Hz</td></tr>
</table></td></tr>
</table>

| **P1OFFSET** | **Base address：** **0x40013800**<br>**offset address：** **78H** |
|---|---|
| | **Bit15…Bit0** |
| **Read:**<br>**Write:** | P1OFFSET[15:0] |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Active power channel1 correction register is in binary complement format.

P1OFFSETH is in aglined to bit[18..3] of 32-bit register PowerP1.

$$P1\_offset = \frac{\text{Preal*EC*HFConst*2\^28 *(-Err\%)}}{\text{2.304* 10\^10}}$$

See table correction process for detailed information.

| **P2OFFSET** | | | **Base address:** 0x40013800<br>**offset address:** 7CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | P2OFFSET[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Active power channel2 correction register is in binary complement format.

P2OFFSETH is in aligned to bit[18..3] of 32-bit register PowerP2.

$$P2\_offset = \frac{\text{Preal*EC*HFConst*2\^28 *(-Err\%)}}{\text{2.304* 10\^10}}$$

| **Q1OFFSET** | | | **Base address:** 0x40013800<br>**offset address:** 80H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | Q1OFFSET[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Reactive power channel1 correction register is in binary complement format.

Q1OFFSETH is in aligned to bit[18..3] of 32-bit register PowerQ1.

$$Q1\_offset = \frac{\text{Qreal*EC*HFConst*2\^28 *(-Err\%)}}{\text{2.304* 10\^10}}$$

| **Q2OFFSET** | | | **Base address:** 0x40013800<br>**offset address:** 84H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | Q2OFFSET[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Reactive power channel2 correction register is in binary complement format.

Q2OFFSETH is in aglined to bit[18..3] of 32-bit register PowerQ2.

$$Q2\_offset = \frac{\text{Qreal*EC*HFConst*2\^28 *(-Err\%)}}{\text{2.304* 10\^10}}$$

| **I1RMSOFFSET** | **Base address:** 0x40013800 |
|---|---|

| | | | offset address： 88H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: Write: | I1RMSOFFSET[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Current channel1 rms correction register is binary unsigned num.

Calculation formula:

When input signal is 0, read I1RMS for several times and average these readings and calculate using the following formula.

I1RMSOFFSET = (I1RMS^2)/ (2^15)

If external noise is excessive, the above the I1RMSOFFSET calculated by above formula will exceed the limit, in this case user has to eliminate the board level excessive noise by customer software; this register cannot eliminate this kind of zero drift noise.

| I2RMSOFFSET | | | Base address： 0x40013800<br>offset address： 8CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: Write: | I2RMSOFFSET[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Current channel2 rms correction register is binary unsigned num.

Calculation formula:

When input signal is 0, read I2RMS for several times and average these readings and calculate using the following formula.

I2RMSOFFSET = (I2RMS^2)/ (2^15)

If external noise is excessive, the above the I2RMSOFFSET calculated by above formula will exceed the limit. In this case user has to eliminate the board level excessive noise by customer software, this register cannot eliminate this kind of zero drift noise.

| URMSOFFSET | | | Base address： 0x40013800<br>offset address： 90H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read: Write: | URMSOFFSET[15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Voltage channel rms correction register is binary unsigned num.

Calculation formula:

When input signal is 0, read URMS for several times and average these readings and calculate using the following formula.

URMSOFFSET = (URMS^2)/ (2^15)

If external noise is excessive, the above the URMSOFFSET calculated by above formula will exceed the limit, in this case user has to eliminate the board level excessive noise by customer software; this register cannot eliminate this kind of zero drift noise.

| **ROSICTRL （ Rogowski coil control register）** | | | **Base address： 0x40013800 offset address： 94H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** / **Write:** | Auto_Tc_ EN | Clkdelay 2 | ClkDelay1 | Clkdelay0 | X | Adci_ctrl 1 | Adci_ctrl 0 | DC_En |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** / **Write:** | DC_S_E n | XboFiter 50Hz 60Hz | LpIdleEn | LpStaCt rl | LpCtrl1 | LpCtrl0 | Rosi_i2_ en | Rosi_i1_ en |
| **Reset:** | 0 | 0 | 0 | 0 | | | 0 | 0 |

Note: enable Rogowski coil to open internal filter, integrate the differential signal the Rogowski coil to recovery it, so open this channel of Rogowski coil if external channel employs Rogowski coil.

| **Bits** | **function description** |
|---|---|
| Auto_Tc_ EN | **Temperature auto compensation control bit**<br>0：close temperature auto-compensation(default)<br>1：open temperature auto-compensation, note, calculation parameter register of EMU will give the compensation value at present temperature after open the function at different temperature. |
| Clkdelay[2:0] | **Clk delay between Analog digital, remain the default value.** |
| Adci_ctr[1:0] | **ADC operation current control bit( corresponding ADC power consumption will increase along with current)**<br>=00（default），in normal calculation mode, use offset bias10uA<br>In low power mode, recommend to configurate it as [01], use offset current 1.5uA.<br>Adci_ctr[1] \| Adci_ctr[0] \| offset current<br>0 \| 0 \| 10uA （default）<br>0 \| 1 \| 1.5 uA<br>1 \| 0 \| 3 uA<br>1 \| 1 \| 5 uA<br>The larger the operating current of ADC, the greater the power consumption of the corresponding ADC. |
| DC_En | **DC_En DC average function enable bit：**<br>= 0，close DC average calculation function<br>= 1，enable DC average calculation function |
| DC_S_En | **DC_S_En DC S apparent power energy enable control bit；**<br>= 0，close DC S apparent |

| | |
|---|---|
| | = 1，enable DC S apparent, both two channels of PowerS use DC average U I1 I2 calculation, S apparent pule and energy both are DC. |
| XboFiter<br>50Hz<br>60Hz | **During DC calculation, notch filter frequency control bit(during AC calculation,remain the default value）:**<br>= 0 , notch filter center frequency is 50Hz<br>= 1, notch filter center frequency is 60Hz |
| LpIdleEn | **Open Low power calculation in time division enable control bit:**<br>= 0，low power calculation is in consecutive mode.<br>= 1，low power calculation in time division, if LpIdleTime = 0, idle time is 1s |
| LpStaCtrl | **In Low power calculation mode, steady points num control bit**<br>= 0，lose fixed 512 points;（default）<br>= 1，lose fixed 1024 points; |
| LpCtrl[1:0] | **In low power calculation mode, average time control bit** |

| LpCtrl [1] | LpCtrl [0] | 平均时间（s） |
|---|---|---|
| 0 | 0 | 0.5 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |

| | |
|---|---|
| Rosi_i2_en | **Rogowski coil of current channel2 enable bit**<br>0： close Rogowski coil of current channel2 function<br>1： enable Rogowski coil of current channel2 function |
| Rosi_i1_en | **Rogowski coil of current channel1 enable bit**<br>0：close Rogowski coil of current channel1 function<br>1：enable Rogowski coil of current channel1 function |

| **ANA_control register（analog control register）** | | | **Base address: 0x40013800**<br>**offset address: 98H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:**<br>**Write:** | ldo_emu_en | X | Vref_ctrl2 | Vref_ctrl1 | Vref_ctrl0 | Chop_Vref_en | Reserved | Adc_ChopEn |
| **Reset:** | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | uchop_en | Reserved 4 | Reserved 3 | Reserved 2 | Reserved 1 | Reserved 0 | Vref_Chop_buf_en | lowVref_isel |
| **Reset:** | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

.

| Bits | function description |
|---|---|
| ldo_emu_en | **EMU Analog supply LDO enable control bit:**<br>= 0 ， close LDO，only for constant calculation mode to lower power consumption（default） |

| | |
|---|---|
| | = 1 ， open LDO； |
| Vref_Ctrl[2:0] | **Vref_Ctrl[2：0]，Vref TC peak adjust code**<br>**Default = 100；** |
| Chop_Vref_en | **Vref chop control bit：**<br>= 0，Vref chop off<br>= 1，Vref chop on；（default） |
| Reserved | **Internal reversed register, keep the default value** |
| Adc_ChopEn | **Adc chop control bit**<br>= 0，close Adc chop，<br>= 1，enable Adc chop，recommend to open it（default） |
| uchop_en | **Voltage channel chop control bit：**<br>=0：close voltage channel chop；<br>=1：enable voltage channel chop；recommend to open it（default） |
| Reserved[6:2] | **Internal reversed register, keep the default value Default = 01010** |
| Vref_Chop_buf_en | **Vref chop buf enable control bit：**<br>= 0，close<br>= 1，open（**default**） |
| lowVref_isel | **Lvref offset current control bit, if choose Lvref as calculation reference, user can configurate offset current to adjust ADC power consumption：**<br>= 0，offset current is 10ua（default）<br>= 1，offset current is 5ua |

| **UCONST** | | | | **Base address：** **0x40013800**<br>**offset address：** **9CH** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | **UCONST** [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Voltage constant register is in binary unsigned num format.

During voltage loss, calculation is in the low power mode. 16 bits of Voltage register is in binary unsigned num format.

The main purpose of this register is replacing Urms register to involve calculation, users can write to the value as needed as well. This register corresponds to Urms register like this:

URms is a 24-bit register; the MSB is always 0 namely only 23 bits of it is significant. However, UCONST is a register who has only 16 significant bits, the Urms need right shift 7 bits namely divided by 2^7 to obtain Uconst when input normal signal.

| **LpIdleTime** | | **Base address：** **0x40013800**<br>**offset address：** **A0H** |
|---|---|---|
| | **Bit10…Bit0** | |

| Read: | **LpIdleTime** [10:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Write: | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In low power metering mode, the time-sharing mode of the idle time setting register.

The default is 0, that is, the free 1S, and then open the fixed time;

Free time can write range 0---599, that is, idle 1S--600S, open another fixed time.

Note:In low power calculation mode, idle time register of time division mode is 0 by default, namely idle for 1S and then open fixed time.

Idle time range is 0-599, namely IDLE for 1S-600S and open fixed time.

| **USAGLVL** | | | **Base address:** 0x40013800<br>**offset address:** A4H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | USAGLVL[15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:This register is unsigned num, and 16bit is aligned to high bits of 19-bit ADC waveform register, compare ADC waveform register absolute value with USAGLVL and within setted SagCyc, if ADC waveform is less than USAGLVL register, SAF flag will be updated in the last SagCyc half period.

Upeak register is in 22-bit signed binary complement format. It aligned to sampled data. USAGLVL register is 16-bit unsigned data, aligned to high bit of waveform sampled absolute data.

Peak value is 1.414 times of rms. So:

Upeak = Urms*1.414/2^5

If set USAGLVL threshold to 80% peak value, then:

USAGLVL= 0.8* Upeak /2^3 = 0.8*Urms*1.414/2^8

| **IpeakCyc**<br>**(I1 I2 PEAK check period setting register)** | | | **Base address:** 0x40013800<br>**offset address:** A8H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | **SagCyc** [15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Default value: 0x0020.

This is a 16-bit unsigned num, Used to set the flow detection data length, that is, the IPeak function detects the data length after setting, and gives the peak value.

The 1 LSB correspond to half cycles.

| UOVLVL | | | Base address: 0x40013800<br>offset address: ACH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | **UOVLVL** [15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register is unsigned num, and 16bit is aligned to high bits of 22-bit ADC waveform register, compare ADC waveform register absolute value with UOVLVL and within setted **OVCyc**, if ADC waveform exceed UOVLVL register , UOVIF flag will be updated in the last **OVCyc** half period and Upeak will be updated after the setted **OVCyc** half periods.

Upd_Half register is in 22-bit signed binary complement format. It aligned to sampled data. UOVLVL register is 16-bit unsigned data, aligned to high bit of waveform sampled absolute data.

Peak value is 1.414 times of rms. So:

Upd_Half = Urms*1.414/2^5

If set UOVLVL threshold to 120% peak value, then:

UOVLVL= 1.2* Upd_Half /2^3 = 1.2*Urms*1.414/2^8

| OvCyc | | | Base address: 0x40013800<br>offset address: B0H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | **OvCyc** [15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The default value of this register is 0x0001. Be careful:

16 bit unsigned number, used to set Udet detection data length, that is, voltage channel Peak/SAG value, in the detection of the data set after the length of the peak.

The 1 LSB correspond to half cycles, and when the detection value exceeds the set threshold, an interrupt flag is given.

| IOVLVL | | | Base address: 0x40013800<br>offset address: B4H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | **IOVLVL** [15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:This register is unsigned num, and 16bit is aligned to high bits of 22-bit ADC waveform register, compare ADC waveform register absolute value with IOVLVL and within setted **IpeakCyc**, if ADC waveform exceed IOVLVL register , UOVIF flag will be updated in the last **IpeakCyc** half period and Ipeak will be updated after the

setted **IpeakCyc** half periods.

Ipeak register is in 22-bit signed binary complement format. It aligned to sampled sata. IOVLVL register is 16-bit unsigned data, aligned to high bit of waveform sampled absolute data.

Peak value is 1.414 times of rms. So:

Ipeak = Irms*1.414/2^5

If set IOVLVL threshold to 120% peak value, then:

IOVLVL= 1.2* Ipeak /2^3 = 1.2*Irms*1.414/2^8

| **ZXILVL** | | | **Base address：0x40013800**<br>**offset address：B8H** | | | | |
|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12 … 3** | **2** | **1** | **Bit0** |
| **Read:**<br>**Write:** | ZX15 | ZX4 | ZX13 | ZC12…ZC3 | ZC2 | ZC1 | ZC0 |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:Compare current rms with ZXILVL. ZXILVL corresponds to low 16 bits Bit15…bit0 of IRMS.

Zero crossing threshold set register, do not output current zero crossing signal if current rms is less than setted current zero crossing threshold set register. Internal output is always 0.

| **PDataCpH** | | | | **Base address：0x40013800**<br>**offset address：BCH** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | IChk[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **PDataCpL** | | | | **Base address：0x40013800**<br>**offset address：C0H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | IChk[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

During constant calculation, the constant active power high and low 16 bits make the 32-bits constant apparent power, if user enable constant calculation function and LoadDataCp, PDATACP will replace active power for pulse output and energy accumulation

| **QDataCpH** | | **Base address：0x40013800**<br>**offset address：C4H** |
|---|---|---|
| | **Bit15…Bit0** | |
| **Read:** | IChk[7:0] | |

| Write: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| QDataCpL | | | Base address: 0x40013800<br>offset address: C8H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | IChk[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

During constant calculation, the constant reactive power high and low 16 bits make the 32-bits constant apparent power, if user enable constant calculation function and LoadDataCp,QDATACP will replace reactive power for pulse output and energy accumulation

| SDataCpH | | | Base address: 0x40013800<br>offset address: CCH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | IChk[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| SDataCpL | | | Base address: 0x40013800<br>offset address: D0H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | IChk[7:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Dering constant calculation, the constant apparent power high and low 16 bits make the 32-bits constant apparent power, if user enable constant calculation function and LoadDataCp, SDATACP will replace apparent power for pulse output and energy accumulation

| Filter control register （filter coefficient control register） | | | Base address: 0x40013800<br>offset address: D4H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit8 |
| Read:<br>Write: | X | X | X | RosiPgaCtr1 | RosiPgaCtr0 | ZXsourceSel | PahseShiftCtr1 | PahseShiftCtr0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | Bit7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit0 |
| Read:<br>Write: | RosiCtr1 | RosiCtr0 | HCtl2 | HCtl1 | HCtl0 | LCtl2 | LCtl1 | LCtl0 |

| **Reset:** | 1 | 1 | 1 | 1 | 0 | 1 | **1** | **1** |
|---|---|---|---|---|---|---|---|---|

| **Bits** | **function description** |
|---|---|
| RosiPgaCtr[1:0] | **Gain of two current channel Rogowski coil control gear**<br><br>RosiPgaCtr1 \| RosiPgaCtr0 \| Rosi gain<br>0 \| 0 \| 1（default）<br>0 \| 1 \| 2<br>1 \| 0 \| 4<br>1 \| 1 \| 8 |
| ZXsourceSel | **Current voltage zero crossing source selection:：**<br>= 0，select the data filtered by high pass filter as voltage current source, = 1，select the data 90de phase lags for the voltage zero crossing source ,data after rosi as current zero crossing source |
| PahseShiftCtr[1:0] | **90de phase shift filter coefficient control gear**<br>PahseShiftCtr1 \| PahseShiftCtr0 \| Phase shift filter gear<br>0 \| 0 \| 11<br>0 \| 1 \| 12<br>1 \| 0 \| 13(default)<br>1 \| 1 \| 14 |
| RosiCtr[1:0] | **Gain of two current channel Rogowski coil filter coefficient control gear**<br>RosiCtr1 \| RosiCtr0 \| Rosi filter gear<br>0 \| 0 \| 11<br>0 \| 1 \| 12<br>1 \| 0 \| 13<br>1 \| 1 \| 14(default) |
| HCtl[2:0] | **Internal coefficient1 selection control bit**<br>HpfCtl2 \| HpfCtl1 \| HpfCtl0 \| High pass filter<br>0 \| 0 \| 0 \| 4<br>0 \| 0 \| 1 \| 5<br>0 \| 1 \| 0 \| 6<br>0 \| 1 \| 1 \| 7<br>1 \| 0 \| 0 \| 8<br>1 \| 0 \| 1 \| 9<br>1 \| 1 \| 0 \| 10（default）<br>1 \| 1 \| 1 \| 11 |
| LCtl[2:0] | **Internal coefficient2 selection control bit**<br>HpfCtl2 \| HpfCtl1 \| HpfCtl0 \| Low pass coefficient |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 4 |
| 0 | 0 | 1 | 5 |
| 0 | 1 | 0 | 6 |
| 0 | 1 | 1 | 7 |
| 1 | 0 | 0 | 8 |
| 1 | 0 | 1 | 9 |
| 1 | 1 | 0 | 10 |
| 1 | 1 | 1 | 11（default） |

| **TUgain** | | | **Base address：0x40013800**<br>**offset address：D8H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | IChk[7:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **TI1gain** | | | **Base address：0x40013800**<br>**offset address：DCH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | IChk[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **TI2gain** | | | **Base address：0x40013800**<br>**offset address：E0H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | TI2gain [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

Temperature manual compensation registers TUgain, TI1gain, and TI2gain. To compensate the influence the temperature has on three channels.

Added manual gain correction for 3 channels of ADC, aims at the ADC changes along with VREF and peripheral changes which is caused by temperature changes.

For example:

Given: U channel rms changes err% if affected by Vref and peripheral. It is rms that changes err%.

Calculation formula:

If gain>=0, TUgain=INT[gain*2^15]

Else if gain<0, TUgain=INT[2^16+gain*2^15]

Temperature auto-compensation curve coefficient register

| **UTCcoffA** | | | **Base address: 0x40013800**<br>**offset address: E4H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | UTCcoffA[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Quadratic term coefficient of U channel temperature auto-compensation curve

| **UTCcoffB** | | | **Base address: 0x40013800**<br>**offset address: E8H** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | UTCcoffB [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Monomial coefficient of U channel temperature auto-compensation curve

Bit15-bit13 is the integer part of monomial, MSB for sign, indicate ±3(it should be -4~+3),bit12-bit0 is the decimal part of monomial, Msb for sign, indicate ±1.

So , UTCoffB = Bit15（sign）bit14—bit13 + bit12（sign）bit11—bit0

For example:

If UTCoffB is 0x1400, the coefficient is：Integer part is 0, decimal part is -（2^13-0x1400）/2^12= -0.75

Then UTCoffB = 0 + -0.75 = -0.75

If UTCoffB is 0x2400, the coefficient is：Integer part is 1, decimal part is 0x400/2^12 = 0.25

Then UTCoffB = 1 + 0.25 = 1.25

| **UTCcoffC** | | | **Base address: 0x40013800**<br>**offset address: ECH** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | UTCcoffC[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Constant of U channel temperature auto-compensation curve

If temperature auto-compensation is enabled, TPScode equals to high 10 bits of chip internal TPS register.

TPS code is in binary complement format.

Temperature auto-compensation voltage output:

| | | | | | |
|---|---|---|---|---|---|
| A0H | AutoUgain | R | 2 | 0x000000 | coefficient of U channel temperature |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | auto-compensation |

Actual internal calculation formula:

AutoUgain = UTCcoffA/32768* (TPSCODE/4)^2 + UTCcoffB* (TPSCODE/4) + UTCcoffC/32768

Note: UTCcoffB contains integer part and decimal part, they has their own sign bit, UTCoffC is signed num,the range -32768-32767.

| **I1TCcoffA** | | | | **Base address: 0x40013800**<br>**offset address: F0H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | I1TCcoffA [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Quadratic term coefficient of I1 channel temperature auto-compensation curve

| **I1TCcoffB** | | | | **Base address: 0x40013800**<br>**offset address: F4H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | I1TCcoffB [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Monomial coefficient of I1 channel temperature auto-compensation curve

Bit15-bit13 is the integer part of monomial, MSB for sign, indicate ±3(it should be -4~+3),bit12-bit0 is the decimal part of monomial, MSB for sign, indicate ±1.

| **I1TCcoffC** | | | | **Base address: 0x40013800**<br>**offset address: F8H** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | I1TCcoffC [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Constant of I1 channel temperature auto-compensation curve

| **I2TCcoffA** | | | | **Base address: 0x40013800**<br>**offset address: FCH** | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | I2TCcoffA [15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Quadratic term coefficient of I2 channel temperature auto-compensation curve

| **I2TCcoffB** | **Base address: 0x40013800**<br>**offset address: 100H** |
|---|---|

| | Bit15…Bit0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Read:** | I2TCcoffB [15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Monomial coefficient of I2 channel temperature auto-compensation curve

Bit15-bit13 is the integer part of monomial, MSB for sign, indicate ±3(it should be -4~+3),bit12-bit0 is the decimal part of monomial, MSB for sign, indicate ±1.

| **I2TCcoffC** | | | **Base address:** 0x40013800<br>**offset address:** 104H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | I2TCcoffC [15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Constant of I1 channel temperature auto-compensation curve

Note:the calculation method of two channels current temperature auto-compensation coefficient is same with voltage channels.

| LoadDataCp | | | **Base address:** 0x40013800<br>**offset address:** 108H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:** | LoadDataCp[15:0] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

During constant calculation, if user writes0x00BC to LoadDataCp, constant power P/Q/SDATACP will be loaded to accumulation source.

| **BufferStart** | | | **Base address:** 0x40013800<br>**offset address:** 10CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15** | **14** | **13** | **12** | **11** | **10** | **9** | **Bit8** |
| **Read:** | BufferStar[15:8] | | | | | | | |
| **Write:** | | | | | | | | |
| **Reset:** | **0** | **0** | **0** | **0** | **0** | 0 | **0** | **0** |
| | **Bit7** | **6** | **5** | **4** | **3** | **2** | **1** | **Bit0** |
| **Read:** | BufferStar[7:4] | | | | X | X | X | X |
| **Write:** | | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: default value is 0x0000. Cache starts control register. After the cache is started, the register is reset to 0x0000.

The BIT15--4 boot bit writes to the 0CCCH and starts the buffer cache

| Bits | Function Description |
|---|---|
| bit[3:2] | Buf dec gain<br>Bit3 / Bit2 / Gain<br>0 / 0 / 1<br>0 / 1 / 2<br>1 / 0 / 4<br>1 / 1 / 8 |
| bit[1:0] | Buffer source select<br>Bit1 / Bit0 / Source select<br>0 / 0 / U<br>0 / 1 / I1<br>1 / 0 / I2<br>1 / 1 / U |

| **BufferCoff** | | | | **Base address:** 0x40013800<br>**offset address:** 110H | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit8…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | BufferCoff [8:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

When the sampling signal is 50Hz, this register is 0x0100 (Default)

Cache data manual adjustment factor bit8-0, write range 196 - 284, greater than 284, value 284 less than 196, value 196

Coefficient calculation method: the actual line frequency.

$$BufferCoff = \frac{EMUCLK}{Freq \times 64}$$

When EMUCLK is 819.2KHz or 409.6KHz, and Freq is line frequency of signal, the frequency range within 45Hz ~ 65Hz, beyond this range does not apply.

The user can manually adjust this factor to obtain cached data for two full cycle signals.

| **SRSTREG** | | | | **Base address:** 0x40013800<br>**offset address:** 114H | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Bit15…Bit0** | | | | | | | |
| **Read:**<br>**Write:** | IChk[15:0] | | | | | | | |
| **Reset:** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:After written 0x55, all table correction parameter will be reset; after written 0xAA, the whole EMU module will be reset.

| PFCntN | | | Base address: 0x40013800<br>offset address: 118H | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | PFCntN [15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Default value is 0x0000.

| QFCntN | | | Base address: 0x40013800<br>offset address: 11CH | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit15…Bit0 | | | | | | | |
| Read:<br>Write: | QFCntN [15:0] | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Default value is 0x0000.

When the count value of the fast pulse count register PFCntN/QFCntN is greater than or equal to HFconst, the corresponding PF/QF will have a pulse overflow, and the value of the energy register will be increased by 1 accordingly. In order to prevent the electric power loss, power down when the register PFCntN/QFCntN can save a pulse energy value of the MCU can be read back and save this value, then in the next power MCU will these values to write to PFCntN/QFCntN to participate in energy accumulation.

## 22.4 Table correction process

- **Step1. high frequency pulse constant set（the meters of same batch need one HFCONST）**

Adjust the user sampled table error accuracy to 10% through HFConst. Two methods of calculation

**First Scheme：**

HFCONST register is 0x0080 by default.

The initial error users observe electrical meter is Err%,adjust the error to 10% by following formula：

HFCONST = 0x0080 * (1+ Err%)

**For example：**

Electrical meters constant is set to 3200,power factor is 1, HFCONST register is 0x0080 by default.the observed error the standard meter shows is 52.8%.

According to formula：HFCONST = 0x0080 * (1+ Err%)

We get：HFCONST = 0x0080 * (1+52.8%) = 0x00C3

Write 0x00C3 to FCONST（4CH） register of HT502X：

Then the displayed error of standard meter wills less than 10%.

**Second scheme：**

If fs = 12.8KHz (Femu = 819.2KHz，OSR = 64，fs = Femu/OSR),
HFConst＝2.332*Vu*Vi*10^10/(EC*Un*Ib)
Vu：Rated voltage input, voltage of voltage channel( voltage on the pin×amplification factor）
Vi：Rated current input, current of current channel( current on the pin×amplification factor）
Un：Rated voltage input
Ib：Rated current input
EC：Electrical meter constant
HFConst is in proportion to femu.

**For example：**
Electrical meters constant is set to 3200,power factor is 1.
Un is 220V, Ib Ib is 5A, Vu is 0.22V
Vi is 1.75mA, internal current channel gain is 16 times, So Vi*16 = 28mV
According to formula：HFConst＝2.332*Vu*Vi*10^10/(EC*Un*Ib)
We get：HFConst = 2.332*0.22 *0.028*10^10 / (3200*220*5) = 0x0028
Write 0x0028 to FCONST（4CH） register of HT502X：
Then the displayed error of standard meter wills less than 10%.

- **Step 2．Active reactive and apparent power correction of first channel**

They are calculated when rated input and power factor is 1 according to active power. Generally, write the same value to active reactive and apparent power gain.
given：
The error readed from standard meter is err%
Calculation formula:

$$Pgain = \frac{-err\%}{1+err\%}$$

If Pgain>=0, GP1=INT[Pgain*$2^{15}$]
If Pgain<0, GP1=INT[$2^{16}$+Pgain*$2^{15}$]
And then, the chip automatically writes the value to GQ1 (reactive power gain) and GS1(appear power gain).

**For example：**
Power source output signal of 220v 5A and power factor=1, the error readed from standard meter is 3.8%
Then Pgain＝-0.038/(1+0.038)=-0.0366
This num is smaller than 0, converted to complement and -0.0366*2^15+2^16=0xFB50H
Write 0xFB50H to all of GP1(0CH) register to complete the resistive gain correction

- **Step 3．Phase correction of first channel**

Compensate the phase after gain correction. Correct it when power factor is 0.5L.
Given: the error readed from standard meter when power factor is 0.5L is err%
Phase compensation formula：

$$\theta = \frac{-err\%}{1.732}$$

If $\theta \geq 0$, $GPhs = \theta * 2^{15}$

Else if $\theta < 0$, $GPhs = 2^{16} + \theta * 2^{15}$

**For example：**

After resistive gain correction, change the power factor to 0.5L, and the error readed from standard meter is -0.4%, then

θ=-(-0.004)/1.732=0.0023

Gphs1=0.0023*2^15=75.3

Round to0x4BH, write it to angle correction register Gphs1(18H).

- **Step 4．Gain correction of Current channe2(necessary for power stealing prevention)**

During power stealing prevention, it need compare the rms or power of to channels, so the register of channel1and cannel2 should be same if current inputs are same

Correct gain of channel2 through register I2GAIN(30H) to ensure the register values are same when the current inputs are same

Given the input same rated current, channel1 rms register reading is I1rms, channel2 rms register reading is I2rms, and channel2 active power is Power2

**If choose current stealing prevention:**

Then Gain＝I1rms/I2rms – 1

**If choose power stealing prevention:**

Then Gain＝Power1/Power2 - 1

If Gain>=0，I2Gain=Gain*2^15; if Gain<0，I2Gain＝Gain*2^15+2^16

- **Step 5．Second channel gain correction and phase correction**

They are similar to that of first channel

- **Step 6．Poffset correct（small signal active power correction）**

After the steps above, the meter error is nearly 0 when Ib=100%

Observe the small signal x%Ib（5%，2%）point, we get the meter error is Err%，

The active power standard meter shows at the x%Ib point is Preal if it is resistive

As formula：

$$P1\_offset = \frac{\text{Preal*EC*HFConst*2\^28 *(-Err\%)}}{\text{2.304* 10\^10}}$$

**For example：**

Rate voltage 220V,rate current(Ib) 5A,meter constant is 3200,fast pulse register（HFCONST）reading 0x66；

The error is corrected to nearly 0 if Ib = 100%, the meter error is 0.5% at the 5% small signal point

The output power readed from standard meter is 55.02 at 5% small signal point (Preal is displayed by standard meter)

We get:

Poffset = （Preal×EC×HFCONST×2^28×(-Err%)）/ （2.3×10^10）

= (55.02×3200×102×2^28×(-0.5%)) / (2.3×10^10)

= -1047.98

Since Poffset < 0, value write to register P1offset is 2^16 + Poffset =63444

Round to 64488 or 0xFBE8.

Write 0xF7D4 to register P1offset(78H) of HT502X

- **Step 7．IRMS gain, URMS gain and power gain convention coefficient correction of two channels**

There is no corresponding register for these registers and user need to get these value through calculation

**For example:**

Take current channel1 for example, the standard output of current channel1 is 5A, the reading of current channel1 rms register RMS_I1（3CH） is 0x03BA55, if 5A is supposed to be displayed on the LCD, the convention coefficient between these is calculated as shown: K = 5/0x03BA55 = 2.046*10^(-5)

Where K is convention coefficient and the K times RMS_I1 reading is correct current.

See rms output and power parameter output chapter for detailed information

# 24. Electrical specification

## 24.1. DC parameter

| symbol | Parameter explanation | Test condition | minimum | typical | maximum | unit |
|---|---|---|---|---|---|---|
| VCC | Input power | The voltage on VCC pin | 2.2 | 3.3 | 5.5 | V |
| Vih | High level input voltage | Reset pin, TEST pin | 0.8VCC | | | V |
| Vih | High level input voltage | PC9,PC10,PE0,PE3,PE6 | 0.6VCC | | | V |
| Vih | High level input voltage | Except the power and ground,pins except for Reset，TEST，PC9,PC10,PE0,PE3, PE6 | 0.7VCC | | | V |
| Vih | low level input voltage | Pins except for power and ground | | | 0.2VCC | V |
| Ioh | High level | VCC=3.3V | 10 | | | mA |

| | | | | | | |
|---|---|---|---|---|---|---|
| | output current | Lower voltage on I/O Vio to 0.9VCC<br>Test pins：PA.6，PA.7，PA.8，PC.0 | | | | |
| Iol | Low level output current | VCC=3.3V<br>Increase voltage on I/O Vio to 0.1VCC<br>Test pins：PA.6，PA.7，PA.8，PC.0 | 20 | | | mA |
| Ioh | High level output current | VCC=3.3V<br>Lower voltage on I/O Vio to 0.9VCC<br>Test pins：<br>PD.4,PD.5,PD.6,PD.7<br>PC.5,PC.6,PC.7,PC.8,PC.9,<br>PC.10,PC.11,PE.1 | 5 | | | mA |
| Iol | Low level output current | VCC=3.3V<br>Lower voltage on I/O Vio to 0.9VCC<br>Test pins：<br>PD.4,PD.5,PD.6,PD.7<br>PC.5,PC.6,PC.7,PC.8,PC.9,<br>PC.10,PC.11,PE.1 | 5 | | | mA |
| Ioh | High level output current | VCC=3.3V<br>Lower voltage on I/O Vio to 0.9VCC<br>Test pins：<br>Pins except for PA.6, PA.7, PA.8, PC.0,<br>PD.4,PD.5,PD.6,PD.7<br>PC.5,PC.6,PC.7,PC.8,PC.9,<br>PC.10,PC.11,PE.1 | 3 | | | mA |
| Iol | Low level output current | VCC=3.3V<br>Lower voltage on I/O Vio to 0.9VCC<br>Test pins：<br>Pins except for PA.6, PA.7, PA.8, PC.0,<br>PD.4,PD.5,PD.6,PD.7<br>PC.5,PC.6,PC.7,PC.8,PC.9,<br>PC.10,PC.11,PE.1 | 3 | | | mA |

## 24.2. Ultimate parameter

| symbol | Parameter explanation | Test condition | minimum | maximum | unit |
|---|---|---|---|---|---|
| VCC | Input power | Voltage on input power pin | 2.2 | 5.5 | V |
| Vi | Input voltage | All digital pins | 0 | 5.5 | V |
| Via | ADC input limit voltage | Test ADC pin input voltage,voltage exceed this probably damage pin | 0 | 3.0 | V |
| Vib | VBAT input limit voltage | Test VBAT pin input voltage,voltage exceed this probably damage pin | 0 | 5.0 | V |
| Idd | Input current | VCC power pin | | 50 | mA |
| Iss | Ground current | All GND pins | | 50 | mA |
| Tstg | Restore temperature | Limit restore temperature of chip | -65 | +150 | ℃ |
| Vesd | Static ESD （HBM） | ADC input pins | -6000 | +6000 | V |
| | | All chip pins | -8000 | +8000 | |

Note: the VCC input system voltage can stands for 1.5 times nominal voltage.

## 24.3. Power consumption parameter

| Test condition explanation | | minimum | typical | maximum | unit |
|---|---|---|---|---|---|
| **Test condition：follow power consumption are gained by testing under 3.3V power supply** | | | | | |
| The minimum power in Hold mode,(specified power depends on the modules users opens in Hold mode) | -45℃ | | 2.2 | | uA |
| | Normal temperature | 1 | 3.9 | 7.5 | |
| | 85℃ | | 34.3 | | |
| Minimum power in Sleep mode<br>Power when open LCD when chip is in low frequency 32768Hz<br>Operating at high frequency RC, high frequency RC is set to 9.6M, close CLKOUT | -45℃ | | 1.9 | | uA |
| | Normal temperature | 1 | 2.9 | 6.9 | |
| | 85℃ | | 16.0 | | |
| Operating at high frequency RC, high frequency RC is set to 8M, close CLKOUT | Normal temperature | | 48 | | uA |
| Operating at high frequency RC, high frequency RC is set to 4.8M, close CLKOUT | Normal temperature | | 12.2~ | | uA |
| Operating at high frequency RC, high frequency RC is set to 2.4M, close CLKOUT | Normal temperature | | 2.85 /1.52 | | mA |
| Operating at high frequency RC, high frequency RC is set to 1.2M, close CLKOUT | Normal temperature | | 2.19 /0.86 | | mA |
| Operating at high frequency RC, high | Normal | | 1.75 | | mA |

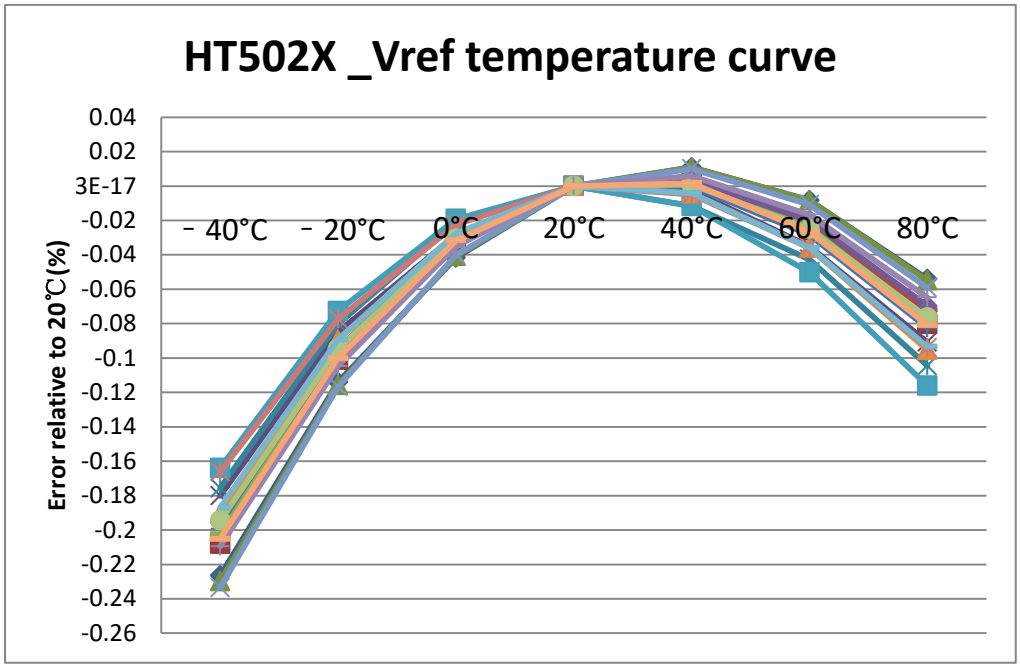
| | | | | | |
|---|---|---|---|---|---|
| frequency RC is set to 600k, close CLKOUT | temperature | | /0.53 | | |
| Operating at high frequency RC, high frequency RC is set to 300k, close CLKOUT | Normal temperature | | 1.42 /0.35 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 39.32M, all digital modules are opened | Normal temperature | | 1.25 /0.27 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 19.66M, all digital modules are opened | Normal temperature | | 1.17 /0.23 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 9.83M, all digital modules are opened | Normal temperature | | 1.13 /0.21 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 4.915M, all digital modules are opened | Normal temperature | | 8.76 /5.64 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 2.45M, all digital modules are opened | Normal temperature | | 5.78 /4.03 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 1.22M, all digital modules are opened | Normal temperature | | 3.63 /2.69 | | mA |
| Operating at PLL clock, CPU operating at PLL clock 620K, all digital modules are opened | Normal temperature | | 2.53 /1.00 | | mA |
| EMU singale channel ADC power（offset current is 1.5uA | Normal temperature | | 1.96 /0.66 | | mA |
| EMU singale channel ADC power（ffset current is 10uA） | Normal temperature | | 1.68 /0.48 | | mA |
| Minimum power in Sleep mode | Normal temperature | | 1.54 /0.39 | | mA |
| Power when open LCD when chip is in low frequency 32768Hz | Normal temperature | | 130 | | uA |
| Operating at high frequency RC, high frequency RC is set to 9.6M, close CLKOUT | Normal temperature | | 600 | | uA |

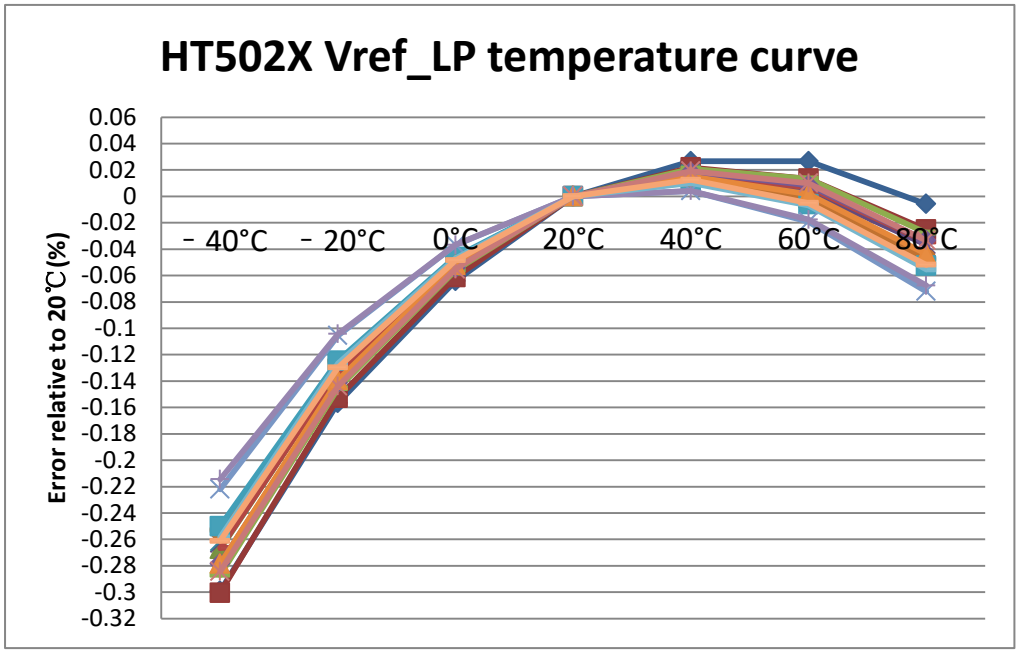
| **Test condition explanation** | **minimum** | **typical** | **maximum** | **unit** |
|---|---|---|---|---|
| **Test condition：follow power consumption are gained by testing under 3.3V power supply** | | | | |
| High frequency RC module power consumption（HRC） | | 116 | | uA |
| Low frequency RC module power consumption（LRC） | | 0.9 | | uA |
| PLL module power consumption | | 205 | | uA |
| LCD module power consumption（fast charge mode） | | 6.9 | | uA |
| LCD module power consumption (small current mode) | | 8.3 | | uA |
| LCD module power consumption (big current mode) | | 35.2 | | uA |
| BOR module power consumption | | 8 | | uA |
| LVDIN module | | 7 | | uA |

## 24.4. Measurement ADC parameter

| stmbol | Parameter introduction | Test condition | minimum | typical | maximum | unit |
|---|---|---|---|---|---|---|
| Via | ADC input voltage | | 0 | | ±800 | mVp |
| Fc | ADC conversion frequency | VCC=3.3V | 1/16 | | 1 | Hz |
| Res | resolution | VCC=3.3V | | 0.012 | | mv/LSB |
| Vref TC | Measurement reference temperature coefficient | | | 10 | | ppm/℃ |
| VrefLP | Auxiliary measuring reference voltage | | | 1.20000 | | V |
| VrefLPTC | Auxiliary measuring reference temperature coefficient | | 15 | 20 | 30 | ppm/℃ |

Temperature drift curve:






## 24.5. ADC of TBS module test parameter

| symbol | Parameter introduction | Test condition | minimum | typical | maximum | unit |
|---|---|---|---|---|---|---|
| VCC | Operating voltage | | 2.4 | | 5.5 | V |
| Iadc | Operating current | | | 350 | | uA |
| Res | resolution | | | | 16 | Bit |
| ENOB | Significant bit | | | 12 | | Bit |
| Vrefadc | reference voltage （Internal） | Normal temperature | -2% | 1.258V | +2% | |
| VREF TC | Vref temperature coefficient | | | 100ppm | | /℃ |
| Fsample | Operating frequency | | | 32.768k | | Hz |
| Cin | Input capacitor | | | 0.515 | | pF |
| Rin | Equivalent input resistance | | | 30M | | Ω |
| ADCINx channels | | | | | | |
| Vadcin | ADC input range | Vrefadc=1.258V, Normal temperature | 0 | | 800 | mV |
| Resadcin | resolution | theory | | 0.0128 | | mV/LSB |
| | Calculation formula | Actual fitting | VADCIN0 = 0.0128*ADC0DAT+425.5623; VADCIN0 is actual ADC measuring voltage. | | | mV |

| | | | | | | |
|---|---|---|---|---|---|---|
| Temperature sensor TPS channel | | | | | | |
| Adjtps | Temperature detect accuracy | | | +-1 | | ℃ |
| Restps | resolution | theory | | 0.00278 | | ℃/LSB |
| | Calibrating formular | | Tr =12.9852 - TMPDAT *0.0028<br>Tr is the actual temperature（℃） | | | ℃ |
| VBAT channel | | | | | | |
| Vbatin | VBAT input range | | 1.0 | | 5.5 | V |
| Resvbat | resolution | theory | | 0.08958 | | mV/LSB |
| Rinbat | VBAT equivalent input resistance | | | 220k | | Ω |
| Div | Equivalent voltage divider coefficient | | | 1/7 | | |
| | Calibrating formular | Actual fitting | VBAT = 0.0897*VBATDAT+2946.2662；<br>VBAT is the actual battery voltage | | | mV |
| VDD channel | | | | | | |
| VDDin | VBAT input range | | 2.4 | | 5.5 | V |
| Restps | resolution | theory | | 0.08958 | | mV/LSB |
| Rinvdd | VBAT equivalent input resistance | | | 42k | | Ω |
| Div | Equivalent voltage dividor coefficient | | | 1/7 | | |
| | Calibrating formular | Actual fitting | VDD = 0.0897*VDDDAT+ 2998.1189；<br>VDD is the actual VDD voltage | | | mV |

Note: when use VBAT and VDD channel, the operating current is the sum of ADC operating current and input resistance to the ground current.

## 24.6. Interval RC analog parameters

| **symbol** | **Parameter introduction** | **Test condition** | **minimum** | **typical** | **maximum** | **unit** |
|---|---|---|---|---|---|---|
| Via | VCC input voltage | @PVT | 1.2 | 5 | 5.5 | V |
| Fc | ADC conversion frequency | @TT | | 1 | | V |
| Res | resolution | | | 400 | 600 | nA |
| $F_{i\text{-}nor}$ | Output initial frequency | | 18 | 32 | 40 | KHz |

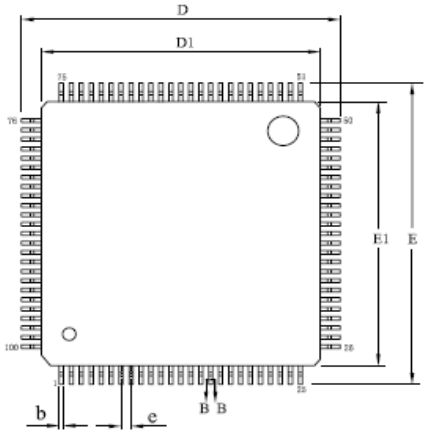
| | | | | | | |
|---|---|---|---|---|---|---|
| $F_{i\text{-}high}$ | Full temperature output initial frequency | | 15 | 32 | 46 | KHz |
| $V_{raise}$ | Minimum operating voltage on power | @VCC=3V | | 1.6(25℃)<br>1.3(-45℃)<br>1.3(85℃) | | V |
| $V_{fall}$ | Minimum operating voltage for power down | @VCC=3V | | 1.4(25℃)<br>1(-45℃)<br>1.1(85℃) | | V |

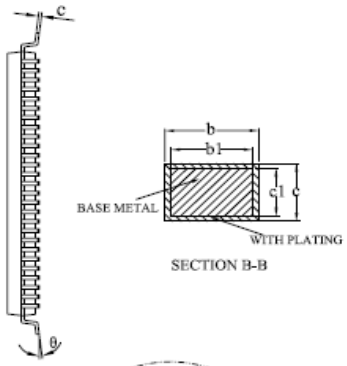
## 24.7. IO interface performance index

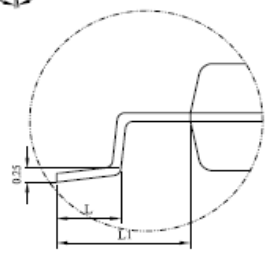
| | -40℃ | 27℃ | 90℃ |
|---|---|---|---|
| Isink@0.1VCC | 7mA | 5mA | 4mA |
| | 14mA | 10mA | 8mA |
| | 56mA | 40mA | 32mA |
| Isource@0.9VCC | 5mA | 4mA | 3.5mA |
| | 10mA | 8mA | 7mA |
| | 15mA | 12mA | 10.5mA |
| The total Isouce cannot exceed 50mA,total Isink cannot exceed 120mA | | | |

# Package

**HT5023：LQFP100**



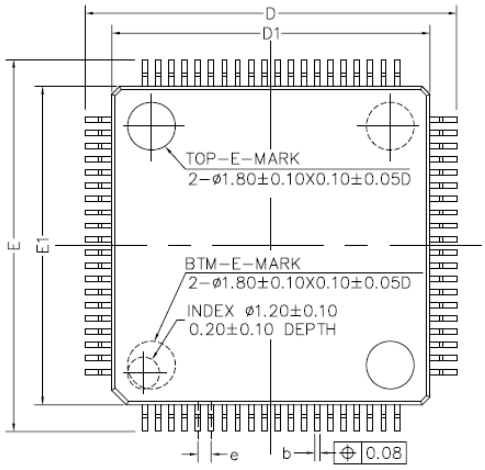



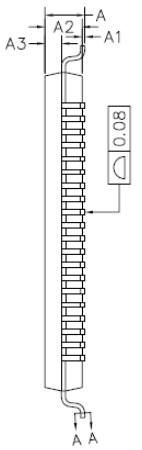




DETAIL: F

| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | — | — | 1.60 |
| A1 | 0.05 | 0.15 | 0.25 |
| A2 | 1.30 | 1.40 | 1.50 |
| A3 | 0.54 | 0.64 | 0.74 |
| b | 0.19 | — | 0.27 |
| b1 | 0.18 | 0.20 | 0.23 |
| c | 0.13 | — | 0.18 |
| c1 | 0.12 | 0.13 | 0.14 |
| D | 15.80 | 16.00 | 16.20 |
| D1 | 13.80 | 14.00 | 14.20 |
| E | 15.80 | 16.00 | 16.20 |
| E1 | 13.80 | 14.00 | 14.20 |
| e | 0.50BSC | | |
| L | 0.40 | 0.60 | 0.80 |
| L1 | 1.00BSC | | |
| θ | 0 | — | 8° |
| L/F载体尺寸 (mil) | 256*256 | | |
| | 276*276 | | |
| | 230*230 | | |
| | 354*354 | | |

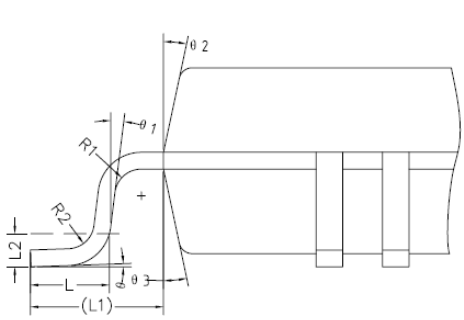
**HT5025：LQFP80**

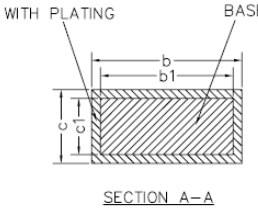



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

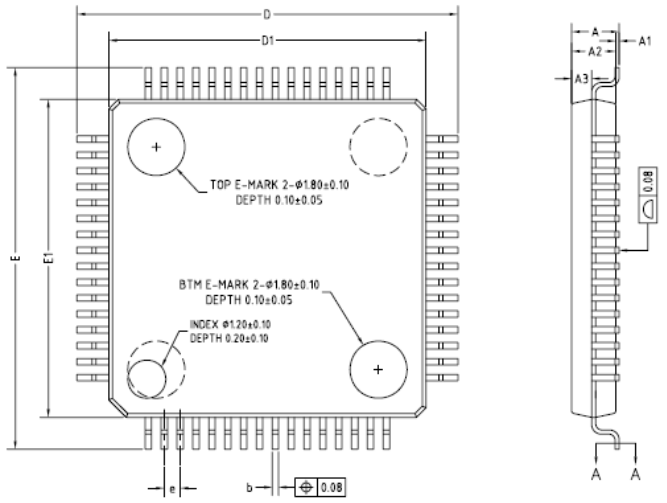
| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | – | – | 1.60 |
| A1 | 0.05 | – | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | – | 0.27 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | – | 0.18 |
| c1 | 0.12 | 0.127 | 0.134 |
| D | 13.80 | 14.00 | 14.20 |
| D1 | 11.90 | 12.00 | 12.10 |
| E | 13.80 | 14.00 | 14.20 |
| E1 | 11.90 | 12.00 | 12.10 |
| e | 0.40 | 0.50 | 0.60 |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00REF | | |
| L2 | 0.25BSC | | |
| R1 | 0.08 | – | – |
| R2 | 0.08 | – | 0.20 |
| θ | 0° | 3.5° | 7° |
| θ 1 | 0° | – | – |
| θ 2 | 11° | 12° | 13° |
| θ 3 | 11° | 12° | 13° |



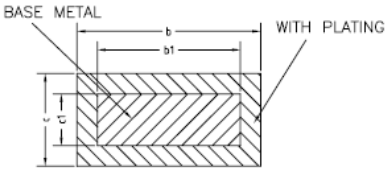

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

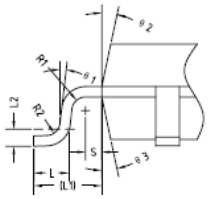
**HT5027：LQFP64**




COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

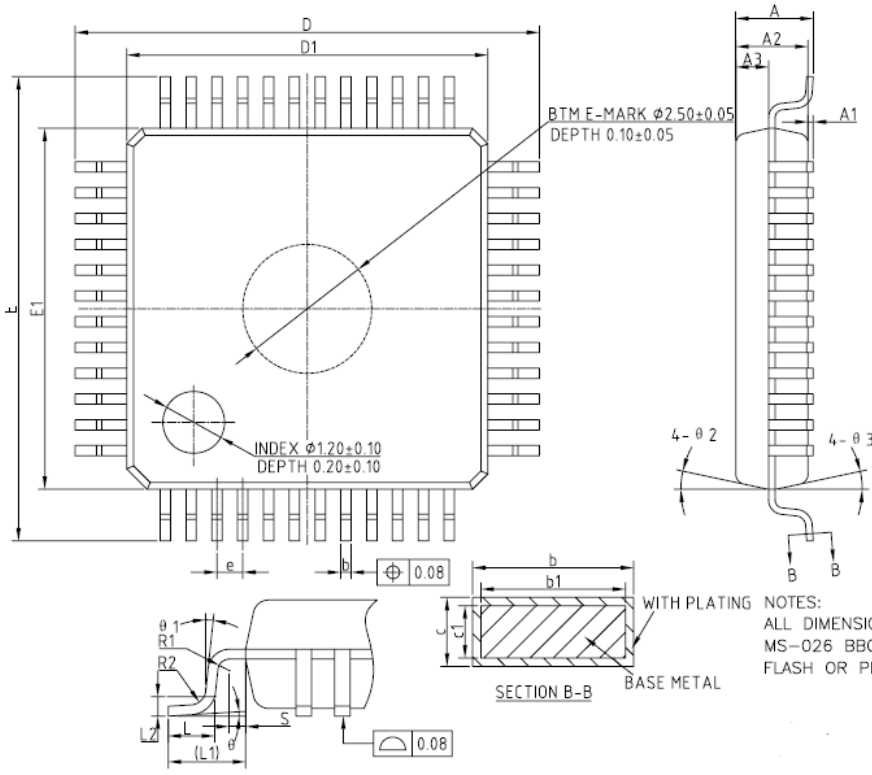
| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | – | – | 1.60 |
| A1 | 0.05 | – | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | – | 0.27 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | – | 0.18 |
| c1 | 0.12 | 0.127 | 0.134 |
| D | 11.80 | 12.00 | 12.20 |
| D1 | 9.90 | 10.00 | 10.10 |
| E | 11.80 | 12.00 | 12.20 |
| E1 | 9.90 | 10.00 | 10.10 |
| e | | 0.50BSC | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | | 1.00REF | |
| L2 | | 0.25BSC | |
| R1 | 0.08 | – | – |
| R2 | 0.08 | – | 0.20 |
| S | 0.20 | – | – |
| θ | 0° | 3.5° | 7° |
| θ 1 | 0° | – | – |
| θ 2 | 11° | 12° | 13° |
| θ 3 | 11° | 12° | 13° |




NOTES:
ALL DIMENSIONS MEET JEDEC STANDARD MS-026 BCD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

HT5029: LQFP48




COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | – | – | 1.60 |
| A1 | 0.05 | – | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | – | 0.27 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | – | 0.18 |
| c1 | 0.12 | 0.127 | 0.134 |
| D | 8.80 | 9.00 | 9.20 |
| D1 | 6.90 | 7.00 | 7.10 |
| E | 8.80 | 9.00 | 9.20 |
| E1 | 6.90 | 7.00 | 7.10 |
| e | | 0.50BSC | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | | 1.00REF | |
| L2 | | 0.25BSC | |
| R1 | 0.08 | – | – |
| R2 | 0.08 | – | 0.20 |
| S | 0.20 | – | – |
| θ | 0° | 3.5° | 7° |
| θ 1 | 0° | – | – |
| θ 2 | 11° | 12° | 13° |
| θ 3 | 11° | 12° | 13° |

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BBC DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.