HY16F196B
HY16F197B
HY16F198B
User's Guide

High-Precision Mixed Signal Micro Controller
4x36 ~ 6x34 LCD Driver
32-bit Low Power Micro Controller
21-bit ENOB ΣΔADC
64KB Flash ROM

Table of Contents

Attention:

1、 HYCON Technology Corp. reserves the right to change the content of this datasheet without further notice. For most up-to-date information, please constantly visit our website: http://www.hycontek.com .

2、 HYCON Technology Corp. is not responsible for problems caused by figures or application circuits narrated herein whose related industrial properties belong to third parties.

3、 Specifications of any HYCON Technology Corp. products detailed or contained herein stipulate the performance, characteristics, and functions of the specified products in the independent state. We does not guarantee of the performance, characteristics, and functions of the specified products as placed in the customer's products or equipment. Constant and sufficient verification and evaluation is highly advised.

4、 Please note the operating conditions of input voltage, output voltage and load current and ensure the IC internal power consumption does not exceed that of package tolerance. HYCON Technology Corp. assumes no responsibility for equipment failures that resulted from using products at values that exceed, even momentarily, rated values listed in products specifications of HYCON products specified herein.

5、 Notwithstanding this product has built-in ESD protection circuit, please do not exert excessive static electricity to protection circuit.

6、 Products specified or contained herein cannot be employed in applications which require extremely high levels of reliability, such as device or equipment affecting the human body, health/medical equipments, security systems, or any apparatus installed in aircrafts and other vehicles.

7、 Despite the fact that HYCON Technology Corp. endeavors to enhance product quality as well as reliability in every possible way, failure or malfunction of semiconductor products may happen. Hence, users are strongly recommended to comply with safety design including redundancy and fire-precaution equipments to prevent any accidents and fires that may follow.

8、 Use of the information described herein for other purposes and/or reproduction or copying without the permission of HYCON Technology Corp. is strictly prohibited.

1. CHIP OVERVIEW

1.1. Brief introduction

The HY16F19xB Series is a low-power and high-precision mixed signal micro controller (MCU) with LCD driver (Liquid Crystal Display), and is applicable to perform high-precision measurement and control; besides, the controller can work in a wide voltage range (2.2V-3.6V) and the clock of the controller can be up to 16MHz; further, the controller has a built-in 64/32/16kbyte embedded Flash ROM and a 8/4/2kbyte SRAM. The HY16F19xB series products integrate a high-precision 24-bit ΣΔ A/D converter, Rail-to-Rail OPAMP, 8-bit D/A converter, Hardware RTC and Multi-Comparator; moreover, the HY16F19xB series products provide high-performance peripheral interfaces, such as the EAURT, SPI, I2C, GPIO and built-in power management system, etc., and support low-voltage detection and multiple peripheral interface wake-up functions. The HY16F19xB series products are of low voltage, low power, low stand-by current, high integrity and high efficient operation, and support the 32-bit micro controller of the C/C++ development platform. Therefore, the HY16F19xB series products can provide various resources for designers to design a low-current and low-cost mixed signal processing system.

The AFE circuit of the controller includes an 8-bit resistance ladder, a Rail-to-Rail OPAMP and a Rail-to-Rail input comparator. In particular, 8-bit resistance ladder has monotonicity, which is a step resistor, and the least significant bit (LSB) is close to 730 ohm and the resistor has the low-temperature coefficient. The Rail-to-Rail OPAMP has an input network, which is applicable for the differential analog circuit configuration, such as integrator, current-to-voltage converter, programmable gain amplifier and successive approximation A/D converter. The Rail-to-Rail input comparator can continuously monitor analog signals by extremely low power consumption; thus, it can serves as a power supply voltage monitor, external wake-up triggering source or capacitive touch key driver.

The 24 bits A/D converter with extremely low noise is embedded. Its maximal output rate is 10.24KSPS, its ENOB (Effective number of bit) is 21, and its minimal resolvable signal is 65nV RMS Noise (Root- Mean- Square). The programmable gain amplifier with low noise is used with the A/D converter together and the maximal gain is 128 times magnification. There is a built-in 4-bit A/D converter at the input of the A/D converter to expand the measurement range.

The power management provides selectable analogous regulating voltage, which can serve as reference voltage source or the power supply of a transducer. The working power source of the core of the CPU is also provided by the internal linear stabilized power supply. The charge pump is used to block the power interference from the system.

64Kbyte embedded Flash ROM can be used to execute programs or store data; the data can be still stored into the Flash ROM even if the Flash ROM is executing a program. A built-in 8kbyte SRAM is provided for the system to use.

The core of the 32-bit high-performance mixed signal micro controller can execute an instruction during each clock cycle, which can be up to 20MIPS (Millions of Instructions per Second) and conform to low power consumption indicator. HYCON Technology provides convenient programming tools for users to write programs by C/C++ language or assembly language in the development platform. The chip has the circuit simulation function and provides a good environment for troubleshooting. The chip can work in 2.2V to 3.6V and -40℃ to 85℃.

1.2. Type description table

The bit type description table of the register

Setting type	Description	Initial value
-	No Use	
RSV.	Reserve	
X	Unknown	
W	Write	
R	Read	
R0	Only Read 0	
R1	Only Read 1	
W0	Only Write 0	
W1	Only Write 0	
RW-0	Read/ Write	Initial 0
RW-1	Read/ Write	Initial 1
R0W-0	Read 0/ Write	Initial 0
R1W-1	Read 1/ Write	Initial 1
R-X	Read	Initial 1 or 0 Unknown
[ ]	Register length	
< >	Register value	
ABC[7:0]	ABC register had 0 to 7bit	
ABC<111>	ABC register had 3bit and value had 111 of binary	
ABC<11x>	x : can be neglected, it can be set as 1 or 0	

2. FUNCTION OVERVIEW

2.1. Block diagram

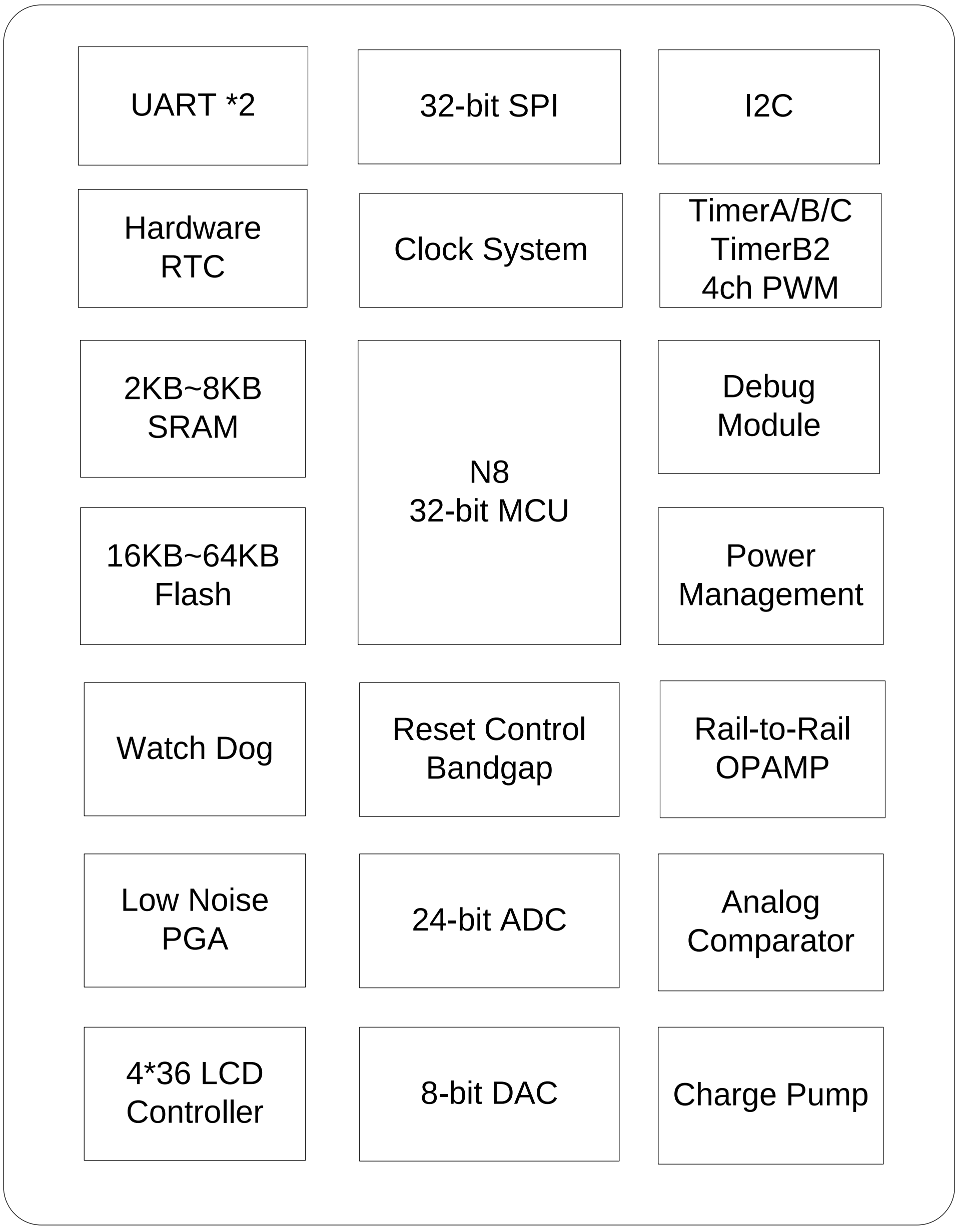


FIG. 2-1 Chip function structure diagram

2.2. CPU core block diagram

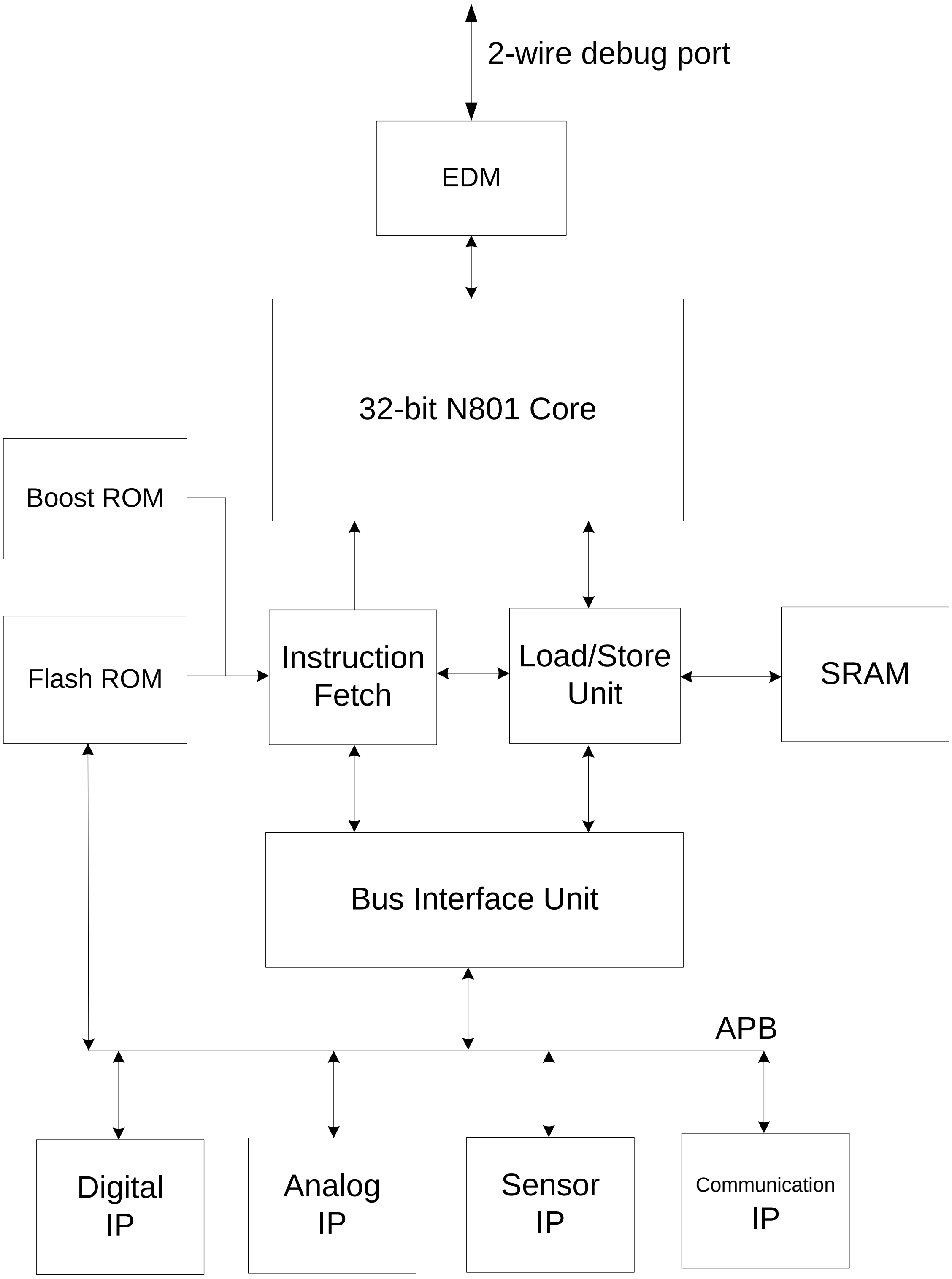


FIG. 2-2 CPU core block diagram

3. MEMORY STRUCTURE

3.1. Memory description

The core of the CPU (Central Processing Unit) of HY16F19xB series products is Andes N801 32-bit CPU. The allocation of the memory addresses of the micro controller is as follows:

0x00000 to 0x01FFF Static random-access memory (SRAM) (8K Byte)
0x40000 to 0x4FFFF SOC Register
0x80000 to 0x81FFF Boot ROM (8K Byte) : Support (Timeout Entry) 4-WIRE and (CheckPin Entry) 5-WIRE UART interface ROM ISP Bootloader function. The default setting is DISABLE status. User can use HY16F Writer(Or IC programming service) to ENABLE ROM ISP Bootloader function.
0x90000 to 0x9FFFF Main Program Flash ROM (64K Byte)

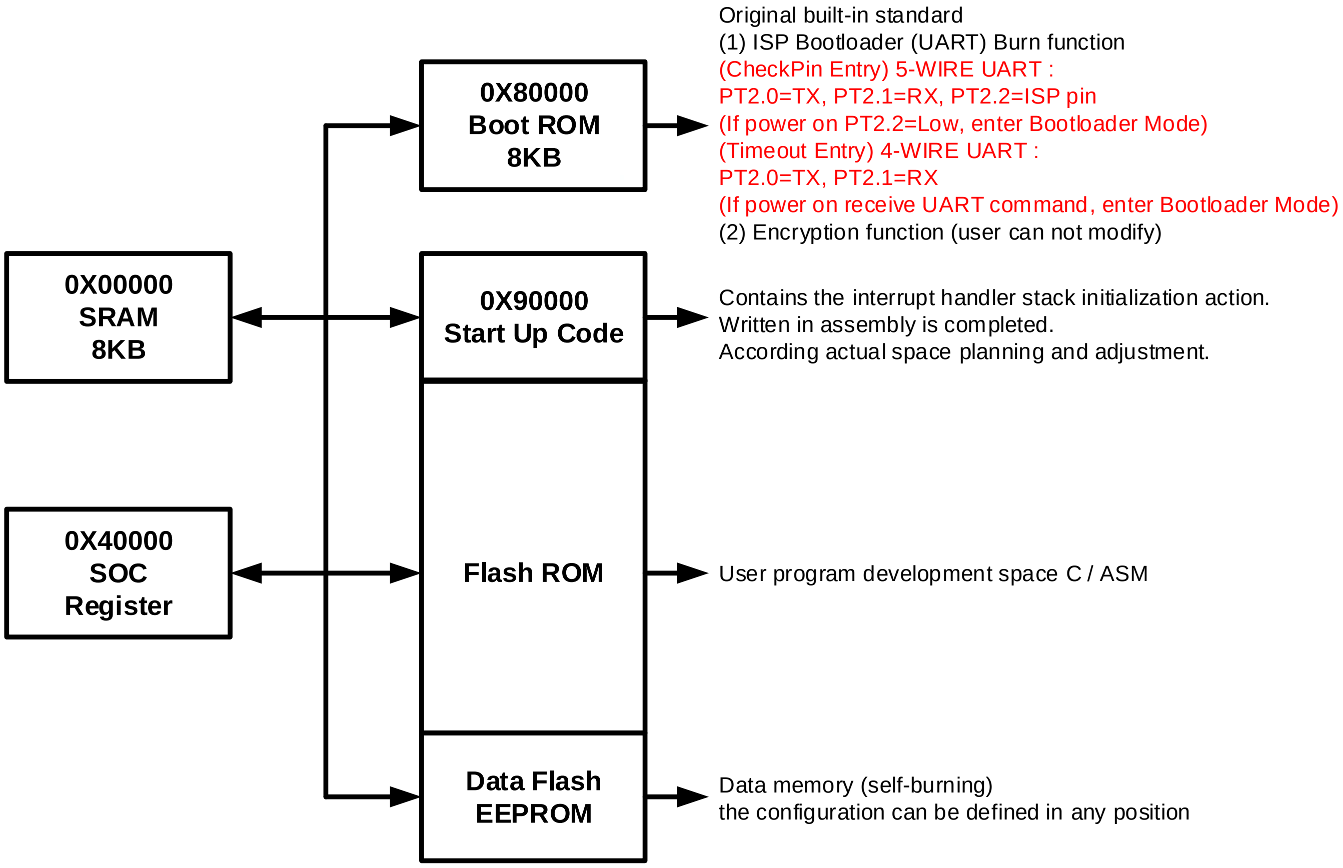

0X80000
Boot ROM
8KB
Original built-in standard
(1) ISP Bootloader (UART) Burn function
(CheckPin Entry) 5-WIRE UART :
PT2.0=TX, PT2.1=RX, PT2.2=ISP pin
(If power on PT2.2=Low, enter Bootloader Mode)
(Timeout Entry) 4-WIRE UART :
PT2.0=TX, PT2.1=RX
(If power on receive UART command, enter Bootloader Mode)
(2) Encryption function (user can not modify)
0X00000
SRAM
8KB
0X90000
Start Up Code
Contains the interrupt handler stack initialization action.
Written in assembly is completed.
According actual space planning and adjustment.
0X40000
SOC
Register
Flash ROM
User program development space C / ASM
Data Flash
EEPROM
Data memory (self-burning)
the configuration can be defined in any position

FIG. 3-1 Memory address allocation diagram

3.2. Memory address

The detailed address allocation of the SOC registers of the micro controller is as follows:

Function module	Description	Base Address
INT	Interrupt control Flag	0x40000
SoC	System control register	0x40100
CLK	Clock system control register	0x40300
PMU	Power management control register	0x40400
MC	Memory controller register	0x40600
PIO	GPIO port control register	0x40800
TMR	Timer register	0x40C00
UART	UART mode control register	0x40E00
SPI	SPI mode control register	0x40F00
I2C	I2C mode control register	0x41000
ADC	Analog-to-Digital module control register	0x41100
8-bit resistance ladder	Digital-to-Analog module control register	0x41700
CMP	Comparator network module control register	0x41800
OPAMP	Operational amplifier control register	0x41900
RTC	Real time clock control register)	0x41A00
LCD	LCD driver control register	0x41B00

Table 3-1 SOC registers

Some important registers have MASK bits, as describe in FIG. 3.3. MASK is used to control written-in bits; only when the MASK bit corresponding to the control bit is <1>, the corresponding control bit can be written in, or the written-in operation will be invalid and cannot actually modify the value of the register, as shown in FIG. 3-2.

The total length of a register is 32-bit and most registers have 16 MASK bits. The MASK bits include two 8-bit groups, and each 8-bit group controls corresponding 8 control register bits. According to the content allocation of a register: BIT [31:24] controls BIT [23:16], and BIT [15:8] controls BIT [7:0]. Only when the MASK bit is <1>, the corresponding control bit can be validly written in.

For example, if a user wants to write 101010b in BIT [5:0] and the write value of the register should be: 0011111100101010b, where 00111111b are the MASK bits of BIT [5:0] and can make written-in corresponding control bits valid; and 00101010b are the values written in BIT [5:0].

INT Base Address + 0x10 (0x40010)									
Symbol	INTPT1 (PT1 Interrupt Control Register)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT17IE	PT16IE	PT15IE	PT14IE	PT13IE	PT12IE	PT11IE	PT10IE
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT17IF	PT16IF	PT15IF	PT14IF	PT13IF	PT12IF	PT11IF	PT10IF
RW	R0W-0	RW0-0							

FIG. 3-2 Basic structure of register

3.3. Static random-access memory (SRAM)

HY16F198B has an 8Kbyte SRAM. The initial address is from 0x00000 to 0x01FFF. MCU can select to access one byte, half word or one word. MCU can access one word during each clock cycle.

3.4. Flash ROM

HY16F198B has a 64Kbyte embedded Flash ROM. The initial value is from 0x90000 to 0x9FFFF. User programmable codes are stored in the Flash ROM. A user needs to use CPU instructions to read and write the Flash ROM if wanting to edit the program codes of the Flash ROM. The user can store data at any positions between the blocks.

3.5. Bus interface unit

Regarding the structure of a bus, the reading and writing of the register are controlled by a 32-bit advanced peripheral bus (APB), which can write in 32-bit data during each clock cycle. In order to prevent from the existing data be covered when writing in new data, the user can use the MASK function to finish the operation.

As described in FIG. 3.3, the original data in BIT[7:0] of the register are 10101010b, and the written-in data are made valid via the MASK bits; when 0000111101010101b are written in BIT[15:0], the result will be 0000000010100101b, which means the MASK bit can only be set as 1b and the read value will be 0b; when 0101b are written in BIT[7:4], but the definition of BIT[15:12] is 0000b; therefore, it means the write values of BIT[7:4] are invalid; when 0101b are written in BIT[3:0] and the definition of MASK BIT[11:8] is 1111b; therefore, it means the write values of BIT[3:0] can be valid.

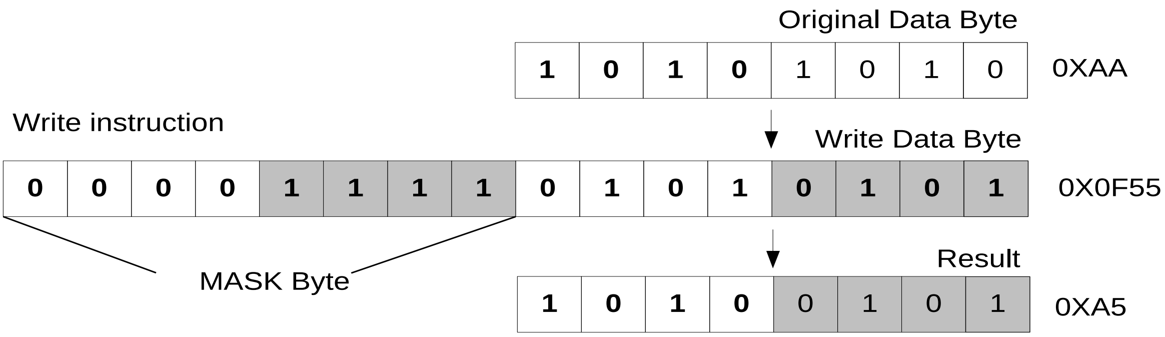


FIG. 3-3 Data flow structure

3.6. Boot ROM

8Kbyte Boot ROM is provided, and the initial value is from 0x80000 to 0x81FFF. The blocks are for boot codes, flash codes and security codes. When the chip is reset, the program timer will start from 0x80000. The software of the Boot ROM includes much information, such as system program protocol, security protocol and the like.

3.7. Embedded debug module (EDM)

The embedded debug module (EDM) is a debug interface which can be used by the chip in the development environment. When the chip has no security protection, the user can transmit instructions to the MCU via EDM interface to read the information of the debug mode. EDM is the bridge of the communication between the chip and the computer The PC USB and the chip EDM are connected via HY16F Mini Link by only using a two-wire protocol interface. EDM can access the control register, general GPR register, SRAM DLM and Flash ROM ILM of a chip.

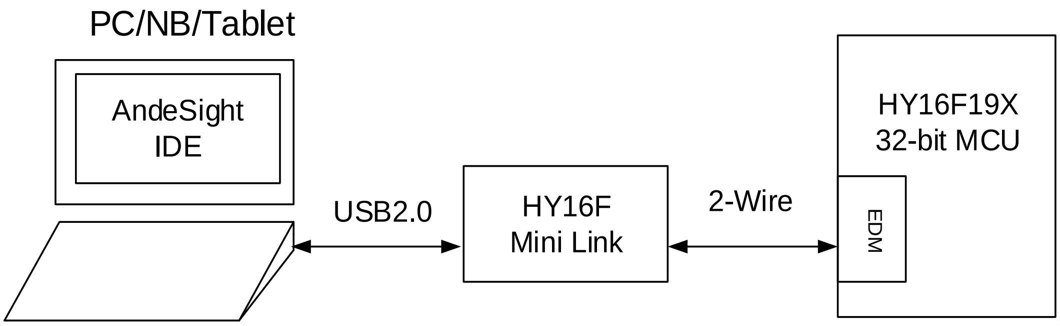


FIG. 3-4 Chip development connection diagram

4. SOC REGISTER

4.1. Overall description

Manage the operating mode of the system and the reset status of the chip, such as WDT, external reset, under voltage reset, etc.

4.2. Register address

SoC Status Register Address	31	24	23	16	15	8	7	0
SoC Status Base Address + 0x04(0x40104)	-		-		MASK0		REG0	

-Reserved

4.3. Register function

Operate the register SoC 0x40104 [4] can set the operating mode of the system as SLEEP mode/IDLE mode. The user can check the register SoC 0x40104[3] to understand what the current operating mode of the system is. The setting of the operating mode of the chip will be specified at the chapter 31.

4.3.1. SOC register

SoC Status Base Address + 0x04 (0x40104)									
Symbol	SoC Status Register								
Bit	[31:24]					[23:16]			
Name	ICE Configuration					SoC Configuration			
RW	R-0x0F					R-0x1C			
Bit	[15:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	FPRG	FCRst	IDLE	FSLP/IDLE	FWDog	FRST	FBOR
RW	R0W-0	RW-0	RW0-0		RW-0	RW0-0			RW0-1

Bit	Name	Description	
Bit[06]	FPRG	Power Good Flag	
		0	Normal
		1	Power Good has already been triggered before.
Bit[05]	FCRST	CPU Core Reset Flag	
		0	Normal
		1	ICP Core has already been triggered before.
Bit[04]	IDLE	IDLE Mode Control Bit	
		0	Sleep Mode
		1	IDLE Mode
Bit[03]	FSLP/IDLE	Sleep/Idle Flag (Low voltage reset or reset circuit reset can reset the bit.)	

		0	Normal
		1	Sleep Mode or Idle Mode
Bit[02]	FWDT	WDT Flag (Low voltage reset or external reset can clear the bit.)	
		0	Normal
		1	WDT is reset or interrupted.
Bit[01]	FRST	External Reset Flag (Low voltage reset (BOR) can clear the bit.)	
		0	Normal
		1	Reset PIN or ICP software reset has occurred.
Bit[00]	FBOR	Low Voltage Reset (BOR) Flag (The bit will be automatically cleared after the voltage of the chip is higher than 1.8V.)	
		0	Normal
		1	Low voltage reset has occurred.

5. POWER MANAGEMENT

5.1. Overall description

Power management module includes a charge pump regulator, a wide BandGap reference, a narrow BandGap reference, a VDDA LDO, a VDD18 LDO and a reference output buffer. Chip VDD3V can work by only one voltage source between 2.2V and 3.6V. The power system can be classified into three parts: I/O circuit, analog circuit, and digital circuit. The power supply of the I/O circuit is driven by VDD3V. The power supply of the analog circuit is driven by the internal VDDA LDO. Finally, the power supply of the digital circuit is driven by VDD18 LDO.

When the MCU is under IDLE mode, it will use lowest power consumption to perform the memory operation of the register and the SRAM. During the IDLE mode, the wide BandGap reference, BOR and VDD18 LDO are enabled. If the MCU is under the automatic wake-up mode, the low-speed oscillator should be enabled.

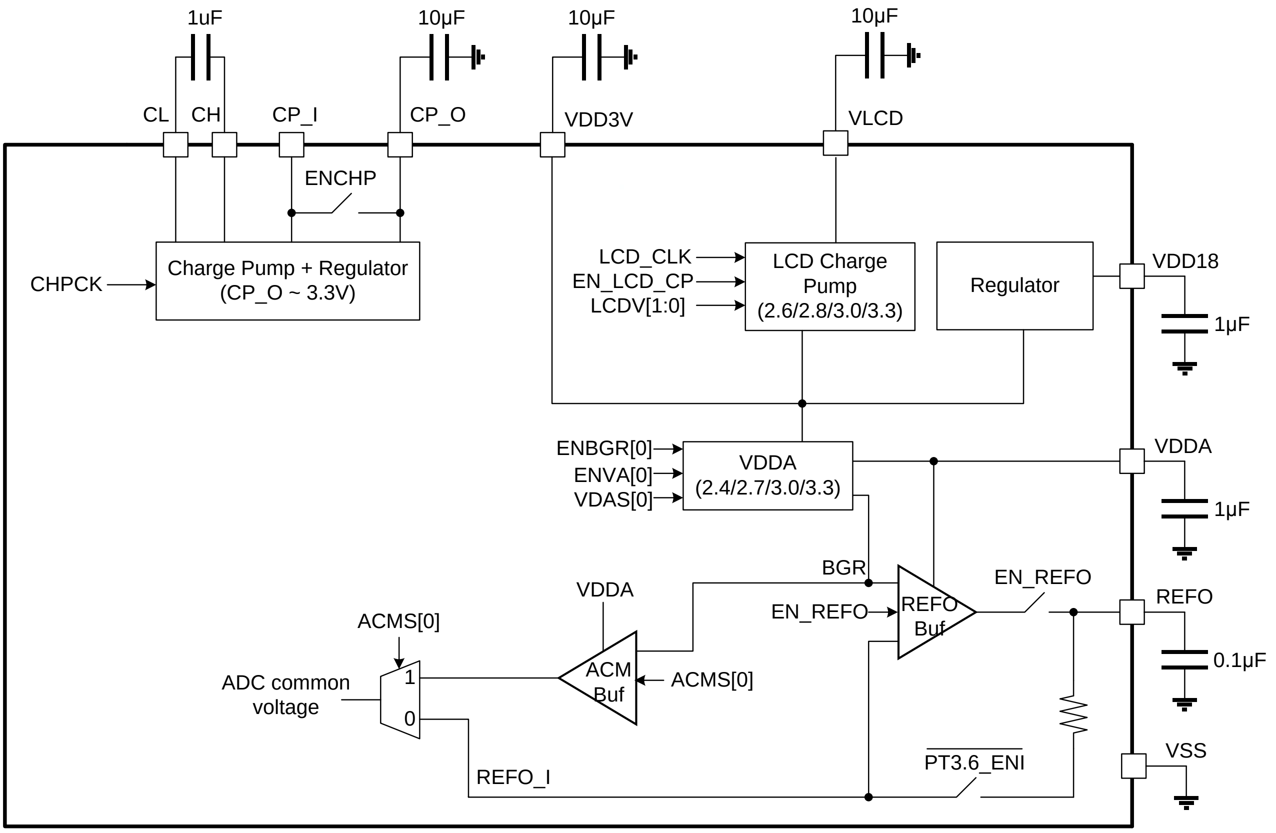


FIG. 5-1 Function block diagram

Chip operating voltages VDD3V and VDD18

The operating voltage of the chip is inputted via the pin VDD3V, and the voltage range is 2.2V~3.6V; besides, the pin should be connected to a 10uF ground capacitor, which can make VDD3V become more stable. If the operating voltage of the chip is used to drive a high current load, the operating voltage of the chip may be decreased and the chip may be reset; in this situation, it is necessary to enable the charge pump boost circuit to output a stable voltage to VDD3V via the CP_O pin so as to make sure the chip can work normally. The VDD18 LDO output a stable voltage 1.8V via the VDD18 pin and the pin should be connected to a 1uF bypass capacitor. The VDD18 LDO has a low-power voltage mode; the register PMU [0] (VDDLP) should be set as 1 to achieve lowest power consumption. Before the control bit enters the IDLE mode, the user can set it as 1; after the MCU is wakened, the bit will be cleared to be 0

VDDA voltage

The chip has a voltage regulator circuit LDO: VDDA and the VDDA voltage should be enabled when using ADC. It can have different operating modes and different output voltages. It has four different operating modes; the first mode is to be short-circuited to the VDD3V; and the VDDA is close to the VDD3V during the mode. The second mode is Weak pull down; during the mode, the VDDA is close to the VSS. The third mode is High Z; and it is possible to input the voltage into the VDDA from outside but the inputted voltage should not exceed VDD3V. The four modes is adjustable voltage regulating mode LDO; during the mode, the VDDA can output four different voltages: 2.4V, 2.7V, 3.0V and 3.3V. For better performance, the voltage difference between VDD3V and VDDA. Should be higher than 0.2V and can drive at most 10mA. Additionally, it also needs to be connected to a 1uF bypass capacitor. Note that when connecting a 1uF VDDA to-ground capacitance, stabilization time VDDA at least require more than 0.5ms, when VDDA access a 10uF capacitance to ground, settling time of at least VDDA greater than 1ms.

Low voltage detection circuit (BOR)

The BOR circuit is used to monitor the stability of the power system and the MCU. When the BOR detects the VDD3V and VDD18 are lower the detecting voltage of the BOR, the BOR will be triggered to reset the system and the chip; the chip will work normally until the BOR detects the operating voltage of the chip exceeds the voltage of the BOR.

Charge Pump

The charge pump regulator provides stable voltage for the chip, which can also be used to separate the power supply of the system from the power supply of the chip. Some applications will need to use high current external circuit, such as driving a DC motor; in this case, it is necessary to enable the charge pump to make sure the operating voltage of the chip is stable and the interference caused by the surge current from the inductors of the motor can be reduced.

So as to enable the charge pump, the register PMU0x40400 [2] should be set as <1> and the ADC clock source should be enabled; the external circuit needs connect an external capacitor (Ccp2) in series between the CH/CL pins, and respectively connect an external ground capacitor to the CP_I/CP_O pins; the voltage regulating output end CP_O of the charge pump should be short-circuited to the operating voltage pin VDD3V of the chip. When the charge pump is working, the power supply is inputted from the CP_I to generate and output stable voltage from CP_O to the input of the VDD3V. If the charge pump is not enabled, the voltage of the CP_O will be equal to the voltage of the CP_I; and the voltage will be also outputted from the CP_O to the input of the VDD3V. If the charge pump boot circuit will not be used, it is not necessary to install the external capacitors Ccp2, Ccp1 and Ccp3; the operating voltage of the chip will be directly provided by VDD3V.

When the charge pump is working, it is suggested that the capacitances of the CH/CL pins are 1uF and cannot be lower than 0.1uF. It is suggested that the capacitance of the CP_O is at least ten times the capacitance of the capacitors connected in series between the CH/CL pins. If the capacitance of the CH/CL pins is 1uF, it is suggested that the capacitance of the CP_O is higher than or equal to 10uF. If the capacitance of the CP_O is large, the system will be more stable. The connection of the output of the CP_O and the pins of VDD3V should be done by short circuit from the external PCB. Note that when a 10uF CP_O access to ground capacitance, CP_O stable for at least requires more than 10ms.

BandGap reference voltage and common mode voltage (REFO)

When the VDDA is higher than 2.4V, the analog circuit can work. However, the analog circuit needs the current offset and the reference voltage. Therefore, the BandGap reference voltage should be enabled before the analog circuit is enabled; the BandGap reference voltage can be enabled by setting the register PMU [4] (ENBGR) as 1. Only after the BandGap reference voltage is enabled, the common mode voltage (REFO) can effectively output 1.2V.

It is necessary to provide a common mode voltage (REFO) for the ADC to enable it. If the user wants to use the internal power supply, the ACMS should be set as 1; if the user wants to the external power supply, the ACMS should be set as 0 to output a common mode voltage (REFO). The user will need to use a reference voltage to drive the external circuit; therefore, the ENRFO should be set as 1 to output the common mode voltage to the pin; besides, the REFO is the BandGap reference voltage with buffer. The output voltage of the REFO pin is about 1.2V and has +/-1mA push-pull driving ability. It can drive a 22~1000nF big capacitor load. If the external REFO voltage output is used, the common mode voltage for the ADC can be provided by an external power supply; in this case, the ACMS can be set as 0 to save more power. Note that when a 0.1uF REFO access to ground capacitance, REFO stabilization time at least more than 0.1ms.

The following table shows the voltage sources for all modules.

Block name	Voltage source	Block name	Voltage Source
32-bit CPU Core N801	VDD18	Timer A/B/C PWM	VDD18
08KB SRAM	VDD18	GPIO Port	VDD3V
64KB Flash ROM	VDD3V/ VDD18	24-bit SD ADC	VDDA
Clock System	VDD18	08-bit Resistance ladder	VDDA
Watch Dog Timer	VDD18	Rail-to-Rail OPAMP	VDDA
Hardware RTC	VDD18	Analog Comparator	VDD3V
Charge Pump	VIN		
BOR	VDD3V/ VDD18		
Band Gap/Reference	VDDA		
Hardware EUART	VDD18/VDD3V		
32-bit Hardware SPI	VDD18/VDD3V		
Hardware I2C	VDD18/VDD3V		

Table 5-1 Chip Power supply distribution

5.2. Register address

Power Management Register Address	31	24	23	16	15	8	7	0
PMU Base Address + 0x00 (0x40400)	MASK1		REG1		MASK0		REG0	

5.3. Register function

5.3.1. Power management register (PMU)

Power Management Base Address + 0X00 (0X40400)							
Symbol	PMU (PMU Control Register)						
Bit	[31:24]	[23:20]	[19:18]		[17:16]		
Name	MASK	-	VDAS		ENVA		
RW	R0W-0	RW-0					
Bit	[15:08]	[7:5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	ENBGR	ACMS	ENCHP	ENRFO	VDDLP
RW	R0W-0	RW-0					

Bit	Name	Description	
Bit[19~18]	VDAS	VDDA output voltage selection	
		00	VDDA =2.4V
		01	VDDA =2.7V
		10	VDDA =3.0V
		11	VDDA =3.3V @ VDD >=3.5V
Bit[17~16]	ENVA	VDDA LDO voltage source configuration for controlling The output voltage range of the VDDA.	
		00	High impedance (High Z)
		01	Internally short-circuited to the VDD3V; the output of the VDDA is close to the VDD3V
		10	Weak pull down; the output of the VDDA is close to the VSS.
		11	VDDA LDO; the output of the VDDA can be adjusted , which is decided by the VDAS.
Bit[04]	ENBGR	Band Gap voltage enablement control	
		0	Disable
		1	Enable
Bit[03]	ACMS	ADC analog ground input source selection	
		0	External analog ground
		1	Internal analog ground (used with the ADC)
Bit[02]	ENCHP	Charge Pump enablement control	
		0	Disable
		1	Enable
Bit[01]	ENRFO	Common mode voltage (REFO) enablement control	

		0	Disable
		1	Enable
Bit[00]	VDDLP	VDD18 LDO low-power control	
		0	Normal (the bit should be set as 0 after the SLEEP mode)
		1	Low-power

6. CLOCK SYSTEM

6.1. Overall description

The clock control system provides the clocks for the whole chip, including the system clocks (CPU clock, APB clock) and all peripheral operating clocks (timer, communication interface, RTC, analog circuit, etc.) Each function module has a clock switch controller, clock source selection and frequency divider. Under the SLEEP mode, the controller always closes the external crystal oscillators, internal crystal oscillators and system clocks to minimize the system power consumption.

The operating clock sources include the external crystal oscillators, internal HAO and LPO oscillators; with the frequency divider, the frequency sources of the CPU and the peripheral devices can be flexibly allocated and managed to adjust the power consumption of the chip in order to save the energy.

6.1.1. External oscillators

There are two external oscillators, including the high-speed crystal oscillator (HSXT) and the low-speed crystal oscillator (LSXT). The chip has two independent input pins for the external high-speed crystal oscillator and low-speed crystal oscillator; thus, the user can connect the two external oscillators to the chip at the same time. The external oscillator should be connected to a resistor in parallel, or the crystal oscillator will not work even if it is soldered at the pin; besides, the crystal oscillator is connected to two 0~20pF ground capacitors and the capacitance of each capacitor is subject to the parasitic capacitor caused by the layout of the PCB.

The parallel resistor between the pins of the oscillator and the capacitor C2/C1 parameters of each pin of the oscillator will vary with the frequency, brand of the external crystal oscillator and the layout of the PCB. The following table lists suggested allocation of the R1/C1/C2 parameters and the frequency sources for your reference. In the absence of special circumstances, it can also be capacitive default.

Type	Symbol	External crystal oscillator parameters				Instruction execution status	
		Frequency	R1/Ω	C1	C2	Sleep mode	Idle mode
Low-speed oscillator	LSXT	32768Hz	10M	10pF	10pF	Stop	Available
High-speed oscillator	HSXT	2~16MHz	1M	10pF	10pF	Stop	Available

6-1 Suggest external crystal oscillator configuration

Note: The external oscillator pin capacitance (C1 / C2 parameters) can be in accordance with the actual PCB board layout with a different crystal as the case may choose to adjust the capacitance value of the size, it is recommended in the range of 0 ~ 20pF.

Using an external crystal oscillator parameter Note:

(01) when using 16MHz, chip operating voltage must be greater than 3.0V.

(02) The external crystal shock 4MHz / 8MHz stabilization time is about 30ms.

(03) External 32768Hz crystal shock stabilization time is about 1.3s.

(04) After Sleep instruction execution, external crystal earthquake shock all stops.

(05) When External crystal oscillator parameter, note that the pin input / output configuration, the use shall not be required to set the configuration pin internal pullup resistor, in order to avoid abnormal operation. And the external resistor R1 must not default.

(06) To use an external oscillator (HSXT), recommended to choose the MCU clock / 2, can reduce the oscillator frequency source interference, and strengthen anti-jamming capability.

6.1.2. Internal crystal oscillators HAO and LPO

The HAO is an external high-speed RC oscillator of the chip and its typical output frequency is 2MHz/4MHz/10MHz/16MHz; besides, it has several features, such as quick start, high anti-interference and low power consumption, etc. The output frequency of the HAO is adjustable; therefore, the user can adjust the output frequency of the HAO by software.

Matter needing attentions of using internal crystal oscillators:

- The operating voltage of the chip should be kept high when using the 16MHz HAO.
- The output frequency of the HAO can be adjusted by modifying the register HAOTR 0x40304[7:0] Example: When set HAO work at 2MHz, if the actual output is only 1.99MHz, it can be controlled by adjusting the position HAOTR [7: 0] to adjust the frequency of the output, HAOTR default is 0x80, adjustments can be increased up HAO actual operating frequency.
- The default oscillator of the chip is the internal 2MHz HAO; the user can modify the default settings register 0x40300[4:3] to change the output frequencies of other HAOs.
- The stabilization time of the 4MHz HAO is about 0.5ms;
- After the SLEEP instruction is executed, all HAO oscillators will stop and enter the SLEEP mode.
- From Sleep mode to wake up time of about 1024 * HAO + 2048 * LPO = <64ms. Note: If you meet the above description wake-up time, it should be before entering Sleep Mode, CPU frequency source is selected as HAO.
- After the IDLE instruction is executed, all HAO oscillators will not stop, but the CPU will enter the IDLE mode.
- From Idle Mode to wake-up time of about 500 cycles, when using a preset HAO = CPU Clock when 2MHz, wake-up time is about 250us.

The LPO is the internal low-speed RC oscillator of the chip; its output frequency is 35 kHz and has low power consumption; it will immediately start after the chip is power-on or wakened; besides, it cannot be enabled; in other words, the LPO will keep working during the whole operation process of the chip.

The stabilization time of the LPO is about 510us and it is the only operating clock source of the WDT.

- After the SLEEP instruction is executed, all LPO oscillators will stop.
- After the IDLE instruction is executed, all LPO oscillators will not stop, but the CPU will enter the IDLE mode.
- From Idle Mode to wake-up time of about 500 cycles, when the CPU Clock using a preset LPO = 35 kHz, wake-up time of about 14ms.

Typical output frequencies of the HAO and LPO are as shown in following Table 6-2.

Symbol	Frequency	Frequency source configuration			Instruction execution status	
		ENHAO[1]	HAO[1:0]	CKHS[1]	Sleep	Idle
HAO	2MHz	1	00B	0	Stop	Workable
	4MHz	1	01B	0	Stop	Workable
	10MHz	1	10B	0	Stop	Workable
	16MHz	1	11B	0	Stop	Workable
LPO	35KHz	Start after the chip is power-on		CKLS=0	Stop	Oscillate

Table 6-2 internal crystal oscillator configuration

HAO calibration method:

Chip HAO will have about +/- 10% error range, If the user wants a more accurate HAO operating frequency, the HYCON C library can be calibrated (DrvCLOCK_CalibrateHAO this function), This function can be set to control the HAO oscillation frequency error within the range of +/- 2%, Detailed HAO frequency specifications, refer to the document DS-HY16F198B_EN Note, the function can refer to the following or document APD-HY16IDE007_EN:

-function

void DrvCLOCK_CalibrateHAO (short int uMHz)

- Function

HAO calibration function; Note To correspond with the selected HAO frequency use; setting register 0x40304 [7: 0] value

-Input parameters

uMHz [in] pending correction of HAO frequency mode selection

0: Correction 2MHz; 1: Correction 4MHz; 2: Correction 10MHz; 3: Correction 16MHz;

6.1.3. CPU and external peripheral operating frequency sources configuration

Both of the external and internal crystal oscillators can provide the frequency sources for the CPU and the frequency sources will be provided for the CPU after passing the frequency dividers. The chip can determine the frequency source of the CPU is the HS_CK or LS_CK via the frequency selector MCUCKS [1] and perform the frequency division via the frequency divider ENMCD [1]. Thus, there are multiple operating frequency modes for the CPU to select from to determine the instruction cycle of the chip.

Similarly, the external peripheral operating frequency sources are also provided by the external, internal crystal oscillators and the HS_CK or LS_CK passing the frequency dividers; or the frequency sources can be directly provided by the crystal oscillators, such as the WDT. As the external peripheral operating frequency configuration may vary with the different operations, please refer to the following figure for more information.

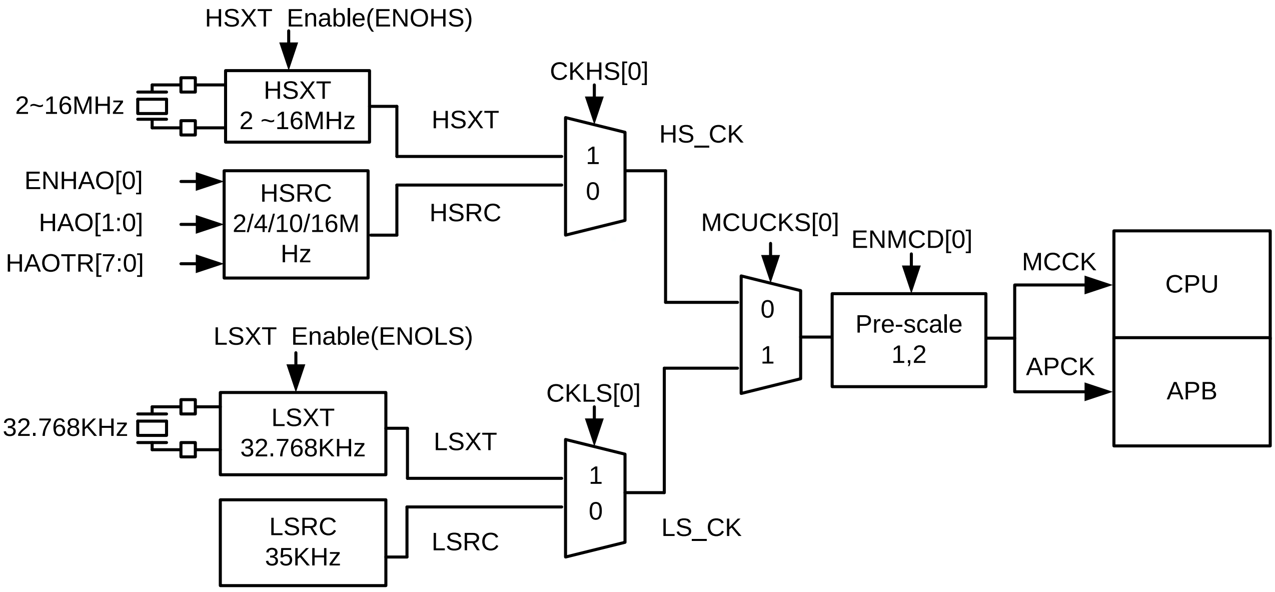


Table 6-1 CPU operating frequency source configuration diagram

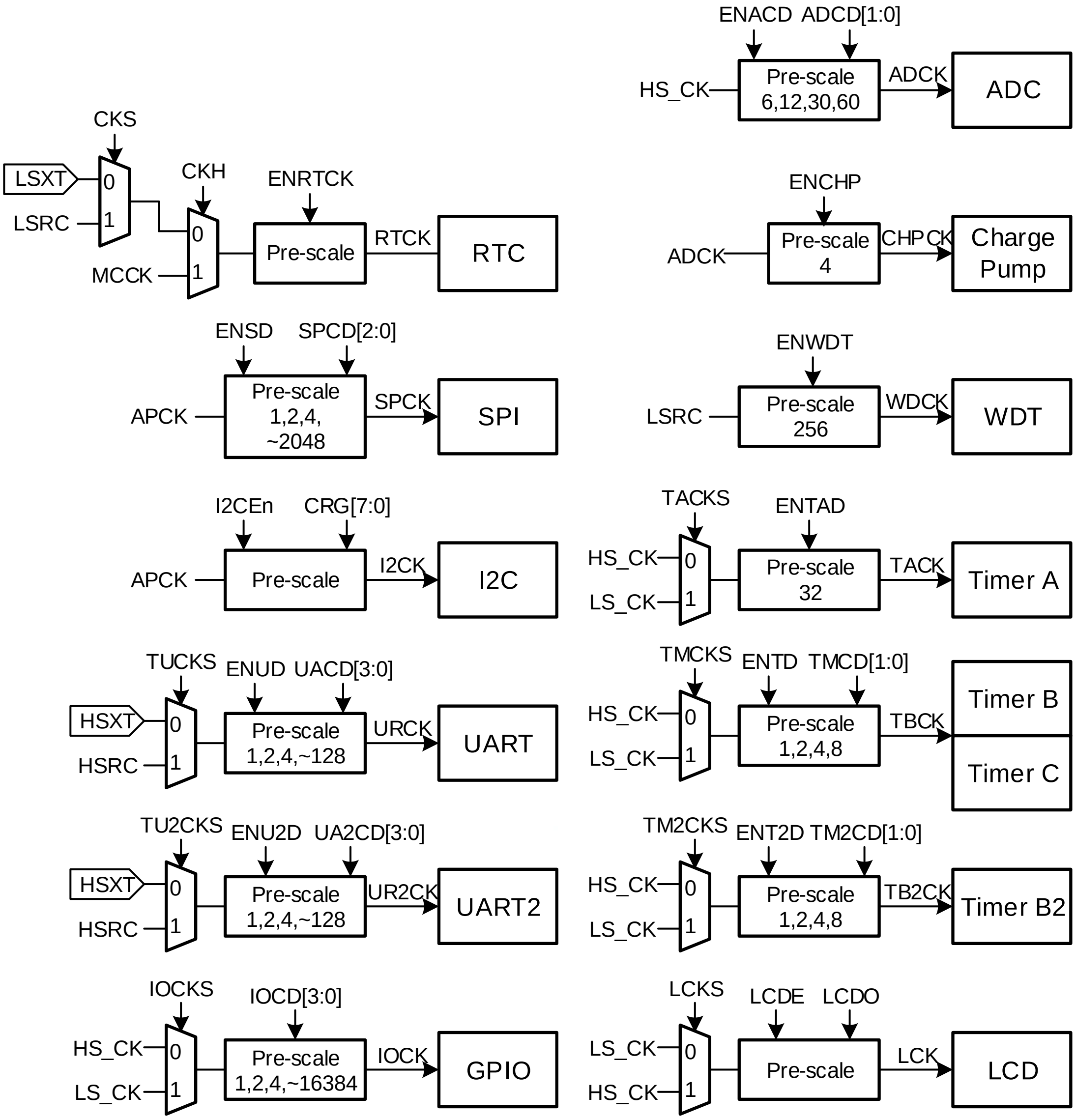
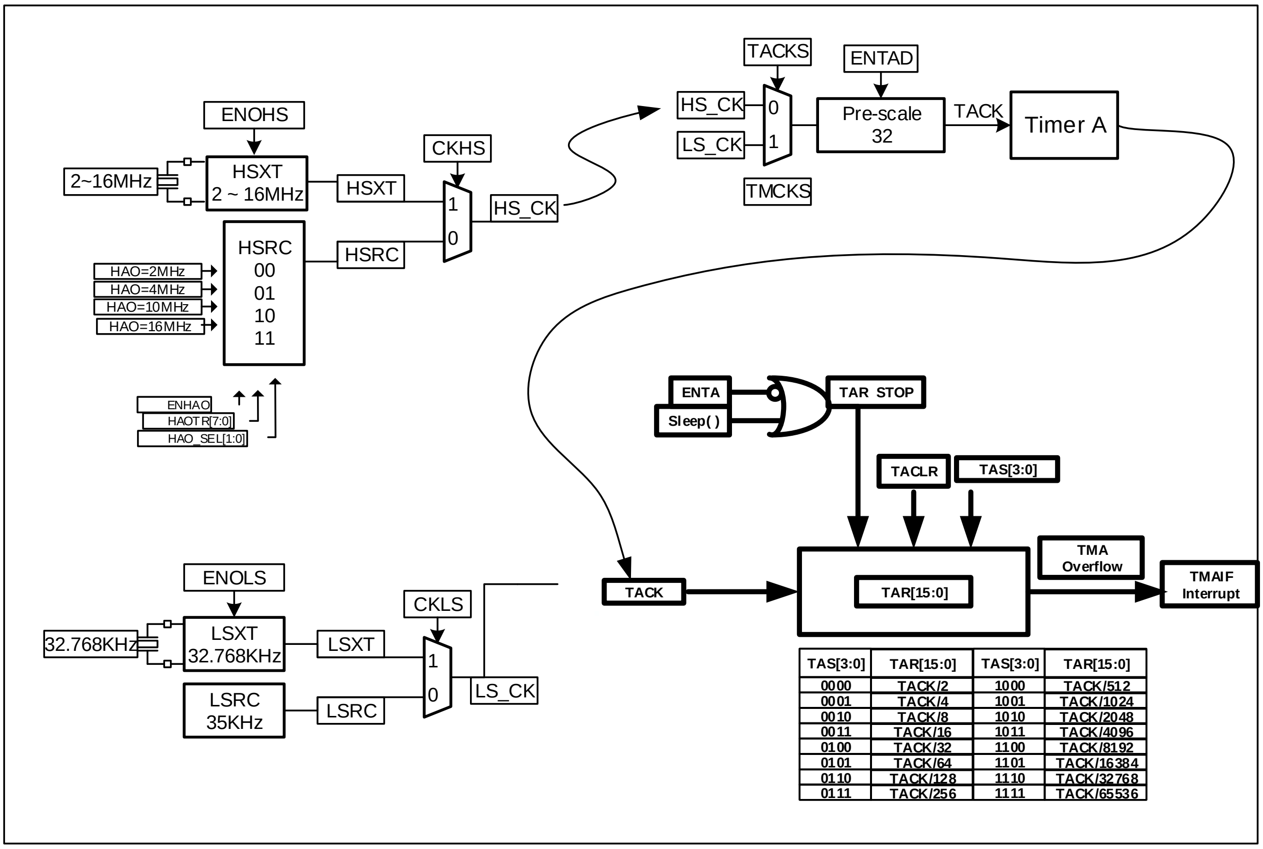


Table 6-2 External peripheral operating frequency configuration diagram

Frequency Configuration Instructions:

All HY16F19xB surrounding IP frequency allocation, you can refer to Figure 6-1 and Figure 6-2 to do the relevant register configurations.

Figure 6-3 below is a TimerA operating frequency arrangement view this Figure with text to explain how to configure TimerA do every second counts overflow settings. First select the CPU frequency source is internal high-speed 2MHz, namely register HAO 0x40300 [4: 3] control bit is set to <00>, register CKHS 0x40300 [5] is set to <0>, after the setting is completed, in this HS_CK which represents 2MHz. (Note: HY16F9xB preset CPU operating frequency and for this setting). Register TACKS 0x40308 [3] is set to <0>, representing the selected clock source HS_CK as TimerA (TACKS). Set register ENTAD 0x40308 [2] <1>, which do TimerA clock source divider action (Timer A Clock TACKS / 32), complete this setting; on behalf of TimerA input clock source (TACK) is 2MHz / 32. After doing TimerA count overflow is set, setting register TAS 0x40C00 [3: 0] control bits <1111>, that do TimerA input clock source except 65536 TACK action, complete this setting, TimerA count overflow is set to 2MHz / 32/65536 = 0.95, approximately once every second counts overflow.



TAS[3:0]	TAR[15:0]	TAS[3:0]	TAR[15:0]
0000	TACK/2	1000	TACK/512
0001	TACK/4	1001	TACK/1024
0010	TACK/8	1010	TACK/2048
0011	TACK/16	1011	TACK/4096
0100	TACK/32	1100	TACK/8192
0101	TACK/64	1101	TACK/16384
0110	TACK/128	1110	TACK/32768
0111	TACK/256	1111	TACK/65536

Figure 6-3 TimerA frequency setting instructions Figure

6.2. Register address

Clock Register Address	31	24	23	16	15	8	7	0
CLK Base Address + 0x00 (0x40300)	-		-		MASK0		REG0	
CLK Base Address + 0x04 (0x40304)	-		-		-		HAOTR	
CLK Base Address + 0x08 (0x40308)	MASK1		REG1		MASK2		REG2	
CLK Base Address + 0x0C (0x4030C)	MASK3		REG3		MASK4		REG4	
CLK Base Address + 0x10 (0x40310)	MASK5		REG5		MASK6		REG6	
CLK Base Address + 0x14 (0x40314)	-		-		MASK7		REG7	

-Reserved

6.3. Register function

6.3.1. Clock system register CLKCR0

Clock Base Address + 0x00 (0x40300)								
Symbol	CLK0 (Clock Control Register 0)							
Bit	[31:16]							
Name	RSV(Reserved)							
RW	R-0							
Bit	[15:8]	[7]	[6]	[5]	[4:3]	[2]	[1]	[0]
Name	MASK	OHS_HS	CKLS	CKHS	HAO	ENOLS	ENOHS	ENHAO
RW	R0W-0	RW-0						RW-1

Bit	Name	Description	
Bit[7]	OHS_HS	External oscillator mode selection	
		0	HSXT<4MHz
		1	HSXT>4MHz
Bit[6]	CKLS	Chip low-speed frequency source selection	
		0	Internal low-speed oscillator (OSC_LSRC)
		1	External low-speed oscillator (OSC_LSXT)
Bit[5]	CKHS	Chip high-speed frequency source selection	
		0	External low-speed oscillator (OSC_HSRC)
		1	External high-speed oscillator (OSC_HSXT)
Bit[4~3]	HAO	Internal high-speed oscillator mode configuration	
		[00]	2MHz
		[01]	4MHz
		[10]	10MHz
		[11]	16MHz
Bit[02]	ENOLS	External low-speed oscillator enablement control	
		0	Disable
		1	Enable
Bit[01]	ENOHS	External high-speed oscillator enablement control	
		0	Disable
		1	Enable
Bit[00]	ENHAO	Internal high-speed oscillator enablement control	
		0	Disable
		1	Enable

Precautions:

HS_CK, LS_CK clock source foolproof control:

When using CKHS (or CKLS) switch to the HS_CK (or LS_CK) clock source, it judges the corresponding oscillator Enable or Disable. If it is Disable, it will not perform switching clock source.

Precautions:

HS_CK, LS_CK clock source foolproof closed:

If user would like to disable one of the oscillator, have to switch HS_CK (or LS_CK) to another one enabling oscillator. In order to avoid the system no clock source that cause system crash.

Precautions:

Disable high-speed oscillator foolproof control:

Two sets of high-speed oscillators can be disabled at the same time, but user needs to switch the CPU Core to low-speed clock source. Otherwise, user cannot disable the two sets of high-speed oscillators at the same time.

6.3.2. Clock system register CLKCR1

Clock Base Address + 0x04 (0x40304)		
Symbol	CLK1 (Clock Control Register 1)	
Bit	[31:16]	
Name	Reserved	
RW	R-0	
Bit	[15:8]	[7:0]
Name	Reserved	HAOTR
RW	R-0	RW-80H

Bit	Name	Description	
Bit[7:0]	HAOTR	Internal high-speed oscillator calibration control register	
		0	Set 0
		1	Set 1

1*LSB.Step = 0.125%

0000_0000 is the lowest speed.

1000_0000 is the default speed.

1111_1111 is the higher speed.

6.3.3. Clock system register CLKCR2

Clock Base Address + 0x08 (0x40308)								
Symbol	CLK2 (Clock Control Register 2)							
Bit	[31:24]	[23]	[22]	[21]	[20]		[19:16]	
Name	MASK	ENRTCK	-	TUCKS	ENUD		UACD	
RW	R0W-0	RW-0	-	RW-0				
Bit	[15:08]	[7]	[6]	[5:4]	[3]	[2]	[1]	[0]
Name	MASK	TMCKS	ENTD	TMCD	TACKS	ENTAD	ENMCD	MCUCKS
RW	R0W-0	RW-0						

Bit	Name	Description	
Bit[23]	ENRTCK	RTC clock source control	
		0	Disable (The RTC register cannot be written in and unlocked.)
		1	Enable (The RTC register can be unlocked.)
Bit[21]	TUCKS	EUART clock source selection	
		0	HSXT: External high-speed oscillator
		1	HSRC: Internal high-speed oscillator
Bit[20]	ENUD	EUART clock source enablement control	
		0	Disable
		1	Enable

Bit[19~16]	UACD	EUART clock source frequency divider configuration	
		0000	EUART clock source/ 1
		0001	EUART clock source/ 2
		0010	EUART clock source/ 4
		0011	EUART clock source/ 8
		0100	EUART clock source/ 16
		0101	EUART clock source/ 32
		0110	EUART clock source/ 64
		0111	EUART clock source/ 128
		1000 ~1111	Reserved
Bit[07]	TMCKS	Timer B,C clock source selection	
		0	HS_CK
		1	LS_CK
Bit[06]	ENTD	Timer B,C clock source enablement control	
		0	Disable
		1	Enable
Bit[5~4]	TMCD	Timer B,C clock source frequency divider configuration	
		00	clock/1
		01	clock/2
		10	clock/4
		11	clock/8
Bit[03]	TACKS	Timer A clock source selection	
		0	HS_CK
		1	LS_CK
Bit[02]	ENTAD	Timer A clock source frequency divider configuration	
		0	Disable the frequency divider
		1	Timer A clock/32
Bit[01]	ENMCD	MCU input clock source frequency divider configuration	
		0	MCU clock/1
		1	MCU clock/2
Bit[00]	MCUCKS	MCU input clock source selection	
		0	HS_CK
		1	LS_CK

6.3.4. Clock system register CLKCR3

Clock Base Address + 0x0C (0x4030C)						
Symbol	CLK3 (Clock Control Register 3)					
Bit	[31:24]	[23:21]		[20]		[19:16]
Name	MASK	-		IOCKS		IOCD
RW	R0W-0	-		RW-0		
Bit	[15:08]	[7]	[6]	[5:4]	[3]	[2:0]
Name	MASK	ADCKP	ENACD	ADCD	ENSD	SPCD
RW	R0W-0	RW-0				

Bit	Name	Description	
Bit[20]	IOCKS	GPIO input clock source selection	
		0	HS_CK
		1	LS_CK
Bit[19~16]	IOCD	GPIO clock frequency divider configuration	
		0000	Disable
		0001	GPIO clock source/ 1
		0010	GPIO clock source/ 2
		0011	GPIO clock source/ 4
		0100	GPIO clock source/ 8
		0101	GPIO clock source/ 16
		0110	GPIO clock source/ 32
		0111	GPIO clock source/ 64
		1000	GPIO clock source/ 128
		1001	GPIO clock source/ 256
		1010	GPIO clock source/ 512
		1011	GPIO clock source/ 1024
		1100	GPIO clock source/ 2048
		1101	GPIO clock source/ 4096
		1110	GPIO clock source/ 8192
		1111	GPIO clock source/ 16384
Bit[7]	ADCKP	ADC clock phase shift(effective only at Core Clock/2 and Core Clock is HS_CK)	
		0	ADC Clock Rising Edge generates from Core Clock Low
		1	ADC Clock Rising Edge generates from Core Clock High
Bit[6]	ENACD	ADC clock switch	
		0	Disable
		1	Enable
Bit[5~4]	ADCD	ADC clock frequency divider configuration	

		00	ADC clock source/ 6
		01	ADC clock source/ 12
		10	ADC clock source/ 30
		11	ADC clock source/ 60
Bit[3]	ENSD	SPI clock switch	
		0	Disable
		1	Enable
Bit[2~0]	SPCD	SPI clock frequency divider configuration	
		000	Reserved
		001	SPI clock source/ 2
		010	SPI clock source/ 4
		011	SPI clock source/ 8
		100	SPI clock source/ 32
		101	SPI clock source/ 128
		110	SPI clock source/ 512
		111	SPI clock source/ 2048

6.3.5. Clock system register CLKCR4

Clock Base Address + 0x10 (0x40310)							
Symbol	CLK4 (Clock Control Register 4)						
Bit	[31:24]	[23:22]	[21]	[20]	[19]	[18:16]	
Name	MASK	LCDCPD	UT2CKS	ENU2D	-	UA2CD	
RW	R0W-0	RW-0			-	RW-0	
Bit	[15:08]	[7]	[6:4]		[3:1]		[0]
Name	MASK	-	LCDO		LCDE		LCKS
RW	R0W-0	-	RW-0				

Bit	Name	Description	
Bit[23~22]	LCDCPD	LCD charge pump regulator clock source selection	
		0	LS_CK / 1 or HS_CK/8 (LCKS determines LS_CK or HS_CK)
		1	LS_CK / 2 or HS_CK/16 (LCKS determines LS_CK or HS_CK)
		2	LS_CK / 4 or HS_CK/32 (LCKS determines LS_CK or HS_CK)
		3	LS_CK / 4 or HS_CK/32 (LCKS determines LS_CK or HS_CK)
Bit[21]	UT2CKS	UART2 clock source selection	
		0	HSXT: External high-speed oscillator
		1	HSRC: Internal high-speed oscillator
Bit[20]	ENUD2D	UART2 clock source enablement control	
		0	Disable

		1	Enable
Bit[18~16]	UA2CD	UART2 clock source frequency divider configuration	
		0	UART2 clock source/ 1
		1	UART2 clock source/ 2
		2	UART2 clock source/ 4
		3	UART2 clock source/ 8
		4	UART2 clock source/ 16
		5	UART2 clock source/ 32
		6	UART2 clock source/ 64
		7	UART2 clock source/ 128
Bit[6~4]	LCDO	LCD clock source 2-stage frequency divider configuration	
		0	LCD clock source/ 1
		1	LCD clock source/ 3
		2	LCD clock source/ 5
		3	LCD clock source/ 7
		4	LCD clock source/ 9
		5	LCD clock source/ 11
		6	LCD clock source/ 13
		7	LCD clock source/ 15
Bit[3~1]	LCDE	LCD clock source 1-stage frequency divider configuration	
		0	Disable
		1	LCD clock source/ 1
		2	LCD clock source/ 2
		3	LCD clock source/ 4
		4	LCD clock source/ 8
		5	LCD clock source/ 16
		6	LCD clock source/ 32
		7	Disable
Bit[0]	LCKS	LCD clock source selection	
		0	LS_CK(always÷8)
		1	HS_CK(always÷64)

6.3.6. Clock system register CLKCR5

Clock Base Address + 0x014 (0x40314)					
Symbol	CLK5 (Clock Control Register 4)				
Bit	[31:16]				
Name	Reserved				
RW	R-0				
Bit	[15:8]	[7]	[6]	[5:4]	[3:0]
Name	Mark	TM2CKS	ENT2D	TM2CD	-
RW	R0W-0	RW-0			-

Bit	Name	Description	
Bit[7]	TM2CKS	Timer B2 clock source selection	
		0	HS_CK
		1	LS_CK
Bit[6]	ENT2D	Timer B2 clock source enablement control	
		0	Disable
		1	Enable
Bit[5~4]	TM2CD	Timer B2 clock source frequency divider configuration	
		0	Timer B2 clock source/ 1
		1	Timer B2 clock source/ 2
		2	Timer B2 clock source/ 4
		3	Timer B2 clock source/ 8

7. INTERRUPT CONTROL SYSTEM

7.1. Overall description

Interrupt vectors and interrupt priority Description:

The interrupt module includes the interrupt startup controller, interrupt enable controller and interrupt event flag register to manage the overall interrupt service, such as communication interrupt, timer interrupt, ADC interrupt, comparator interrupt and IO external interrupt. The chip provides 9-stage interrupt source and also provides 4-stage interrupt vector priorities, including HW0, HW1…HW8 (from high priority to low priority). The interrupt service is composed of the interrupt event flag (INTF), interrupt event service intelligent startup (INTE), interrupt general control GIE and vector addresses HW0~HW8. When the interrupt event occurs and the interrupt event service is enabled, the program counter PC will turn to the interrupt service vector addresses HW0~HW8 of the program memory at the next instruction cycle to execute the interrupt service program.

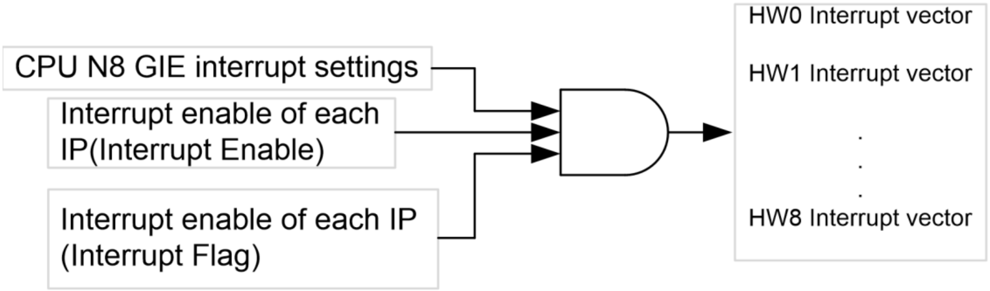

CPU N8 GIE interrupt settings
Interrupt enable of each IP(Interrupt Enable)
Interrupt enable of each IP (Interrupt Flag)
HW0 Interrupt vector
HW1 Interrupt vector
.
.
.
HW8 Interrupt vector

FIG. 7-1 Interrupt service structure diagram

Interrupt Vector Address	Vector	Interrupt Function
INT Base Address + 0X00 (I2C/UART/SP interface)	HW0	void HW0_ISR(void)
INT Base Address + 0X04 (Timer ABC /WDT/ HW RTC)	HW1	void HW1_ISR(void)
INT Base Address + 0X08 (ADC)	HW2	void HW2_ISR(void)
INT Base Address + 0X0C (CMP/OPA)	HW3	void HW3_ISR(void)
INT Base Address + 0X10 (PT1)	HW4	void HW4_ISR(void)
INT Base Address + 0X14 (PT2)	HW5	void HW5_ISR(void)
INT Base Address + 0X18 (UART2)	HW7	void HW7_ISR(void)
INT Base Address + 0X1C (TMB2)	HW8	void HW8_ISR(void)

NOTE: INT HW6 belong SW INT

Interrupt group HW0 ~ HW8 has priority can be set to provide four kinds of priority level (0-3).

0: Priority level to the highest level

1: priority level to the second highest level

2: a lower level of priority level

3: The priority level is the lowest level

Preset HW0 ~ HW8 are set to level 0 (priority level is the highest level).

When the priority levels are set at the same time, the priority for the HW0> HW1> HW2 ...> HW8.

for example :

HW0 priority setting to level 1, HW1 priority level 0, when the two interrupts occur simultaneously, then set the relationship because of the priority level, first enter HW1 interrupted. If you set HW0 priority to level 0, HW1 priority for level 0, when the two interrupts occur simultaneously, then the two set the same interrupt level though, but will give priority to enter HW0 interrupt.

Detail operation description:

The user can set the corresponding interrupt enable bit to be 1 or clear the bit 0 to enable or disable the corresponding interrupt function. The interrupt function can be enabled by setting the corresponding interrupt enable bit to be 1.

After the interrupt event takes place, the interrupt flag will be generated; the user can clear the flag to cancel the interrupt request.

It is necessary to set the global interrupt enable bit GIE=1, or any interrupt cannot be enabled.

The interrupt vector priority will be determined when multiple interrupt requests take place at the same time; the interrupt vector with high priority should be replied first.

During the execution of the interrupt vector service program, the high-level interrupt vector can terminate the current interrupt service to execute the high-level interrupt service; the high-level interrupt request can be executed only after the current interrupt vector service is finished.

After the interrupt service program is finished, it will automatically return to the program address where the interrupt occurred and continuously execute the program. Note that when the program enters the interrupt vector services, GIE will automatically be set to 0, so it is necessary to first GIE is set to 1, to meet the conditions of service set up high-level interrupt to enter the high-level interrupt service routine, when the advanced interrupt service executed, the program will Back to the original interrupt service program, continue down the implementation program

7.2. Register address

Interrupt Register Address	31	24	23	16	15	8	7	0
INT Base Address + 0x00 (COM) (0x40000)	MASK0		REG0		MASK1		REG1	
INT Base Address + 0x04 (TMR) (0x40004)	MASK2		REG2		MASK3		REG3	
INT Base Address + 0x08 (ADC) (0x40008)	MASK4		REG4		MASK5		REG5	
INT Base Address + 0x0C (CMP) (0x4000C)	MASK6		REG6		MASK7		REG7	
INT Base Address + 0x10 (PT1) (0x40010)	MASK8		REG8		MASK9		REG9	
INT Base Address + 0x14 (PT2) (0x40014)	MASK10		REG10		MASK11		REG11	
INT Base Address + 0x18 (UART2) (0x40018)	MASK12		REG12		MASK13		REG13	
INT Base Address + 0x1C (TMB2) (0x4001C)	MASK14		REG14		MASK15		REG15	

7.3. Register function

7.3.1. Interrupt control register INTCOM

INT Base Address + 0X00 (0X40000)														
Symbol	INTCOM (Interrupt Control Register 0)													
Bit	[31:24]							[23:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK							-	I2CEIE	I2CIE	UTxIE	URxIE	STxIE	SRxIE
RW	R0W-0							-	RW-0					
Bit	[15:14]	[13]	[12]	[11]	[10]	[09]	[08]	[07:06]	[05]	[04]	[03]	[02]	[01]	[00]
Name	MASK							-	I2CEIF	I2CIF	UTxIF	URxIF	STxIF	SRxIF
	-	I2CEIR	I2CIR	UTxIR	URxIR	STxIR	SRxIR							
RW	R-0							-	RW0-0					

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit15~8 are general registers.)

Bit	Name	Description	
Bit[21]	I2CEIE	I2C error interrupt enable control	
		0	Disable
		1	Enable
Bit[20]	I2CIE	I2C interrupt enable control	
		0	Disable
		1	Enable
Bit[19]	UTxIE	UART transmits (TX) interrupt enable control	
		0	Disable
		1	Enable
Bit[18]	URxIE	UART receives (RX) interrupt enable control	
		0	Disable
		1	Enable
Bit[17]	STxIE	SPI transmits (TX) interrupt enable control	
		0	Disable
		1	Enable
Bit[16]	SRxIE	SPI receives (RX) interrupt enable control	
		0	Disable
		1	Enable
Bit[13]	I2CEIR	I2C interrupt error request	
		0	Normal
		1	Interrupt
Bit[12]	I2CIR	I2C interrupt request	
		0	Normal
		1	Interrupt

Bit[11]	UTxIR	UART Tx interrupt request	
		0	Normal
		1	Interrupt
Bit[10]	URxIR	UART Rx interrupt request	
		0	Normal
		1	Interrupt
Bit[9]	STxIR	SPI Tx interrupt request	
		0	Normal
		1	Interrupt
Bit[8]	SRxIR	SPI Rx interrupt request	
		0	Normal
		1	Interrupt
Bit[05]	I2CEIF	I2C error interrupt flag (level-trigger)	
		0	Normal
		1	I2C error takes place and interrupt occurs
Bit[04]	I2CIF	I2C interrupt flag (level-trigger)	
		0	Normal
		1	I2C interrupt occurs
Bit[03]	UTxIF	UART transmits (TX) interrupt flag (level-trigger)	
		0	Normal
		1	UART transmission (TX) interrupt occurs.
Bit[02]	URxIF	UART receives (RX) interrupt flag (level-trigger)	
		0	Normal
		1	UART receives (RX) interrupt occurs.
Bit[01]	STxIF	SPI transmission (TX) interrupt flag (level-trigger)	
		0	Normal
		1	SPI transmission (TX) interrupt occurs.
Bit[00]	SRxIF	SPI reception (RX) interrupt flag (level-trigger)	
		0	Normal
		1	SPI reception (RX) interrupt occurs.

7.3.2. Interrupt control register INTTMR

INT Base Address + 0x04 (0x40004)														
Symbol	INTTMR (Interrupt Control Register 1)													
Bit	[31:24]							[23:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK							-	RTCIE	WDTIE	TMC1IE	TMC0IE	TMBIE	TMAIE
RW	R0W-0							-	RW-0					
Bit	[15:14]	[13]	[12]	[11]	[10]	[9]	[8]	[7:6]	[5]	[4]	[3]	[2]	[1]	[0]

Name	MASK							-	RTCIF	WDTIF	TMC1IF	TMC0IF	TMBIF	TMAIF
	-	RTCIR	WDTIR	TMC1IR	TMC0IR	TMBIR	TMAIR							
RW	R0W-0							-	RW0-0					

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit15~8 are general registers)

Bit	Name	Description	
Bit[21]	RTCIE	Real-time clock RTC interrupt enable control	
		0	Disable
		1	Enable
Bit[20]	WDTIE	WDT interrupt enable control	
		0	Disable
		1	Enable
Bit[19]	TMC1IE	TMC1 interrupt enable control	
		0	Disable
		1	Enable
Bit[18]	TMC0IE	TMC0 interrupt enable control	
		0	Disable
		1	Enable
Bit[17]	TMBIE	Timer TMB interrupt enable control	
		0	Disable
		1	Enable
Bit[16]	TMAIE	Timer TMA interrupt enable control	
		0	Disable
		1	Enable
Bit[13]	RTCIR	RTC interrupt request	
		0	Disable
		1	Enable
Bit[12]	WDTIR	WDT interrupt request	
		0	Normal
		1	Enable
Bit[11]	TMC1IR	Timer C channel 1 interrupt request	
		0	Normal
		1	Enable
Bit[10]	TMC0IR	Timer C channel 0 interrupt request	
		0	Normal
		1	Enable
Bit[9]	TMBIR	TMB interrupt request	

		0	Normal
		1	Enable
Bit[8]	TMAIR	TMA interrupt request	
		0	Normal
		1	Enable
Bit[05]	RTCIF	Real-time clock RTC interrupt flag	
		0	Normal
		1	Real-time clock RTC interrupt occurs.
Bit[04]	WDTIF	WDT interrupt flag	
		0	Normal
		1	WDT interrupt occurs.
Bit[03]	TMC1IF	TMC1 interrupt flag	
		0	Normal
		1	TMC1 interrupt occurs.
Bit[02]	TMC0IF	TMC0 interrupt flag	
		0	Normal
		1	TMC0 interrupt occurs
Bit[01]	TMBIF	Timer TMB interrupt flag	
		0	Normal
		1	Timer TMB interrupt occurs
Bit[00]	TMAIF	Timer TMA interrupt flag	
		0	Normal
		1	Timer TMA interrupt occurs.

7.3.3. Interrupt control register INTADC

INT Base Address + 0x08 (0x40008)				
Symbol	INTADC (Interrupt Control Register 2)			
Bit	[31:24]		[23:17]	[16]
Name	MASK		-	ADCIE
RW	R0W-0		-	RW-0
Bit	[15:9]	[8]	[07:01]	[00]
Name	MASK		-	ADCIF
	-	ADCIR		
RW	R-0		-	RW0-0

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit 8 is general register.)

Bit	Name	Description	
Bit[16]	ADCIE	ADC converter interrupt enable control	
		0	Disable
		1	Enable
Bit[08]	ADCIR	ADC interrupt request	
		0	Normal
		1	Interrupt
Bit[00]	ADCIF	ADC converter interrupt flag	
		0	Normal
		1	ADC converter interrupt occurs.

7.3.4. Interrupt control register INTCMP

INT Base Address + 0x0C (0x4000C)						
Symbol	INTCMP (Interrupt Control Register 3)					
Bit	[31:24]			[23:18]	[17]	[16]
Name	MASK			-	CPOIE	OPOIE
RW	R0W-0			-	RW-0	
Bit	[15:10]	[9]	[8]	[07:02]	[01]	[00]
Name	MASK			-	CPOIF	OPOIF
	-	CPOIR	OPOIR			
RW	R-0			-	RW0-0	

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit9~8 are general registers.)

Bit	Name	Description	
Bit[17]	CPOIE	Multi-function comparator output (CPO) interrupt enable control	
		0	Disable
		1	Enable

Bit[16]	OPOIE	Low-noise OP amplifier output (OPO) interrupt enable control	
		0	Disable
		1	Enable
Bit[09]	CPOIR	Multi-function comparator output (CPO) interrupt request	
		0	Normal
		1	Interrupt
Bit[08]	OPOIR	Low-noise OP amplifier output (OPO) interrupt request	
		0	Normal
		1	Interrupt
Bit[01]	CPOIF	Multi-function comparator output (CPO) interrupt flag	
		0	Normal
		1	Multi-function comparator output (CPO) interrupt occurs.
Bit[00]	OPOIF	Low-noise OP amplifier output (OPO) interrupt flag	
		0	Normal
		1	Low-noise OP amplifier output (OPO) interrupt occurs

7.3.5. Interrupt control register INTPT1

INT Base Address + 0x10 (0x40010)									
Symbol	INTPT1 (Interrupt Control Register 4)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT17IE	PT16IE	PT15IE	PT14IE	PT13IE	PT12IE	PT11IE	PT10IE
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK PT1 IR	PT17IF	PT16IF	PT15IF	PT14IF	PT13IF	PT12IF	PT11IF	PT10IF
RW	R-0	RW0-0							

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit15~8 are general registers.)

Bit	Name	Description	
Bit[23]	PT17IE	PT17 external interrupt enable control	
		0	Disable
		1	Enable
Bit[22]	PT16IE	PT16 external interrupt enable control	
		0	Disable
		1	Enable
Bit[21]	PT15IE	PT15 external interrupt enable control	
		0	Disable

		1	Enable
Bit[20]	PT14IE	PT14 external interrupt enable control	
		0	Disable
		1	Enable
Bit[19]	PT13IE	PT13 external interrupt enable control	
		0	Disable
		1	Enable
Bit[18]	PT12IE	PT12 external interrupt enable control	
		0	Disable
		1	Enable
Bit[17]	PT11IE	PT11 external interrupt enable control	
		0	Disable
		1	Enable
Bit[16]	PT10IE	PT10 external interrupt enable control	
		0	Disable
		1	Enable
Bit[15:8]	PT1 IR	PT1 Bit7-0 interrupt request	
		0	Disable
		1	Enable
Bit[07]	PT17IF	PT17 external interrupt flag	
		0	Normal
		1	PT17 external interrupt occurs
Bit[06]	PT16IF	PT16 external interrupt flag	
		0	Normal
		1	PT16 external interrupt occurs
Bit[05]	PT15IF	PT15 external interrupt flag	
		0	Normal
		1	PT15 external interrupt occurs
Bit[04]	PT14IF	PT14 external interrupt flag	
		0	Normal
		1	PT14 external interrupt occurs
Bit[03]	PT13IF	PT13 external interrupt flag	
		0	Normal
		1	PT13 external interrupt occurs
Bit[02]	PT12IF	PT12 external interrupt flag	
		0	Normal

		1	PT12 external interrupt occurs
Bit[01]	PT11IF	PT11 external interrupt flag	
		0	Normal
		1	PT11 external interrupt occurs
Bit[00]	PT10IF	PT10 external interrupt flag	
		0	Normal
		1	PT10 external interrupt occurs

7.3.6. Interrupt control register INTPT2

INT Base Address + 0x14 (0x40014)									
Symbol	INTPT2 (Interrupt Control Register 5)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT27IE	PT26IE	PT25IE	PT24IE	PT23IE	PT22IE	PT21IE	PT20IE
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK PT2 IR	PT27IF	PT26IF	PT25IF	PT24IF	PT23IF	PT22IF	PT21IF	PT20IF
RW	R0W-0	RW0-0							

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit15~8 are general registers.)

Bit	Name	Description	
Bit[23]	PT27IE	PT27 external interrupt enable control	
		0	Disable
		1	Enable
Bit[22]	PT26IE	PT26 external interrupt enable control	
		0	Disable
		1	Enable
Bit[21]	PT25IE	PT25 external interrupt enable control	
		0	Disable
		1	Enable
Bit[20]	PT24IE	PT24 external interrupt enable control	
		0	Disable
		1	Enable
Bit[19]	PT23IE	PT23 external interrupt enable control	
		0	Disable
		1	Enable
Bit[18]	PT22IE	PT22 external interrupt enable control	
		0	Disable

		1	Enable
Bit[17]	PT21IE	PT21 external interrupt enable control	
		0	Disable
		1	Enable
Bit[16]	PT20IE	PT20 external interrupt enable control	
		0	Disable
		1	Enable
Bit[15:8]	PT2 IR	PT2 Bit 7-0 interrupt request	
		0	Disable
		1	Enable
Bit[07]	PT27IF	PT27 external interrupt flag	
		0	Normal
		1	PT27 external interrupt occurs
Bit[06]	PT26IF	PT26 external interrupt flag	
		0	Normal
		1	PT26 external interrupt occurs
Bit[05]	PT25IF	PT25 external interrupt flag	
		0	Normal
		1	PT25 external interrupt occurs
Bit[04]	PT24IF	PT24 external interrupt flag	
		0	Normal
		1	PT24 external interrupt occurs
Bit[03]	PT23IF	PT23 external interrupt flag	
		0	Normal
		1	PT23 external interrupt occurs
Bit[02]	PT22IF	PT22 external interrupt flag	
		0	Normal
		1	PT22 external interrupt occurs
Bit[01]	PT21IF	PT21 external interrupt flag	
		0	Normal
		1	PT21 external interrupt occurs
Bit[00]	PT20IF	PT20 external interrupt flag	
		0	Normal
		1	PT20 external interrupt occurs

7.3.7. Interrupt control register INTUART2

<table>
<tr><td colspan="9">INT Base Address + 0x18 (0x40018)</td></tr>
<tr><td>Symbol</td><td colspan="8">INTUART2 (Interrupt Control Register 6)</td></tr>
<tr><td>Bit</td><td colspan="4">[31:24]</td><td>[23:20]</td><td>[19]</td><td>[18]</td><td>[17:16]</td></tr>
<tr><td>Name</td><td colspan="4">MASK</td><td>-</td><td>U2TxIE</td><td>U2RxIE</td><td>-</td></tr>
<tr><td>RW</td><td colspan="4">R0W-0</td><td>-</td><td colspan="2">RW-0</td><td>-</td></tr>
<tr><td>Bit</td><td>[15:12]</td><td>[11]</td><td>[10]</td><td>[9:8]</td><td>[7:4]</td><td>[3]</td><td>[2]</td><td>[1:0]</td></tr>
<tr><td rowspan="2">Name</td><td colspan="4">MASK</td><td rowspan="2">-</td><td rowspan="2">U2TxIF</td><td rowspan="2">U2RxIF</td><td rowspan="2">-</td></tr>
<tr><td>-</td><td>U2TxIR</td><td>U2RxIR</td><td>-</td></tr>
<tr><td>RW</td><td colspan="4">R-0</td><td>-</td><td colspan="2">RW-0</td><td></td></tr>
</table>

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit11~10 are general registers.)

Bit	Name	Description	
Bit[19]	U2TxIE	UART2 transmits (TX) interrupt enable control.	
		0	Disable
		1	Enable
Bit[18]	U2RxIE	UART2 receives (RX) interrupt enable control.	
		0	Disable
		1	Enable
Bit[11]	U2TxIR	UART Tx interrupt request	
		0	Normal
		1	Interrupt
Bit[10]	U2RxIR	UART Rx interrupt request	
		0	Normal
		1	Interrupt
Bit[03]	U2TxIF	UART transmits (TX) interrupt flag (level-trigger).	
		0	Normal
		1	UART transmission (TX) interrupt occurs.
Bit[02]	U2RxIF	UART receives (RX) interrupt flag (level-trigger).	
		0	Normal
		1	UART reception (RX) interrupt occurs.

7.3.8. Interrupt control register INTTMB2

INT Base Address + 0x1C (0x4001C)						
Symbol	INTTMB2 (Interrupt Control Register 7)					
Bit	[31:24]			[23:18]	[17]	[16]
Name	MASK			-	TMB2IE	-
RW	R0W-0			-	RW-0	-
Bit	[15:10]	[9]	[8]	[7:2]	[1]	[0]
Name	MASK			-	TMB2IF	-
	-	TMB2IR	-			
RW	R-0			-	RW-0	

(When writing the register, the Bit15~8 are Mask; when reading the register, the Bit9 is general register.)

Bit	Name	Description	
Bit[17]	TMB2IE	Timer B2 interrupt enable control	
		0	Disable
		1	Enable
Bit[9]	TMB2IR	Timer B2 interrupt request	
		0	Normal
		1	Interrupt
Bit[1]	TMB2IF	Timer B2 interrupt flag (level-trigger)	
		0	Normal
		1	TMB2 transmission interrupt occurs.

8. WATCH DOG TIMER (WDT)

8.1. Overall description

The watch dog timer (WDT) is, as the name implies, the watcher of the chip, and its main function is to generate the wake-up event or execute basic reset function after the chip crashes accidentally.

Operation mode:

The WDT overflows and then generate the reset signal to reset the chip.

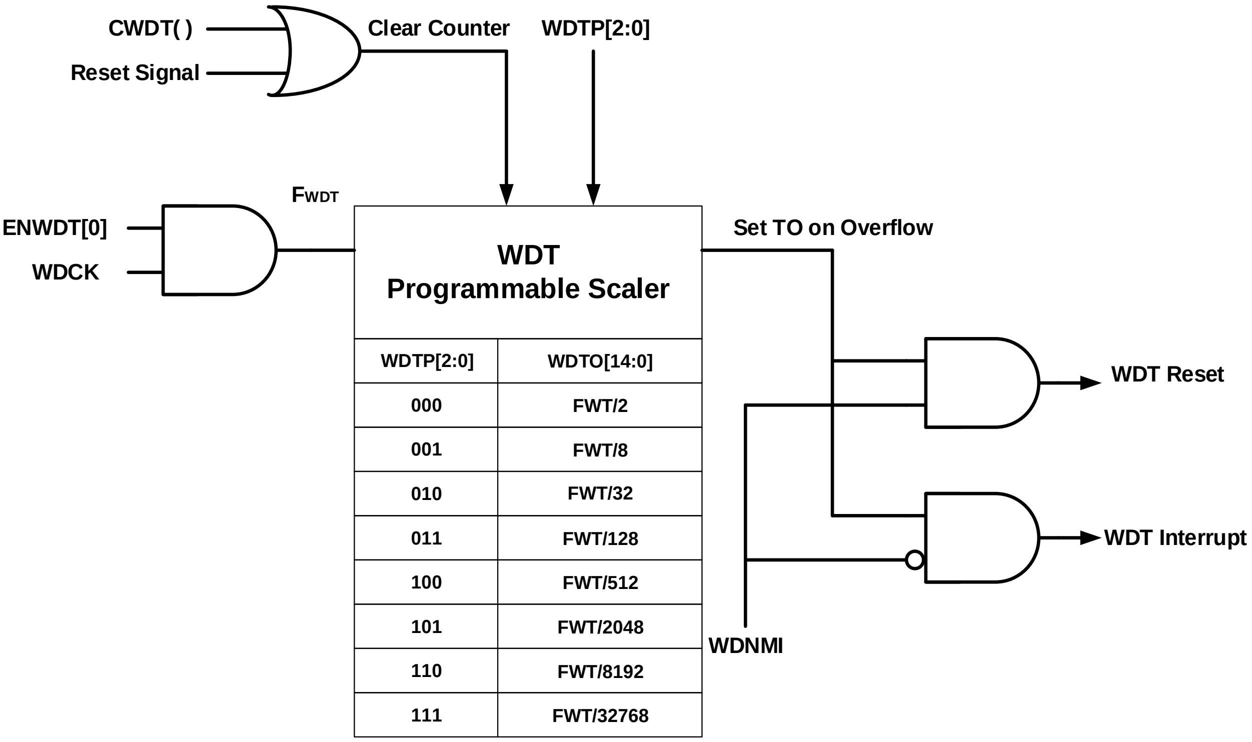
The WDT can be cleared by using software.

SLEEP mode:

The WDT is disabled, and cannot work.

IDLE mode:

The WDT overflows and then generate the interrupt event to wake up the chip.



WDTP[2:0]	WDTO[14:0]
000	FWT/2
001	FWT/8
010	FWT/32
011	FWT/128
100	FWT/512
101	FWT/2048
110	FWT/8192
111	FWT/32768

FIG. 8-1 WDT block diagram

8.1.1. WDT operating instruction

Setting the frequency divider WDTP 0x40108 [2:0] can determine the operating frequency and the overflow value of the WDT. After the WDT overflows, the WDT reset signal or interrupt event can be generated. The control bit WDNMI 0x40108[6] determines the reset signal or the interrupt request signal will be generated after the WDT overflows; if 0 is written in the bit, the WDT will generate the interrupt request signal. Please refer to the chapter about the interrupt control chapter for more information about the interrupt mode. The WDT can start up only when the chip is in operation; the WDT can start up by setting the control bit ENWDT 0x40108 [4] as <1>. It is necessary to enable the global interrupt control bit GIE before enabling the interrupt function.

The operating frequency source of the WDT is LSRC; therefore, the calculation of the theoretical values of the operating frequency and the overflow value of the WDT is as follows:

WDT= LSRC /256 /WDTP[2:0] （Equation 8-1）

LSRC is the frequency of the internal low-speed RC oscillator; and WDTP is the frequency divider；

Assuming that LSRC=33.9 KHz and WDTP=32768, the operating frequency of the WDT is:
33900Hz/256/WDT_PS (32768) =0.00404Hz

8.2. Register address

SoC Register Address	31	24	23	16	15	8	7	0
SoC Base Address + 0x08(0x40108)	WDTO1		WDTO0		MASK0		REG0	

8.3. Register function

8.3.1. WDT register WDTCR

SoC Base Address + 0x08 (0x40108)								
Symbol	WDTCR (WDT Control Register)							
Bit	[31]	[30:16]						
Name	-	WDTO						
RW	-	R-0						
Bit	[15]	[14:8]	[7]	[6]	[5]	[4]	[3]	[2:0]
Name	-	Mask	-	WDNMI	CLRWDT	ENWDT	-	WDTP
RW	-	R0W-0	-	RW1-0	RW-0	RW1-0	-	RW-7

Bit	Name	Description	
Bit[30~16]	WDTO	counter register of WDT	
		0	Set 0
		1	Set 1
Bit[06]	WDNMI	WDT interrupt operating mode selection	
		0	Timer mode
		1	Reset Mode **(As long as the Reset Mode is set, the Timer Mode cannot be switched)**
Bit[05]	CLRWDT	WDT reset control	
		0	Disable
		1	Enable
Bit[04]	ENWDT	WDT enable control	
		0	Disable
		1	Enable**(As long as the setting is on, it will not turn off)**
Bit[2~0]	WDTP	WDT overflow value configuration	
		000	0 : WCLK / 2
		001	1 : WCLK / 8
		010	2 : WCLK / 32
		011	3 : WCLK / 128
		100	4 : WCLK / 512
		101	5 : WCLK / 2048
		110	6 : WCLK / 8192
		111	7 : WCLK / 32768

Note : After WDT Reset, the PC counter will jump to 0x80000 ROM area. It is no executing register initialization.

BOR Reset : H.W. IP & register initialization -> Jump to 0x80000 ROM area

WDT Reset : Jump to 0x80000 ROM area.

9. TIMER A

9.1. Overall description

Timer A is a 16-bit up counter and can be operated in Operation mode, IDLE mode, and Wait mode. It can be used to generate different output frequencies.

Main features:

Up counter

16-stage overflow values are available to be selected.

Overflow generates an interrupt event.

The values of the counter can be read.

Initial configuration of Timer A (TMA):

TMA is a 16-bit up counter. Its input clock source is the TACK and it will perform the counting according to each rising edge of the TACK and the frequency of the input clock source is controlled by the clock system management module. The function of the TMA can be enabled or disabled by setting the control bit ENTA 0x40C00 [5] as 1 or 0.

The overflow value of the TMA can be adjusted by the frequency divider TAS 0x40C00 [3:0]; the user can change the overflow value by modifying the value of the frequency divider TAS to generate the counting values with different frequencies. The control bit TACLR 0x40C00[4] is set as 1 but the TMA is reset and the counter register becomes 0; after the counter register is cleared, the control bit TACLR will automatically become 0.

After the TMA overflows, the interrupt request will be generated and the TMA interrupt flag TMAIF 0x40004[0] will be set as <1>; if the TMA interrupt function is enabled and the global interrupt control bit is set as `, the chip will enter the TMA interrupt service event in response to the TMA interrupt request. The TMA interrupt request can be cancelled by clearing the TMA interrupt flag; in this way, the chip will not reply the TMA interrupt. Note, TMAIF although interrupt flag may be set to <0>, but TMA after counting overflows, as it will interrupt occurs, the interrupt flag TMAIF this time or will automatically be set to <1>
Under the IDLE mode, the TMA interrupt can be used to wake up the chip. Under the SLEEP mode, the TMA interrupt is not available.

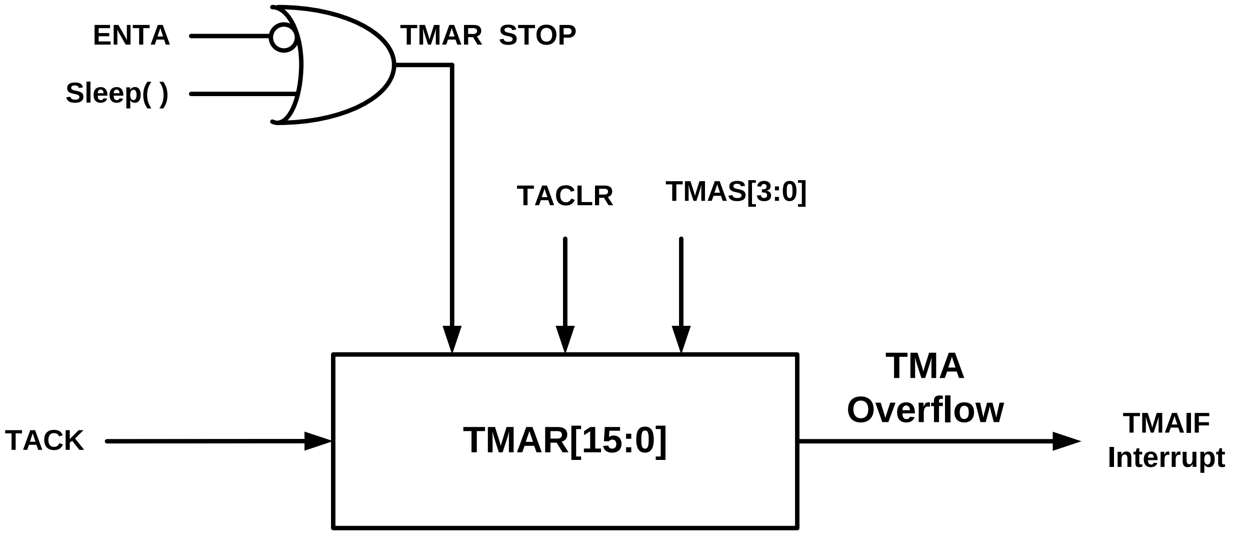
The TMA has a 16-stage frequency dividing configuration, which allows the TMA to have a wide counting range; the calculation of the overflow value of the TMA is as follows:

TAR[15:0]=1/(TACK/32/TAS[3:0])　（Equation 9-1）

The TACK is the input clock source of the TMA and the TAS [3:0] is the frequency dividing value;

Assuming the TMA selects the LS_CK, and the LS_CK is from the LPO; then TACK=35 KHz, TAS [3:0] =1001B=/1024 and the theoretical value of the overflow value of the Timer A:

35000Hz/32/TAS (1024)=35000Hz/32/1024=1.068Hz



TMAS[3:0]	TMAR[15:0]	TMAS[3:0]	TMAR[15:0]
0000	TACK/2	1000	TACK/512
0001	TACK/4	1001	TACK/1024
0010	TACK/8	1010	TACK/2048
0011	TACK/16	1011	TACK/4096
0100	TACK/32	1100	TACK/8192
0101	TACK/64	1101	TACK/16384
0110	TACK/128	1110	TACK/32768
0111	TACK/256	1111	TACK/65536

FIG. 9-1 Timer A block diagram

9.2. Register address

TMA Register Address	31	24	23	16	15	8	7	0
TMA Base Address + 0x00(0x40C00)	TMAR1		TMAR0		MASK0		REG0	

9.3. Register function

9.3.1. Timer A register TMACR

TMA Base Address + 0X00 (0X40C00)					
Symbol	TMACR(TMA Control Register)				
Bit	[31:16]				
Name	TMAR				
RW	R-0				
Bit	[15:8]	[07:06]	[05]	[04]	[03:00]
Name	MASK	-	ENTA	TACLR	TAS
RW	R0W-0	-	RW-0		RW-0XF

Bit	Name	Description	
Bit[31-16]	TMAR	Timer A Counter counting value	
		TAR[31:16] are the counting values of the 16-bit Timer A, And the output value can be MSB to LSB.	
Bit[5]	ENTA	Enable the Timer A	
		0	Disable
		1	Enable
Bit[4]	TACLR	Clear the counting value of the Timer A	
		0	Normal
		1	Clear (After the bit is cleared, the bit will automatically become 0).
Bit[3~0]	TAS	Timer A frequency divider configuration	
		0000	Timer A clock/2
		0001	Timer A clock/4
		0010	Timer A clock/8
		0011	Timer A clock/16
		0100	Timer A clock/32
		0101	Timer A clock/64
		0110	Timer A clock/128
		0111	Timer A clock/256
		1000	Timer A clock/512
		1001	Timer A clock/1024
		1010	Timer A clock/2048
		1011	Timer A clock/4096
		1100	Timer A clock/8192
		1101	Timer A clock/16384
		1110	Timer A clock/32768
		1111	Timer A clock/65536

10. TIMER B

10.1. Overall description

The Timer B is a 16-bit counter, which can be used to perform time counting, time controlling, clock generating and time delaying, etc. It will generate the interrupt signal when the counting flow takes place, and the program can read the current counting value of the TMB; besides, the TMB can be also used to generate the waveform of the PWM. It can be operated under the Operation mode, IDLE mode and the Wait mode.

The 16-bit counter register of the Timer B can be separated into two independent 8-bit counter registers; thus, the TMB has four counting methods:

16-bit up counting method, which can generate the interrupt signal;

16-bit counting method; it will increase to the overflow value and then decrease to 0, which can generate the interrupt signal;

Two independent 8-bit up counting methods; the low 8-bit counter overflows and then the high 8-bit counter is automatically added by 1, which can generate the interrupt signal;

Moreover, the TMB has three counter overflow controller: TBC0, TBC1 and TBC2. TMB can also serve as the PWM waveform generator, which can provide two PWM waveforms PWM0/PWM1; and each has multiple operation modes and can satisfy different PWM output requirements; the operation modes are as follows:
PWMA /PWMB /PWMC /PWMD /PWME /PWMF /PWMG

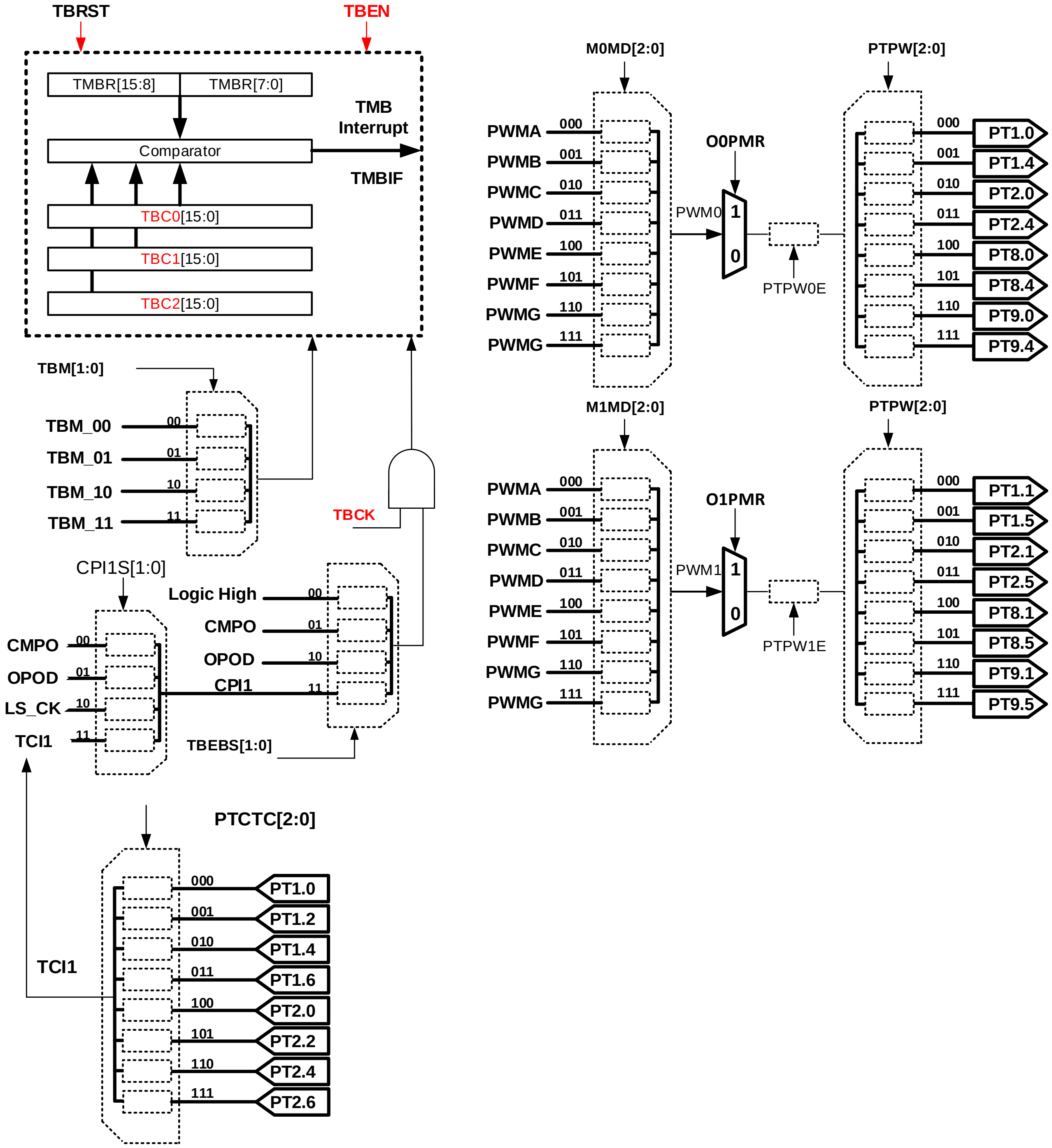


FIG. 10-1 Timer B block diagram

10.1.1. Timer mode

The Timer B is a 16-bit up counter, which can be used to generate the PWM waveforms. It can be used to perform the time counting, time controlling, clock generating, etc., and can generate the interrupt signal when the counter overflow takes place. The TMB can be operated under the Operation mode, IDLE mode and the Wait mode.

It has four different counting methods, and can generate the counting values with different frequencies:

(1)16-bit up counting method, which can generate the interrupt signal;

(2)16-bit counting method; it will increase to the overflow value and then decrease to 0, which can generate the interrupt signal;

(3)Two independent 8-bit up counting methods; the maximum count value 0xFF, can generate an interrupt signal

(4) The low 8-bit counter overflows and then the high 8-bit counter are automatically added by 1, which can generate the interrupt signal;

It has four different counting-trigger signal sources, which can be applied to count different events:

(1)Continuous counting method is always enabled;

(2)The comparator outputs (CMPO) high-potential trigger.

(3)The OP amplifier outputs (OPOD) high-potential trigger.

(4)The Timer C outputs (CPI1) high-potential trigger.

The operating clock source of the TMB is HS_CK or LS_CK, which will pass through the frequency divider to generate the frequency source TBCLK to provide the operating frequency for the TMB. It provides the setting frequency divider TMCD0x40308 [5:4], which can set different counting cycles for the TMB. The clock source of the TMB can be set at the clock system control module.

TMBR：16-bit timer/counter registers

The TMBR is a 16-bit timer/counter register, which can be separated into two independent 8-bit timer/counter registers in order to satisfy the four different counting methods of the TMB. The TMBR will crease or decrease at each rising edge of the TBCLK; under different counting methods, the TMBR will increase or decrease according to different conditions. TMBR can be automatically cleared by setting the control bit TBRST [1] as <1> and the control bit TBRST will automatically become 0 after the TMBR is cleared. The program can also read the current counting value of the TMBR for other purposes.
The TBEN is the enable control signal of the TMB. If the bit is set as 1, the counting function of the TMB will be enabled; if the bit is set as 0, the counting function of the TMB will be disabled.

The TBEBS [1:0] is the counting-trigger signal source controller; the controller can

provide four different counting-trigger signal sources.
TBM [3:2] is the counting method controller of the TMB; the controller can provide four different counting methods. The TBRST is the control bit of the TMB counter register. If the bit is set as <1>, the counter register will be automatically cleared and then the bit will automatically become 0.

Operating configuration when TMB serves as a timer/counter:
Set the operating clock source of the TMB and set the control bits 0x40308[6](ENTD) and 0x40308[5:4](TMCD);
Select the counting mode and set the control bit TBM[1:0];
Select the counting-trigger signal source and set the TBEBS[1:0]; as a timer, it can be set as 00b, which means it is always enabled and continuously perform counting;
Set the timer/counter overflow value is TBC0[15:0];
Set the control bit TBRST as <1> to clear the counting register;
Enable the TMB and the control bit TBEN is set as <1>.

TMB as the timing counter operation initialization settings:
(1) Select the clock source is HS_CK TMB work or LS_CK (control bit TMCKS 0x40308 [7]), and do clock divider set and open source movement (control bit ENTD 0x40308 [6] and control bits TMCD 0x40308 [5: 4])
(2) Select the count mode, set the register control bits TBM 0x40C04 [3: 2]
(3) Select the trigger count source, set the control bit register TBEBS 0x40C04 [1: 0], as the timer can be set to <00>, which is always enabled, continuous counting;
(4) Set the timer count overflow value, setting register control bits TBC0 0x40C0C [15: 0];
(5) Set the register 0x40C04 [4] = 1, i.e. control bit TBRST set <1>, clearing the count register;
(6) Set register 0x40C04 [5] = 1, i.e. control bit TBEN is set to <1>, enable TMB.
(7) TMB start counting after TMB count overflows, it will generate an interrupt request, TMB interrupt flag register TMBIF 0X40004 [1] is set to <1>, if open TMB interrupt function, that register control bits TMBIE0X40004 [17] is set to <1>, and the global interrupt control bit (GIE) has been set to <1>, the chip will be in response to an interrupt request to enter TMB TMB interrupt service events. TMB interrupt flag is cleared, an interrupt request to cancel the TMB, when they do not respond to the wafer TMB interruption.
(8) Note, TMBIF although interrupt flag may be set to <0>, but TMB after the count overflows, as it will interrupt occurs, the interrupt flag TMBIF this time or will automatically be set to <1>. In standby mode, TMB interrupt can be used to wake up. In sleep mode, TMB interrupts unavailable.

The calculation of the theoretical overflow value of the Timer B:

T = TBC0*1 / TBCLK; TBCLK=HS_CK (or LS_CK) / TMCD; (Equation 10-1)

Then

T=TBC0*TMCD / HS_CK (or LS_CK); (Equation 10-2)

The TMB has four different counting methods, and different counting method have different overflow conditions, which will be specified later.

TMB counting method 0

When register TBM 0x40C04 [3:2] =00b, the register control bit TMBR 0x40C08 [15:0] serves as a 16-bit up counter. Under the mode, the TMBR will be automatically added by 1 at each rising edge of the TBCLK; if the counting value of the TMBR is higher than Register control bits 0x40C0C TBC0 [15: 0], the TMBR will become 0 at the next rising edge and the timer interrupt flag TBCLK is set as <1>;(as 0X40004[1] =1) if the interrupt function of the TMB and the global interrupt function are enabled, the chip will reply the TMB interrupt. Then, the TMBR will restart the up counting. The schematic view of the counting waveform of the mode is as shown in the follow figure.

The counting cycle calculation method of the TMB under the mode: T=TBC0*TMCD / HS_CK (or LS_CK)

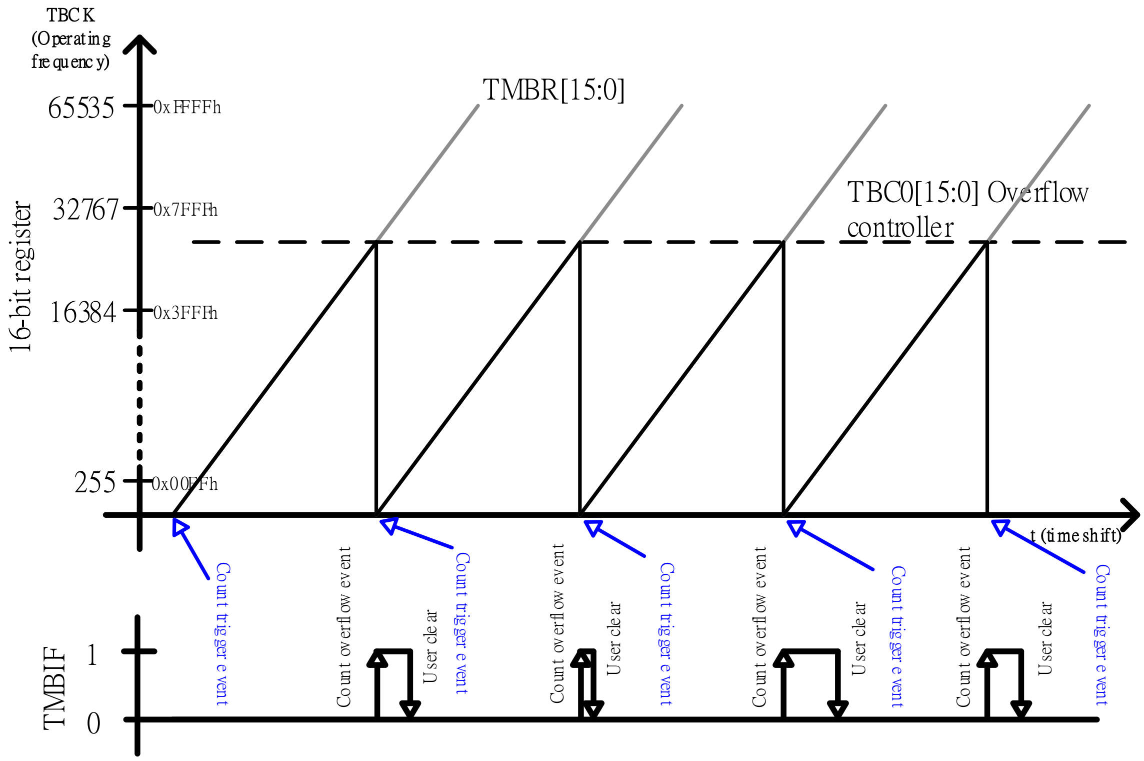


FIG. 10-2 Schematic view of counting waveform of counting method 0

TMB counting method 1

When TBM 0x40C04 [3:2] =01b, the TMB will perform incremental counting and then perform decrement counting; the TMBR is a 16-bit counter. After enabled, the TMB will perform incremental counting, and the TMBR will automatically be added 1 at each rising edge of the TBCLK. When the TMBR is equal to TBC0, the TMBR will be changed to downward mode, but the interrupt flag TMBIF is still 0; at the next rising edge of the TBCLK, the TMBR will be changed to perform decrement counting; the interrupt request will take place until the TMBR is decreased to 0 and the interrupt flag TMBIF is set as <1>, and then the TMBR will start to perform incremental counting at the next rising edge of the TBCLK. The above process will be kept repeating. The schematic view of the counting waveform of the mode is as shown in the following figure.

In the mode, the calculation method of the counting cycle of the TMB is:
T=2*TBC0*TMCD / HS_CK (or LS_CK)

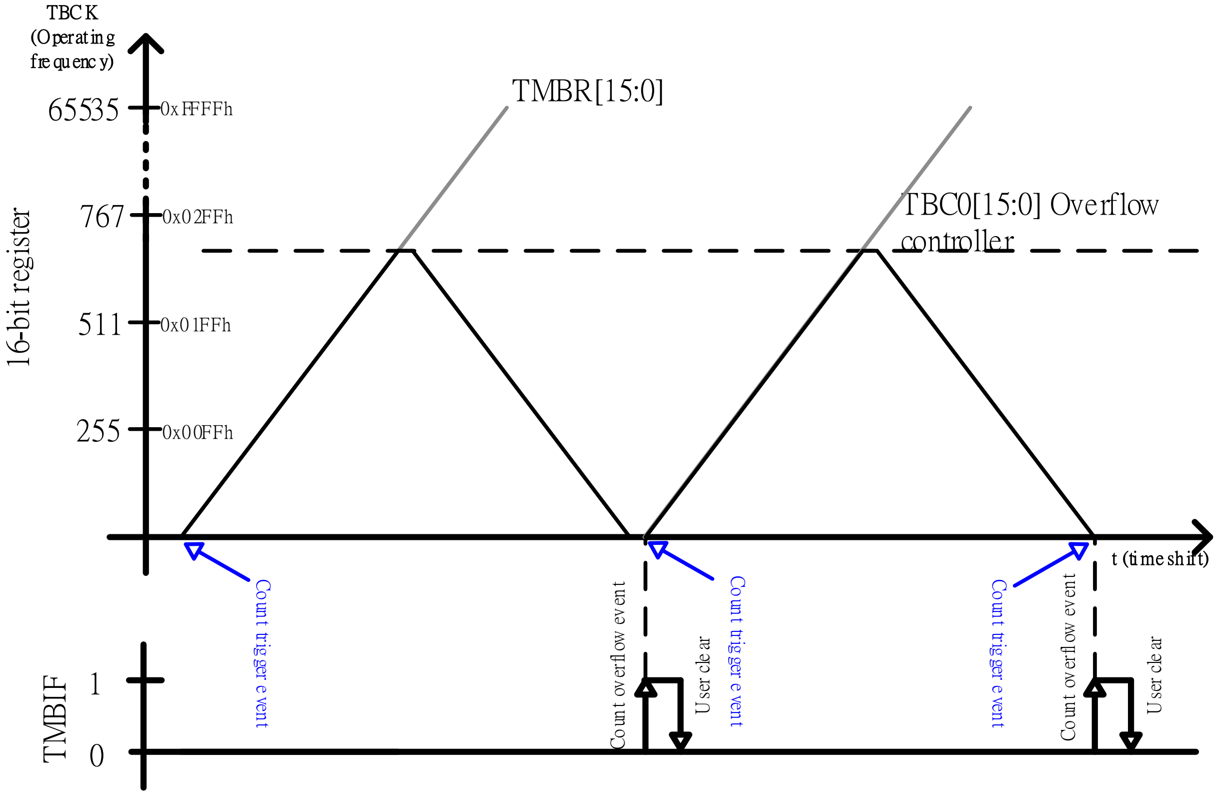


FIG. 10-3 Schematic view of counting waveform of counting method 1

TMB counting method 2

When TBM 0x40C04 [3:2] =10b, the TMB will perform incremental counting, but the TMBR is separated into two independent 8-bit counters: TMBR [15:8] and TMBR [7:0]. Besides, the two independent 8-bit counters perform incremental counting at the same time. The overflow value of the TMBR [15:8] is controlled by the TBC0 [15:8] and the overflow value of the TMBR [7:0] is controlled by TBC0 [7:0]. The two counters will be automatically added by 1 at each rising edge of the TBCLK. If the TMBR [15:8] is equal to the TBC0 [15:8], the TMBR [15:8] will become 0 at the next rising edge of the TBCLK but the interrupt flag TMBIF is still 0; if the TMBR [7:0] is equal to TBC0 [7:0], TMBR [7:0] will become 0 at the next rising edge of the TBCLK and the interrupt flag TMBIF will be set as <1>. At this time, if the TMB interrupt function and the global interrupt enable function are enabled, the chip will reply to the TMB interrupt. Under the mode, the interrupt request is controlled by the counter TMBR [7:0]; therefore, during the mode, please pay attention to set the value of the TBC0 [7:0] in order to control the TMB interrupt vector. The schematic view of the counting waveform of the mode is as shown in the following figure.

In the mode, the calculation method of the counting cycle of the interrupt method of the mode is: T=TBC0 [7:0]*TMCD / HS_CK (or LS_CK);

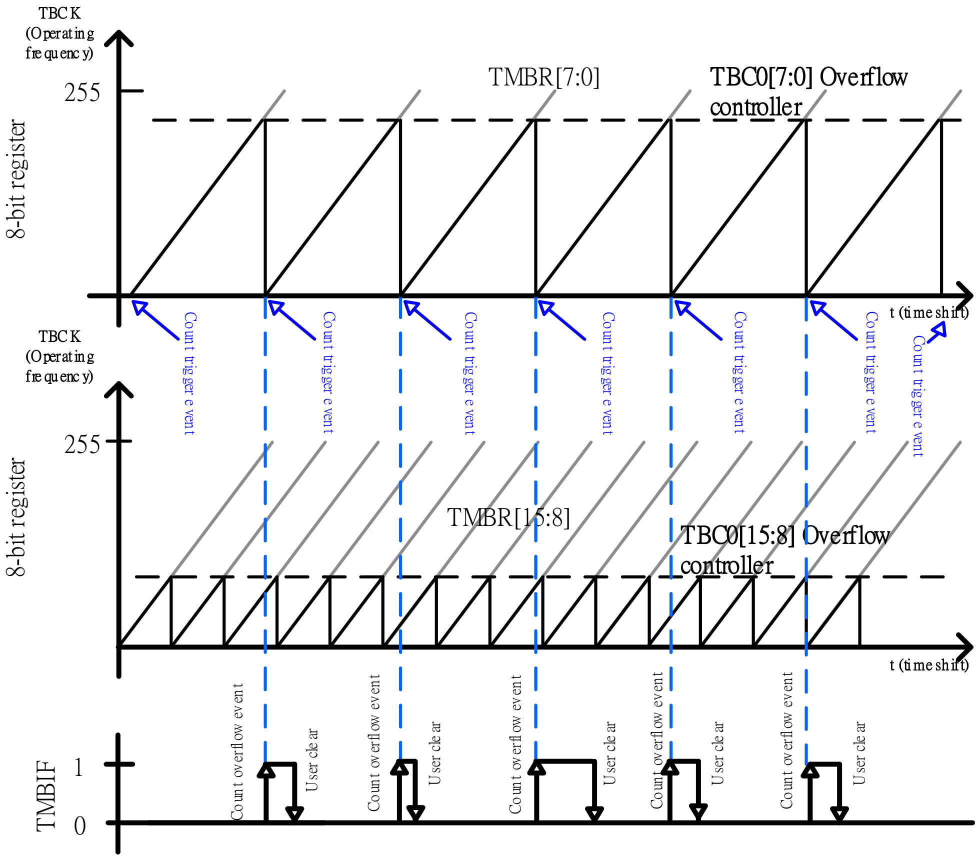


FIG. 10-4 Schematic view of counting waveform of counting method 2

TMB counting method 3

When TBM 0x40C04 [3:2] =11b, the TMB will perform incremental counting, and the TMBR is separated into two counters: TMBR [15:8] and TMBR [7:0]; and both of them are under incremental counting mode. The overflow value of the TMBR [7:0] is controlled by the TBC0 [7:0] and the overflow value of the TMBR [7:0] is controlled by TBC0 [7:0]. TMBR [7:0] will be automatically added by 1 at each rising edge of the TBCLK; if the TMBR [7:0] is equal to the TBC0 [7:0], the TMBR will become 0 at the next rising edge of the TBCLK; besides the TMBIF will become 1 and the TMBR [15:8] will be automatically added by 1. At this time, if the TMB interrupt function and the global interrupt enable function are enabled, the chip will reply to the TMB interrupt. The schematic view of the counting waveform of the mode is as shown in the following figure.

In the mode, the calculation method of the counting cycle of the interrupt method of the mode is: T=TBC0 [7:0]*TMCD / HS_CK (or LS_CK);

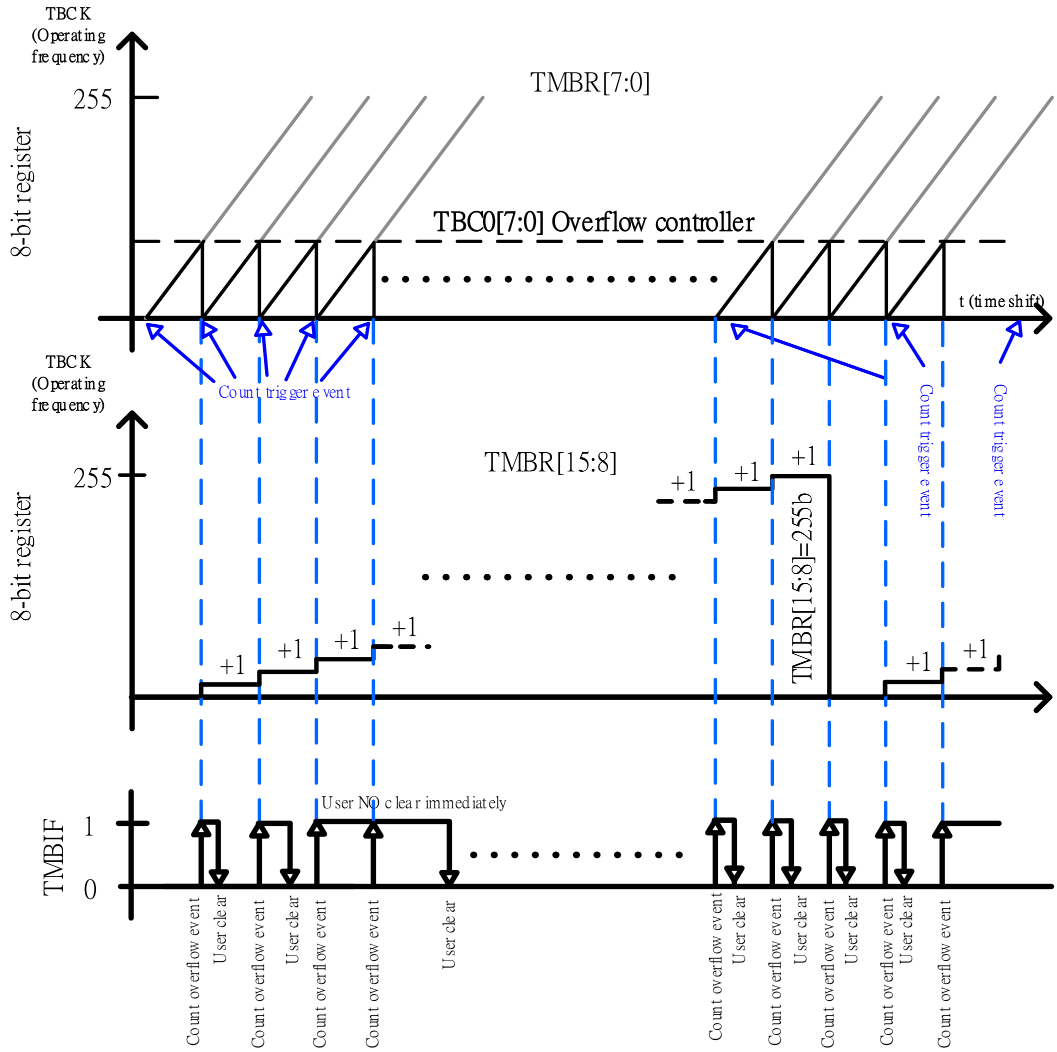


FIG. 10-5 Schematic view of counting waveform of counting method 3

10.1.2. PWM function mode

When the timer B works under the PWM mode, the combinations of the different counting methods and different PWM mode selectors can generate different PWM waveforms. The chip has only two PWMs: PWM0/PWM1, which can be simply considered two PWM waveform generators; the combinations of the different counting methods and different PWM operating modes can generate many kinds of PWM waveforms. The chip provides many output pins for the output of the PWM, and each PWM waveform generator is corresponding to 8 output IO ports; therefore, the usage and output of the PWM is very flexible. However, the TMB is necessary for the function; that is to say, the TMB should be enabled to set the counting cycle of the TMB.

Each of the PWM waveform generators (PWM0/PWM1) has many operating modes: PWMA, PWMB, PWMC, PWMD, PWME, PWMF and PWMG. The operating modes of the PWM0 and PWM1 can be changed by setting the control bits O0MD 0x40C04 [18:16] and 0x40C04 [22:20]. The phase of the output waveform of the PWM can be changed by setting the control bits O1PMR 0x40C04 [23] and O0PMR 0x40C04 [19]. The user can check the current operating mode of the PWM 0x40C08 [21:16] via the PWM operating mode flag register; if the flag is 1, it means the operating mode is enabled. The TBC1 0x40C10 [15:0]/TBC2 0x40C10 [31:16] are the duty cycle controller of the PWM0/PWM1 respectively; the duty cycles of the PWMs can be changed by setting the values of the TBC1/TBC2.

The chip provides 8 output IOs for each PWM, and the corresponding pins are distributed over the PT1/PT2; the selection and enablement of the output pins of the PWM1 and PWM0 are controlled by the controllers PTPW 0x40840[4:2], PTPW1E 0x40840[1] and PTTPW0E 0x40840[0]. The output and disablement of the PWMs can be controlled by the enablement and disablement of the output pins of the PWMs. If the user wants to completely disable the PWMs, it is necessary to disable the output pins of the TMB and the PWMs. The output pins of the PWMs are as shown in Table 10.1.

Serial number PTPW[2:0]	PWM0 Output pins	PWM1 Output pins	Serial number PTPW[2:0]	PWM0 Output pins	PWM1 Output pins
000	PT1.0	PT1.1	100	PT8.0	PT8.1
001	PT1.4	PT1.5	101	PT8.4	PT8.5
010	PT2.0	PT2.1	110	PT9.0	PT9.1
011	PT2.4	PT2.5	111	PT9.4	PT9.5

Table 10-1 PWM output pin distribution

PWM initialize operation description:

(1) Select the operating mode and PWM duty cycle, the output waveform phase, namely setting register 0x40C04 control bit O0MD / O0PMR, O1MD / O1PMR, write timer counter overflow value to register 0x40C10 control bit TBC1 / TBC2.

(2) The IO control output enable and off, can control the PWM output on and off, if you want to completely shut down PWM, you must shut down TMB.

(3) By the control bit register 0x40840 PTPW, PTPW1E, PTPW0E control PWM1, select PWM0 output pin and open.

(4) Select the clock source is HS_CK TMB work or LS_CK (control bit TMCKS 0x40308 [7]), and do clock divider set and open source movement (control bit ENTD 0x40308 [6] and control bits TMCD 0x40308 [5: 4]).

(5) Select the count mode, set the register control bits TBM 0x40C04 [3: 2].

(6) Select the trigger count source, set the control bit register TBEBS 0x40C04 [1: 0], as the timer can be set to 00b, that is always enabled, continuous counting.

(7) Set the timer count overflow value, setting register control bits TBC0 0x40C0C [15: 0].

(8) Set the register 0x40C04 [4] = 1, i.e. control bit TBRST set <1>, the count register is cleared.

(9) Set register 0x40C04 [5] = 1, namely TBEN control bit is set to <1>, enable TMB.

The waveform of the PWM is generated by the combination of the TMBR, TBC0, TBC1 and TBC2; and there are 7 kinds of operating modes; thus, the operating conditions of the operating modes are different from each other. The 7 operating modes will be respectively specified later. The usage conditions and the controls of the two independent PWMs: PWMO0 and PWMO1; therefore, they will not be specified separately.

PWMA mode

The PWMA mode is a 16-bit PWM; the counting value of the TMBR is compared with the TBC1 and the waveform period of the PWM is controlled by the TBC0.

PWM output state controlled conditions:

PWM = 1, when TMBR [15:0] >= TBC1 [15:0];

PWM = 0, when TMBR [15:0] < TBC1 [15:0];

PWM period:

PWM Period = TMBR[15:0]*TMCD / HS_CK(or LS_CK)；

PWMA frequency and duty cycle formula:

$$\text{PWMA Frequency} = \frac{\text{TBCK}}{\text{TBC0}[15:0]+1}$$

$$\text{PWMA Duty Cycle} = \frac{(\text{TBC0}[15:0]+1)-\text{TBC1}[15:0]}{\text{TBC0}[15:0]+1}$$

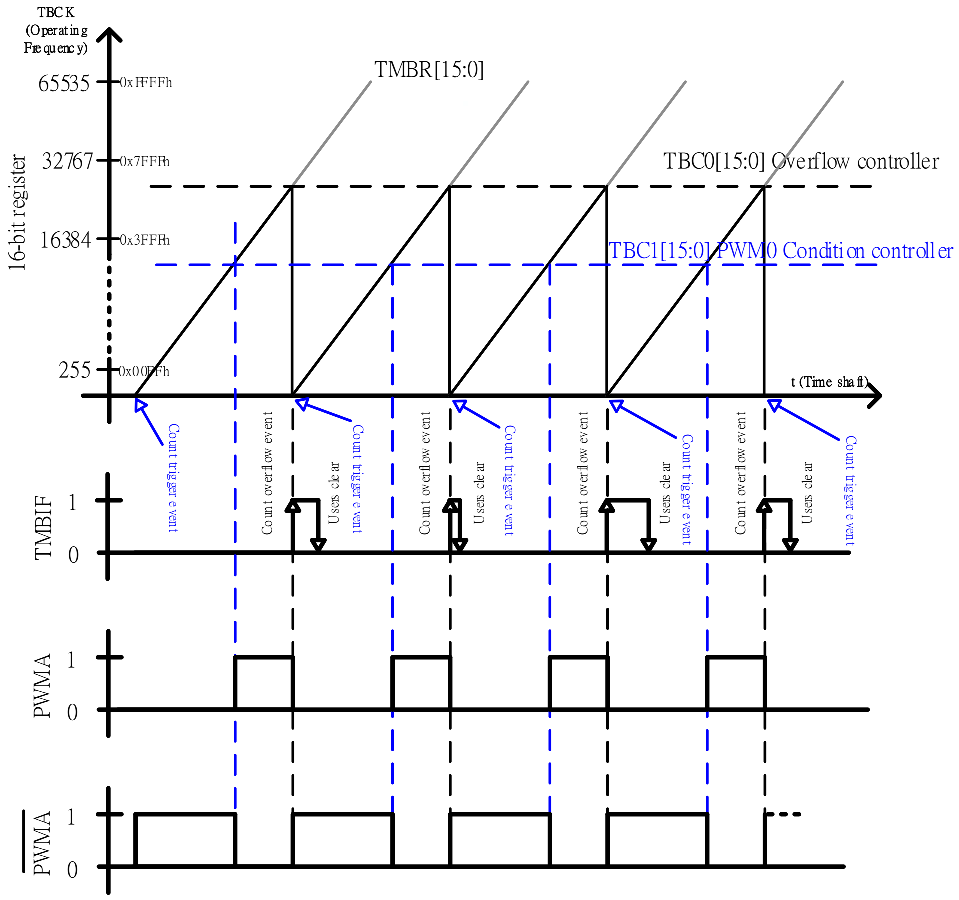


FIG.10-6 Waveform schematic view and counting waveform schematic view of PWM mode A

PWMB mode

The PWMB mode is a 16-bit PWM; the counting value of the TMBR is compared with the TBC2 and the waveform period of the PWM is controlled by the TBC0.

PWM output state controlled conditions:

PWM = 1, when TMBR [15:0] >= TBC2 [15:0];

PWM = 0, when TMBR [15:0] < TBC2 [15:0];

PWM period:

PWM Period = TMBR [15:0]*TMCD / HS_CK (or LS_CK);

PWMB frequency and duty cycle formula:

$$\text{PWMB Frequency} = \frac{\text{TBCK}}{\text{TBC0}[15:0]+1}$$

$$\text{PWMB Duty Cycle} = \frac{(\text{TBC0}[15:0]+1)\text{ - TBC2}[15:0]}{\text{TBC0}[15:0]+1}$$

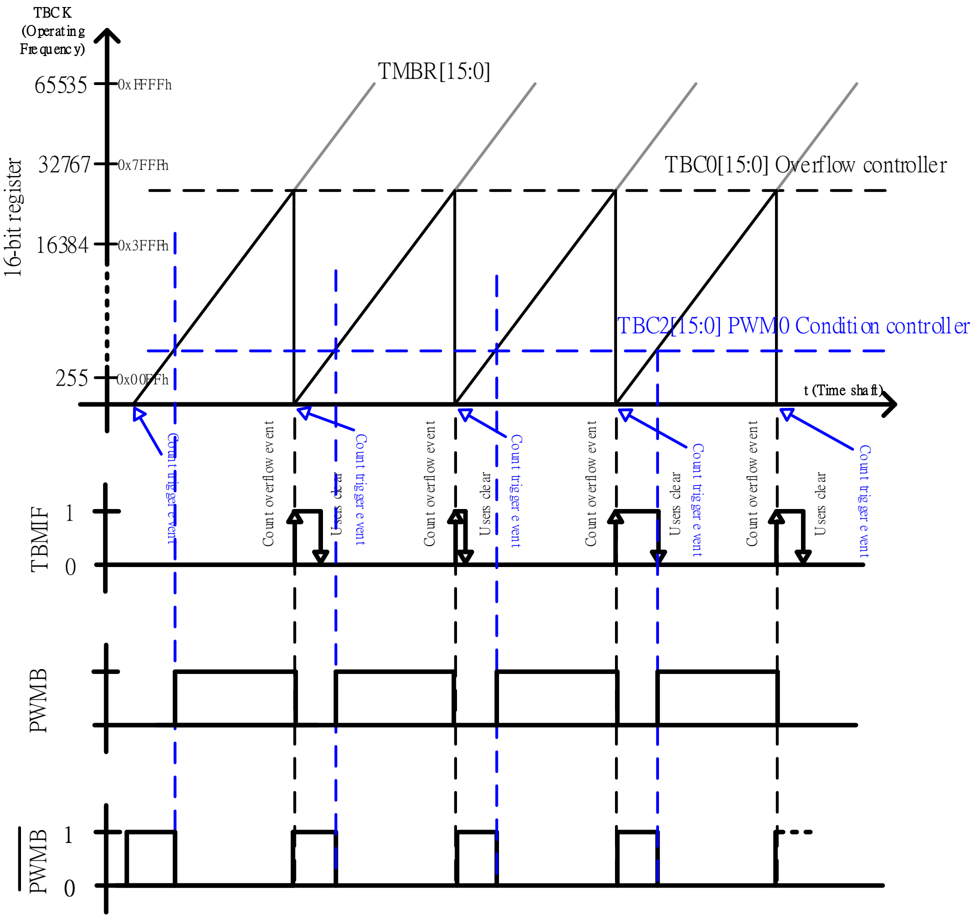


FIG.10-7 Waveform schematic view and counting waveform schematic view of PWM mode B

PWMC mode

The PWMC mode is an 8-bit PWM; the counting value of the TMBR is compared with the TBC1 [7:0] and many PWM waveforms appear within the period of the TBC0.

PWM output status control conditions:

PWM = 1, when TMBR [7:0] >= TBC1 [7:0].

PWM = 0, when TMBR [7:0] < TBC1 [7:0].

PWM period:

PWM Period = TMBR [7:0]*TMCD / HS_CK (or LS_CK);

PWMC frequency and duty cycle formula:

$$\text{PWMC Frequency} = \frac{\text{TBCK}}{\text{TBC0}[7:0]+1}$$

$$\text{PWMC Duty Cycle} = \frac{(\text{TBC0}[7:0]+1)-\text{TBC1}[7:0]}{\text{TBC0}[7:0]+1}$$

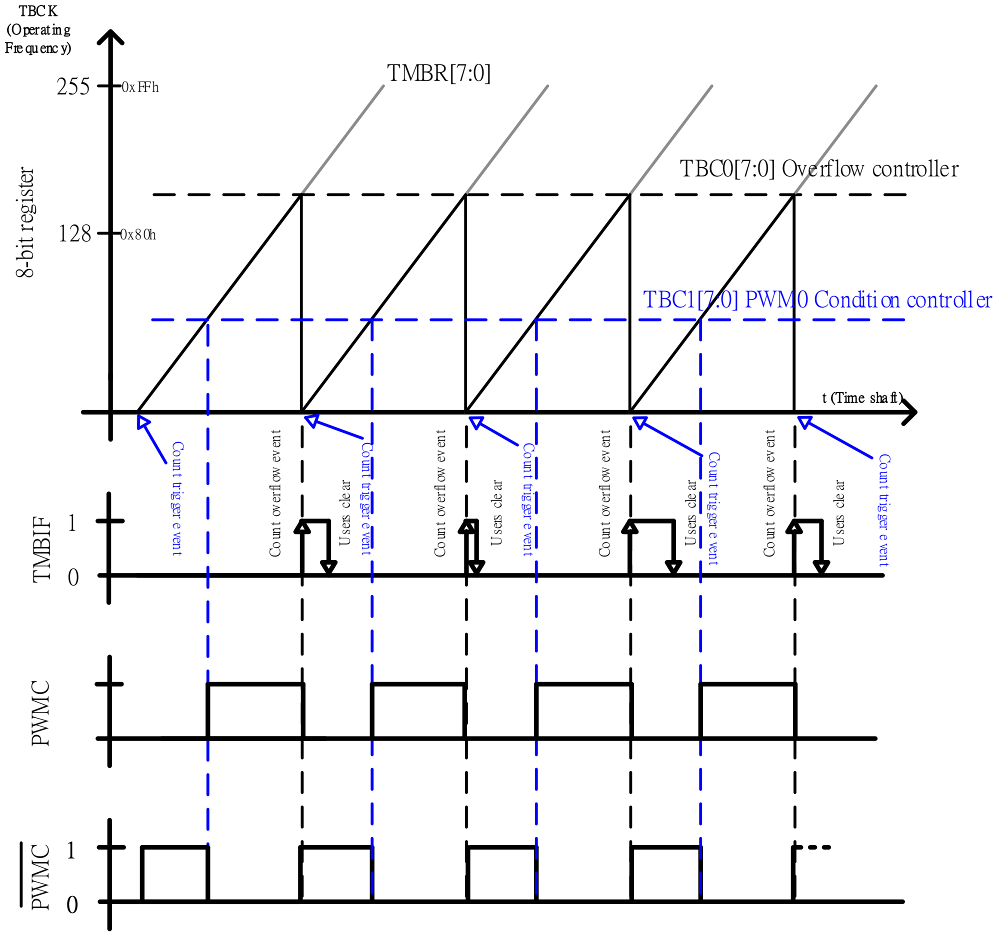


FIG.10-8 Waveform schematic view and counting waveform schematic view of PWM mode C

PWMD mode

The PWMD mode is an 8-bit PWM; the counting value of the TMBR is compared with the TBC2 [7:0] and many PWM waveforms appear within the period of the TBC0.

PWM output status control conditions:

PWM = 1, when TMBR [15:8] >= TBC2 [7:0];

PWM = 0, when TMBR [15:8] < TBC2 [7:0];

PWM period:

PWM Period = TMBR [15:8]*TMCD / HS_CK (or LS_CK);

PWMD frequency and duty cycle formula:

$$\text{PWMD Frequency} = \frac{\text{TBCK}}{\text{TBC0}[15:8]+1}$$

$$\text{PWMD Duty Cycle} = \frac{(\text{TBC0}[15:8]+1)-\text{TBC2}[7:0]}{\text{TBC0}[15:8]+1}$$

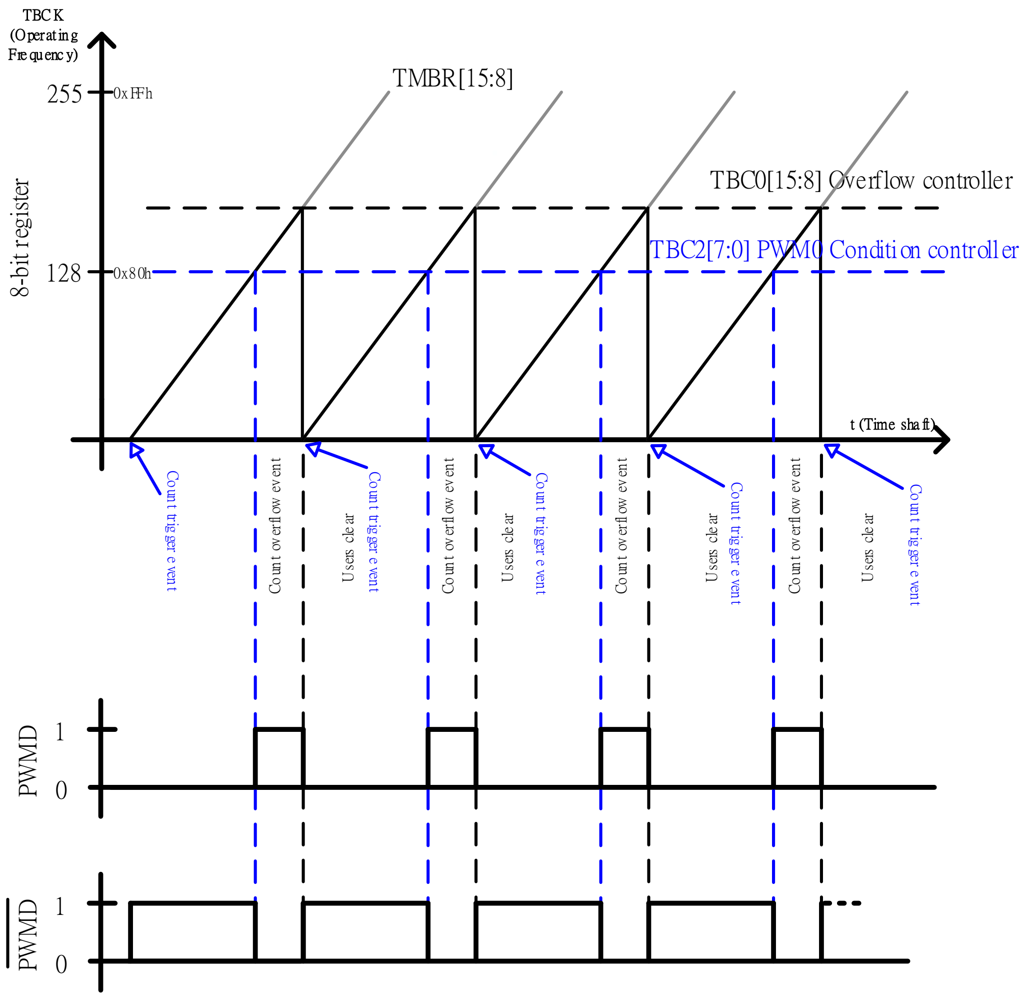


FIG.10-9 Waveform schematic view and counting waveform schematic view of PWM mode D

PWME waveform (8+8-bit PWM)

Set the TMB counter as the 8+8-bit mode and select the PWME as the output waveform of the PWM; then the 8+8bit PWM output is acquired.

The 8+8-bit PWM is composed of the control registers TMBR[7:0], TMBR[15:8], TBC0[7:0], TBC1[7:0] and TBC2[7:0], etc., and the internal digital circuits, where the TMBR[7:0] is the accumulating counter, the TBC0[7:0] is the PWM frequency controller and when the counting of the TMBR[7:0] reaches the TBC0[7:0], the TMBR[15:8] will be added by 1; the TBC1[7:0] is the PWM duty cycle controller and the TBC2[7:0] is 8+8-bit PWM duty cycle adjuster.

(※The following waveform description is under the conditions that the O1PMR or O0PMR

is set as <0> and outputs inversely)

The configuration and description of the 8+8-bit PWM duty cycle adjuster TBC2 [7:0] are as shown in the follow table.

Configuration Weighted quantity	TBC2[7:0]							
	01h	02h	04h	08h	10h	20h	40h	80h
PWM duty cycle fine adjustment	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
Description	TMB overflows 2 times; 1 is N+1, and 1 is N	TMB overflows 4 times; 3 are N, and 1 is N+1	TMB overflows 8 times; 7 are N, and 1 is N+1	TMB overflows 16 times; 15 are N, and 1 is N+1	TMB overflows 32 times; 31 are N, and 1 is N+1	TMB overflows 64 times; 63 are N, and 1 is N+1	TMB overflows 128 times; 127 are N, and 1 is N+11	TMB overflows 256 times; 255 are N, and 1 is N+1

Table 10-1 Configuration table of duty cycle adjuster

- The description of the duty cycle adjuster TBC2[7:0], where N is the width of the duty cycle (PS: *N = TBC1 [7:0]*)
  - Basic type
    - Set the TBC2[7:0] as 01h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 2 output periods as one set, where one outputs N and then the other one outputs N+1.
    - Set the TBC2[7:0] as 02h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 4 output periods as one set, where 3 of them continuously outputs N and the last one outputs N+1

- Set the TBC2[7:0] as 04h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 8 output periods as one set, where 7 of them continuously outputs N and the last one outputs N+1.
- Set the TBC2[7:0] as 08h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 16 output periods as one set, where 15 of them continuously outputs N and the last one outputs N+1
- Set the TBC2[7:0] as 10h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 32 output periods as one set, where 31 of them continuously outputs N and the last one outputs N+1
- Set the TBC2[7:0] as 20h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 64 output periods as one set, where 63 of them continuously outputs N and the last one outputs N+1
- Set the TBC2[7:0] as 40h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 128 output periods as one set, where 127 of them continuously outputs N and the last one outputs N+1
- Set the TBC2[7:0] as 80h, which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 256 output periods as one set, where 255 of them continuously outputs N and the last one outputs N+1

■ Logic calculation OR superposition type

(Only 1/2+1/4,1/2+1/8,~,1/2+1/4+1/8+1/16+1/32+1/64+1/128,1/2+1/4+1/8+1/16+1/32+1/64+1/256 are used to illustrated.)

- Set the TBC2[7:0] as 03h(1/2+1/4), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 4 output periods as one set, where one of them outputs N and then the other 3 output N+1.
- Set the TBC2[7:0] as 05h(1/2+1/8), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 8 output periods as one set, where 3 of them output N and then the other 5 output N+1.
- Set the TBC2[7:0] as 09h(1/2+1/16), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 16 output periods as one set, where 7 of them output N and then the other 9 output N+1.
- Set the TBC2[7:0] as 11h(1/2+1/32), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 32 output periods as one set, where 15 of them output N and then the other 17 output N+1
- Set the TBC2[7:0] as 21h(1/2+1/64), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 64 output periods as one set, where 31 of them output N and then the other 33 output N+1

- Set the TBC2[7:0] as 41h(1/2+1/128), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 128 output periods as one set, where 63 of them output N and then the other 67 output N+1
- Set the TBC2[7:0] as 81h(1/2+1/256), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 256 output periods as one set, where 127 of them output N and then the other 129 output N+1
- Set the TBC2[7:0] as 07h(1/2+1/4+1/8), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 8 output periods as one set, where one of them outputs N and then the other 7 output N+1
- Set the TBC2[7:0] as 07h(1/2+1/4+1/8+1/16), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 16 output periods as one set, where one of them outputs N and then the other 15 output N+1.
- Set the TBC2[7:0] as 1Fh(1/2+1/4+1/8+1/16+1/32), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 32 output periods as one set, where one of them outputs N and then the other 31 output N+1
- Set the TBC2 [7:0] as 3Fh (1/2+1/4+1/8+1/16+1/32+1/64), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 64 output periods as one set, where one of them outputs N and then the other 63 output N+1.
- Set the TBC2 [7:0] as 7Fh (1/2+1/4+1/8+1/16+1/32+1/64+1/128), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 128 output periods as one set, where one of them outputs N and then the other 127 output N+1.
- Set the TBC2 [7:0] as FFh (1/2+1/4+1/8+1/16+1/32+1/64+1/128+1/256), which makes the waveform of the PWM duty cycle generates the N+1 and N outputs, which is to generate a waveform using 256 output periods as one set, where one of them outputs N and then the other 255 output N+1.

◆ The following Table 10.2 and FIG. 10-10 list partial 8+8-bit PWM waveform changes under different configurations of the TBC2[7:0] for your reference.

8+8bit PWM			Overflowing times of TBN																		
型態	TBC2 [7:0]	邏輯運算	0	1	2	3	4	5	6	7	8	9	10	~	127	128	~	252	253	254	255
基本波	01h	1/2	N	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N+1	N	~	N	N+1	N	N+1
	02h	1/4	N	N	N+1	N	N	N	N+1	N	N	N	N+1	~	N	N	~	N	N	N+1	N
	04h	1/8	N	N	N	N	N+1	N	N	N	N	N	N	~	N	N	~	N+1	N	N	N

形	08h	1/16	N	N	N	N	N	N	N	N	N+1	N	N	~	N	N	~	N	N	N	N
	10h	1/32	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	20h	1/64	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	40hh	1/128	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	80h	1/256	N	N	N	N	N	N	N	N	N	N	N	~	N	N+1	~	N	N	N	N
邏	03h	3/4	N	N+1	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N+1	~	N+1	N	~	N	N+1	N+1	N+1
輯	05h	5/8	N	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	N	~	N+1	N	~	N+1	N+1	N	N+1
運	07h	7/8	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	N+1	N	~	N+1	N	~	N+1	N+1	N+1	N+1
算	0Fh	15/16	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	~	N+1	N	~	N+1	N+1	N+1	N+1
疊	85h	161/256	N	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	N	~	N+1	N+1	~	N+1	N+1	N	N+1
核	8Fh	241/256	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	~	N+1	N+1	~	N+1	N+1	N+1	N+1
波 形	FFh	255/256	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	~	N+1	N+1	N+1	N+1

Table 10-2 PWME output waveform table

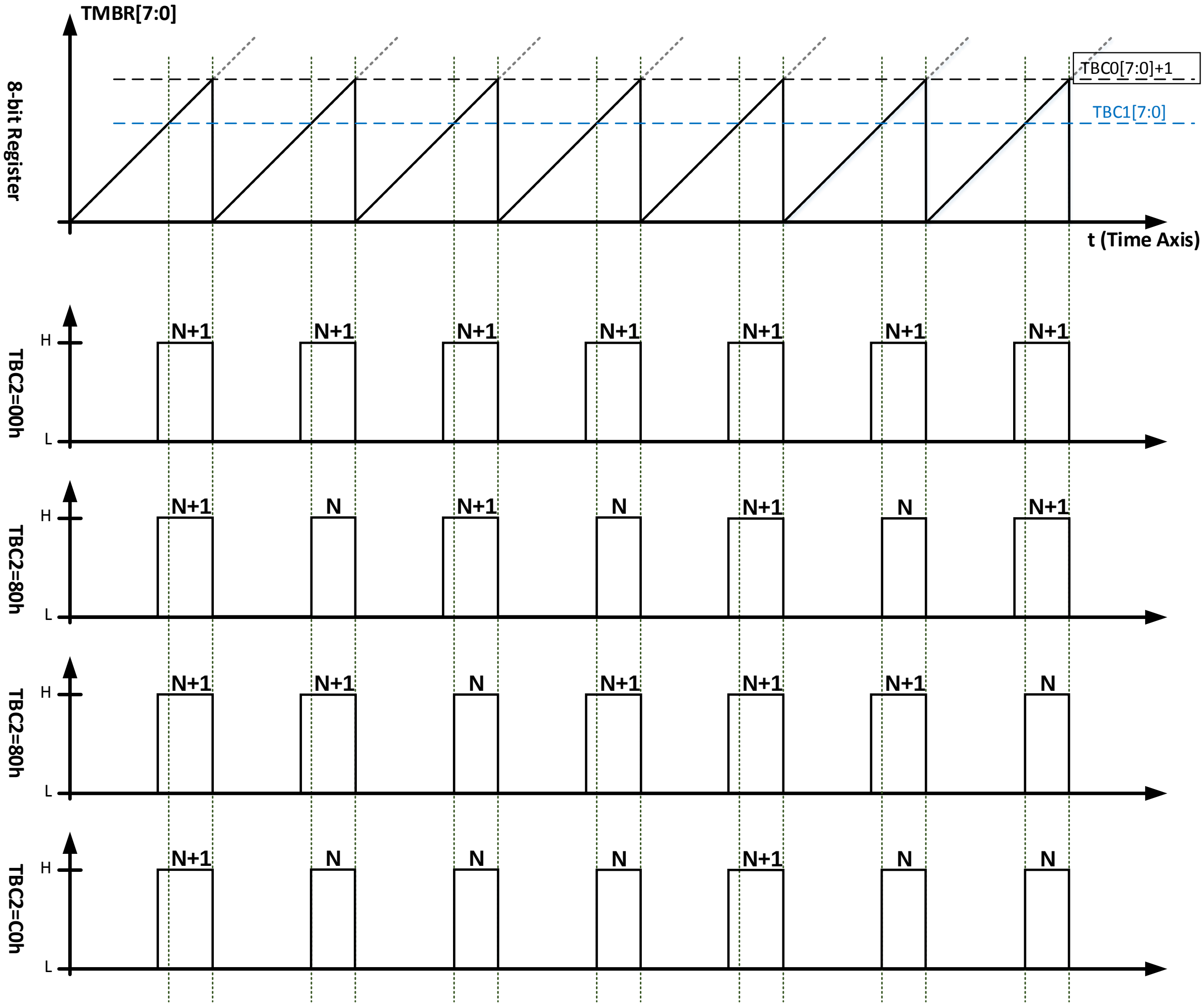


FIG. 10-10 Schematic view of PWME output waveforms

PWME output operation description

- Initialization (The PWME frequency and duty cycle configuration)
  - Setting the register's control bit TMCKS 0x40308[7] can select the operating frequency source of the TMB, and setting the control bit TMCD 0x40308[5:4] can determine the operating frequency of the TMB.
  - When the TBM 0x40C04 [3:2] is set as <11>, the TMB serves as 8+8-bit counter.
  - When the O0MD 0x40C04 [18:16] or control bit O1MD 0x40C04 [22:20] is set as <100>, the output waveform is the PWME.
  - Setting the TBEBS 0x40C04 as <00> can set the counting-trigger signal as "Always Enable", which means cycle counting
  - Write data in the TBC0 [7:0] to determine the frequency of the PWM.
  - Write data in the TBC1 [7:0] to determine the duty cycle of the PWM.
  - Write data in the TBC2 [7:0] to determine the duty cycle fine adjustment method of the PWM.
  - Setting the TBEN 0x40C04 [5] as <1> to enable the counter.
- Generate PWME waveform
  - When the counting value of the TMBR [7:0] is equal to that of the TBC1 [7:0], the PWME will be 0➔1.
  - When the counting value of the TMBR[7:0] is equal to that of the TBC0[7:0] again, the PWME will be 1➔0；
    - ✓ And the overflowing event takes place to set the TMBIF 0x40004[1] as <1>, and reset and restart the incremental counting; at this time, if the TMBIE 0x40004[17] is set as<1>, the interrupt event service will take place.
    - ✓ At this time, the set value of the TBC2[7:0] adjusts the outputs of the PWME to be and N, as shown in the table, where N = TBC1[7:0].

- PWM output control
  - Set the O0PMR 0x40C04 [19] or O1PMR 0x40C04 [23] to determine whether the output waveform of the pins is opposite in phase or not.

    Set the PTPW0E 0x40840 [0] or PTPW1E 0x40840 [1] as <1> to set the pin of the PWM waveform be under output status select appropriate PWM waveform output pin for the PTPW0x40840 [4:2].
- Set the TBEN 0x40C04 [5] as <0> to disable the counter and the PWM output.
- The calculation formula of the frequency and duty cycle of the PWME:

$$\text{PWME Frequency} = \frac{TBCK}{TBC0[7:0]+1}$$

$$\text{PWME Duty Cycle} = \frac{(TBC0[7:0]+1)-TBC1[7:0]-TBC2[7:0]/256}{TBC0[7:0]+1}$$

  - Fine adjustment is effective when the duty cycle is N+1; the formula is as follows: <X>stands for each bit of the TBC2.

$$\text{Fine adjustment} =<0>*128+<1>*64+<2>*32+<3>*16+<4>*8+<5>*4+<6>*2+<7>$$

PWMF mode

The PWMF is a 16-bit PWM. The counting value of the TMBR is compared with the TBC1 and TBC2, and the TBC2 should be larger than TBC1; the TMBR will keep increasing until overflowing.

PWM output status control conditions:

PWM = 1, when TBC1[15:0] =< TMBR[15:0] <= TBC2[15:0];

PWM = 0, when TMBR[15:0] > TBC2[15:0] or TMBR[15:0] <= TBC1[15:0];

PWM=1; the time is: t = tclock × (TBC2 – TBC1);

PWM period:

PWM Period = TMBR[15:0]*TMCD / HS_CK(or LS_CK);

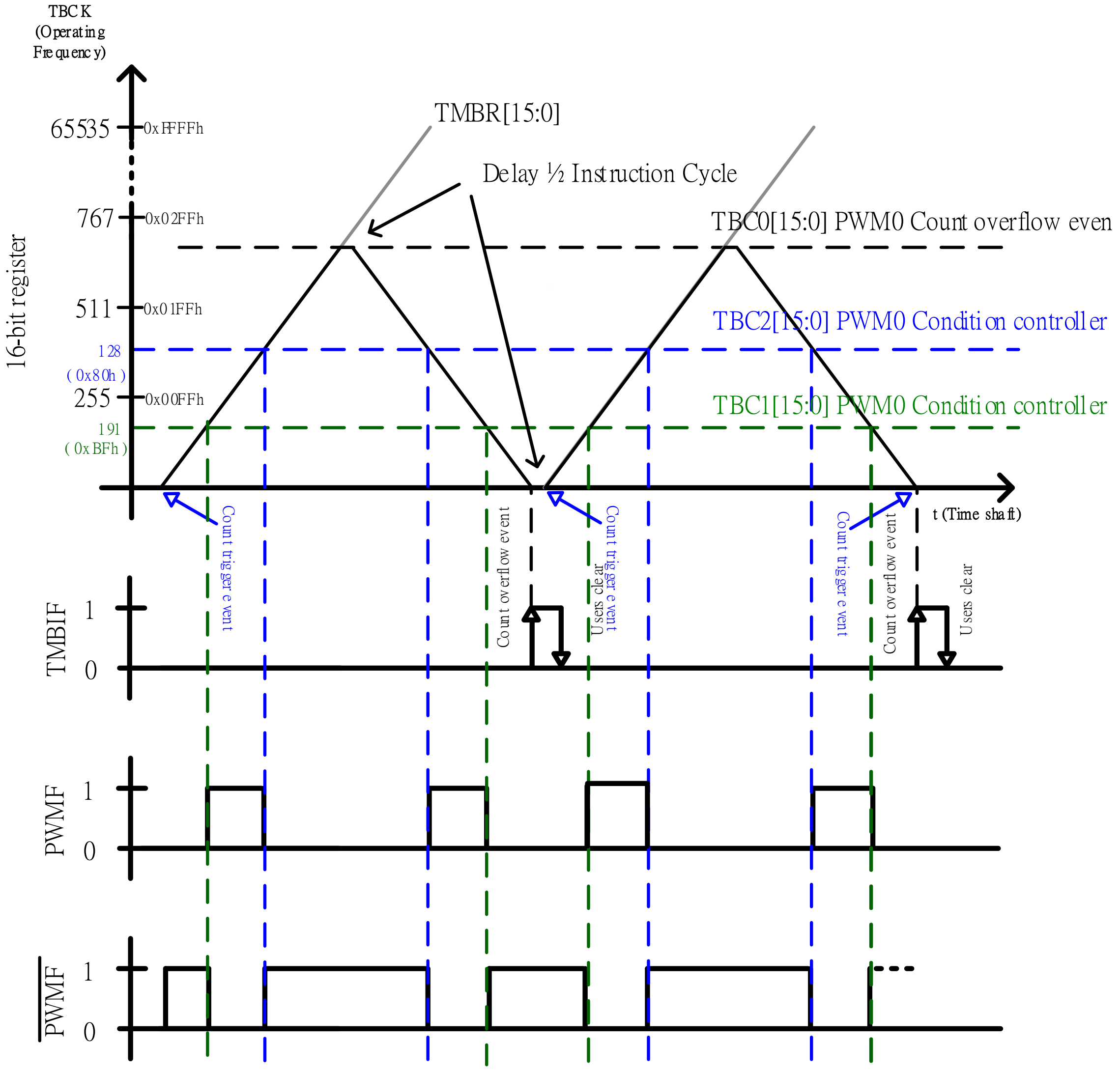


FIG.10-10 Waveform schematic view and counting waveform schematic view of PWM mode F

PWMG mode

The PWMG is a 16-bit PWM mode and the duty cycle of the output waveform is 50%, which is the PFD waveform. The counting value of the TMBR is not compared with the TBC1/TBC2, and the period of the output waveform is related to the TBC0.

PWM period:

PWM Period = TBC0 [15:0]*TMCD / HS_CK (or LS_CK);

PWMG frequency and duty cycle formula:

$$\text{PWMG Frequency} = \frac{\text{TBCK}}{\text{TBC0}[15:0]+1} \div 2$$

$$\text{PWMG Duty Cycle} = 50\%$$

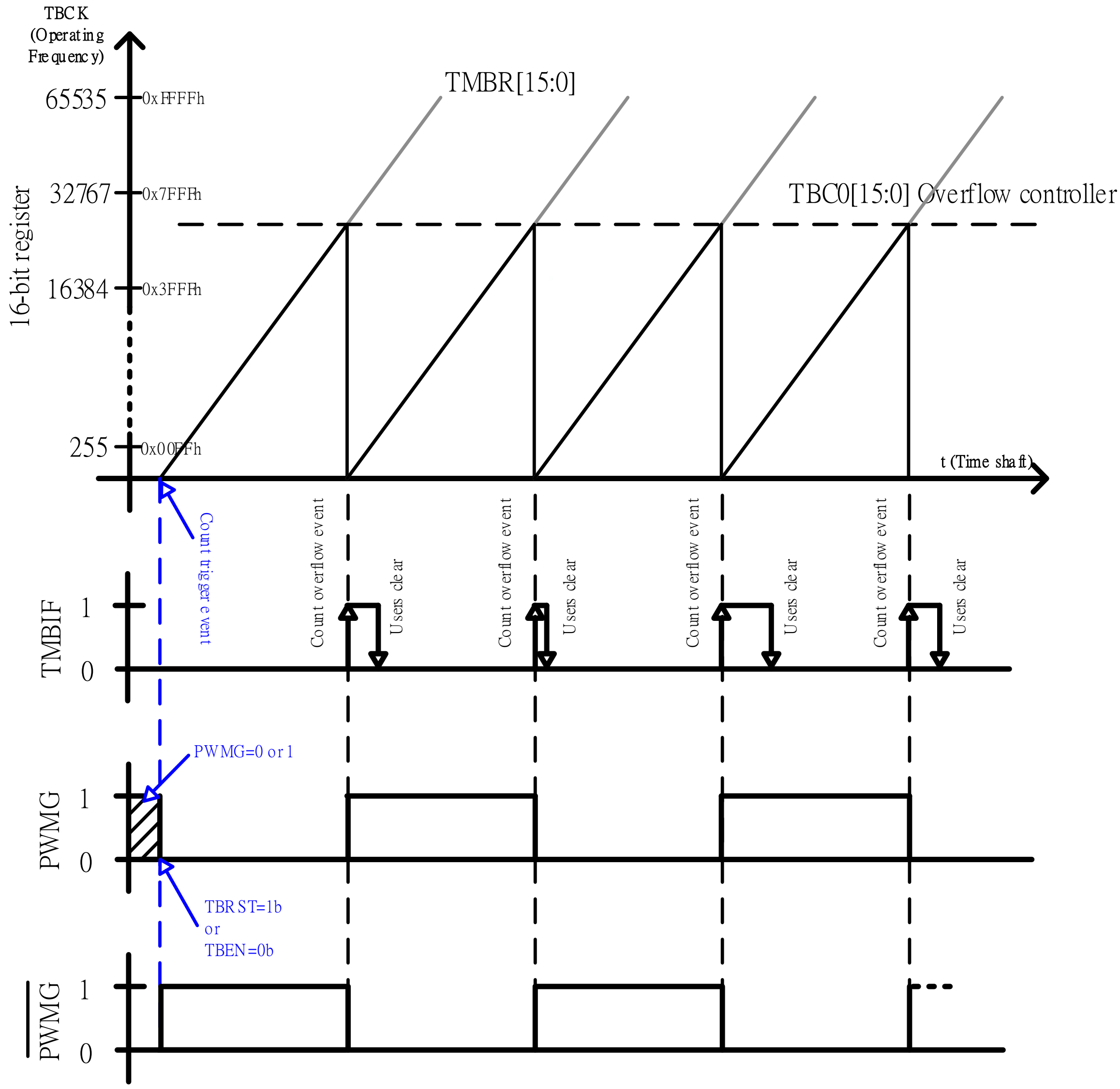


FIG. 10-11 Waveform schematic view and counting waveform schematic view of PWM mode G

10.2. Register address

TMB Register Address	31	24	23	16	15	8	7	0
TMA Base Address + 0x04(0x40C04)	MASK1		REG1		MASK0		REG0	
TMA Base Address + 0x08(0x40C08)	-		REG2		TBCR		TBCR	
TMA Base Address + 0x0C(0x40C0C)	-		-		TBC0		TBC0	
TMA Base Address + 0x10(0x40C10)	TBC2		TBC2		TBC1		TBC1	

- Reserved

10.3. Register function

10.3.1. Timer B register TMBCR0

TMB Base Address + 0X04 (0X40C04)						
Symbol	TMBCR0(TMB Control Register 0)					
Bit	[31:24]	[23]		[22:20]	[19]	[18:16]
Name	MASK	O1PMR		O1MD	O0PMR	O0MD
RW	R0W-0	RW-0				
Bit	[15:08]	[7:6]	[05]	[04]	[03:02]	[01:00]
Name	MASK	-	TBEN	TBRST	TBM	TBEBS
RW	R0W-0	-	RW-0			

Bit	Name	Description	
Bit[23]	O1PMR	PWM1 waveform output phase control	
		0	Inverted output
		1	Normal output
Bit[22-20]	O1MD	PWM1 operating mode selection	
		000	PWMA
		001	PWMB
		010	PWMC
		011	PWMD
		100	PWME
		101	PWMF
		110	PWMG
		111	PWMG
Bit[19]	O0PMR	PWM0 waveform output phase control	
		0	Inverted output
		1	Normal output
Bit[18-16]	O0MD	PWM0 operating mode selection	
		000	PWMA
		001	PWMB
		010	PWMC

		011	PWMD
		100	PWME
		101	PWMF
		110	PWMG
		111	PWMG
Bit[5]	TBEN	Timer B enablement control	
		0	Disable
		1	Enable
Bit[4]	TBRST	Timer B reset	
		0	Normal
		1	Clear the counting register TMBR of the Timer B; it will be automatically set as 0 after finished.
Bit[3~2]	TBM	Timer B counting mode selection	
		00	16-bit up counter; sawtooth-wave type counting method; the counting will increase to the maximum TBC0 on the basis that the step is 1.
		01	16-bit up/down counter; triangle-wave type counting method; the counting will increase to the maximum TBC0 on the basis that the step is 1 and then decrease to 0.
		10	2 independent 8-Bit up counters TMBR[15:8] and TMBR[7:0]; sawtooth-wave type counting method; the two counters will increase to the maximums TBC0[15:8] and TBC0[7:0] at the same time on the basis that the step is 1.
		11	Two 8-Bit up counters TMBR[15:8] and TMBR[7:0], sawtooth-wave type counting method with step being 1; after the counter TMBR[7:0] increases and then overflows, the counter TMBR[15:8] is automatically added by 1 and then the TMBR[7:0] restarts the counting from 0.
Bit[1~0]	TBEBS	Timer B counting-trigger mode selection	
		00	Always enable, continuous counting method
		01	CMPO trigger the multi-function comparator to output high potential
		10	OPOD trigger the OP amplifier to output high potential
		11	CPI1 trigger the output of the Timer C CPI1 to output high potential

10.3.2. Timer B register TMBCR1

TMA Base Address + 0x08 (0x40C08)							
Symbol	TMBCR1(TMB Control Register 1)						
Bit	[31:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	-	PWMF	PWME	PWMD	PWMC	PWMB	PWMA
RW	-	R-X					
Bit	[15:0]						
Name	TMBR						
RW	R-X						

Bit	Name	Description	
Bit[21-16]	PWM Flag	PWM A/B/C/D/F operating mode status flag	
		0	Normal
		1	Enable
Bit[15-0]	TMBR	Timer B 16-bit counting value	

10.3.3. Timer B register TMBCOD

TMA Base Address + 0x0C (0x40C0C)	
Symbol	TMBCOD(TMB Counter overflow condition Register)
Bit	[31:16]
Name	-
RW	-
Bit	[15:0]
Name	TBC0:Timer B Overflow Condition
RW	RW-0xFFFF

Bit	Name	Description
Bit[15-0]	TBC0	Timer B counter overflow threshold value

10.3.4. Timer B register PWMDOD

TMA Base Address + 0x10 (0x40C10)	
Symbol	PWMDOD(PWM counter overflow condition Control Register)
Bit	[31:16]
Name	TBC2: PWM1 duty cycle counter overflow value
RW	RW-FFFFh
Bit	[15:0]
Name	TBC1: PWM0 duty cycle counter overflow value
RW	RW-FFFFh

Bit	Name	Description
Bit[31-16]	TBC2	PWM1 duty cycle counter overflow value
Bit[15-0]	TBC1	PWM0 duty cycle counter overflow value

11. TIMER B2

11.1. General description

Timer B2 is HY16F198B second set Timer B, method of operation is identical with Timer B, using the method detailed reference section CH10 of TimerB.

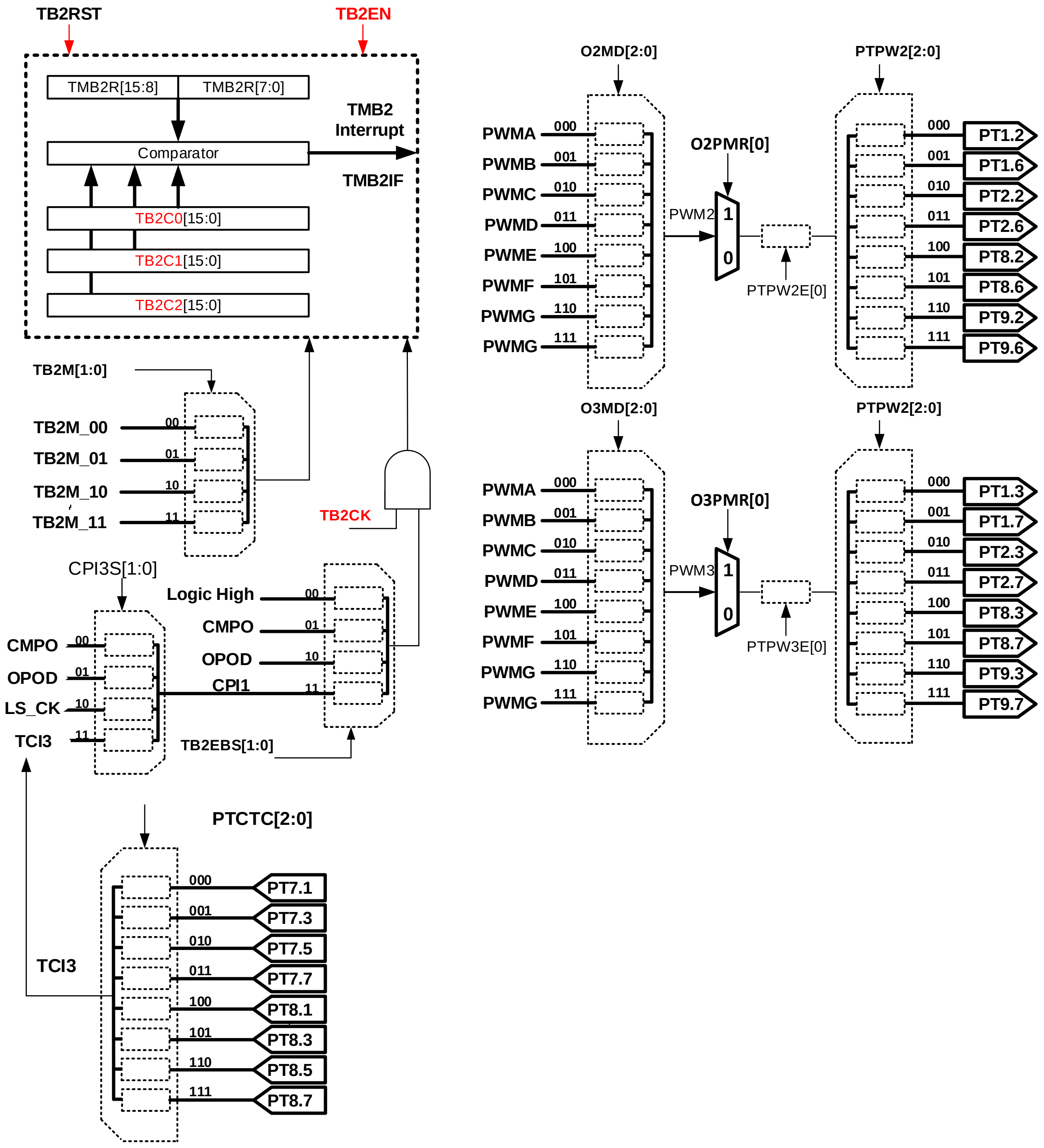


FIG. 11-1 B2 block diagram of the timer counter

11.2. Register address

TMB2 Register Address	31	24	23	16	15	8	7	0
TMA Base Address + 0x24(0x40C24)	MASK1		REG1		MASK0		REG0	
TMA Base Address + 0x28(0x40C28)	-		REG2		TB2CR		TB2CR	
TMA Base Address + 0x2C(0x40C2C)	-		-		TB2C0		TB2C0	
TMA Base Address + 0x30(0x40C30)	TB2C2		TB2C2		TB2C1		TB2C1	

11.3. Register function

11.3.1. Timer B2 register TMB2CR0

TMB2 Base Address + 0X24 (0X40C24)							
Symbol	TMB2CR0(TMB2 Control Register 0)						
Bit	[31:24]	[23]	[22:20]		[19]	[18:16]	
Name	MASK	O3PMR	O3MD		O2PMR	O2MD	
RW	R0W-0	RW-0					
Bit	[15:8]	[7]	[6]	[5]	[4]	[3:2]	[1:0]
Name	MASK	-	-	TB2EN	TB2RST	TB2M	TB2EBS
RW	R0W-0	-	RW-0				

Bit	Name	Description	
Bit[23]	O3PMR	PWM3 waveform output phase control	
		0	Inverted output
		1	Normal output
Bit[22-20]	O3MD	PWM3 operating mode selection	
		0	PWMA
		1	PWMB
		2	PWMC
		3	PWMD
		4	PWME
		5	PWMF
		6	PWMG
		7	PWMG
Bit[19]	O2PMR	PWM2 waveform output phase control	
		0	Inverted output
		1	Normal output
Bit[18-16]	O2MD	PWM2 operating mode selection	
		0	PWMA
		1	PWMB
		2	PWMC
		3	PWMD
		4	PWME

		5	PWMF
		6	PWMG
		7	PWMG
Bit[5]	TB2EN	Timer B2 enablement control	
		0	Disable
		1	Enable
Bit[4]	TB2RST	Timer B2 reset	
		0	Normal
		1	Clear the counting register TB2R of the Timer B2; it will be automatically set as 0 after finished.
Bit[3~2]	TB2M	Timer B2 counting mode selection	
		00	16-bit up counter; sawtooth-wave type counting method; the counting will increase to the maximum TB2C0 on the basis that the step is 1.
		01	16-bit up/down counter; triangle-wave type counting method; the counting will increase to the maximum TB2C0 on the basis that the step is 1 and then decrease to 0.
		10	2 independent 8-Bit up counters TB2R[15:8] and TB2R[7:0]; sawtooth-wave type counting method; the two counters will increase to the maximums TB2C0[15:8] and TB2C0[7:0] at the same time on the basis that the step is 1.
		11	Two 8-Bit up counters TB2R[15:8] and TB2R[7:0], sawtooth-wave type counting method with step being 1; after the counter TB2R[7:0] increases and then overflows, the counter TB2R[15:8] is automatically added by 1 and then the TB2R[7:0] restarts the counting from 0.
Bit[1~0]	TB2EBS	Timer B2 counting-trigger mode selection	
		00	Always enable, continuous counting method
		01	CMPO trigger the multi-function comparator to output high potential
		10	OPOD trigger the OP amplifier to output high potential
		11	CPI3 trigger the output of the Timer C CPI2 to output high potential

11.3.2. Timer B2 register TMB2CR1

TMA Base Address + 0x28 (0x40C28)							
Symbol	TMB2CR1(TMB2 Control Register 1)						
Bit	[31:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	-	PWMF	PWME	PWMD	PWMC	PWMB	PWMA
RW	-	R-X					
Bit	[15:0]						
Name	TMB2R						
RW	R-X						

Bit	Name	Description	
Bit[21-16]	PWM Flag	PWM A/B/C/D/F operating mode status flag	
		0	Normal
		1	Enable
Bit[15-0]	TMB2R	Timer B2 16-bit counting value	

11.3.3. Timer B2 register TMB2COD

TMA Base Address + 0x2C (0x40C2C)	
Symbol	TMB2COD(TMB2 Counter overflow condition Register)
Bit	[31:16]
Name	-
RW	-
Bit	[15:0]
Name	TB2C0:Timer B Overflow Condition
RW	RW-0xFFFF

Bit	Name	Description
Bit[15-0]	TB2C0	Timer B2 counter overflow threshold value

11.3.4. Timer B2 register PWM2DOD

TMA Base Address + 0x30 (0x40C30)	
Symbol	PWM2DOD(PWM counter overflow condition Control Register)
Bit	[31:16]
Name	TB2C2: PWM3 duty cycle counter overflow value
RW	RW-FFFFh
Bit	[15:0]
Name	TB2C1: PWM2 duty cycle counter overflow value
RW	RW-FFFFh

Bit	Name	Description
Bit[31-16]	TB2C2	PWM3 duty cycle counter overflow value
Bit[15-0]	TB2C1	PWM2 duty cycle counter overflow value

11.3.5. Timer B2 register TMB2CR2

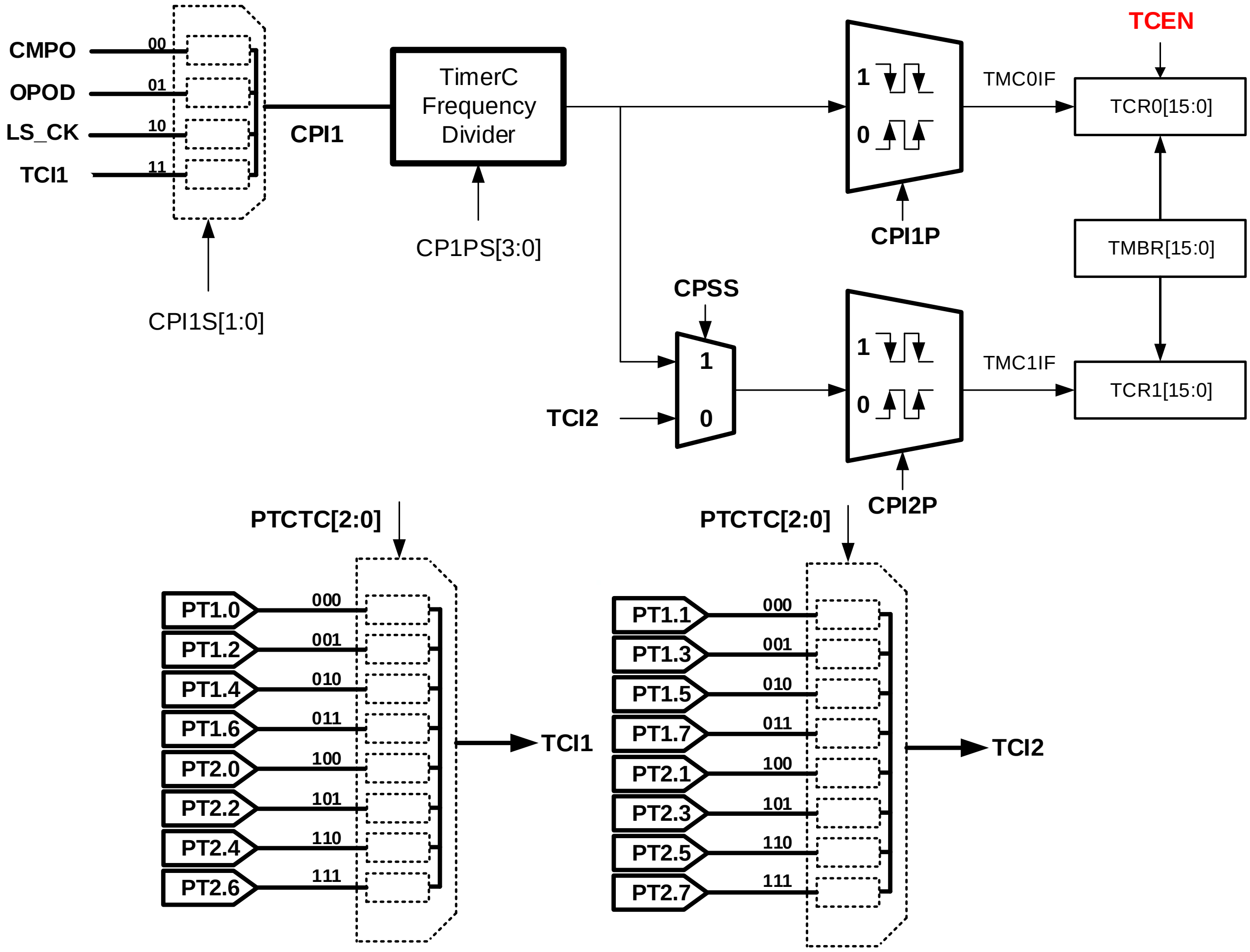
TMB2 Base Address + 0X34 (0X40C34)					
Symbol	TMB2CR1(TMB2 Control Register 2)				
Bit	[31:24]	[23]	[22]	[21:20]	[19:16]
Name	-	CPI3R	RSV	CPI3S	RSV
RW	-	RW-0	-	RW-0	-
Bit	[15:00]				
Name	RSV				
RW	R-0				

Bit	Name	description	
Bit[23]	CPI3R	Timer B2 TCI3 Input Mode Control	
		0	Level Trigger
		1	Rising edge triggered
Bit[21:20]	CPI3S	Timer C Channel3 Trigger source control	
		00	Comparator output
		01	Rail-to-Rail OPA output
		10	LS_CKLow volatility
		11	TCI3 From GPIO

12. TIMER C

12.1. Overall description

The timer C is designed to execute the capture function, which can be used to perform frequency measurement, event counting, interval time measurement, etc. It can generate the interrupt signal when the counter overflow takes place; and it should be used together with the TMB counter register.



CP1PS[3:0]	CPI1 Divider	CP1PS[3:0]	CPI1 Divider
0000	CPI1/1	1000	CPI1/256
0001	CPI1/2	1001	CPI1/512
0010	CPI1/4	1010	CPI1/1024
0011	CPI1/8	1011	CPI1/2048
0100	CPI1/16	1100	CPI1/4096
0101	CPI1/32	1101	CPI1/8192
0110	CPI1/64	1110	CPI1/16384
0111	CPI1/128	1111	CPI1/32768

Fig. 12-1 TMC Function block diagram

TMC clock source selection

The clock source of the TMC is equal to that of the TMB; all of them are generate by make the HS_CK or LS_CK pass the frequency divider to generate the clock source TMBCLK. The capture function of the TMC can be enabled or disabled by setting the control bit TCEN [0].

TMC capture counting value

The capture counting value of the TMC is finished by the counter register's control bit TMBR 0x40C08 [15:0] of the TMB. When Timer B start TMBR began counting, after CPI1P trigger occurs, the value of TMBR placed TCR0 and interruption (TMC0IF), after CPI2P trigger occurs, the value of TMBR placed TCR1 and interruption (TMC1IF).

When the TMC captures the first trigger edge of the input signal, the TMBR will start the counting; when the TMC captures the second trigger edge of the input signal, the TMBR will stop the counting. After the capture event is finished, the hardware will automatically write the value of the TMBR into the register's control bit TCR0 0x40C18 [15:0] or TCR1 0x40C18 [31:16], and generate the interrupt flags TMC0IF and TMC1IF. The user can read the values of the TCR0/TCR1 by program.

Capture comparator 1

The capture comparator 1 has four capture signal input sources, and the input signal source can be set by setting the selector CPI1S 0x40C14 [21:20]; and the input signal should further pass the frequency divider CP1PS 0x40C14 [19:16]; the frequency divider can perform the frequency dividing on the input signal to slow the input signal; in this way, the input signals with high frequency can be measured. The setting of the controller CPI1P 0x40C14 [1] can determine the trigger edge of the capture signal is the rising edge or the falling edge. After the capture event is finished, the interrupt signal can be generated and the interrupt flag TMC0IF 0x40004[2] is set as <1>.

The capture signal input source of the capture comparator 1:

Input signal source symbol	Function description
CMPO	The output status of the comparator
OPOD	The output status of the OP amplifier
LS_CK	Chip low-speed frequency source
TCI1	Input from the IO

The input of the capture comparator 1(When the control bit CPI1S 0x40C14 [21:20] = 11b time):

Serial number	TCI1	TCI2	Serial number	TCI1	TCI2
000	PT1.0	PT1.1	100	PT2.0	PT2.1
001	PT1.2	PT1.3	101	PT2.2	PT2.3
010	PT1.4	PT1.5	110	PT2.4	PT2.5
011	PT1.6	PT1.7	111	PT2.6	PT2.7

Initial Operation of capture comparator 1:

(1)Select the operating clock source TMBCLK of the TMC;

(2)Set the capture signal input source and the input signal source frequency dividing value, which is to set the values of the CPI1S[1:0] and CP1PS [3:0];

(3)Set the capture signal trigger edge, which is to set the value of the CPI1P;

(4)If the TCI1 is selected to be the capture signal input source, it is necessary to set the input IO to select the corresponding IO as the input mode;

(5)If the interrupt function is used, it is necessary to enable TMC0IE 0x40004[18] =<1> and enable the global interrupt function GIE=<1>;

(6)Enable the TMC and enable the TCEN 0x40C14 [0] =<1>.

Capture comparator 2

The capture comparator 2 has 2 capture signal input sources, and the input signal source can be set by setting the selector CPSS 0x40C14 [22]; and the input signal does not have to pass the frequency divider. The setting of the controller CPI2P 0x40C14 [2] can determine the trigger edge of the capture signal is the rising edge or the falling edge. After the capture event is finished, the interrupt signal can be generated and the interrupt flag TMC1IF 0x40004[3].

The capture signal input of the capture comparator 2 is:

Input from IO port;

It is consistent with the input source of the capture comparator 1;

Initial Operation of the capture comparator 2:

(1)Select the operating clock source TMBCLK of the TMC;

(2)Set the capture signal input source, which is to set the values of the CPSS 0x40C14 [22];

(3)Set the capture signal trigger edge, which is to set the value of the CPI2P 0x40C14 [2];

(4)If the TCI2 is selected to be the capture signal input source, it is necessary to set the input IO to select the corresponding IO as the input mode;

(5)If the interrupt function is used, it is necessary to enable TMC1IE 0x40004[19] =<1> and enable the global interrupt function GIE=<1>;

(6)Enable the TMC and enable the TCEN 0x40C14 [0] =<1>.

12.2. Register address

TMC Register Address	31	24	23	16	15	8	7	0
TMA Base Address + 0x14(0x40C14)	MASK1		REG1		MASK0		REG0	
TMA Base Address + 0x18 (0x40C18)	TCR1		TCR1		TCR1		TCR1	

12.3. Register function

12.3.1. Timer C register TMCCR0

TMA Base Address + 0x14 (0x40C14)							
Symbol	TMCCR0(TMC Control Register 0)						
Bit	[31:24]	[23]	[22]	[21:20]	[19:16]		
Name	MASK	CPI1R	CPSS	CPI1S	CP1PS		
RW	R0W-0	RW-0					
Bit	[15:08]	[7:3]			[2]	[1]	[0]
Name	MASK	-			TCPI2P	TCPI1P	TCEN
RW	R0W-0	-			RW-0		

Bit	Name	Description

Bit[23]	CPI1R	Timer B TCI1 input mode control	
		0	Level trigger
		1	Rising edge trigger
Bit[22]	CPSS	Capture 2 (Timer C Channel 2) Capture trigger source selection	
		0	The input of the TCI2 from the IO port
		1	The input source the same with the capture 1
Bit[21~20]	CPI1S	Capture 1 (Timer C Channel 1) Capture trigger source selection	
		00	CMPO comparator output
		01	(Rail-to-Rail OPAMP)OPOD output
		10	Low-frequency clock LS_CK
		11	The input of the TCI1 from the GPIO port
Bit[19~16]	CP1PS	The frequency divider configuration of the input signal of the Capture1	
		0000	CPI1 frequency/1
		0001	CPI1 frequency/2
		0010	CPI1 frequency/4
		0011	CPI1 frequency/8
		0100	CPI1 frequency/16
		0101	CPI1 frequency/32
		0110	CPI1 frequency/64
		0111	CPI1 frequency/128
		1000	CPI1 frequency/256
		1001	CPI1 frequency/512
		1010	CPI1 frequency/1024
		1011	CPI1 frequency/2048
		1100	CPI1 frequency/4096
		1101	CPI1 frequency/8192
		1110	CPI1 frequency/16384
		1111	CPI1 frequency/32768
Bit[02]	CPI2P	Capture2 trigger edge configuration	
		0	Rising edge trigger
		1	Falling edge trigger
Bit[01]	CPI1P	Capture1 trigger edge configuration	
		0	Rising edge trigger
		1	Falling edge trigger
Bit[00]	TCEN	Timer C enablement control	
		0	Disable (but the TCR0 and TCR1 are not cleared)
		1	Enable

12.3.2. Timer C register TMCCR1

TMA Base Address + 0x18 (0x40C18)	
Symbol	TMCCR1(TMC Control Register 1)
Bit	[31:16]
Name	TCR1
RW	R-X
Bit	[15:00]
Name	TCR0
RW	R-X

Bit	Name	Description
Bit[31-16]	TCR1	Capture2 frequency capture counter
Bit[15-00]	TCR0	Capture1 frequency capture counter

13. GPIO PT1 MANAGEMENT

13.1. Overall description

The PT1 has 8 IO pins, which can be used as the common universal IO ports, and can also be reused as the input or output IO ports of the capture comparator, SPI, IIC, UART comparator, PWM and external interrupt modules, etc. Different reuses need different configurations.

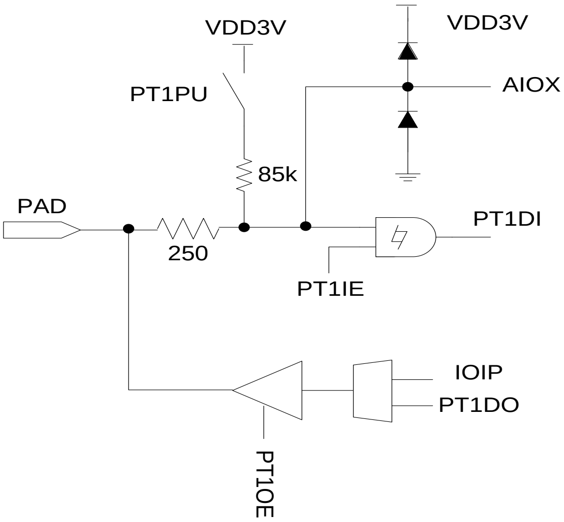


FIG. 13-1 PT1 function block diagram

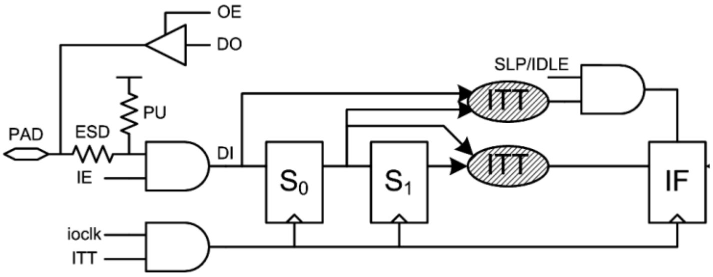


FIG. 13-2 PT1 interruption function block diagram

PT1 has the functions of the input, output, internal pull-up resistors and the external interrupt input port; and different functions should be set by different controllers.

Internal pull-up resistor

The controller PT1PU 0x40800[23:16] can enable or disable the internal pull-up resistor of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the internal 85k pull-up resistor will be enabled; if it is set as <0>, the internal 85k pull-up resistor will be disabled. If the IO port is under the input mode and there is no external pull-up resistor, the internal pull-up resistor should be enabled, especially in low power consumption mode, which can prevent from electric leakage and increase the power consumption. If it serves as the analog signal input port, it is not necessary to enable the internal pull-up resistor.

Output mode

The controller PT1OE 0x40800 [7:0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT1DO 0x40804 [7:0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Input mode

The controller PT1IE 0x40804[23:16] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. The control bit PT1DI 0x40808 can determine whether the input status of the pin of the corresponding IO port is 1 or 0. If the IO is set as the input mode and the chip is not connected to the external pull-up resistor, the internal pull-up resistor should be enabled; the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

External interrupt input

The PT1 has 8 IO pins, and all of them can be reused as external interrupt input pins. The mode should set the IO port to be the input mode and enable the internal pull-up resistor.   It is necessary to set the external interrupt trigger edge by the controller PT1#ITT 0x4080C [23:00] and enable the control bit PT1ITD [0] to enable the interrupt trigger edge. The controller PT1IDF 0x4080C[31:24] can enable the interrupt response function of the corresponding IO pin INTPT1 0x40010; when the external interrupt signal generates, the interrupt flag of the corresponding IO pin is set as 1. When the global interrupt GIE and the

IO external interrupt function are enabled, the chip will stop the current program right away and execute the IO external interrupt program.

13.2. Register address

GPIO Register Address	31	24	23	16	15	8	7	0
GPIO base address + 0x00(0x40800)	MASK1		PT1PU		MASK0		PT1OE	
GPIO base address + 0x04 (0x40804)	MASK3		PT1IE		MASK2		PT1DO	
GPIO base address + 0x08(0x40808)	-		-		-		PT1DI	
GPIO base address + 0x0C (0x4080C)	PT1IDF		PT1#TT		PT1#TT		PT1#TT	

-Reserved

Description: On the list # represents 0 to 7

13.3. Register function

13.3.1. PT1 register PT1CR0

GPIO Base Address + 0x00 (0x40800)									
Symbol	PT1CR0 (PT1 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT1PU7	PT1PU6	PT1PU5	PT1PU4	PT1PU3	PT1PU2	PT1PU1	PT1PU0
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT1OE7	PT1OE6	PT1OE5	PT1OE4	PT1OE3	PT1OE2	PT1OE1	PT1OE0
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23~16]	PT1PU	Port 1 internal pull-up control	
		0	Disable the internal pull-up
		1	Enable the internal pull-up
Bit[07~00]	PT1OE	Port 1 PAD output mode enable control	
		0	Disable the output mode
		1	Enable the output mode

PT1PU: PT1 Pull High Enable
PT1OE: PT1 Output Enable

13.3.2. PT1 register PT1CR1

GPIO Base Address + 0x04 (0x40804)									
Symbol	PT1CR1 (PT1 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT1IE7	PT1IE6	PT1IE5	PT1IE4	PT1IE3	PT1IE2	PT1IE1	PT1IE0
RW	R0W-0	RW-0							

Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT1DO7	PT1DO6	PT1DO5	PT1DO4	PT1DO3	PT1DO2	PT1DO1	PT1DO0
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23~16]	PT1IE	Port 1 PAD input mode control	
		0	Disable the input mode
		1	Enable the input mode
Bit[07~00]	PT1DO	Port 1 PAD output status value	
		0	Output low potential
		1	Output high potential

PT1IE: PT1 Input Enable

PT1DO: PT1 Output Data

13.3.3. PT1 register PT1CR2

GPIO Base Address + 0x08 (0x40808)									
Symbol	PT1CR2(PT1 Control Register 2)								
Bit	[31:16]								
Name	-								
RW	-								
Bit	[15:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	-	PT1DI7	PT1DI6	PT1DI5	PT1DI4	PT1DI3	PT1DI2	PT1DI1	PT1DI0
RW	-	R-0							

Bit	Name	Description	
Bit[7~0]	PT1DI	Port1 PAD input status value	
		0	input low potential
		1	input high potential

PT1DI: PT1 Data Input

13.3.4. PT1 register PT1CR3

GPIO Base Address + 0X0C (0X4080C)						
Symbol	PT1CR3 (PT1 Control Register 3)					
Bit	[31:24]		[23:21]	[21:18]		[17:16]
Name	PT17IDF~ PT10IDF		PT17ITT	PT16ITT		PT15ITT
RW	R-0		RW-0			
Bit	[15]	[14:12]	[11:9]	[8:6]	[5:3]	[2:0]
Name	PT15ITT	PT14ITT	PT13ITT	PT12ITT	PT11ITT	PT10ITT
RW	RW-0					

Bit	Name	Description

Bit[31]	PT17IDF	PT1.7 Interrupt condition flag For example: check this flag before entering the Sleep Mode: When Bit = 1b, can be PT1.7 pin wake up MCU. When Bit = 0b, PT1.7 pin cannot be awakened MCU.	
		When PT17ITT=0	Always 0. Explanation： When PT17ITT set to 000, the Bit [31] = 0b
		When PT17ITT=1	Inverse DI. Explanation： Before entering Sleep Mode, if PT1.7 status is Low, the Bit [31] = 1b
		When PT17ITT=2	Same as DI. Explanation: Before entering Sleep Mode, if PT1.7 status is High, the Bit [31] = 1b
		When PT17ITT=3	Same as S1. Explanation: When PT1.7 Potential change, which produce an interrupt is triggered
		When PT17ITT=4	Same as DI. Explanation: Before entering Sleep Mode, if PT1.7 status is High, the Bit [31] = 1b
		When PT17ITT=5	Inverse DI. Explanation: Before entering Sleep Mode, if PT1.7 status is Low, the Bit [31] = 1b
		When PT17ITT=6	Same as DI. Explanation: Before entering Sleep Mode, if PT1.7 status is High, the Bit [31] = 1b
		When PT17ITT=7	Inverse DI. Explanation: Before entering Sleep Mode, if PT1.7 status is Low, the Bit [31] = 1b

Bit[30~24]	PT1NIDF	PT1.N Interrupt condition flag (N represent 6~0)	
		When PT1NITT=0	Always 0
		When PT1NITT=1	Inverse DI
		When PT1NITT=2	Same as DI
		When PT1NITT=3	Same as S1
		When PT1NITT=4	Same as DI
		When PT1NITT=5	Inverse DI
		When PT1NITT=6	Same as DI
		When PT1NITT=7	Inverse DI

Bit	Name	Description

Bit[23~0]	PT1#ITT	Port 1.# select the interrupt trigger method (# represent 7~0)			
		000	Disable the GPIO interrupt trigger to not reply to the interrupt.		
		001	Rising edge trigger	101	High potential trigger
		010	Falling edge trigger	110	Low potential trigger
		011	Potential change trigger	111	High potential trigger
		100	Low potential trigger		

13.4. Analog to digital multiplexing function Switchover Considerations

PT1.0 ~ PT1.6 addition When the general use of digital functions can also be set as analog multiplexing function, and in doing analog to digital time multiplexed switching function, it should be noted that the relevant register is set to avoid affecting the use the normal function of this pin.

Examples PT1.0 / CH1 multiplexed pins: (PT1.0 ~ PT1.6 the same as those described below)
- Designed to CH1 input; PT1PU0 = PT1OE0 = PT1IE0 = 0b
- Designed to PT1.0 GPIO Output; PT1IE0 = 0b, PT1PU0 = 0b, PT1OE0 = 1b.
- Designed to PT1.0 GPIO input; PT1IE0 = 1b, PT1PU0 = 1b, PT1OE0 = 0b (PT1PU0 = set 1b is input not floating).

14. GPIO PT2 MANAGEMENT

14.1. Overall description

The PT2 has 8 IO pins, and can be used as common universal IO ports or reused as the input or output IO ports of many function modules, such as capture comparator, SPI, IIC, PWM, external crystal oscillator and external interrupt input, etc.　Different reuses need different configurations.

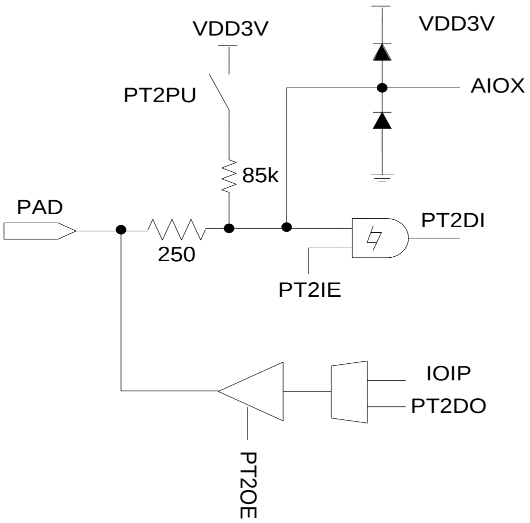


FIG. 14-1 PT2 function block diagram

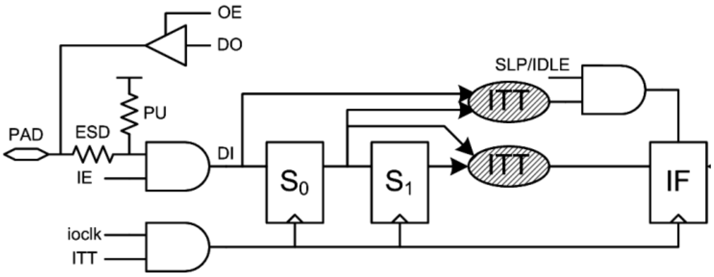


FIG 14-2 PT2 interruption function diagram

The PT2 has the functions of the input, output, internal pull-up resistor and external interrupt input port; and different functions need to be set by different controllers.

Internal pull-up resistor

The controller PT2PU 0x40810[23:16] can enable or disable the internal pull-up resistor of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the internal 85k pull-up resistor will be enabled; if the corresponding bit of the IO port is set as <0>, the internal 85k pull-up resistor will be disabled. If the IO port is under the input mode and there is no external pull-up resistor, the internal pull-up resistor should be enabled, especially in low power consumption mode, which can prevent from electric leakage and increase the power consumption. If it serves as the analog signal input port, it is not necessary to enable the internal pull-up resistor.

PS: When PT2.4~PT2.7 serves as the external crystal oscillator input pins, the internal pull-up resistor cannot be enabled, or the crystal oscillator cannot work normally.

Output mode

The controller PT2OE 0x40810 can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT2DO0x40814 can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

PS: When the PT2.4~PT2.7 serves as the external crystal oscillator input pins, the output mode should be disabled.

Input mode

The controller PT2IE0x40814 [23:16] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether the current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT2DI0x40818. If the IO is set as the input mode and the chip is not connected to the external pull-up resistor, the internal pull-up resistor should be enabled; the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

External interrupt input

The PT2 has 8 IO pins, and all of them can be reused as external interrupt input pins. The mode should set the IO port to be the input mode and enable the internal pull-up resistor. It is necessary to set the external interrupt trigger edge by the controller PT2#ITT

0x4081C [23:00] and enable the control bit PT2IDF 0x4081C [31:24] to enable the interrupt trigger edge. The controller INTPT2 0x40014can enable the interrupt response function of the corresponding IO pin; when the external interrupt signal generates, the interrupt flag of the corresponding IO pin is set as 1. When the global interrupt GIE and the IO external interrupt function are enabled, the chip will stop the current program right away and execute the IO external interrupt program.

14.2. Register address

GPIO Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x10(0x40810)	MASK1		PT2PU		MASK0		PT2OE	
GPIO Base Address + 0x14 (0x40814)	MASK3		PT2IE		MASK2		PT2DO	
GPIO Base Address + 0x18(0x40818)	-		-		-		PT2DI	
GPIO Base Address + 0x1C (0x4081C)	PT2IDF		PT2#ITT		PT2#ITT		PT2#ITT	

-Reserved

Description: above talbel ,the # represent 0~7

14.3. Register function

14.3.1. PT2 register PT2CR0

GPIO Base Address + 0x10 (0x40810)									
Symbol	PT2CR0 (PT2 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT2PU7	PT2PU6	PT2PU5	PT2PU4	PT2PU3	PT2PU2	PT2PU1	PT2PU0
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT2OE7	PT2OE6	PT2OE5	PT2OE4	PT2OE3	PT2OE2	PT2OE1	PT2OE0
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23~16]	PT2PU	Port 2 internal pull-up control	
		0	Disable the internal pull-up
		1	Enable the internal pull-up
Bit[07~00]	PT2OE	Port 2 PAD output mode enable control	
		0	Disable the output mode
		1	Enable the output mode

PT2PU: PT2 Pull High Enable
PT2OE: PT2 Output Enable

14.3.2. PT2 register PT2CR1

GPIO Base Address + 0x14 (0x40814)									
Symbol	PT2CR1 (PT2 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT2IE7	PT2IE6	PT2IE5	PT2IE4	PT2IE3	PT2IE2	PT2IE1	PT2IE0
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT2DO7	PT2DO6	PT2DO5	PT2DO4	PT2DO3	PT2DO2	PT2DO1	PT2DO0
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23~16]	PT2IE	Port 2 PAD input mode enable control	
		0	Disable the input mode
		1	Enable the input mode
Bit[7~0]	PT2DO	Port 2 PAD output status value	
		0	Output low potential
		1	Output high potential

PT2IE: PT2 Input Enable
PT2DO: PT2 Output Data

14.3.3. PT2 register PT2CR2

GPIO Base Address + 0x18 (0x40818)									
Symbol	PT2CR2 (PT2 Control Register 2)								
Bit	[31:16]								
Name	-								
RW	-								
Bit	[15:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	-	PT2DI[7]	PT2DI[6]	PT2DI[5]	PT2DI[4]	PT2DI[3]	PT2DI[2]	PT2DI[1]	PT2DI[0]
RW	-	R-0							

Bit	Name	Description	
Bit[7~0]	PT2DI	Port2 PAD input status value	
		0	Input low potential
		1	Input high potential

PT2DI: PT2 Data Input

14.3.4. PT2 register PT2CR3

<table>
<tr><td colspan="13">GPIO Base Address + 0x1C (0x4081C)</td></tr>
<tr><td>Symbol</td><td colspan="12">PT2CR3 (PT2 Control Register 3)</td></tr>
<tr><td>Bit</td><td colspan="3">[31:24]</td><td colspan="3">[23:21]</td><td colspan="3">[21:18]</td><td colspan="3">[17:16]</td></tr>
<tr><td>Name</td><td colspan="3">PT2IDF</td><td colspan="3">PT27ITT</td><td colspan="3">PT26ITT</td><td colspan="3">PT25ITT</td></tr>
<tr><td>RW</td><td colspan="12">RW-0</td></tr>
<tr><td>Bit</td><td colspan="2">[15]</td><td colspan="2">[14:12]</td><td colspan="2">[11:9]</td><td colspan="2">[8:6]</td><td colspan="2">[5:3]</td><td colspan="2">[2:0]</td></tr>
<tr><td>Name</td><td colspan="2">PT25ITT</td><td colspan="2">PT24ITT</td><td colspan="2">PT23ITT</td><td colspan="2">PT22ITT</td><td colspan="2">PT21ITT</td><td colspan="2">PT20ITT</td></tr>
<tr><td>RW</td><td colspan="12">RW-0</td></tr>
</table>

Bit	Name	Description	
Bit[31]	PT27IDF	PT2.7 Interrupt condition flag For example: check this flag before entering the Sleep Mode: When Bit = 1b, can be PT2.7 pin wake up MCU. When Bit = 0b, PT2.7 pin cannot be awakened MCU.	
		When PT27ITT=0	Always 0. Explanation : When PT27ITT set to 000, the Bit [31] = 0b
		When PT27ITT=1	Inverse DI. Explanation : Before entering Sleep Mode, if PT2.7 status is Low, the Bit [31] = 1b
		When PT27ITT=2	Same as DI. Explanation: Before entering Sleep Mode, if PT2.7 status is High, the Bit [31] = 1b
		When PT27ITT=3	Same as S1. Explanation: When PT2.7 Potential change, which produce an interrupt is triggered
		When PT27ITT=4	Same as DI. Explanation: Before entering Sleep Mode, if PT2.7 status is High, the Bit [31] = 1b
		When PT27ITT=5	Inverse DI. Explanation: Before entering Sleep Mode, if PT2.7 status is Low, the Bit [31] = 1b
		When PT27ITT=6	Same as DI. Explanation: Before entering Sleep Mode, if PT2.7 status is High, the Bit [31] = 1b
		When PT27ITT=7	Inverse DI. Explanation: Before entering Sleep Mode, if PT2.7 status is Low, the Bit [31] = 1b

Bit[30~24]	PT2NIDF	PT2.N Interrupt condition flag (N represent 6~0)	
		When PT2NITT=0	Always 0
		When PT2NITT=1	Inverse DI
		When PT2NITT=2	Same as DI

		When PT2NITT=3	Same as S1
		When PT2NITT=4	Same as DI
		When PT2NITT=5	Inverse DI
		When PT2NITT=6	Same as DI
		When PT2NITT=7	Inverse DI

Bit	Name	Description			
Bit[23~0]	PT2#ITT	Port 2.# select the interrupt trigger method (# represent 7~0)			
		000	Disable the GPIO interrupt trigger to not reply to the interrupt.		
		001	Rising edge trigger	101	High potential trigger
		010	Falling edge trigger	110	Low potential trigger
		011	Potential change trigger	111	High potential trigger
		100	Low potential trigger		

14.4. Analog to digital multiplexing function Switchover Considerations

PT2.0~PT2.3 not only could be used as normal digital function, but can beset and analog function too. While doing the switch of analog function, should notice the setting of register to avoid the normal function using.
EX: PT2.0/CL5 Multiplexed pin: (PT2.0 ~ PT2.3 the same as those under the Notes)

Design as CL5 input; PT2PU0=PT2OE0=PT2IE0=0b
Design asPT2.0 GPIO Output; PT2IE0=0b, PT2PU0=0b, PT2OE0=1b.
Design asPT2.0 GPIO input; PT2IE0=1b, PT2PU0=1b, PT2OE0=0b. (PT2PU0=1b setting is input not floating)

15. GPIO PT3 MANAGEMENT

15.1. Overall description

The PT3 has 8 IO pins, and can be used as common universal IO ports or reused as the input or output IO ports of many function modules, such as OP amplifier, 8-bit resistance ladder and ADC converters, etc. Different reuses need different configurations.

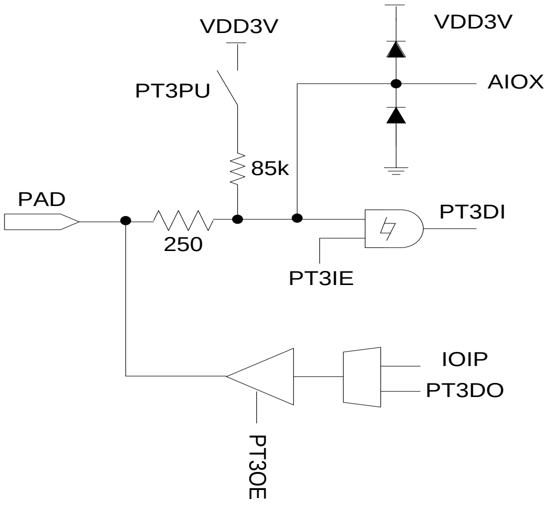


FIG. 15-1 PT3 function block diagram

The PT3 has the functions of the input, output and internal pull-up resistors; and different functions need to be set by different controllers.

Internal pull-up resistor

The controller PT3PU 0x40820[23:16] can enable or disable the internal pull-up resistor of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the internal 85k pull-up resistor will be enabled; if it is set as <0>, the internal 85k pull-up resistor will be disabled. If the IO port is under the input mode and there is no external pull-up resistor, the internal pull-up resistor should be enabled, especially in low power consumption mode, which can prevent from electric leakage and increase the power consumption. If it serves as the analog signal input port, it is not necessary to enable the internal pull-up resistor.

Output mode

The controller PT3OE0x40820 [7:0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT3DO0x40824 [7:0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Input mode

The controller PT3IE0x40824 [23:16] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether the current input status of the corresponding IO pin is 1 or 0 can be read via the controller PT3DI0x40828 [7:0]. If the IO is set as the input mode and the chip is not connected to the external pull-up resistor, the internal pull-up resistor should be enabled; the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

15.2. Register address

<table>
<tr><td>GPIO Register Address</td><td>31</td><td>24</td><td>23</td><td>16</td><td>15</td><td>8</td><td>7</td><td>0</td></tr>
<tr><td>GPIO Base Address + 0x20(0x40820)</td><td colspan="2">MASK1</td><td colspan="2">PT3PU</td><td colspan="2">MASK0</td><td colspan="2">PT3OE</td></tr>
<tr><td>GPIO Base Address + 0x24 (0x40824)</td><td colspan="2">MASK3</td><td colspan="2">PT3IE</td><td colspan="2">MASK2</td><td colspan="2">PT3DO</td></tr>
<tr><td>GPIO Base Address + 0x28(0x40828)</td><td colspan="2">-</td><td colspan="2">-</td><td colspan="2">REG4</td><td colspan="2">PT3DI</td></tr>
</table>

-Reserved

15.3. Register function

15.3.1. PT3 register PT3CR0

GPIO Base Address + 0x20 (0x40820)									
Symbol	PT3CR0 (PT3 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT3PU7	PT3PU6	PT3PU5	PT3PU4	PT3PU3	PT3PU2	PT3PU1	PT3PU0
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT3OE7	PT3OE6	PT3OE5	PT3OE4	PT3OE3	PT3OE2	PT3OE1	PT3OE0
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23~16]	PT3PU	Port 3 internal pull-up enable control	
		0	Disable the internal pull-up
		1	Enable the internal pull-up
Bit[7~0]	PT3OE	Port 3 PAD output mode enable control	
		0	Disable the output mode
		1	Enable the output mode

PT3PU: PT3 Pull High Enable
PT3OE: PT3 Output Enable

15.3.2. PT3 register PT3CR1

GPIO Base Address + 0x24 (0x40824)									
Symbol	PT3CR1 (PT3 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	PT3IE7	PT3IE6	PT3IE5	PT3IE4	PT3IE3	PT3IE2	PT3IE1	PT3IE0
RW	R0W-0	RW-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	PT3DO7	PT3DO6	PT3DO5	PT3DO4	PT3DO3	PT3DO2	PT3DO1	PT3DO0
RW	R0W-0	RW-0							

PT3IE: PT3 Input Enable

PT3DO: PT3 Output Data

Bit	Name	Description	
Bit[23~16]	PT3IE	Port 3 PAD input mode enable control	
		0	Disable the input mode
		1	Enable the input mode
Bit[7~0]	PT3DO	Port 3 PAD output status value	
		0	Output low potential
		1	Output high potential

15.3.3. PT3 register PT3CR2

GPIO Base Address + 0x28 (0x40828)									
Symbol	PT3CR2 (PT3 Control Register 2)								
Bit	[31:24]	[23:18]						[17]	[16]
Name	MASK	-						PT3AO	-
RW	R0W-0	-						R-X	-
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	-	PT3DI[7]	PT3DI[6]	PT3DI[5]	PT3DI[4]	PT3DI[3]	PT3DI[2]	PT3DI[1]	PT3DI[0]
RW	-	R-0							

Bit	Name	Description	
Bit[17]	PT3A0	DAO outputs to the PT3.1 enable control	
		0	Disable
		1	Enable
Bit[7~0]	PT3DI	Port3 PAD input status value	
		0	input low potential
		1	input high potential

15.4. Analog to digital multiplexing function Switchover Considerations

PT3.0~PT3.7 not only could be used as normal digital function, but can beset and analog function too. While doing the switch of analog function, should notice the setting of register to avoid the normal function using.

PT3.7/OPO Multiplexed pins:
Design as OPO output; control resister OPOE 0x41900[1]=1b,
PT3PU7=PT3OE7=PT3IE7=0b
Design as PT3.7 GPIO input; control resister OPOE 0x41900[1]=0b, PT3IE7=1b(Even not set as input, still must be forcibly set)
Design as PT3.7 GPIO output; control resister OPOE 0x41900[1]=0b, PT3IE7=1b(Even not set as input, still must be forcibly set), PT3OE7=1b

PT3.6/REFO Multiplexed pins:
Design as REFO output; control resister ENRFO 0x40400[1]=1b,
PT3PU6=PT3OE6=PT3IE6=0b
Design as REFO input; control resister ENRFO 0x40400[1]=0b,
PT3PU6=PT3OE6=PT3IE6=0b
Design as PT3.6 GPIO input; control resister ENRFO 0x40400[1]=0b, PT3IE6=1b(Even not set as input, still must be forcibly set)
Design as PT3.6 GPIO output; control resister ENRFO 0x40400[1]=0b, PT3IE6=1b(Even not set as input, still must be forcibly set), PT3OE6=1b

Other GPIO use method: (PT3.5~PT3.0 all as follows description)
EX: PT3.5/AIO7 Multiplexed pins:
Design as AIO7 input; PT3PU5=PT3OE5=PT3IE5=0b
Design as PT3.5 GPIO Output; PT3IE5=0b, PT3PU5=0b, PT3OE5=1b.
Design as PT3.5 GPIO input; PT3IE5=1b, PT3PU5=1b, PT3OE5=0b. (PT3PU5=1b set as input not floating)

16. GPIO PT6 MANAGEMENT

16.1. Overall description

The PT6 has 8 IO pins, which can be used as the common universal IO ports, and can also be reused as the LCD function output port. Different reuses need different configurations.

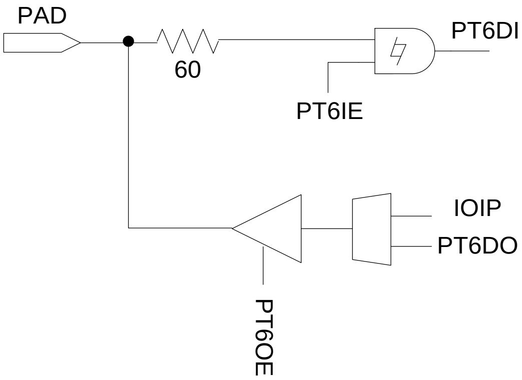


FIG. 16-1 PT6 function block diagram

The PT6 has input and output functions; and different functions should be set by different controllers.

Output mode

The controller PT6xOE [0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT6xDO [0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Description: Above x represent 0~7, reflect for PT6.0~PT6.7

Input mode

The controller PT6xIE [0] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether the current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT6xDI [0]. If the IO is set as the input mode and the chip should be connected to the external pull-up resistor; and the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

Description: Above x represent 0~7, reflect for PT6.0~PT6.7

LCD mode

The controller SEGx [5:0] determines the output data of the LCD SEGMENT. If the LCD is under the 1/6 duty mode, the SEGx[5:0] can determine the data content of the 1/6 duty data content; if the LCD is under the 1/5 duty mode, the SEGx[4:0] can determine the 1/5 duty data content; if the LCD is under the 1/4 duty mode, the SEGx[3:0] can determine the 1/4 duty data content; if the LCD is under the 1/3 duty mode, the SEGx[2:0] can determine the 1/3 duty data content.

Description: Above x represent 2~9, reflect for SEG2~SEG9

16.2. Register address

GPIO Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x50(0x40850)	MASK1		PT61CFG		MASK0		PT60CFG	
GPIO Base Address + 0x54(0x40854)	MASK3		PT63CFG		MASK2		PT62CFG	
GPIO Base Address + 0x58(0x40858)	MASK5		PT65CFG		MASK4		PT64CFG	
GPIO Base Address + 0x5C(0x4085C)	MASK7		PT67CFG		MASK6		PT66CFG	

LCD Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x50(0x40850)	MASK1		SEG3		MASK0		SEG2	
GPIO Base Address + 0x54(0x40854)	MASK3		SEG5		MASK2		SEG4	
GPIO Base Address + 0x58(0x40858)	MASK5		SEG7		MASK4		SEG6	
GPIO Base Address + 0x5C(0x4085C)	MASK7		SEG9		MASK6		SEG8	

LCD Register Address 0x41B04 can determine the setting is the GPIO Mode or the LCD Mode.

16.3. Register function

16.3.1. PT6.0/PT6.1 register

When GPIO Mode.

GPIO Base Address + 0x50 (0x40850)									
Symbol	PT60CFG/ PT61CFG (PT6 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT61OE	PT61IE	PT61DO	PT61DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT60OE	PT60IE	PT60DO	PT60DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT61OE	PT6.1 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT61IE	PT6.1 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT61DO	PT6.1 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT61DI	PT6.1 Input Data	
		0	Input Low

		1	Input High
Bit[3]	PT60OE	PT6.0 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT60IE	PT6.0 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT60DO	PT6.0 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT60DI	PT6.0 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x50 (0x40850)									
Symbol	SEG2/SEG3 (PT6 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG3 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG2 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 3 Data	LCD Segment 3 Data
		Segment Data
Bit[5~0]	SEG 2 Data	LCD Segment 2 Data
		Segment Data

16.3.2. PT6.2/PT6.3 register

When GPIO Mode.

GPIO Base Address + 0x54 (0x40854)									
Symbol	PT62CFG/ PT63CFG (PT6 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT63OE	PT63IE	PT63DO	PT63DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT62OE	PT62IE	PT62DO	PT62DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT63OE	PT6.3 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT63IE	PT6.3 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT63DO	PT6.3 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT63DI	PT6.3 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT62OE	PT6.2 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT62IE	PT6.2 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT62DO	PT6.2 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT62DI	PT6.2 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x54 (0x40854)									
Symbol	SEG4/SEG5 (PT6 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG5 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG4 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 5 Data	LCD Segment 5 Data
		Segment Data
Bit[5~0]	SEG 4 Data	LCD Segment 4 Data
		Segment Data

16.3.3. PT6.4/PT6.5 register

When GPIO Mode.

GPIO Base Address + 0x58 (0x40858)									
Symbol	PT64CFG/ PT65CFG (PT6 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT65OE	PT65IE	PT65DO	PT65DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT64OE	PT64IE	PT64DO	PT64DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT65OE	PT6.5 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT65IE	PT6.5 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT65DO	PT6.5 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT65DI	PT6.5 Input Data	
		0	Input Low

		1	Input High
Bit[3]	PT64OE	PT6.4 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT64IE	PT6.4 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT64DO	PT6.4 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT64DI	PT6.4 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x58 (0x40858)									
Symbol	SEG6/SEG7 (PT6 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG7 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG6 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 7 Data	LCD Segment 7 Data
		Segment Data
Bit[5~0]	SEG 6 Data	LCD Segment 6 Data
		Segment Data

16.3.4. PT6.6/PT6.7 register

When GPIO Mode.

GPIO Base Address + 0x5C (0x4085C)									
Symbol	PT66CFG/ PT67CFG (PT6 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT67OE	PT67IE	PT67DO	PT67DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT66OE	PT66IE	PT66DO	PT66DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT67OE	PT6.7 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT67IE	PT6.7 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT67DO	PT6.7 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT67DI	PT6.7 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT66OE	PT6.6 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT66IE	PT6.6 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT66DO	PT6.6 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT66DI	PT6.6 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x5C (0x4085C)									
Symbol	SEG8/SEG9 (PT6 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG9 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG8 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 9 Data	LCD Segment 9 Data
		Segment Data
Bit[5~0]	SEG 8 Data	LCD Segment 8 Data
		Segment Data

17. GPIO PT7 MANAGEMENT

17.1. Overall description

The PT7 has 8 IO pins, which can be used as the common universal IO ports, and can also be reused as the LCD function output port, capture comparators. Different reuses need different configurations.

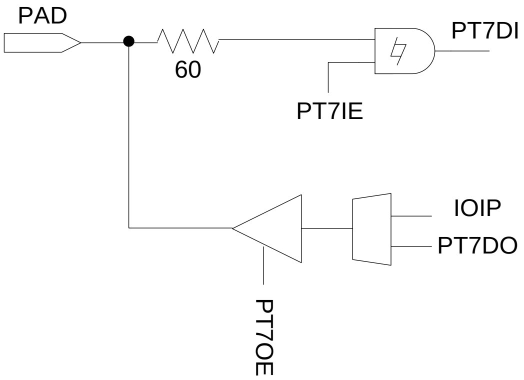


FIG. 17-1 PT7 function block diagram

The PT7 has input and output functions; and different functions should be set by different controllers.

Output mode

The controller PT7xOE [0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT7xDO [0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Description: Above x represent 0~7, reflect for PT7.0~PT7.7

Input mode

The controller PT7xIE [0] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether the current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT7xDI [0]. If the IO is set as the input mode and the chip should be connected to the external pull-up resistor; and the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

Description: Above x represent 0~7, reflect for PT7.0~PT7.7

LCD mode

The controller SEGx[5:0] determines the output data of the LCD SEGMENT. If the LCD is under the 1/6 duty mode, the SEGx[5:0] can determine the data content of the 1/6 duty data content; if the LCD is under the 1/5 duty mode, the SEGx[4:0] can determine the 1/5 duty data content; if the LCD is under the 1/4 duty mode, the SEGx[3:0] can determine the 1/4 duty data content; if the LCD is under the 1/3 duty mode, the SEGx[2:0] can determine the 1/3 duty data content.

Description: Above x represent 10~17, reflect for SEG10~SEG17

17.2. Register address

GPIO Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x60(0x40860)	MASK1		PT71CFG		MASK0		PT70CFG	
GPIO Base Address + 0x64(0x40864)	MASK3		PT73CFG		MASK2		PT72CFG	
GPIO Base Address + 0x68(0x40868)	MASK5		PT75CFG		MASK4		PT74CFG	
GPIO Base Address + 0x6C(0x4086C)	MASK7		PT77CFG		MASK6		PT76CFG	

LCD Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x60(0x40860)	MASK1		SEG11		MASK0		SEG10	
GPIO Base Address + 0x64(0x40864)	MASK3		SEG13		MASK2		SEG12	
GPIO Base Address + 0x68(0x40868)	MASK5		SEG15		MASK4		SEG14	
GPIO Base Address + 0x6C(0x4086C)	MASK7		SEG17		MASK6		SEG16	

LCD Register Address 0x41B04 can determine the setting is the GPIO Mode or the LCD Mode.

17.3. Register function

17.3.1. PT7.0/PT7.1 register

When GPIO Mode.

GPIO Base Address + 0x60 (0x40860)									
Symbol	PT70CFG/ PT71CFG (PT7 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT71OE	PT71IE	PT71DO	PT71DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT70OE	PT70IE	PT70DO	PT70DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT71OE	PT7.1 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT71IE	PT7.1 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT71DO	PT7.1 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT71DI	PT7.1 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT70OE	PT7.0 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT70IE	PT7.0 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT70DO	PT7.0 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT70DI	PT7.0 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x60 (0x40860)									
Symbol	SEG10/SEG11 (PT7 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG11 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG10 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 11 Data	LCD Segment 11 Data
		Segment Data
Bit[5~0]	SEG 10 Data	LCD Segment 10 Data
		Segment Data

17.3.2. PT7.2/PT7.3 register

When GPIO Mode.

GPIO Base Address + 0x64 (0x40864)									
Symbol	PT72CFG/ PT73CFG (PT7 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT73OE	PT73IE	PT73DO	PT73DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT72OE	PT72IE	PT72DO	PT72DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT73OE	PT7.3 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT73IE	PT7.3 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT73DO	PT7.3 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT73DI	PT7.3 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT72OE	PT7.2 Output Enable	
		0	Disable

		1	Enable
Bit[2]	PT72IE	PT7.2 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT72DO	PT7.2 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT72DI	PT7.2 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x64 (0x40864)									
Symbol	SEG12/SEG13 (PT7 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG13 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG12 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 13 Data	LCD Segment 13 Data
		Segment Data
Bit[5~0]	SEG 12 Data	LCD Segment 12 Data
		Segment Data

17.3.3. PT7.4/PT7.5 register

When GPIO Mode.

GPIO Base Address + 0x68 (0x40868)									
Symbol	PT74CFG/ PT75CFG (PT7 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT75OE	PT75IE	PT75DO	PT75DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT74OE	PT74IE	PT74DO	PT74DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT75OE	PT7.5 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT75IE	PT7.5 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT75DO	PT7.5 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT75DI	PT7.5 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT74OE	PT7.4 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT74IE	PT7.4 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT74DO	PT7.4 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT74DI	PT7.4 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x68 (0x40868)									
Symbol	SEG14/SEG15 (PT7 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG15 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG14 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 15 Data	LCD Segment 15 Data
		Segment Data
Bit[5~0]	SEG 14 Data	LCD Segment 14 Data
		Segment Data

17.3.4. PT7.6/PT7.7 register

When GPIO Mode.

GPIO Base Address + 0x6C (0x4086C)									
Symbol	PT76CFG/ PT77CFG (PT7 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT77OE	PT77IE	PT77DO	PT77DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT76OE	PT76IE	PT76DO	PT76DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT77OE	PT7.7 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT77IE	PT7.7 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT77DO	PT7.7 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT77DI	PT7.7 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT76OE	PT7.6 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT76IE	PT7.6 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT76DO	PT7.6 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT76DI	PT7.6 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x6C (0x4086C)									
Symbol	SEG16/SEG17 (PT7 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG17 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG16 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 17 Data	LCD Segment 17 Data
		Segment Data
Bit[5~0]	SEG 16 Data	LCD Segment 16 Data
		Segment Data

18. GPIO PT8 MANAGEMENT

18.1. Overall description

The PT8 has 8 IO pins, which can be used as the common universal IO ports, and can also be reused as the LCD function output port, capture comparator, SPI, UART, and PWM. Different reuses need different configurations.

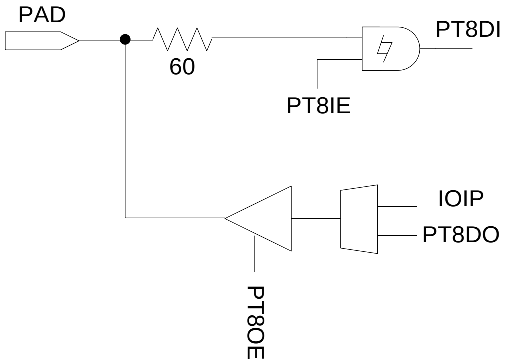


FIG. 18-1 PT8 function block diagram

The PT8 has input and output functions; and different functions should be set by different controllers.

Output mode

The controller PT8xOE [0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT8xDO [0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Description: Above x represent 0~7, reflect for PT8.0~PT8.7

Input mode

The controller PT8xIE [0] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether the current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT8xDI [0]. If the IO is set as the input mode and the chip should be connected to the external pull-up resistor; and the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

Description: Above x represent 0~7, reflect for PT8.0~PT8.7

LCD mode

The controller SEGx[5:0] determines the output data of the LCD SEGMENT. If the LCD is under the 1/6 duty mode, the SEGx[5:0] can determine the data content of the 1/6 duty data content; if the LCD is under the 1/5 duty mode, the SEGx[4:0] can determine the 1/5 duty data content; if the LCD is under the 1/4 duty mode, the SEGx[3:0] can determine the 1/4 duty data content; if the LCD is under the 1/3 duty mode, the SEGx[2:0] can determine the 1/3 duty data content.

Description: Above x represent 18~25, reflect for SEG18~SEG25

18.2. Register address

GPIO Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x70(0x40870)	MASK1		PT81CFG		MASK0		PT80CFG	
GPIO Base Address + 0x74(0x40874)	MASK3		PT83CFG		MASK2		PT82CFG	
GPIO Base Address + 0x78(0x40878)	MASK5		PT85CFG		MASK4		PT84CFG	
GPIO Base Address + 0x7C(0x4087C)	MASK7		PT87CFG		MASK6		PT86CFG	

LCD Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x70(0x40870)	MASK1		SEG19		MASK0		SEG18	
GPIO Base Address + 0x74(0x40874)	MASK3		SEG21		MASK2		SEG20	
GPIO Base Address + 0x78(0x40878)	MASK5		SEG23		MASK4		SEG22	
GPIO Base Address + 0x7C(0x4087C)	MASK7		SEG25		MASK6		SEG24	

LCD Register Address 0x41B04 can determine the setting is the GPIO Mode or the LCD Mode.

18.3. Register function

18.3.1. PT8.0/PT8.1 register

When GPIO Mode.

GPIO Base Address + 0x70 (0x40870)									
Symbol	PT80CFG/ PT81CFG (PT8 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT81OE	PT81IE	PT81DO	PT81DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT80OE	PT80IE	PT80DO	PT80DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT81OE	PT8.1 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT81IE	PT8.1 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT81DO	PT8.1 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT81DI	PT8.1 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT80OE	PT8.0 Output Enable	

		0	Disable
		1	Enable
Bit[2]	PT80IE	PT8.0 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT80DO	PT8.0 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT80DI	PT8.0 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x70 (0x40870)									
Symbol	SEG18/SEG19 (PT8 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG19 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG18 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 19 Data	LCD Segment 19 Data
		Segment Data
Bit[5~0]	SEG 18 Data	LCD Segment 18 Data
		Segment Data

18.3.2. PT8.2/PT8.3 register

When GPIO Mode.

GPIO Base Address + 0x74 (0x40874)									
Symbol	PT82CFG/ PT83CFG (PT8 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT83OE	PT83IE	PT83DO	PT83DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT82OE	PT82IE	PT82DO	PT82DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT83OE	PT8.3 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT83IE	PT8.3 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT83DO	PT8.3 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT83DI	PT8.3 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT82OE	PT8.2 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT82IE	PT8.2 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT82DO	PT8.2 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT82DI	PT8.2 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x74 (0x40874)									
Symbol	SEG20/SEG21 (PT8 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG21 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG20 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 21 Data	LCD Segment 21 Data
		Segment Data
Bit[5~0]	SEG 20 Data	LCD Segment 20 Data
		Segment Data

18.3.3. PT8.4/PT8.5 register

When GPIO Mode.

GPIO Base Address + 0x78 (0x40878)									
Symbol	PT84CFG/ PT85CFG (PT8 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT85OE	PT85IE	PT85DO	PT85DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT84OE	PT84IE	PT84DO	PT84DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT85OE	PT8.5 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT85IE	PT8.5 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT85DO	PT8.5 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT85DI	PT8.5 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT84OE	PT8.4 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT84IE	PT8.4 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT84DO	PT8.4 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT84DI	PT8.4 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x78 (0x40878)									
Symbol	SEG22/SEG23 (PT8 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG23 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG22 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 23 Data	LCD Segment 23 Data
		Segment Data
Bit[5~0]	SEG 22 Data	LCD Segment 22 Data
		Segment Data

18.3.4. PT8.6/PT8.7 register

When GPIO Mode.

GPIO Base Address + 0x7C (0x4087C)									
Symbol	PT86CFG/ PT87CFG (PT8 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT87OE	PT87IE	PT87DO	PT87DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT86OE	PT86IE	PT86DO	PT86DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT87OE	PT8.7 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT87IE	PT8.7 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT87DO	PT8.7 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT87DI	PT8.7 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT86OE	PT8.6 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT86IE	PT8.6 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT86DO	PT8.6 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT86DI	PT8.6 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x7C (0x4087C)									
Symbol	SEG24/SEG25 (PT8 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG25 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG24 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 25 Data	LCD Segment 25 Data
		Segment Data
Bit[5~0]	SEG 24 Data	LCD Segment 24 Data
		Segment Data

19. GPIO PT9 MANAGEMENT

19.1. Overall description

The PT9 has 8 IO pins, which can be used as the common universal IO ports, and can also be reused as the LCD function output, SPI, UART, and PWM. Different reuses need different configurations.

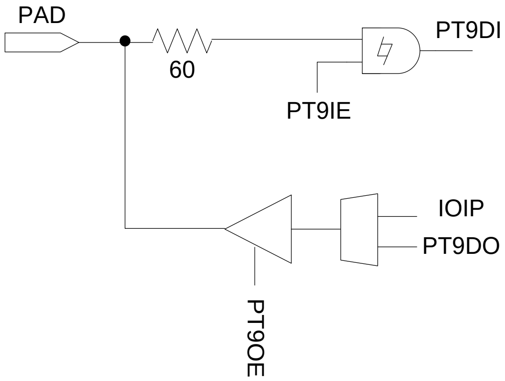


FIG. 19-1 PT9 function block diagram

The PT9 has input and output functions; and different functions should be set by different controllers.

Output mode

The controller PT9xOE [0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT9xDO [0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Description: Above x represent 0~7, reflect for PT9.0~PT9.7

Input mode

The controller PT9xIE [0] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT9xDI [0]. If the IO is set as the input mode and the chip should be connected to the external pull-up resistor; and the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

Description: Above x represent 0~7, reflect for PT9.0~PT9.7

LCD mode

The controller SEGx[5:0] determines the output data of the LCD SEGMENT. If the LCD is under the 1/6 duty mode, the SEGx[5:0] can determine the data content of the 1/6 duty data content; if the LCD is under the 1/5 duty mode, the SEGx[4:0] can determine the 1/5 duty data content; if the LCD is under the 1/4 duty mode, the SEGx[3:0] can determine the 1/4 duty data content; if the LCD is under the 1/3 duty mode, the SEGx[2:0] can determine the 1/3 duty data content.

Description: Above x represent 26~33, reflect for SEG26~SEG33

19.2. Register address

GPIO Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x80(0x40880)	MASK1		PT91CFG		MASK0		PT90CFG	
GPIO Base Address + 0x84(0x40884)	MASK3		PT93CFG		MASK2		PT92CFG	
GPIO Base Address + 0x88(0x40888)	MASK5		PT95CFG		MASK4		PT94CFG	
GPIO Base Address + 0x8C(0x4088C)	MASK7		PT97CFG		MASK6		PT96CFG	

LCD Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x80(0x40880)	MASK1		SEG27		MASK0		SEG26	
GPIO Base Address + 0x84(0x40884)	MASK3		SEG29		MASK2		SEG28	
GPIO Base Address + 0x88(0x40888)	MASK5		SEG31		MASK4		SEG30	
GPIO Base Address + 0x8C(0x4088C)	MASK7		SEG33		MASK6		SEG32	

LCD Register Address 0x41B04 can determine the setting is the GPIO Mode or the LCD Mode.

19.3. Register function

19.3.1. PT9.0/PT9.1 register

When GPIO Mode.

GPIO Base Address + 0x80 (0x40880)									
Symbol	PT90CFG/ PT91CFG (PT9 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT91OE	PT91IE	PT91DO	PT91DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT90OE	PT90IE	PT90DO	PT90DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT91OE	PT9.1 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT91IE	PT9.1 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT91DO	PT9.1 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT91DI	PT9.1 Input Data	
		0	Input Low
		1	Input High

Bit[3]	PT90OE	PT9.0 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT90IE	PT9.0 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT90DO	PT9.0 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT90DI	PT9.0 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

<table>
<tr><td colspan="10">GPIO Base Address + 0x80 (0x40880)</td></tr>
<tr><td>Symbol</td><td colspan="9">SEG26/SEG27 (PT9 Control Register 0)</td></tr>
<tr><td>Bit</td><td>[31:24]</td><td>[23]</td><td>[22]</td><td>[21]</td><td>[20]</td><td>[19]</td><td>[18]</td><td>[17]</td><td>[16]</td></tr>
<tr><td>Name</td><td>MASK</td><td>-</td><td>-</td><td colspan="6">SEG27 Data</td></tr>
<tr><td>RW</td><td>R0W-0</td><td colspan="7">RW-0</td><td>RW-1</td></tr>
<tr><td>Bit</td><td>[15:08]</td><td>[7]</td><td>[6]</td><td>[5]</td><td>[4]</td><td>[3]</td><td>[2]</td><td>[1]</td><td>[0]</td></tr>
<tr><td>Name</td><td>MASK</td><td>-</td><td>-</td><td colspan="6">SEG26 Data</td></tr>
<tr><td>RW</td><td>R0W-0</td><td colspan="7">RW-0</td><td>RW-1</td></tr>
</table>

Bit	Name	Description
Bit[21~16]	SEG 27 Data	LCD Segment 27 Data
		Segment Data
Bit[5~0]	SEG 26 Data	LCD Segment 26 Data
		Segment Data

19.3.2. PT9.2/PT9.3 register

When GPIO Mode.

GPIO Base Address + 0x84 (0x40884)									
Symbol	PT92CFG/ PT93CFG (PT9 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT93OE	PT93IE	PT93DO	PT93DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT92OE	PT92IE	PT92DO	PT92DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT93OE	PT9.3 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT93IE	PT9.3 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT93DO	PT9.3 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT93DI	PT9.3 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT92OE	PT9.2 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT92IE	PT9.2 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT92DO	PT9.2 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT92DI	PT9.2 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x84 (0x40884)									
Symbol	SEG28/SEG29 (PT9 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG29 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG28 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 29 Data	LCD Segment 29 Data
		Segment Data
Bit[5~0]	SEG 28 Data	LCD Segment 28 Data
		Segment Data

19.3.3. PT9.4/PT9.5 register

When GPIO Mode.

GPIO Base Address + 0x88 (0x40888)									
Symbol	PT94CFG/ PT95CFG (PT9 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT95OE	PT95IE	PT95DO	PT95DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT94OE	PT94IE	PT94DO	PT94DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT95OE	PT9.5 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT95IE	PT9.5 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT95DO	PT9.5 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT95DI	PT9.5 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT94OE	PT9.4 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT94IE	PT9.4 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT94DO	PT9.4 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT94DI	PT9.4 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x88 (0x40888)									
Symbol	SEG30/SEG31 (PT9 Control Register 2)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG31 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG30 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 31 Data	LCD Segment 31 Data
		Segment Data
Bit[5~0]	SEG 30 Data	LCD Segment 30 Data
		Segment Data

19.3.4. PT9.6/PT9.7 register

When GPIO Mode.

GPIO Base Address + 0x8C (0x4088C)									
Symbol	PT96CFG/ PT97CFG (PT9 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT97OE	PT97IE	PT97DO	PT97DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT96OE	PT96IE	PT96DO	PT96DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT97OE	PT9.7 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT97IE	PT9.7 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT97DO	PT9.7 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT97DI	PT9.7 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT96OE	PT9.6 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT96IE	PT9.6 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT96DO	PT9.6 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT96DI	PT9.6 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x8C (0x4088C)									
Symbol	SEG32/SEG33 (PT9 Control Register 3)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG33 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG32 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 33 Data	LCD Segment 33 Data
		Segment Data
Bit[5~0]	SEG 32 Data	LCD Segment 32 Data
		Segment Data

20. GPIO PT10 MANAGEMENT

20.1. Overall description

The PT10 has 4 IO pins, which can be used as the common universal IO ports, and can also be reused as the LCD function output port. Different reuses need different configurations.

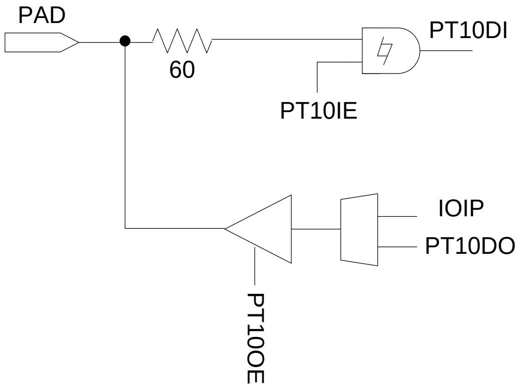


FIG. 20-1 PT10 function block diagram

The PT10 has input and output functions; and different functions should be set by different controllers.

Output mode

The controller PT10xOE [0] can enable or disable the output mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit of the IO port is set as <1>, the output mode of the corresponding IO port will be enabled; if it is set as <0>, the output mode of the corresponding IO port will be disabled. The control bit PT10xDO [0] can determine whether the output status of the pin of the corresponding IO port is 1 or 0. Under the low-power mode, if the IO should enable the output mode, the output status can be set according to the peripheral circuit to decrease the power consumption of the chip. During the mode, the internal pull-up resistor of the IO cannot be enabled, and the input mode and the output mode cannot be enabled at the same time; therefore, when the output mode is enabled, the input mode of the IO port should be disabled.

Description: Above x represent 0~1, reflect for PT10.0~PT10.1

Notice: PT10.2~PT10.3 only works in LCD Mode

Input mode

The controller PT10xIE [0] can enable or disable the input mode of each IO port, and each bit is corresponding to each IO port pin. If the corresponding bit controller is set as <1>, the input mode of the corresponding IO port will be enabled; if it is set as <0>, the input mode of the corresponding IO port will be disabled. Whether current input mode of the corresponding IO pin is 0 or 1 can be read via the controller PT10xDI [0]. If the IO is set as the input mode and the chip should be connected to the external pull-up resistor; and the IO pin is not allowed to be floating in order to prevent from the electric leakage of the chip. Especially in the low-power mode, it is suggested the IO pin should be set as the input mode. If it serves as the analog signal input port, it is not necessary to set the corresponding IO pin as the input mode. The output mode of the IO pin should be disabled before its input mode is enabled.

Description: Above x represent 0~1, reflect for PT10.0~PT10.1

Notice: PT10.2~PT10.3 only works in LCD Mode

LCD mode

The controller SEGx[5:0] determines the output data of the LCD SEGMENT. If the LCD is under the 1/6 duty mode, the SEGx[5:0] can determine the data content of the 1/6 duty data content; if the LCD is under the 1/5 duty mode, the SEGx[4:0] can determine the 1/5 duty data content; if the LCD is under the 1/4 duty mode, the SEGx[3:0] can determine the 1/4 duty data content; if the LCD is under the 1/3 duty mode, the SEGx[2:0] can determine the 1/3 duty data content; but the SEG0 only supports the 1/3 duty and 1/4duty; and the SEG1 supports the 1/3 duty, 1/4 duty and 1/5duty.

Description: Above x represent 0~1 and 34~35, reflect for SEG0~SEG1and SEG34~SEG35.

20.2. Register address

GPIO Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x90(0x40890)	MASK1		PT101CFG		MASK0		PT100CFG	
GPIO Base Address + 0x94(0x40894)	MASK3		PT103CFG		MASK2		PT102CFG	

LCD Mode Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x90(0x40890)	MASK1		SEG35		MASK0		SEG34	
GPIO Base Address + 0x94(0x40894)	MASK3		SEG1		MASK2		SEG0	

LCD Register Address 0x41B04 can determine the setting is the GPIO Mode or the LCD Mode.

20.3. Register function

20.3.1. PT10.0/PT10.1 register

When GPIO Mode.

GPIO Base Address + 0x90 (0x40890)									
Symbol	PT100CFG/ PT101CFG (PT10 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT101OE	PT101IE	PT101DO	PT101DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT100OE	PT100IE	PT100DO	PT100DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT101OE	PT10.1 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT101IE	PT10.1 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT101DO	PT10.1 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT101DI	PT10.1 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT100OE	PT10.0 Output Enable	
		0	Disable
		1	Enable

Bit[2]	PT100IE	PT10.0 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT100DO	PT10.0 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT100DI	PT10.0 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x90 (0x40890)									
Symbol	SEG34/SEG35 (PT10 Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	SEG35 Data					
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	SEG34 Data					
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[21~16]	SEG 35 Data	LCD Segment 35 Data
		Segment Data
Bit[5~0]	SEG 34 Data	LCD Segment 34 Data
		Segment Data

20.3.2. PT10.2/PT10.3 register SEG1/SEG0

When GPIO Mode only.

GPIO Base Address + 0x94 (0x40894)									
Symbol	PT102CFG/ PT103CFG (PT10 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	-	PT103OE	PT103IE	PT103DO	PT103DI
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-	-	-	PT102OE	PT102IE	PT102DO	PT102DI
RW	R0W-0	RW-0							RW-1

Bit	Name	Description	
Bit[19]	PT103OE	PT10.3 Output Enable	
		0	Disable
		1	Enable
Bit[18]	PT103IE	PT10.3 Input Enable	
		0	Disable
		1	Enable
Bit[17]	PT103DO	PT10.3 Output Data	
		0	Output Low
		1	Output High
Bit[16]	PT103DI	PT10.3 Input Data	
		0	Input Low
		1	Input High
Bit[3]	PT102OE	PT10.2 Output Enable	
		0	Disable
		1	Enable
Bit[2]	PT102IE	PT10.2 Input Enable	
		0	Disable
		1	Enable
Bit[1]	PT102DO	PT10.2 Output Data	
		0	Output Low
		1	Output High
Bit[0]	PT102DI	PT10.2 Input Data	
		0	Input Low
		1	Input High

When LCD Mode

GPIO Base Address + 0x94 (0x40894)									
Symbol	SEG0/SEG1 (PT10 Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-		SEG1 Data				
RW	R0W-0	RW-0							RW-1
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	-			SEG0 Data			
RW	R0W-0	RW-0							RW-1

Bit	Name	Description
Bit[20~16]	SEG 1 Data	LCD Segment 1 Data (support 1/3 or 1/4 or 1/5 duty mode)
		Segment Data
Bit[3~0]	SEG 0 Data	LCD Segment 0 Data (support 1/3 or 1/4 duty mode)
		Segment Data

21. GPIO MANAGEMENT

21.1. Overall description

The chip has multiple universal IO ports, and most of them have reuse functions; their reuse functions should be controlled by the registers. The chapter will introduce the control of the reuse functions of the IO ports.

Each IO port has multiple reuse functions but only one of these functions can work at a time; thus, if it is not necessary to use the reuse functions, please remember to disable them for other functions. However, some reuse functions can work together, such as PT1/PT2; when they are set as the external input ports, they can be set as the input ports of the IIC, SPI and UART, etc. Please note that the above situation is on the condition that they are set as the input ports; in this way, the external interrupt functions generated by communication can be realized by the communication signals and the external interrupt functions.

In general, the reuse functions should be used on a group basis and only one group can work at a time. If the SPI function is used, the CS_1, CK_1, MISO_1 and MOSI_1 are the first group, and the CS_2, CK_2, MISO_2 and MOSI_2 are the second group, and so on.   When using the SPI, the user can select the first group or the second group according to the actual requirements but only one group can work at a time. When the user needs use the communication SPI, I2C, UART and the like, the user can set the SPI to use the first group (CS_1, CK_1, MISO_1 and MOSI_1), set the I2C to use the third group (SCL_3 and SDA_3), and set the UART to use the third group (Tx_3 and Rx_3). In this way, the desired effect can be achieved by the different configuration of the pins.

The following table lists the reuse functions of all IO pins and their priority level; 0 stands for the highest level and 6 stands for the lowest level.

Function	INT	Timer C Capture	Special Function	SPI	I2C	UART	Analog IP	Analog	Timer B/B2 PWM
Output Priority	I/P	I/P	0	1	2	3	4	5	6
PT1.0	INT1.0	TCI1_1		CS_1	SCL_1	Tx_1		CH1	PWM0_1
PT1.1	INT1.1	TCI2_1		CK_1	SDA_1	Rx_1		CH2	PWM1_1
PT1.2	INT1.2	TCI1_2		MISO_1	SCL_2	Tx2_1		CH3	PWM2_1
PT1.3	INT1.3	TCI2_2		MOSI_1	SDA_2	Rx2_1		CL1	PWM3_1
PT1.4	INT1.4	TCI1_3		CS_2	SCL_3	Tx_2		CL2	PWM0_2
PT1.5	INT1.5	TCI2_3		CK_2	SDA_3	Rx_2		CL3	PWM1_2
PT1.6	INT1.6	TCI1_4		MISO_2	SCL_4	Tx2_2		CL4	PWM2_2
PT1.7	INT1.7	TCI2_4		MOSI_2	SDA_4	Rx2_2	CMPO		PWM3_2
PT2.0	INT2.0	TCI1_5		CS_3	SCL_5	Tx_3		CL5	PWM0_3
PT2.1	INT2.1	TCI2_5		CK_3	SDA_5	Rx_3		CL6	PWM1_3

PT2.2	INT2.2	TCI1_6		MISO_3	SCL_6	Tx2_3		CL7	PWM2_3
PT2.3	INT2.3	TCI2_6		MOSI_3	SDA_6	Rx2_3		CL8	PWM3_3
PT2.4	INT2.4	TCI1_7	LS_XOUT	CS_4	SCL_7	Tx_4			PWM0_4
PT2.5	INT2.5	TCI2_7	LS_XIN	CK_4	SDA_7	Rx_4			PWM1_4
PT2.6	INT2.6	TCI1_8	HS_XIN	MISO_4	SCL_8	Tx2_4			PWM2_4
PT2.7	INT2.7	TCI2_8	HS_XOUT	MOSI_4	SDA_8	Rx2_4			PWM3_4
PT3.0							OPO1	AIO8	
PT3.1							OPO2	DAO	
PT3.2								AIO4	
PT3.3								AIO5	
PT3.4								AIO6	
PT3.5								AIO7	
PT3.6								REFO	
PT3.7								OPO	
AIO0								AIO0	
AIO1								AIO1	
AIO2								AIO2	
AIO3								AIO3	
COM0			COM 0						
COM1			COM 1						
COM2			COM 2						
COM3			COM 3						
PT10.2			COM 4/SEG 0						
PT10.3			COM 5/SEG 1						
PT6.0			SEG 2						
PT6.1			SEG 3						
PT6.2			SEG 4						
PT6.3			SEG 5						
PT6.4			SEG 6						
PT6.5			SEG 7						
PT6.6			SEG 8						
PT6.7			SEG 9						
PT7.0			SEG 10						
PT7.1		TCI3_1	SEG 11						
PT7.2			SEG 12						
PT7.3		TCI3_2	SEG 13						
PT7.4			SEG 14						

PT7.5		TCI3_3	SEG 15						
PT7.6			SEG 16						
PT7.7		TCI3_4	SEG 17						
PT8.0			SEG 18	CS_5		Tx_5			PWM0_5
PT8.1		TCI3_5	SEG 19	CK_5		Rx_5			PWM1_5
PT8.2			SEG 20	MISO_5		Tx2_5			PWM2_5
PT8.3		TCI3_6	SEG 21	MOSI_5		Rx2_5			PWM3_5
PT8.4			SEG 22	CS_6		Tx_6			PWM0_6
PT8.5		TCI3_7	SEG 23	CK_6		Rx_6			PWM1_6
PT8.6			SEG 24	MISO_6		Tx2_6			PWM2_6
PT8.7		TCI3_8	SEG 25	MOSI_6		Rx2_6			PWM3_6
PT9.0			SEG 26	CS_7		Tx_7			PWM0_7
PT9.1			SEG 27	CK_7		Rx_7			PWM1_7
PT9.2			SEG 28	MISO_7		Tx2_7			PWM2_7
PT9.3			SEG 29	MOSI_7		Rx2_7			PWM3_7
PT9.4			SEG 30	CS_8		Tx_8			PWM0_8
PT9.5			SEG 31	CK_8		Rx_8			PWM1_8
PT9.6			SEG 32	MISO_8		Tx2_8			PWM2_8
PT9.7			SEG 33	MOSI_8		Rx2_8			PWM3_8
PT10.0			SEG 34						
PT10.1			SEG 35						

Table 21-1 IO pin reuse functions and priority levels

21.2. Register address

GPIO Register Address	31	24	23	16	15	8	7	0
GPIO Base Address + 0x40(0x40840)	MASK1		GPIOMCR1		MASK0		GPIOMCR0	
GPIO Base Address + 0x44(0x40844)	MASK3		GPIOMCR3		MASK2		GPIOMCR2	
GPIO Base Address + 0x48(0x40848)	MASK5		GPIOMCR5		MASK4		GPIOMCR4	
GPIO Base Address + 0x4C(0x4084C)	MASK7		GPIOMCR7		MASK6		GPIOMCR6	

21.3. Register function

21.3.1. GPIO reuse function control register GPIOMCR0/ GPIOMCR1

GPIO Base Address + 0x40 (0x40840)								
Symbol	GPIOMCR0/ GPIOMCR1 (GPIO multiplex Control Register 0)							
Bit	[31:24]	[23:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	-	-	PTCOPS	PTCOPE	-	PTCCPE
RW	R0W-0	-	-	-	RW-0	RW-0	-	RW-0
Bit	[15:08]	[7:5]		[4:2]			[1]	[0]
Name	MASK	PTCTC[2:0]		PTPW[2:0]			PTPW1E	PTPW0E

RW	R0W-0	RW-0

Bit	Name	Description			
Bit[19]	PTCOPS	Rail-to-Rail OPAMP digital signal output port selection			
		0	Port 3.0=Rail-to-Rail Output		
		1	Port 3.1=Rail-to-Rail Output		
Bit[18]	PTCOPE	Rail-to-Rail OPAMP digital signal output port enable control			
		0	Disable, no output		
		1	Enable, output to the set target port		
Bit[16]	PTCCPE	Comparator output end IO port enable control			
		0	Disable, only used as a common IO, and no signal outputted		
		1	Enable		
Bit[7~5]	PTCTC	Capture comparator signal input end IO selection			
		000	Port 1.0 =TCI1	Port 1.1 =TCI2	Port 7.1 =TCI3
		001	Port 1.2 =TCI1	Port 1.3 =TCI2	Port 7.3 =TCI3
		010	Port 1.4 =TCI1	Port 1.5 =TCI2	Port 7.5 =TCI3
		011	Port 1.6 =TCI1	Port 1.7 =TCI2	Port 7.7 =TCI3
		100	Port 2.0 =TCI1	Port 2.1 =TCI2	Port 8.1 =TCI3
		101	Port 2.2 =TCI1	Port 2.3 =TCI2	Port 8.3 =TCI3
		110	Port 2.4 =TCI1	Port 2.5 =TCI2	Port 8.5 =TCI3
		111	Port 2.6 =TCI1	Port 2.7 =TCI2	Port 8.7 =TCI3
Bit[4~2]	PTPW	PWM output end IO port selection			
		000	Port 1.0 =PWM0	Port 1.1 =PWM1	
		001	Port 1.4 =PWM0	Port 1.5 =PWM1	
		010	Port 2.0 =PWM0	Port 2.1 =PWM1	
		011	Port 2.4 =PWM0	Port 2.5 =PWM1	
		100	Port 8.0 =PWM0	Port 8.1 =PWM1	
		101	Port 8.4 =PWM0	Port 8.5 =PWM1	
		110	Port 9.0 =PWM0	Port 9.1 =PWM1	
		111	Port 9.4 =PWM0	Port 9.5 =PWM1	
Bit[1]	PTPW1E	PWM 1 IO port input enable control			
		0	Disable (no output from the IO port)		
		1	Enable (the output port is set by the PTPW)		
Bit[0]	PTPW0E	PWM 0 IO port input enable control			
		0	Disable (no output for the IO port)		
		1	Enable (the output port is set by the PTPW)		

21.3.2. GPIO reuse function control register GPIOMCR2/ GPIOMCR3

GPIO Base Address + 0x44 (0x40844)					
Symbol	GPIOMCR2/ GPIOMCR3 (GPIO Multiplex Control Register 1)				
Bit	[31:24]	[23:20]		[19:17]	[16]
Name	MASK	-		I2CPTS	I2CPTEn
RW	R0W-0	-		RW-0	
Bit	[15:08]	[7:5]	[4]	[3:1]	[0]
Name	MASK	PTCSP	PTSPE	PTUR	PTURE
RW	R0W-0	RW-0			

Bit	Name	Description		
Bit[19~17]	I2CPTS	I2C communication IO port selection		
		000	Port 1.0 =SCL	Port 1.1 =SDA
		001	Port 1.2 =SCL	Port 1.3 =SDA
		010	Port 1.4 =SCL	Port 1.5 =SDA
		011	Port 1.6 =SCL	Port 1.7 =SDA
		100	Port 2.0 =SCL	Port 2.1 =SDA
		101	Port 2.2 =SCL	Port 2.3 =SDA
		110	Port 2.4 =SCL	Port 2.5 =SDA
		111	Port 2.6 =SCL	Port 2.7 =SDA
Bit[16]	I2CPTEn	I2C communication IO port reuse function enable control		
		0	Disable (no signal outputted)	
		1	Enable (the IO port is reused as the I2C communication port and the IO port is set by the I2CPTS)	
Bit[7~5]	PTCSP	SPI communication IO port selection		
		000	Port1.0 =CS, Port1.1 =CK, Port1.2 = MISO, Port1.3 =MOSI	
		001	Port1.4 =CS, Port1.5 =CK, Port1.6 = MISO, Port1.7 =MOSI	
		010	Port2.0 =CS, Port2.1 =CK, Port2.2 = MISO, Port2.3 =MOSI	
		011	Port2.4 =CS, Port2.5 =CK, Port2.6 = MISO, Port2.7 =MOSI	
		100	Port8.0 =CS, Port8.1 =CK, Port8.2 = MISO, Port8.3 =MOSI	
		101	Port8.4 =CS, Port8.5 =CK, Port8.6 = MISO, Port8.7 =MOSI	
		110	Port9.0 =CS, Port9.1 =CK, Port9.2 = MISO, Port9.3 =MOSI	
		111	Port9.4 =CS, Port9.5 =CK, Port9.6 = MISO, Port9.7 =MOSI	
Bit[4]	PTSPE	SPI communication IO reuse function enable control		
		0	Disable (only used as a common IO port)	
		1	Enable (The IO port is reused as the SPI communication port, and the communication IO port is set by the PTCSP.)	
Bit[3~1]	PTUR	UART communication IO port selection		
		000	Port 1.0 =TX	Port 1.1 =RX

		001	Port 1.4 =TX	Port 1.5 =RX
		010	Port 2.0 =TX	Port 2.1 =RX
		011	Port 2.4 =TX	Port 2.5 =RX
		100	Port 8.0 =TX	Port 8.1 =RX
		101	Port 8.4 =TX	Port 8.5 =RX
		110	Port 9.0 =TX	Port 9.0 =RX
		111	Port 9.4 =TX	Port 9.5 =RX
Bit[0]	PTURE	EUART communication IO reuse function enable control		
		0	Disable (only used as a common IO port)	
		1	Enable (The IO port is reused as the EUART communication port and the communication IO port is set by the PTUR.)	

MISO: Master input mode, Slave output mode.

MOSI: Master output mode, Slave input mode.

21.3.3. GPIO reuse function control register GPIOMCR4/ GPIOMCR5

GPIO Base Address + 0x48 (0x40848)						
Symbol	GPIOMCR4/GPIOMCR5 (GPIO Multiplex Control Register 2)					
Bit	[31:16]					
Name	-					
RW	-					
Bit	[15:08]	[7:6]	[5]	[4:2]	[1]	[0]
Name	MASK	-	PTCI3E	PTPW2	PTPW3E	PTPW2E
RW	R0W-0	-	RW-0			

Bit	Name	Description		
Bit[5]	PTCI3E	TCI 3 mode control		
		0	The TCI3 is the same with the TCI1.	
		1	The TCI3 configuration is set by the PTCTC.	
Bit[4~2]	PTPW2	PWM communication IO port selection		
		0	Port 1.2 =PWM2	Port 1.3 =PWM3
		1	Port 1.6 =PWM2	Port 1.7 =PWM3
		2	Port 2.2 =PWM2	Port 2.3 =PWM3
		3	Port 2.6 =PWM2	Port 2.7 =PWM3
		4	Port 8.2 =PWM2	Port 8.3 =PWM3
		5	Port 8.6 =PWM2	Port 8.7 =PWM3
		6	Port 9.2 =PWM2	Port 9.3 =PWM3
		7	Port 9.6 =PWM2	Port 9.7 =PWM3
Bit[1]	PTPW3E	GPIO PWM3 control switch		
		0	Disable	

		1	Enable
Bit[0]	PTPW2E	GPIO PWM2 control switch	
		0	Disable
		1	Enable

21.3.4. GPIO reuse function control register GPIOMCR6/ GPIOMCR7

GPIO Base Address + 0x4C (0x4084C)				
Symbol	GPIOMCR6/GPIOMCR7 (GPIO Multiplex Control Register 3)			
Bit	[31:16]			
Name	-			
RW	-			
Bit	[15:08]	[7:4]	[3:1]	[0]
Name	MASK	-	PTUR2	PTUR2E
RW	R0W-0	-	RW-0	

Bit	Name	Description		
Bit[3~1]	PTUR2	UART2 communication IO port selection		
		0	Port 1.2 = Tx2	Port 1.3 = Rx2
		1	Port 1.6 = Tx2	Port 1.7 = Rx2
		2	Port 2.2 = Tx2	Port 2.3 = Rx2
		3	Port 2.6 = Tx2	Port 2.7 = Rx2
		4	Port 8.2 = Tx2	Port 8.3 = Rx2
		5	Port 8.6 = Tx2	Port 8.7 = Rx2
		6	Port 9.2 = Tx2	Port 9.3 = Rx2
		7	Port 9.6 = Tx2	Port 9.7 = Rx2
Bit[0]	PTUR2E	GPIO UART2 control switch		
		0	Disable	
		1	Enable	

Note: PTSPE and PTCSP related SPI I/O Port has the highest priority, when the associated I/O Port is selected as SPI purposes; the other IP and GPIO setting are invalid.

Note: I2CPTEn and I2CPTS related I2C I/O Port has the highest priority, when the associated I/O Port is selected as I2C purposes, Addition to other IP SPI and GPIO settings are invalid.

Note: Only limited I/O Port of output section, UART for the third priority, CMP is the fourth priority, ADC fifth priority, PWM for the sixth priority, GPIO is the lowest priority.

22. ΣΔ 24-BIT A/D CONVERTER (ADC)

22.1. Overall description

The chip has an embedded high-performance 24-bit A/D converter (24-bitΣΔADC). The ADC has a pre low-noise programmable gain amplifier (Low Noise PGA), which can be used to amplify input signals. The gain programmable setting range is 1~128. The sampling rate of the ADC can be programmed by the register. The highest designed sampling rate is 350 KHz per second. It has a 3-stage regulator for filtering the quantized noise of the regulator. The programmable range of the over-sampling rate of the ADC is 32~32768. It is designed to measure the sensors with extremely small signals, such as strain meter, pressure gauge and industry process control.

Note that although the ADC sampling rate can be set HACK frequency source to do a higher sampling rate selection, but in excess of 350KHz ADC sample rate, the ADC cannot guarantee stable operation at high resolution, it is proposed to set the sampling frequency at the 350KHz.

Features:

The settable sampling rate is 40 KHz~350 KHz;

The resolution of the effective number (ENOB) of bits is up to 21 bits;

The lowest input noise is 65nV RMS;

The settable over-sampling rate is 32~32768;

The highest output rate is 10 KHz;

The multiplier gain of the built-in low-noise programmable gain amplifier is 1~128;

Built-in temperature sensor is provided;

Built-in 4-bit DAC is provided to adjust the offset;

3-stage comb filter is provided.

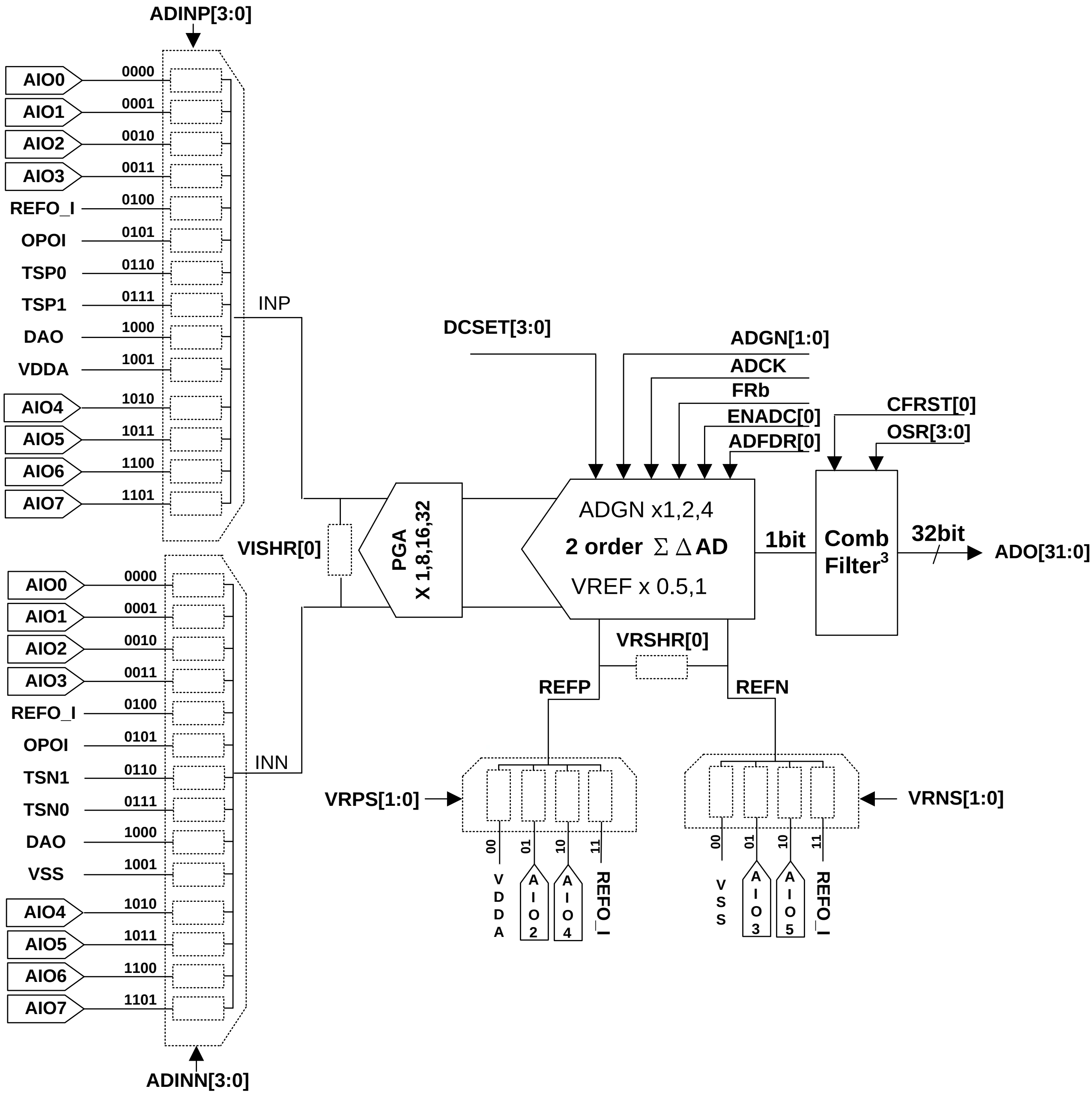
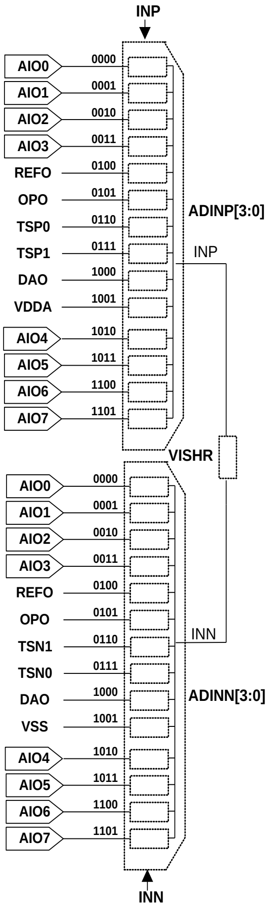


FIG. 22-1 ADC function block diagram

22.1.1. Fully differential signal input end

The input signal of the ADC is fully differential input mode; in other words, the input end is composed of the positive input end and the negative input end. The positive and negative signal input channels comprises 8 external signal input channels and 6 internal signal input channels. When the magnifying power of the ADC is 1, the input impedance of the signal input end of the ADC is 200K. The positive and negative signal input channels can be selected via the controllers ADINP [3:0] and ADINN [3:0]; however, the positive input end can only select one signal input channel at a time, and the negative input end can only select one signal input channel at a time. The positive and negative input ends can select the same input channel; in this way, the differential signal is close to 0 and only the offset is left. The ADC has an internal signal input channel short-circuit switch; the positive and negative the input ends can be set short-circuit via the control bit VISHR. The following figure lists the signal input channels of the positive and negative ends.



ADINP[3:0]	Positive input channel	ADINN[3:0]	Negative input channel
0000	AIO0	0000	AIO0
0001	AIO1	0001	AIO1
0010	AIO2	0010	AIO2
0011	AIO3	0011	AIO3
0100	REFO	0100	REFO
0101	OPO	0101	OPO
0110	TSP0	0110	TSN1
0111	TSP1	0111	TSN0
1000	DAO	1000	DAO
1001	VDDA	1001	VSS
1010	AIO4	1010	AIO4
1011	AIO5	1011	AIO5
1100	AIO6	1100	AIO6
1101	AIO7	1101	AIO7

FIG. 22-2 ADC signal input channel

The input signal is internally amplified and transferred, so the voltage range of the input signal is also limited. So as to obtain high resolution and linearity of the ADC outputs, it is suggested the differential voltage of the input signal be$\Delta SI=\pm 0.9*\Delta VREF$ ($\Delta SI=INP-INN$). The input signal voltages are as shown in the following table.

External input channel	Voltage input range
ADINP	VSS-0.2V ≦ INP ≦ VDDA
ADINN	VSS-0.2V ≦ INN ≦ VDDA

Table 22-1 Input signal voltage range table

22.1.2. Built-in gain amplifier

The ADC has two built-in gain amplifiers: one is the low-noise, low temperature coefficient programmable gain amplifier PGA whose magnifying power is 8/16/32; the other one is the programmable gain amplifier ΣAD whose magnifying power is 1, 2 and 4. Thus, the maximal magnifying power of the combination of the two gain amplifiers is 128. However, the magnifying power is in inverse proportion to the effective number of bits (ENOB) of the ADC output; if the magnifying power is larger, the ENOB will be smaller. Thus, the magnifying power should be set according to the actual requirements. The magnifying power of the PGA can be set by the controller PGA [2:0], and the selection of the magnifying power of the PGA is as shown in the following table. The magnifying power of the ADC modulator can be set by the controller ADGN [1:0], and the selection of the magnifying power of the ADC modulator is as shown in the following table.

	PGA					ADC Modulator			
PGA[2:0]	000	001	011	111	ADGN[1:0]	00	01	10	11
Magnifying power	x1	x 8	x16	x32	Magnifying power	x1	x2	RSV	x4

Table 22-2 internal gain magnifying power

22.1.3. Reference voltage input channel

The reference voltage input of the ADC is fully differential input mode; in other words, the reference voltage input end is composed of the positive input end and negative input end. Both of the positive input end and negative input end respectively have two external input channels and two internal input channels. The positive input end can only select one input channel at a time, and the negative input end can only select one input channel at a time. The reference voltage end further has a short-circuit switch and the short-circuit switch can be enabled or disabled by the control bit VRSHR to achieve the short circuit between the positive input end and the negative input end.

The reference voltage can be generated after the △VREF voltage difference generated after the inputs from the VREFP and VREFN and then pass the programmable reference voltage attenuator. The attenuation power of the reference voltage can be set by the controller FRb[0], and the attenuation power of the reference voltage is as shown in the following table.

The calculation of the reference voltage is as follows:

ΔVREF=VREFP-VREFN (Equation 22-1)

VREF=Gain x ΔVREF (Equation 22-2)

ΔVREF: the voltage difference of the reference voltage;

VREF: the internal reference voltage of the ADC

VREFP/VREFN: input reference voltage

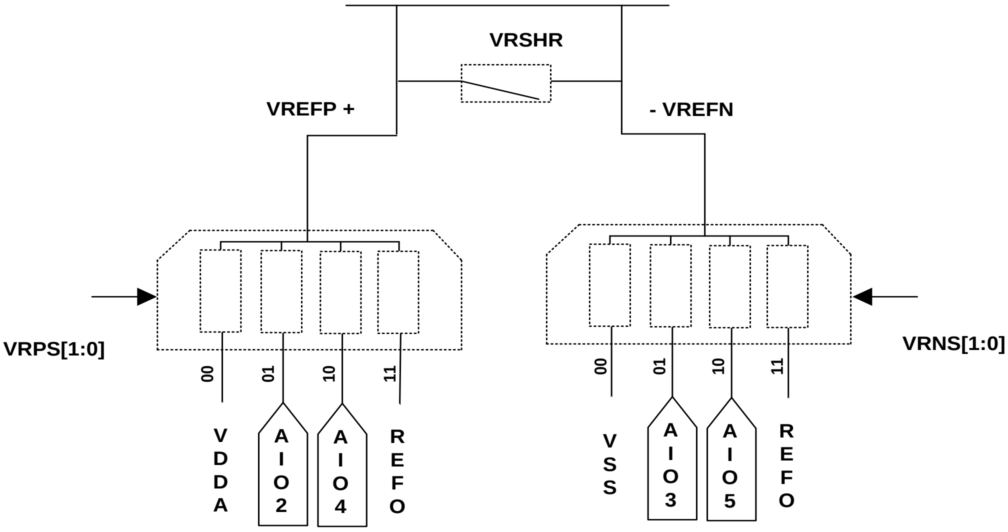


FIG. 22-3 Reference voltage input channel

Reference voltage attenuation power		
FRb[0]	0	1
Gain	1	1/2

Table 22-3 Reference voltage attenuation power

The input impedance of the positive input channels and the negative input channels is 500kΩ, and the input voltage of the VREFP or VREFN cannot be lower than the VSS and VDDA; if it is set as external input channel by the controller, the input impedance can be increased. However, it is necessary to pay attention to the voltage range of the external input channel.

External input channel	Voltage input range
AIO2 / AIO4	VREFN≦ VREFP ≦ VDDA

AIO3 /AIO5	VSS ≦ VREFN ≦ VREFP

Table 22-4 Voltage input range of reference voltage external input channel

22.1.4. Input bias of input signal

The ADC has a zero point bias translation controller, and the zero point bias translation controller DCSET [3:0] can change the position of the zero point of the signal to prevent the voltage of the input signal from being too high to exceed the maximal measurement range. After the signal to be measured adjusted via the pre PGA, the ADC modulator and the zero point bias translation, the calculation formula of the equivalent signal to be measured ΔSI_I is as follows:

ΔSI_I = PGA x ADGN x ΔSI± + (DCSET x ΔVREF) (Equation 18-3)

	DCSET[3:0]							
Setting value	0000	0001	0010	0011	0100	0101	0110	0111
Translation value	0*VREF	+1/8* VREF	+1/4* VREF	+3/8* VREF	+1/2* VREF	+5/8* VREF	+3/4* VREF	+7/8* VREF
Setting value	1000	1001	1010	1011	1100	1101	1110	1111
Translation value	0*VREF	-1/8* VREF	-1/4* VREF	-3/8* VREF	-1/2* VREF	-5/8* VREF	-3/4* VREF	-7/8* VREF

Table 22-5 Zero bias configuration conversion table of input signal to be measured

22.1.5. Comb filter

ΣΔADC adopts the 3-stage comb filter, and different over-sampling rates can be obtained by setting the controller OSR [3:0] and the different combinations of the sampling rates of the ADC so as to realize different ADC conversion output frequencies. The configuration parameters of the OSR [3:0] are as follows:

	OSR[3:0]										
Setting value	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
Frequency dividing value	32768	16384	8192	4096	2048	1024	512	256	128	64	32

Table 22-6 Frequency dividing table of over-sampling rates

The A/D conversion results are stored in the register ADCO[23:0], and the highest bit is the symbol bit, so the relations of the conversion results and the input signals are as shown in the following table.

2-polarity output 2's complement format	Equivalent signals to be measured	ADCO[23:0]	
		Hexadecimal system	Binary system
	ΔVR	7F FF FF	0111-1111 1111-1111 1111-1111
	$\Delta VR \times \frac{1}{2^{23}}$	00 00 01	0000-0000 0000-0000 0000-0001
		00 00 00	0000-0000 0000-0000 0000-0000
	$-\Delta VR \times \frac{1}{2^{23}}$	FF FF FF	1111-1111 1111-1111 1111-1111
	-ΔVR	80 00 00	1000-0000 0000-0000 0000-0000

Table 22-7 Relation table of ADCO[23:0] and input signals

The comb filter provides the reset control function; when the control bit CFRST is set as <0>, the comb filter will be reset, and then the comb filter will be enabled by setting the CFRST=<1>. In this way, the ΣΔADC will automatically throw the first 2 pieces of data. When the user is waiting for the interrupt taking place, the first piece of the ADC conversion data which is read is the effective ADC value.

22.1.6. Temperature sensor TPS

The temperature sensor is composed of a bipolar junction transistor, etc., and the change of the voltage signal to the temperature has passed the 0K curve; thus, it has the following features:

When the environmental temperature is 0K, the output voltage of the temperature sensor is $V_{TPS@0K}$ = 0V;

The asymmetry between the offset voltage ($V_{ADC\text{-}OFFSET}$) of the ADC and the BJT can be automatically cancelled by the measuring;

The temperature calibration only needs single point calibration to satisfy the ±2℃error;

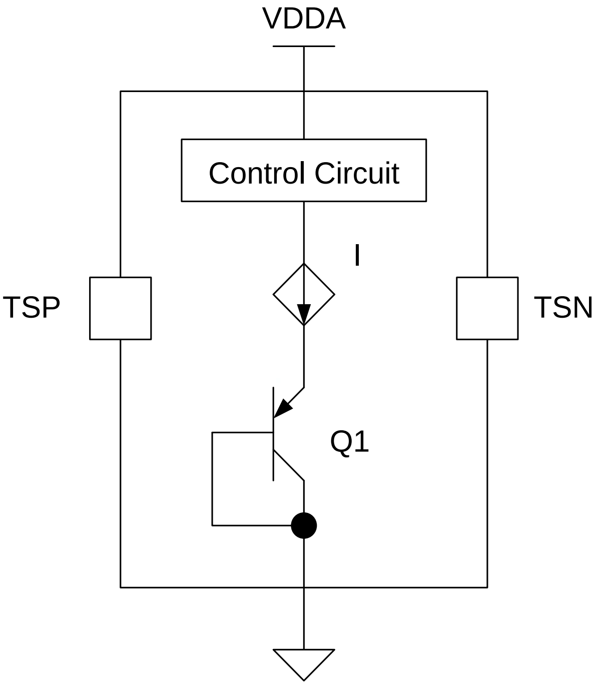


FIG. 22-4 Temperature sensor application block diagram

The TPS initialization configuration and calculation method are as follows:

Enable the ADC and the function of the TPS can be automatically enabled right away. Fix the related configuration of the ADC and the system operating frequencies, and the configurations for the TPS calibration and measurement should be the same with each other.

When it is under the same temperature Ta (°C) and the values of the V_{TSP0} / V_{TSN0} and V_{TSP1} / V_{TSN1} are measured, subtract the two values and calculate the average to obtain the corresponding voltage $V_{TS@Ta}$ of the TPS under the temperature Ta.

When measuring the V_{TSP0} / V_{TSN0}, set the ADINP [3:0] as <0111> and set the ADINN [3:0] as <0110>.

When measuring the V_{TSP1} / V_{TSN1}, set the ADINP [3:0] as <0110> and set the ADINN [3:0] as <0111>.
The value of V_{TSH0}/V_{TSL0} and V_{TSH1}/V_{TSL1} value subtraction operation and then divided by 2. To obtain the $ADC_{TPS@Ta}$
The variation of the output voltage V_{TS} of the TPS to the temperature is a linear curve; therefore, the gain G_{TPS} can be derived (or called slope).

$$G_{TPS} = \frac{ADC_{TPS@Ta}}{(273.15 + T_{offset} + T_A)K} \text{ ………………………………… (Equation 22-4)}$$

G_{TPS} : The gain or slope of the TPS sensor $(\frac{\text{ADC count}}{K})$

$ADC_{TPS@Ta}$: ADC values measured at calibratio n temperat ure

K : °C + 273.15

T_{offset} : Temperatur e offset

TPS in the temperature conversion is not ideal, so in fact not at ℃ = K-273.15

Instead ℃ = K + KT = K + (- 273.15-T_{offset})

For the KT values, refer to the TPS specification in the IC Data sheet ADC section.

HY16F19xB KT value of -288, ℃ = K-288 -> K=℃+288

TPS Example description

It is assumed that TPS calibration will be performed at 25 ° C. After calibration, the IC was moved to a higher temperature environment (65 ° C), Test the temperature in the environment.

(1) Set ADINP [3: 0] to set <0111> and ADINN [3: 0] to set <0110>, The ADC measures a digital code ADC_{TPS0} = 5897634.

(2) Set ADINP [3: 0] to set <0110> and ADINN [3: 0] to set <0111>, The ADC measures a digital code ADC_{TPS1} = -5827679.

(3) Calculate $ADC_{TPS @ 25}$ = (ADC_{TPS0} - ADC_{TPS1}) / 2 = 5862656. This action eliminates the Offset of the Temperature Sensor.

(4) Calculate G_{TPS}

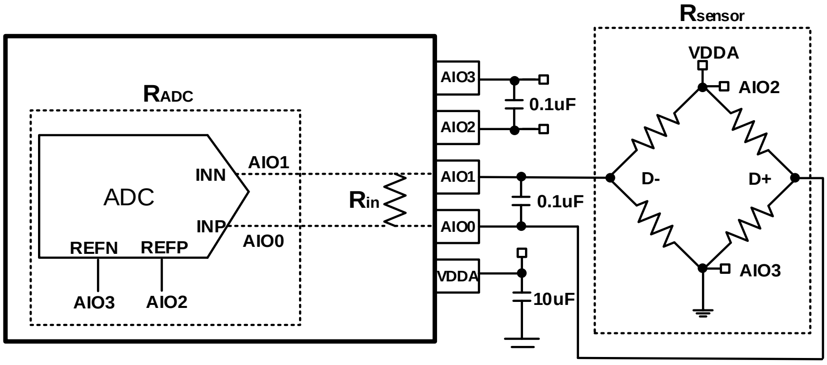
$$G_{TPS} = \frac{ADC_{TPS@Ta}}{(273.15 + T_{offset} + T_A)K} = \frac{5862656}{(288 + 25)K} = 18730.53$$

(5) After the IC was moved to a high temperature (65 ° C), Refer to steps (1) to (3) again to measure $ADC_{TPS\ @\ 65}$: 6630103

$$T_X = \frac{ADC_{TPS@65}}{G_{TPS}} - \left[273.15 + T_{offset}\ \right] = \frac{6630103}{18730.53} - 288 = 65.97^{o}C$$

22.1.7. ADC input impedance (R_{ADC}) description

The following figure shows: ADC input impedance (R_{ADC}) and the sensor output impedance (R_{sensor}) and the actual input to the chip input impedance (R_{in}) diagram. Users can follow the Sensor characteristics to assess whether the Sensor can be directly connected with the ADC input channel, to avoid the measurement of the impedance effect.



(R_{in}) and (R_{ADC}) and (R_{sensor}) relationship is: R_{in}= R_{sensor} // R_{ADC}

R_{in} : R_{in} is R_{sensor} parallel R_{ADC}

R_{ADC} : ADC input impedance

R_{sensor}: sensor output impedance

Note: (R_{ADC}) is not equivalent to the actual output impedance of the Sensor that can be connected to the actual HY16F ADC. When PGA and ADGN = 1 times, R_{ADC} = 2.5M, But this value is not equivalent to connect the Sensor's maximum output impedance value (R_{sensor}). Reference recommendations in the PGA and ADGN = 1 times the time, can be connected to the sensor maximum output impedance of 200k.

ADC input impedance (R_{ADC}) table

R_{ADC}(ohm) @ ADCK= 333kHz			
PGA	ADGN=1	ADGN=2	ADGN=4
1	2.5M	1.25M	626k
8	125k	125k	125k
16	62.5k	62.5k	62.5k
32	31.25k	31.25k	31.25k

Sensor output impedance (R_{sensor}) table

R_{sensor}(ohm) @ ADCK= 333kHz			
PGA	ADGN=1	ADGN=2	ADGN=4
1	200k	100k	50k
8	10k	10k	10k
16	5k	5k	5k
32	2.5k	2.5k	2.5k

22.1.8. ADC operation description

The ADC is the △-Σ structure of 24-bit resolution. If the user wants to enable the functions of the ADC, some peripheral circuits should be correct set. The power supply of the ADC is the VDDA voltage. Thus, the VDDA should be higher than 2.4V. If the user wants to better the performance of the ADC, a stable VDDA power supply is a must. As the VDDA needs some time to get ready, the ADD should start the measurement after the VDDA is ready. The offset and BandGap voltages can be enabled by setting the ENBGR is <1>. Nest, it still needs a 1.2V common mode voltage to enable the ADC. The common mode voltage can be selected from the inside or outside. The ADC also needs an ADCK clock input whose maximum is 350 KHz. The inputted clock should be set to be at least higher than 40 KHz.

Detail of the ADC initial configuration as follows:

(1) Select the input channels of the signals to be measured of the ADC, Including positive input channel ADINP 0x41104 [7:4], the negative input channel ADINN 0x41104 [3:0], ADC input of the short-circuit switch control VISHR 0x41100[21] and the ADC reference voltage input of the short-circuit switch control VRSHR 0x41100[20].

(2) Configure ADC internal gain magnifications, PGA 0x41104 [18:16] and ADGN 0x41104 [21:20], according to the actual situation, let ΔSI in the range 0.9 * VREF.

(3) Set the zero point bias DCSET 0x41104[27:24], if it is not necessary, please set 0* VREF.

(4) Select the ADC reference voltage input channel VRPS 0x41100 [19:18] and VRNS 0x41100 [17:16], and select the reference voltage attenuation rate FRb 0x41104 [19].

(5) Set the ADC conversion value of the output frequency OSR 0x41100 [5:2] need to be set according to the actual ENOB.

(6) Starts comb filter, CFRST 0x41100 [1] = <1>; set this bit hardware can automatically throws the first 2 pieces of the data.

(7) Set up and start ADC clock source (register 0x4030C [7: 4]), it recommended that the ADC sampling frequency is set at about 330 KHz.

(8) Open VDDA voltage VDAS 0x40400 [19:18] and set VDDA LDO voltage source ENVA 0x40400 [17:16] and BandGap reference voltage ENBGR 0x40400 [4] = <1>, open common-mode voltage reference ENERFO 0x40400 [1] = <1> and analog sources ACMS 0x40400 [3] = <1>, and wait for voltage stabilization time.

(9) According to need to open the ADC interrupt function ADCIE 0x40008 [16] = <1>, and enables the global interrupt GIE = <1>.

(10) Open ADC function ENADC 0x41100 [0] <1>, wait for the first ADC interrupt signal occurs, sampling ADC output data can be read register ADCO 0x41108 [31:8].

22.2. Register address

ADC Register Address	31	24	23	16	15	8	7	0
ADC Base Address + 0x00 (0x41100)	MASK0		REG0		MASK1		REG1	
ADC Base Address + 0x04 (0x41104)	REG2		REG3		MASK4		REG4	
ADC Base Address + 0x08 (0x41108)	ADO3		ADO2		ADO1		0x00	

22.3. Register function

22.3.1. Analog ADC register ADCCR0

ADC Base Address + 0x00 (0x41100)						
Symbol	ADCCR0 (ADC Control Register 0)					
Bit	[31:24]	[23:22]	[21]	[20]	[19:18]	[17:16]
Name	MASK	-	VISHR	VRSHR	VRPS	VRNS
RW	R0W-0	-	RW-0			
Bit	[15:08]	[7]	[6]	[5:2]	[1]	[0]
Name	MASK	-	ADFDR	OSR	CFRST	ENADC
RW	R0W-0	-	RW-0			

Bit	Name	Description	
Bit[21]	VISHR	ADC signal input (positive and negative) short-circuit switch control	
		0	Short-circuit switch opens.
		1	Short-circuit switch closes.
Bit[20]	VRSHR	ADC reference voltage input (positive and negative) short-circuit switch control	
		0	Short-circuit switch opens.
		1	Short-circuit switch closes.
Bit[19~18]	VRPS	Reference voltage positive input source selection	
		00	VDDA
		01	AIO2
		10	AIO4
		11	Reference buffer output(REFO_I)
Bit[17~16]	VRNS	Reference voltage negative input source selection	
		00	VSS
		01	AIO3
		10	AIO5
		11	Reference buffer output(REFO_I)
Bit[06]	ADFDR	Fast chopper stable mode configuration	
		0	Normal mode, the frequency of the chopper = ADCK/128
		1	Fast chopper mode, the frequency= ADCK/32, suitable for OSR<512
Bit[5~2]	OSR	ADC over-sampling output frequency configuration	

		(on the condition that the clock source of the ADC is 327680Hz)		
		0000	32768	Data output frequency 10sps
		0001	16384	Data output frequency 20sps
		0010	8192	Data output frequency 40sps
		0011	4096	Data output frequency 80sps
		0100	2048	Data output frequency 160sps
		0101	1024	Data output frequency 320sps
		0110	512	Data output frequency 640sps
		0111	256	Data output frequency 1280sps
		1000	128	Data output frequency 2560sps
		1001	64	Data output frequency 5120sps
		1010	32	Data output frequency 10240sps
		1011	Reserved (32768)	
		1100	Reserved (32768)	
		1101	Reserved (32768)	
		1110	Reserved (32768)	
		1111	Reserved (32768)	
Bit[01]	CFRST	Comb filter enable control		
		0	Reset (Level reset)	
		1	Enable	
Bit[00]	ENADC	ADC enable control		
		0	Disable	
		1	Enable	

22.3.2. Analog ADC register ADCCR1

ADC Base Address + 0x04 (0x41104)							
Symbol	ADCCR1 (ADC Control Register 1)						
Bit	[31:29]	[28]	[27:24]	[23:22]	[21:20]	[19]	[18:16]
Name	-	ADCMP	DCSET	-	ADGN	FRb	PGA
RW	-	RW-0	RW-0	-	RW-0		
Bit	[15:08]			[7:4]		[3:0]	
Name	MASK			ADINP		ADINN	
RW	R0W-0			RW-0			

Bit	Name	Description	
Bit[28]	ADCMP	CMP V12 voltage selection (when CPPS=11b)	
		0	V12 reference
		1	Vaccy reference
Bit[27~24]	DCSET	DC zero point translation input voltage selection (VREF = REFP-REFN)	
		0000	0 VREF
		0001	+1/8 VREF
		0010	+1/4 VREF
		0011	+3/8 VREF
		0100	+1/2 VREF
		0101	+5/8 VREF
		0110	+3/4 VREF
		0111	+7/8 VREF
		1000	0 VREF
		1001	-1/8 VREF
		1010	-1/4 VREF
		1011	-3/8 VREF
		1100	-1/2 VREF
		1101	-5/8 VREF
		1110	-3/4 VREF
		1111	-7/8 VREF
Bit[21~20]	ADGN	ADC input signal magnifying power Gain adjustor configuration	
		00	Gain = 1
		01	Gain = 2
		10	Reserved
		11	Gain = 4
Bit[19]	FRb	Reference voltage range selection	
		0	Full reference voltage input; it is VREF*1

		1	1/2 reference voltage input; VREF*1/2
Bit[18~16]	PGA	ADC input signal magnifying power PGA adjustor configuration	
		000	Gain = 1
		001	Gain = 8
		010	Reserved
		011	Gain = 16
		100	Reserved
		101	Reserved
		110	Reserved
		111	Gain = 32
Bit[7~4]	ADINP	ADC positive signal input end selection	
		0000	AIO0
		0001	AIO1
		0010	AIO2
		0011	AIO3
		0100	REFO_I
		0101	OPO
		0110	TSP0
		0111	TSP1
		1000	DAO
		1001	VDDA
		1010	AIO4
		1011	AIO5
		1100	AIO6
		1101	AIO7
		1110	Reserved
		1111	Reserved
Bit[3~0]	ADINN	ADC negative signal input end selection	
		0000	AIO0
		0001	AIO1
		0010	AIO2
		0011	AIO3
		0100	REFO_I
		0101	OPO
		0110	TSN1
		0111	TSN0
		1000	DAO

		1001	VSS
		1010	AIO4
		1011	AIO5
		1100	AIO6
		1101	AIO7
		1110	Reserved
		1111	Reserved

22.3.3. Analog ADC register ADCCR2

ADC Base Address + 0x08 (0x41108)		
Symbol	ADCCR2 (ADC Control Register 2)	
Bit	[31:16]	
Name	ADCO	
RW	R-0	
Bit	[15:8]	[7:0]
Name	ADCO	0X00
RW	R-0	R-0

ADCO [31:0] ADC transformed value output register; only data higher than 24-bit are effective.

23. RAIL-TO-RAIL OPAMP

23.1. Overall description

The chip has an embedded Rail-to-Rail OPAMP network, which is mainly used to deal with analog signals. The input range and the output range are from VSSA to VDDA. When the input signal range is between VSSA +0.1 V and VDDA - 0.1V, the open loop gain is higher than 80dB. When the output load is 50pF, the unit gain bandwidth is 1MHz. It has the 1mA input and output push-pull driving ability. The maximal drivable capacitor load is 100pF. The positive input end has 7 independent selection switches and the negative input end has 8 independent selection switches. The OPAMP network has a built-in 10pF capacitor. It can serve as input sampling capacitor or integrator. Different input channel configurations and 8-bit DAC configurations can achieve different applications. The output end of the OPAMP can be connected to an I/O pin, or used by other internal IPs. When it serves as comparator, its output is digital format. The user can set the output of the OPAMP to pass a 2us peak pulse filter. Besides, the output of the comparator can be on-and-off or opposite in phase.

The features of the OPAMP include:

Rail-to-Rail input range, and Rail-to-Rail output range;
Under a 22pF load, it can provide a 1MHz unit gain bandwidth and 60 phase margin;
The DC gain can be higher than 80dB;
1mA push-pull output driving ability;
The positive input end has 7 independent selection switches and the negative input end has 8 independent selection switches.
Built-in 10pF capacitor;
It can serve as comparator with the function of a chopper;
Built-in peak pulse digital low-pass filter;

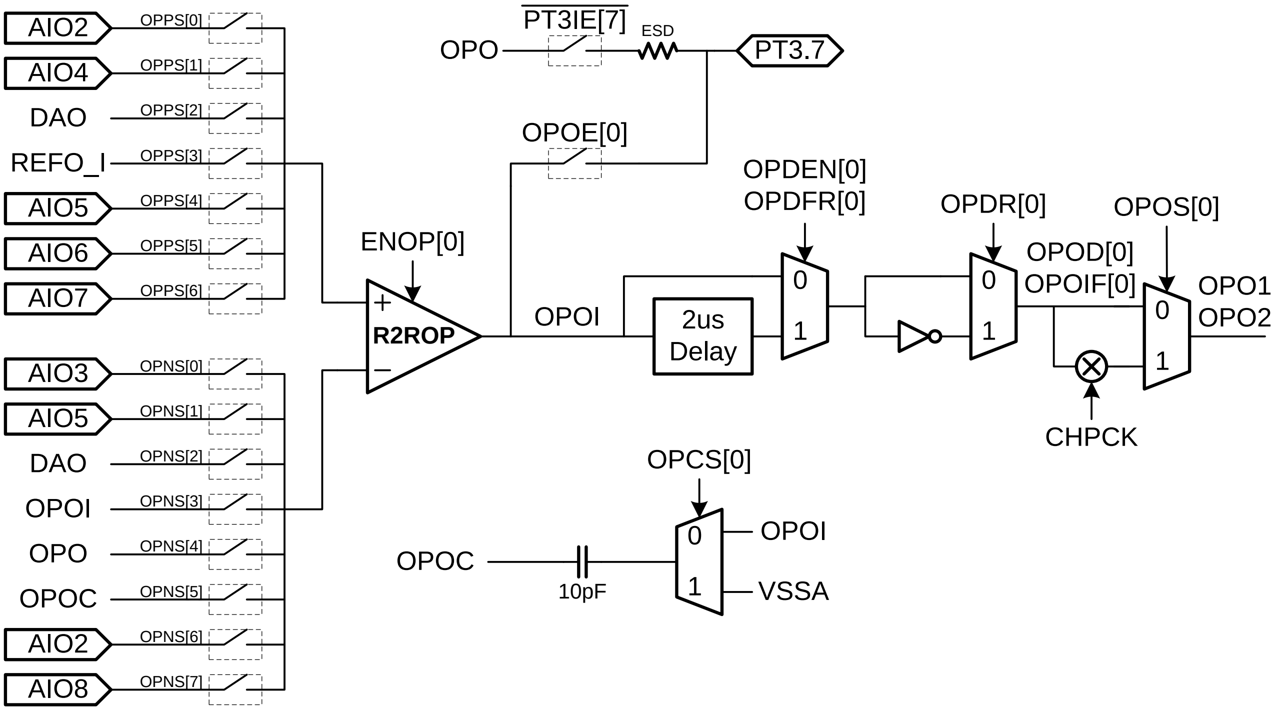


FIG. 23-1 OPAMP function block diagram

23.1.1. Input channel independent selection switch

The input channel selector of the OPAMP is not a multiplexer but an independent selection switch. The positive input channel of the OPAMP is controlled by 7 switches: AIO 2, AIO 4, DAO, REFO_I, AIO5, AIO6 and AIO7, which can be respectively controlled by the control bits OPPS[0], OPPS[1], OPPS[2], OPPS[3], OPPS[4], OPPS[5] and OPPS[6]; besides. The negative input channel of the OPAMP is controlled by 8 switches: AIO3, AIO5, DAO OPOI, OPO, OPOC, AIO2 and AIO8, which can be respectively controlled by the control bits OPNS[0], OPNS[1], OPNS[2], OPNS[3], OPNS[4], OPNS[5], OPNS[6] and OPNS[7].

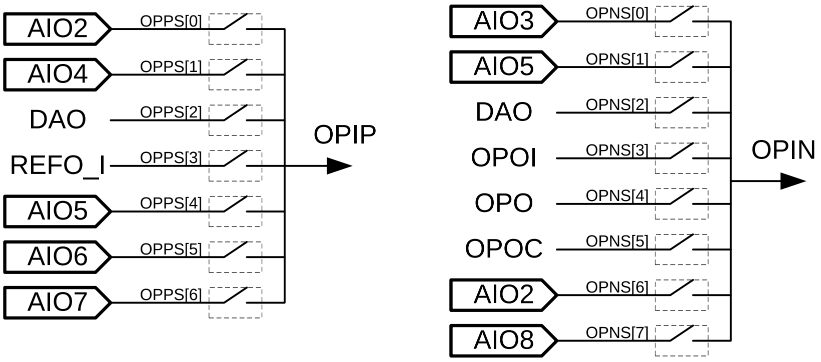


FIG. 23-2 Input channel configuration diagrams

23.1.2. Built-in 10pF capacitor

The OPAMP has a built-in 10uF capacitor, which can have different functions under different configurations. The upper end of the capacitor is connected to the OPOC and can be connected to the negative input end; the switch is set by the control bit OPNS [6]; the lower end of the capacitor can be connected to the OPOI or VSSA, which can be set by the control bit OPCS[0]. There are two methods to sample the analog inputs. One is the open loop sampling technique, and the method requires the analog signals are inputted from the AIO 3 or AIO 5. The configuration of the channel switch is as follows: first, set the OPNS[5] as 1 and set the OPCS[0] as 1; then, set the OPNS[0]=1 (select the AIO3) or set the OPNS[1]=1 (select the AIO5); after the sampling is finished, set the OPNS [5]=0; the voltage data are stored in the capacitor corresponding to the VSSA. The other one is the close loop sampling technique: the method should enable the OPAMP first, which means setting the ENOP=1; then, enable the OPOI and OPO, which means setting the OPNS[4]=1 and OPNS[3]=1; afterward, the lower end of the capacitor is connected to the OPOI, which means the OPCS=1; enable the AIO2 and AIO 4, which means setting the OPPS[0]=1 and OPPS[1]=1; after the sampling is finished, disable the OPOC, which means setting the OPNS[5]=0; the voltage data are also stored in the capacitor corresponding to the VSSA. The close loop method can store the offset of the OPAMP in the capacitor. Additionally, if the applications have the sensors with very high output impedance, the close loop sampling technique is a better choice. Finally, the lower end of the capacitor can be connected to the output end of the OPAMP, which means setting the OPCS=0. Meanwhile, the AIO3 and the AIO5 pins can be used to perform cumulative charge.

23.1.3. Comparator function

If the configuration of the OPAMP is set as the open loop function, the OPAMP can serve as comparator. The 1-bit binary codes can be outputted by the OPOD. If the positive input is higher than the negative input, the OPOD outputs 1; if the positive input is smaller than the negative input, the OPOD outputs 0. In order to prevent from the peak pulse interference, the outputs of the OPOD can further pass the 2us low-pass filter. If any peak pulse is smaller than 2us, the outputs of the comparator will not change. The outputs of the comparator can be changed by setting the control bit OPDR.

The output of the comparator can be also connected to the I/O pins; The PT3.0/PT3.1 is respectively the output pins of the OPO1/OPO2. The output results of the comparators can further be multiplied by the clock frequency of the charge pump (CHPCK) to output a high-frequency signal, which can serve as the LED driver.

23.1.4. Operation description

The OPAMP is a more universal Rail-to-Rail OP amplifier. It can be used to deal with analog signals. When it is used as OP amplifier, the voltage of the VDDA is higher than 2.4V and the reference voltage of the BandGap should be enabled in advance. Within the effective input range, the OPAMP is Rail-to-Rail. However, in order to achieve better performance, it is suggested that the input common mode voltage range is between VSSA+0.1V~VDDA-0.1V. The input impedance of the OPAMP is higher than 1GΩ.

Initialization configuration:

1. Open VDDA voltage VDS 0x40400 [19:18] and set VDDA LDO voltage source ENVA 0x40400 [17:16]. open common-mode voltage reference ENRFO 0x40400 [1] = <1>, the voltage of the VDDA should be higher than 2.4V, and wait until the voltages are stable.
2. Select the output pins of the OPO1/OPO2, and set the corresponding IO pins as the input mode. It is not necessary to set the above configuration if the above function is not used;
3. Select the positive input channel, negative input channel according to the actual applications;
4. Set the 2us low-pass filter and enable or disable it according to the actual requirements;
5. Set the clock frequency of the charge pump and determine whether it is multified by the frequency or not;
6. Enable the analog output of the OPAMP, which means enabling the OPOE;
7. Enable the digital output of the OPAMP according to the actual requirements, which means enabling the OPDEN;

8. If the digital output of the OPAMP is enabled, the output result should be set to be opposite in phase or not according to the actual requirements, which means setting the OPDR.
9. Enable the OPAMP to enable the OP amplifier, which means enabling the ENOP;

23.2. Register address

OPAMP Register Address	31	24	23	16	15	8	7	0
OPAMP Base Address + 0x00 (0x41900)	-		-		MASK0		REG0	
OPAMP Base Address + 0x04 (0x41904)	OPPSM		OPPS		OPNSM		OPNS	

-Reserved

23.3. Register function

23.3.1. Analog OPA register 0

OPA Base Address + 0x00 (0x41900)									
Symbol	OPAMP0 (OPAMP Control Register 0)								
Bit	[31:16]								
Name	RSV								
RW	R-0								
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	OPOD	OPOS	OPDR	OPCS	OPDFR	OPDEN	OPEO	ENOP
RW	R0W-0	R-0	RW-0						

Bit	Name	Description	
Bit[7]	OPOD	OPAMP digital output value, read-only	
		0	Negative input end signal > positive input end signal;
		1	Positive input end signal > Negative input end signal;
Bit[6]	OPOS	Set the outputs of the OPO1/OPO2 pass through the CHPCK multi-functioner.	
		0	If the outputs fail to pass the CHPCK multi-functioner, the outputs of the OPO1/OPO2 are equivalent to the OPOD.
		1	If the outputs pass the CHPCK multi-functioner, the OPO1/OPO2 are equivalent to the OPOD+CHPCK.
Bit[5]	OPDR	OPAMP digital output phase selection	
		0	Normal output
		1	Reverse output
Bit[4]	OPCS	OPAMP built-in capacitor purpose configuration	
		1	The capacitor serves as integrated capacitor, and the lower end is connected to the OPOI.

		0	The capacitor serves as sampling capacitor, and the lower end is connected to the VSSA.
Bit[3]	OPDFR	OPAMP output digital filter enable control	
		0	Disable
		1	Enable (Pass through the 2us deglitch)
Bit[2]	OPDEN	OPAMP digital output function control	
		0	Disable
		1	Enable
Bit[1]	OPOE	OPAMP analog output function control	
		0	Disable
		1	Enable
Bit[0]	ENOP	OPAMP function enable control	
		0	Disable
		1	Enable

23.3.2. Analog OPA register 1

OPA Base Address + 0x04 (0x41904)			
Symbol	OPAMP1 (OPAMP Control Register 1)		
Bit	[31:24]	[23]	[22:16]
Name	MASK	-	OPPS[6:0]
RW	R0W-0	-	RW-0
Bit	[15:08]	[07:00]	
Name	MASK	OPNS[5:0]	
RW	R0W-0	RW-0	

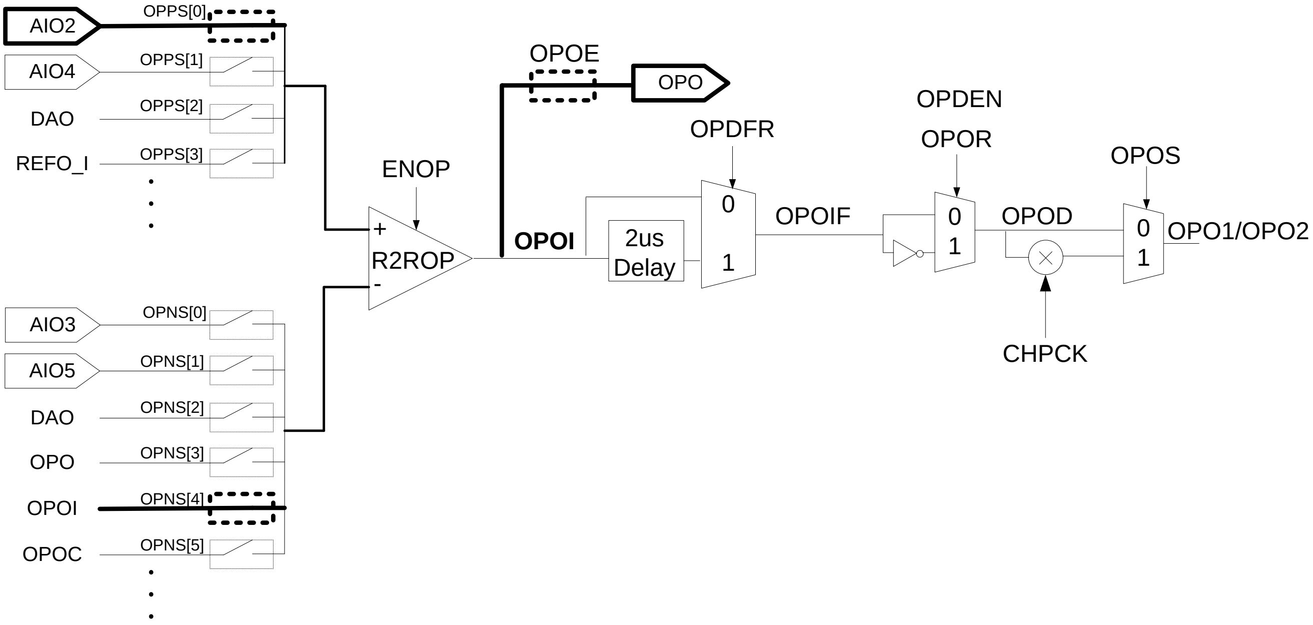
Bit	Name	Description	
Bit[22]	OPPS[6]	OPAMP positive input channel 6	
		0	Disable, high impedance
		1	Enable and connect to the AIO7
Bit[21]	OPPS[5]	OPAMP positive input channel 5	
		0	Disable, high impedance
		1	Enable and connect to the AIO6
Bit[20]	OPPS[4]	OPAMP positive input channel 4	
		0	Disable, high impedance
		1	Enable and connect to the AIO5
Bit[19]	OPPS[3]	OPAMP positive input channel 3	
		0	Disable, high impedance
		1	Enable and connect to the REFO_I
Bit[18]	OPPS[2]	OPAMP positive input channel 2	
		0	Disable, high impedance
		1	Enable and connect to the DAO
Bit[17]	OPPS[1]	OPAMP positive input channel 1	
		0	Disable, high impedance
		1	Enable and connect to the AIO4
Bit[16]	OPPS[0]	OPAMP positive input channel 0	
		0	Disable, high impedance
		1	Enable and connect to the AIO2
Bit[7]	OPNS[7]	OPAMP negative input channel 7	
		0	Disable, high impedance
		1	Enable and connect to the AIO8
Bit[6]	OPNS[6]	OPAMP negative input channel 6	
		0	Disable, high impedance
		1	Enable and connect to the AIO2

Bit[5]	OPNS[5]	OPAMP negative input channel 5	
		0	Disable, high impedance
		1	Enable and connect to the OPC: internal 10pF capacitor
Bit[4]	OPNS[4]	OPAMP negative input channel 4	
		0	Disable, high impedance
		1	Enable and connect to the OPO: internal OPAMP output
Bit[3]	OPNS[3]	OPAMP negative input channel 3	
		0	Disable, high impedance
		1	Enable and connect to the OPOI: internal OPAMP output
Bit[2]	OPNS[2]	OPAMP negative input channel 2	
		0	Disable, high impedance
		1	Enable and connect to the DAO
Bit[1]	OPNS[1]	OPAMP negative input channel 1	
		0	Disable, high impedance
		1	Enable and connect to the AIO5
Bit[0]	OPNS[0]	OPAMP negative input channel 0	
		0	Disable, high impedance
		1	Enable and connect to the AIO3

Application Circuit

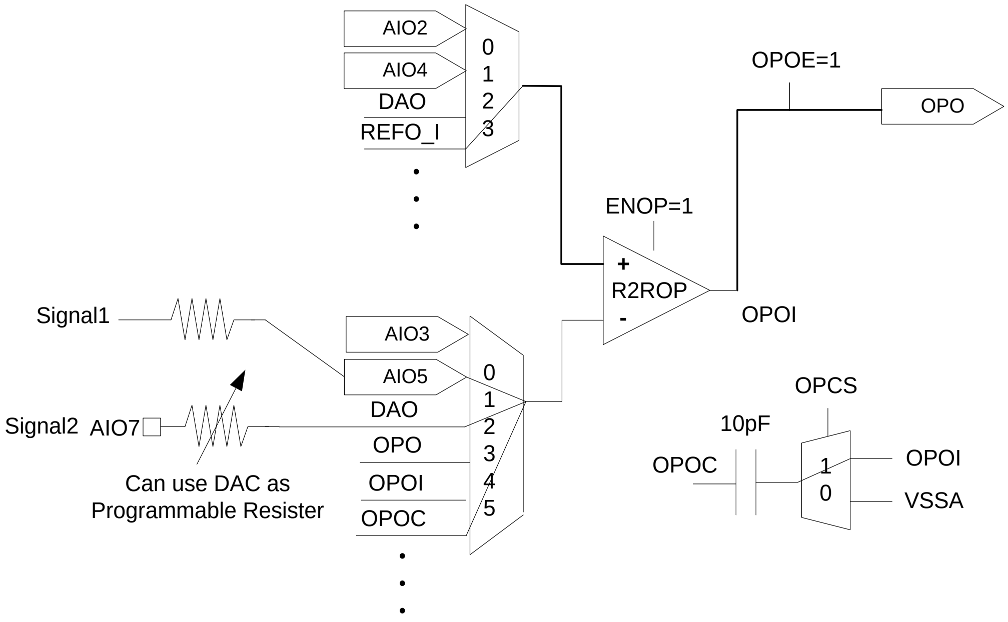
23.3.3. OPAMP application circuit system I

Use as a Unit Gain Buffer



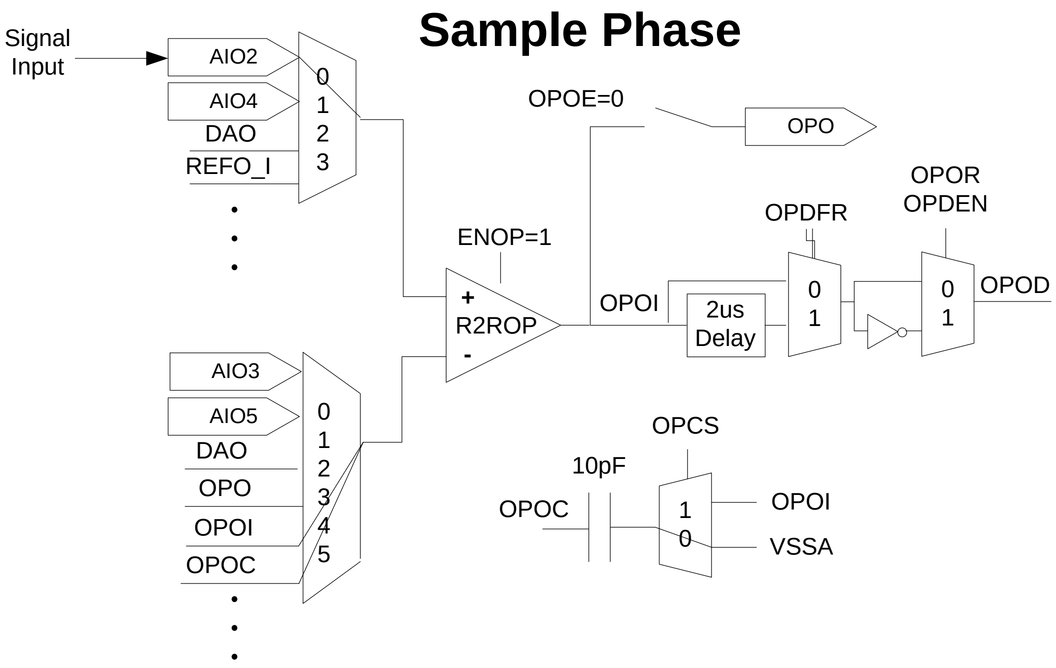
23.3.4. OPAMP application circuit system II

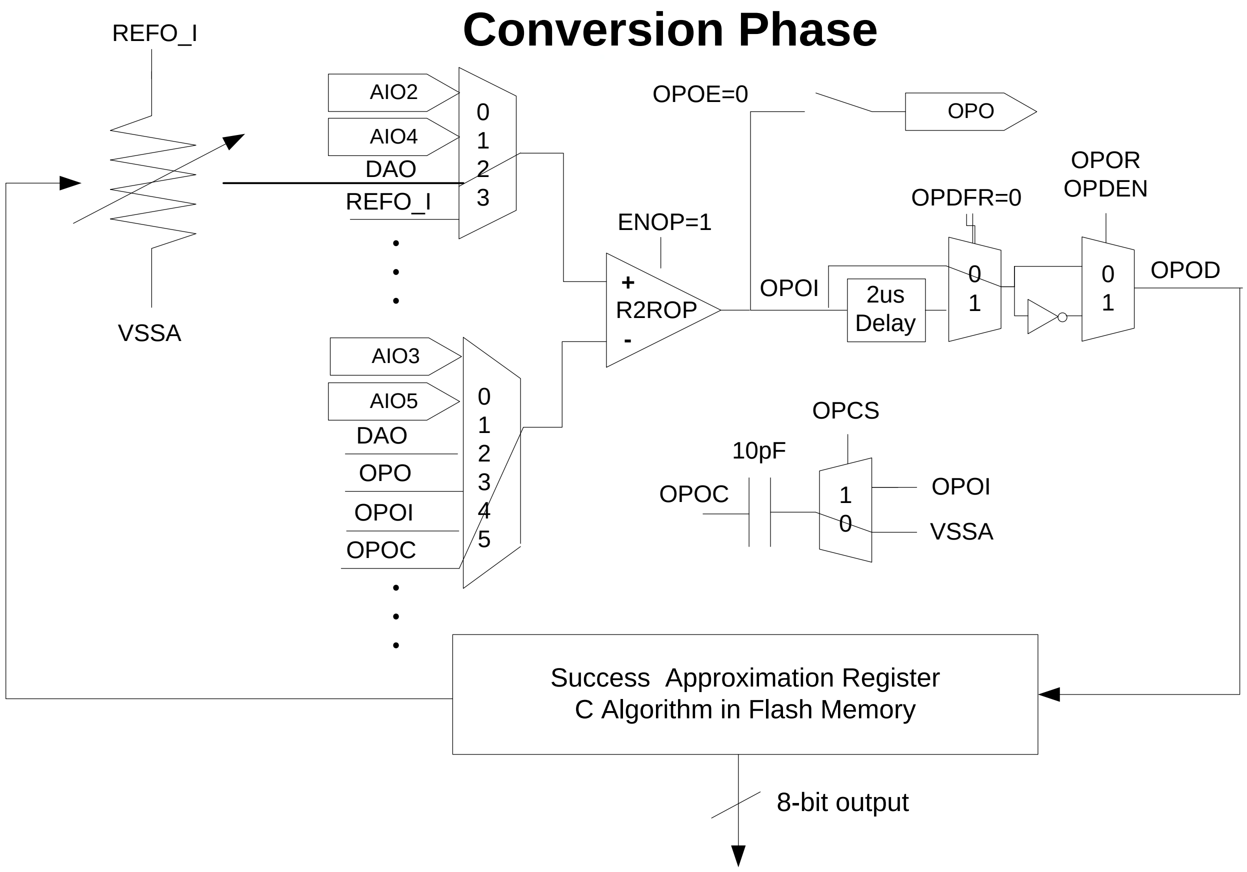
Use as an Integrator



23.3.5. OPAMP application circuit system III

Use as a 8-bit SAR ADC





24. 8-bit Resistance Ladder Network

24.1. Overall description

The chip has an embedded 8-bit Resistance Ladder Network, which is composed of a step resistor with absolute monotonicity.

Features of 8-bit Resistance Ladder include:

8-bit monotonic output

Internal or external reference programmable selection

It can serve as programmable resistor.

Operation of 8-bit Resistance Ladder:

When the ENDA is 0, the 8-bit Resistance Ladder will be disabled and no power consumption will be incurred. The Vrefp multiplexer is disabled and becomes a high impedance node. However, the Vrefn is still enabled, and connects to one of the sources. If the DAOE is set as 1, it will become a programmable resistor able to mark the ohm values.

8-bit Resistance Ladder output:

The DAO can generate the output voltages according to the data stored in the DABIT and VDA_Vrefp – VDA_Vrefn.

DABIT is based on straight binary system; the following figure is the transmission function diagram.

$$DAO = (V_{DAC\_Vrefp} - V_{DAC\_Vrefn}) \times \frac{DAbit\_in}{256} + V_{DAC\_Vrefn}$$

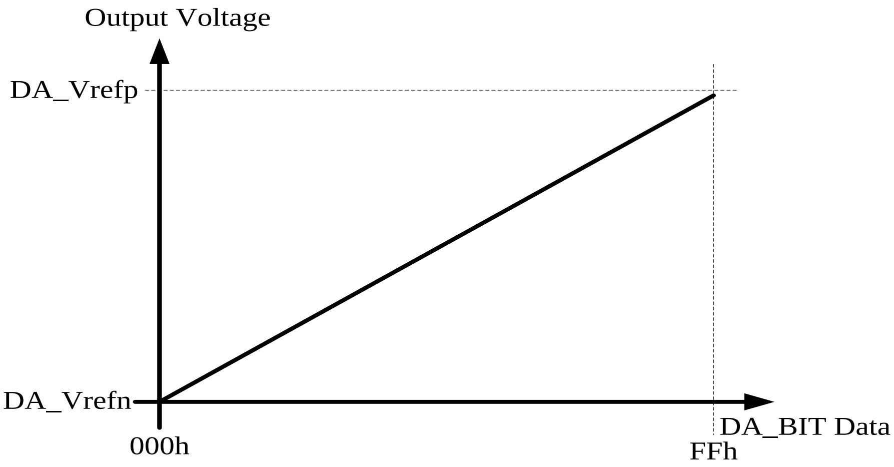
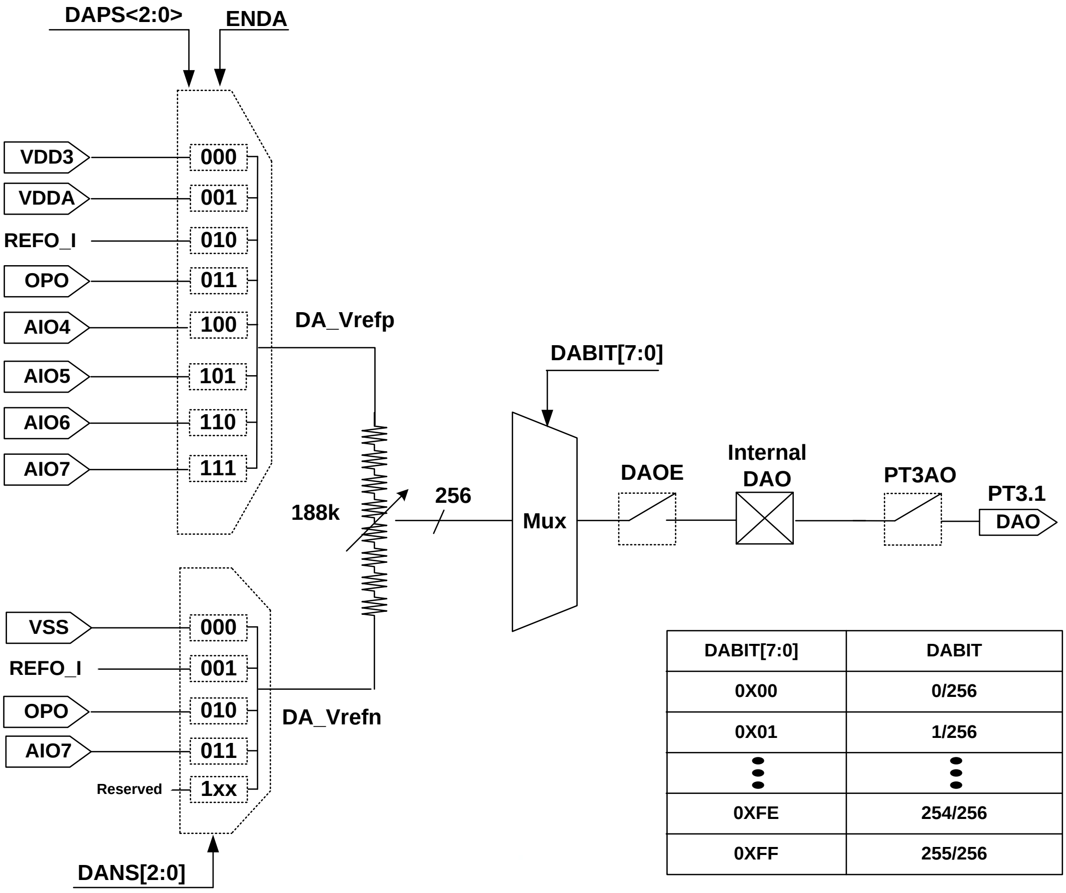


FIG. 24-1 8-bit Resistance Ladder conversion diagram



DABIT[7:0]	DABIT
0X00	0/256
0X01	1/256
⋮	⋮
0XFE	254/256
0XFF	255/256

FIG. 24-2 8-bit Resistance Ladder function block diagram

8-bit resistance ladder initial configuration:

(1) Open VDDA voltage VDAS 0x40400 [19:18] and set the input source voltage VDDA Regulator ENVA 0x40400 [17:16] open the common-mode voltage reference ENRFO 0x40400 [1] = <1>, VDDA voltage is greater than 2.4V, wait for stabilization time.
(2) Set the 8-bit resistance ladder positive and negative reference voltage input (register address 0x41700 [7: 0]), and set the initial scale value 8-bit resistance ladder output voltage (register bits address 0x41700 [7: 0]).
(3) Set the 8-bit resistance ladder output IO interface, set PT3.1 as 8-bit resistance ladder output (0x40828 [17] = <1>).
(4) Open 8-bit resistance ladder output switching control, set DAOE 0x41700 [1] = <1>.
(5) 8-bit resistance ladder feature is turned on, set ENDA 0x41700 [0] = <1>.

24.2. Register address

DAC Register Address	31	24	23	16	15	8	7	0
DAC Base Address + 0x00 (0x41700)	-		-		MASK0		REG0	
DAC Base Address + 0x04 (0x41704)	-		-		MASK1		REG1	

-Reserved

24.3. Register function

24.3.1. 8-bit resistance ladder register 0

8-bit resistance ladder Base Address + 0x00 (0x41700)						
Symbol	8-bit resistance ladder 0 (8-bit resistance ladder Control Register 0)					
Bit	[31:16]					
Name	RSV					
RW	R-0					
Bit	[15:8]	[7]	[6:4]	[3:2]	[1]	[0]
Name	MASK	DANS[2]	DAPS[2:0]	DANS[1:0]	DAOE	ENDA
RW	R0W-0	RW-0				

Bit	Name	Description	
Bit[6~4]	DAPS	8-bit resistance ladder positive input source selection	
		000	VDD3V
		001	VDDA
		010	REFO_I
		011	OPO
		100	AIO4
		101	AIO5
		110	AIO6
		111	AIO7
Bit[3~2 & 7]	DANS	8-bit resistance ladder negative input source selection	
		000	VSS
		001	REFO_I
		010	OPO
		011	AIO7
		100~ 111	Rsv
Bit[1]	DAOE	8-bit resistance ladder output enable control.	
		0	Disable, under high impedance
		1	Enable, the 8-bit resistance ladder outputs corresponding voltage.

Bit[0]	ENDA	8-bit resistance ladder function enable control	
		0	Disable
		1	Enable

24.3.2. 8-bit resistance ladder register 1

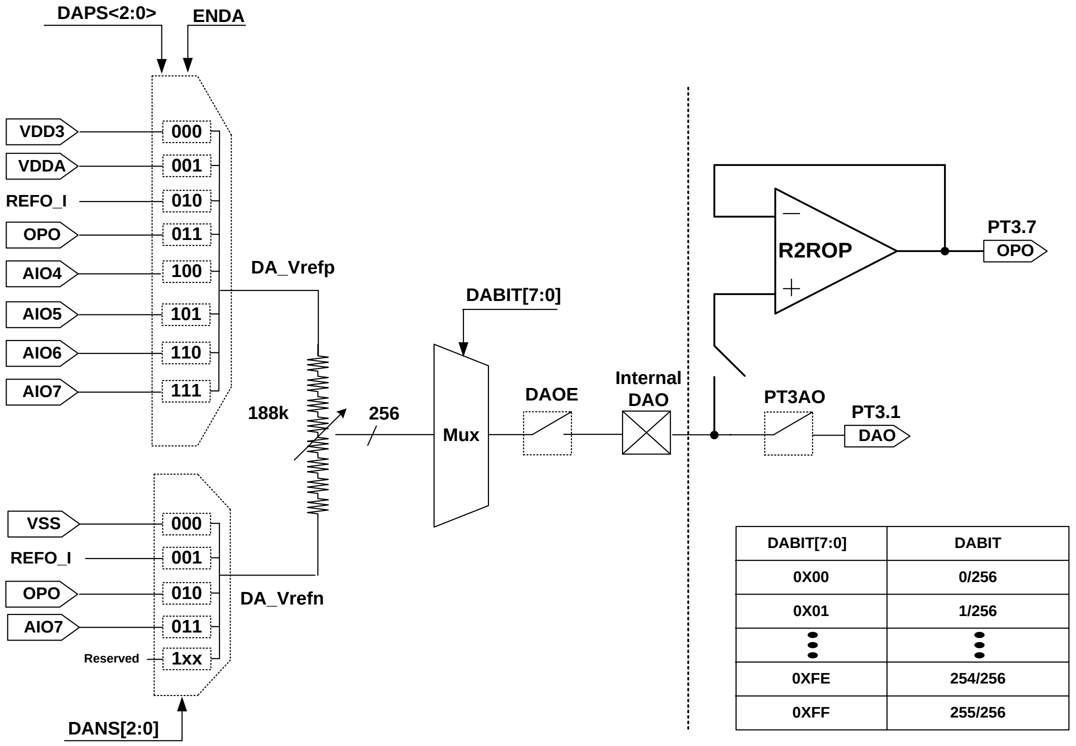
8-bit resistance ladder Base Address + 0x00 (0x41704)		
Symbol	8-bit resistance ladder 1 (8-bit resistance ladder Control Register 1)	
Bit	[31:16]	
Name	RSV	
RW	R-0	
Bit	[15:8]	[7:0]
Name	MASK	DABIT[7:0]
RW	R0W-0	RW-0

Bit	Name	Description
Bit[7~0]	DABIT	DABIT [7:0] the ratio setting of the output voltage; That is DABIT [7:0]/256.

24.4. DAC application circuit

24.4.1. application circuit system 01

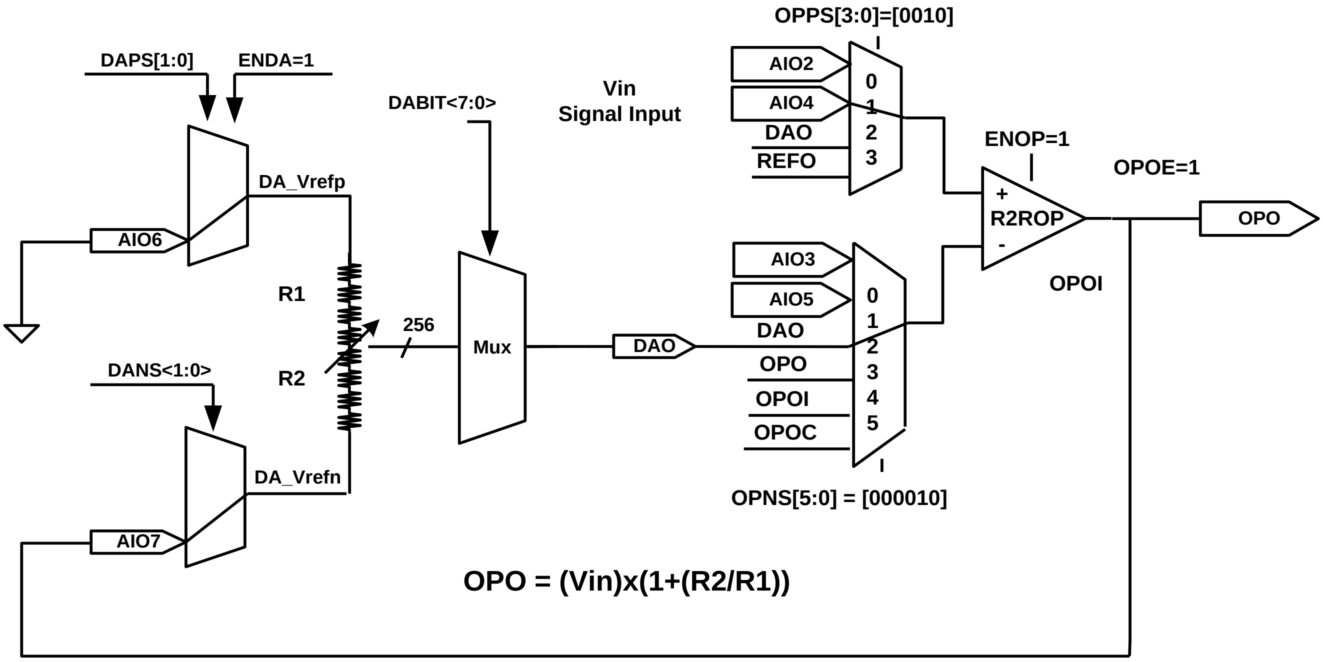
Use DAO Output by R2ROP



DABIT[7:0]	DABIT
0X00	0/256
0X01	1/256
⋮	⋮
0XFE	254/256
0XFF	255/256

24.4.2. DAC application circuit 02

Use as Programmable Gain Amplifier



25. MULTIPLE FUNCTION COMPARATOR CMP

25.1. Overall description

The chip has an embedded low-power, Rail-to-Rail multi-function comparator CMP for the comparing analog signals. It has the interrupt function; when the comparison result generates, the interrupt signal also generates; and it can increase the operability for users. It has different configuration settings for different applications.

Features of CMP include:

Rail-to-Rail input range

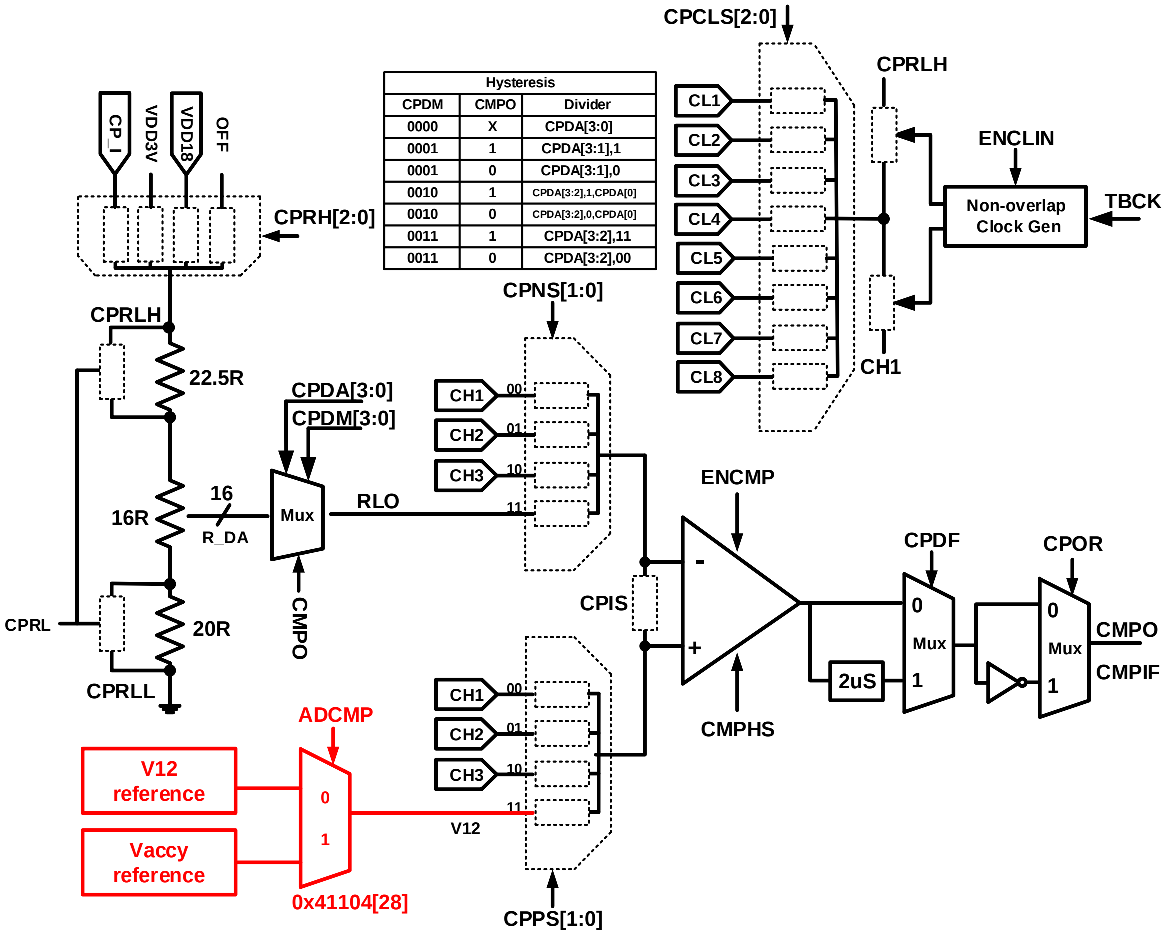
Low operating current

2us peak pulse filter

Buit-in 16 nodes 4-bit step resistor DAC with different comparison sets.

Change and discharge paths measured by touch buttons.

Interrupt signals can be generated, which belong to the interrupt vector HW3.



Hysteresis		
CPDM	CMPO	Divider
0000	X	CPDA[3:0]
0001	1	CPDA[3:1],1
0001	0	CPDA[3:1],0
0010	1	CPDA[3:2],1,CPDA[0]
0010	0	CPDA[3:2],0,CPDA[0]
0011	1	CPDA[3:2],11
0011	0	CPDA[3:2],00

FIG. 25-1 CMP network diagram

25.1.1. Multiplexing input channel selector

The input channel of the comparator is composed two parts; one is the input channel of the comparator, which can be set by the controller CPPS[1:0]/CPNS[1:0] to respectively set the positive input channel and the negative input channel of the comparator; the other one is the touch button input channel, which can be set by the controller CPCLS[2:0]. Via proper configuration and the combination of the input channels of the two parts, the applications of the touch button can be realized. When using the comparator, the user can set the control bit CPIS as <1> to realize the short-circuit between the positive input end and the negative input end; on the contrary, if the CPIS is set as <0>, the short-circuit will not be realized.

25.1.2. Built-in multi-node resistor and resistor node selection

The comparator has a built-in multi-node resistor, and the resistor includes three parts: 22.5R, 16R and 20R. The 16R resistor is connected to a 16-stage resistor node selector; the selector divides the 16R resistor into 16 nodes, which can be set by the controllers CPDA [3:0] and CPDM [3:0] to select different resistor nodes to output different voltages to the input channel RLO of the comparator. If the control bits CPRLH and CPRLL are set as <1>, the short circuit between the 22.5R resistor and 20R resistor can be achieved, which can adjust the resistor node voltage. The voltage sources of the multi-node resistor are VDD18/VDD3V/CP_I, and the controller CPRH [1:0] can be used to select different voltage sources to increase the output range of the node voltage.

The hysteresis controller CPDM [3:0] is linked up with the node selector CPDA [3:0]; each bit of the hysteresis controller CPDM [3:0] is corresponding to the control of the enablement and disablement of the each bit of the controller CPDA [3:0] respectively. When the corresponding bit of the hysteresis controller CPDM[3:0] is set as <1>, the hysteresis function of the corresponding bit of the node controller CPDA[3:0] will be enabled and the status of the bit is consistent with the output status of the comparator; that is CPDA[X]=CMPO. In this way, the node selector will be switched between the two nodes.

'u' means no change.

CPDM[3:0]	CMPO	CPDA[3:0]	CPDM[3:0]	CMPO	CPDA[3:0]
	Output status	Hysteresis switch period		Output status	Hysteresis switch period
0000	0	uuuu	1000	0	0uuu
	1	uuuu		1	1uuu
0001	0	uuu0	1001	0	0uu0

	1	uuu1		1	1uu1
0010	0	uu0u	1010	0	0u0u
	1	uu1u		1	1u1u
0011	0	uu00	1011	0	0u00
	1	uu11		1	1u11
0100	0	u0uu	1100	0	00uu
	1	u1uu		1	11uu
0101	0	u0u0	1101	0	00u0
	1	u1u1		1	11u1
0110	0	u00u	1110	0	000u
	1	u11u		1	111u
0111	0	u000	1111	0	0000
	1	u111		1	1111

Table 25-1 Hysteresis control CPDM[3:0] configuration and values

25.1.3. Comparator output

The output of the comparator is digital output, and it will reach the IO pin PT1.7; therefore, the output of the comparator should set the IO to serve as the output mode. The output of the comparator can be set to pass through the 2us low-pass filter to eliminate the peak pulse interference. If the control bit CPDF is set as <1>, the output of the comparator will pass through the 2us low-pass filter; if the control bit CPDF is set as <0>, it will not pass through the filter. The polarity of the comparator can be set by the control bit CPOR. If the CPOR is set as <1>, the output of the comparator will be opposite in phase; if the CPOR is set as <0>, the output of the comparator will be normal.

25.1.4. Application of touch button

The comparator has a special function: measuring the touch button. The major principle is to set the comparison voltage via the multi-node resistor and then input which into the RLO; the multi-node resistor provides voltage to charge the touch button and then the charges of the touch button charges the external reference capacitor of the negative input channel CH1; next, the TMB counts the charge time that the voltage of the CH1 is higher than the voltage of the RLO and then determine the status that the touch button is touched or not according to the charge time.

Two switches need to be used to control the charging of the corresponding touch button and the charging of the touch button to the reference capacitor; besides, if one of the switches is close, the other one must be open. The comparator has a built-in non-overlap controller to control the switches to ensure one of them is close and the other one is open.

The operating frequency of the non-overlap controller is provided by the operating clock of the TMB. Therefore, if the function should be used, it is necessary to enable the counting function of the TMB and clear the counter register of the TMB.

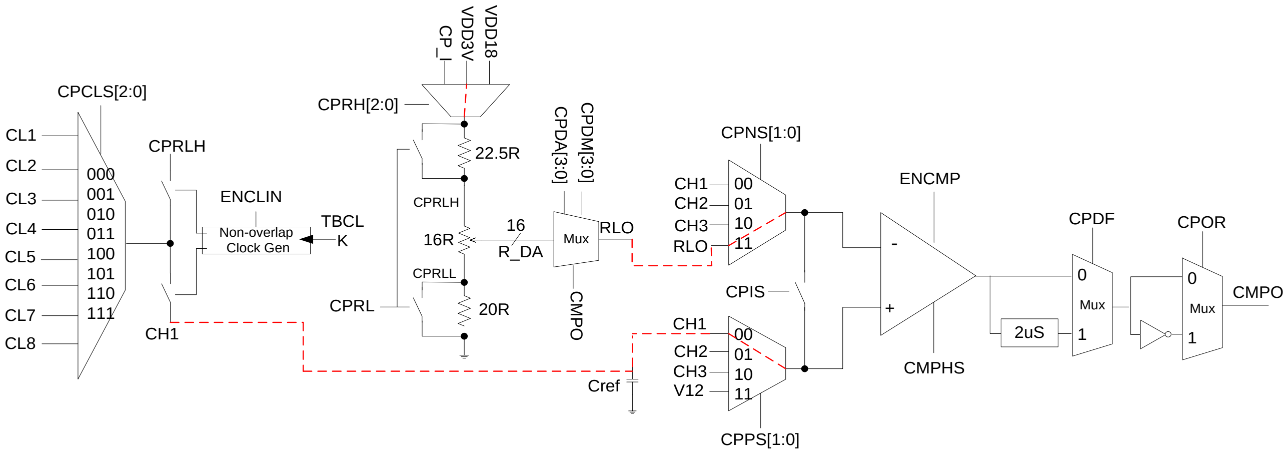


FIG. 25-4 Touch button connection diagram (One possible configuration)

25.1.5. Comparator operation initialization

The main function of the comparator is to compare the input signals; however, different modular combinations need different configurations to achieve different applications.

As a simple signal comparator:

1. Set the operating mode of the CMP to be low-power or normal.
2. Select the input channel, including the positive input channel, negative input channel;
3. If the RLO is selected as the positive input channel, the reference voltage source and the voltage node of the multi-node resistor should be set;
4. Enable the output function of the comparator;
5. Set whether the output passes through the low-pass filter and the output is opposite in phase;
6. If the CMP comparison interrupt vector is used, the interrupt function of the comparator should be enabled;
7. Enable the function of the comparator.

Touch button application initialization:

1. Set the TMB: set the operating mode of the TMB is mode 0, set the counting-trigger source is CMPO,
2. Set the TMB operating clock and overflow;
3. Set the CMP operating mode to be low-power or normal operating mode.

4. Select the input channel, including the positive input channel, negative input channel; the positive input channel is CH1 and the negative input channel is RLO;
5. Set the reference voltage source and the voltage node of the multi-mode resistor; and the resistor short-circuit switch of the resistor;
6. Enable the output function of the comparator;
7. Set whether the output passes through the low-pass filter and the output is opposite in phase;
8. If the CMP comparison interrupt vector is used, the interrupt function of the comparator should be enabled;
9. Enable the function of the comparator;
10. Release the charges of the touch button and the reference capacitor before charging;
11. Disable the non-overlap controller first, and then disable the reference voltage source of the non-overlap controller, and enable the resistor short-circuit switch of the resistor;
12. Enable the input end short-circuit switch; discharge from the reference capacitor of the CH1to ground via the resistor;
13. Set the corresponding IO pin of the touch button as the output mode and the output status is 0 to discharge from the touch button to ground;
14. Then enable the charging function;
15. Disconnect the input end short-circuit switch, disconnect the resistor short-circuit of the resistor and enable the reference voltage source of the resistor;
16. Disable the IO output mode of the touch button;
17. Clear the counter register of the TMB;
18. Enable the non-overlap function and select the touch button to be charged;
19. Read the counting value of the TMB after the charging is finished.

25.2. Register address

CMP Register Address	31	24	23	16	15	8	7	0
CMP Base Address + 0x00 (0x41800)	-		-		MASK1		REG1	
CMP Base Address + 0x04 (0x41804)	MASK2		REG2		MASK3		REG3	
CMP Base Address + 0x08 (0x41808)	MASK4		REG4		MASK5		REG5	

-Reserved

25.3. Register function

25.3.1. CMP register 0

CMP Base Address + 0x00 (0x41800)							
Symbol	CMPCR0 (CMP Control Register 0)						
Bit	[31:17]						[16]
Name	-						CMPO
RW	-						R-0
Bit	[15:08]	[7:5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	-	CPIS	CPOR	CPDF	CMPHS	ENCMP
RW	R0W-0	-	RW-0				

Bit	Name	Description	
Bit[16]	CMPO	The comparison result input transferring out status of the comparator	
		0	Negative input signal > Positive input signal
		1	Positive input signal > Negative input signal
Bit[4]	CPIS	The short-circuit switch control of the comparator	
		0	Short-circuit switch opens. (Open=OFF)
		1	Short-circuit switch closes. (Closed=ON)
Bit[3]	CPOR	The digital output phase control of the comparator	
		0	Normal output
		1	Inversed output
Bit[2]	CPDF	The output low-pass filer enable control of the comparator	
		0	Disable, the output of the comparator does not pass through the 2us low-pass filter.
		1	Enable, the output of the comparator passes through the 2us low-pass filter.
Bit[1]	CMPHS	The high-speed mode enable control of the comparator	
		0	Low power mode
		1	Normal mode
Bit[0]	ENCMP	Comparator function enable control	
		0	Disable (the output status is 0)
		1	Enable

25.3.2. CMP register 1

CMP Base Address + 0x04 (0x41804)					
Symbol	CMPCR1 (CMP Control Register 1)				
Bit	[31:24]	[23:20]		[19:16]	
Name	MASK	CPDM[3:0]		CPDA[3:0]	
RW	R0W-0	RW-0			
Bit	[15:08]	[7:6]	[5:4]	[3:2]	[1:0]
Name	MASK	-	CPPS	-	CPNS
RW	R0W-0	-	RW-0	-	RW-0

Bit	Name	Description	
Bit[23]	CPDM[3]	The CPDA[3] outputs hystersis enable control，and the value of the CPDA[3] is controlled by the CMPO, and keep consistent.	
		0	Disable
		1	Enable, CPDA[3]=CMPO
Bit[22]	CPDM[2]	The CPDA[2] outputs hystersis enable control，and the value of the CPDA[2] is controlled by the CMPO, and keep consistent.	
		0	Disable
		1	Enable, CPDA[2]=CMPO
Bit[21]	CPDM[1]	The CPDA[1] outputs hystersis enable control，and the value of the CPDA[1] is controlled by the CMPO, and keep consistent.	
		0	Disable
		1	Enable, CPDA[1]=CMPO
Bit[20]	CPDM[0]	The CPDA[0] outputs hystersis enable control，and the value of the CPDA[0] is controlled by the CMPO, and keep consistent.	
		0	Disable
		1	Enable, CPDA[0]=CMPO
Bit[19~16]	CPDA	Voltage division node configuration of the built-in multi-node resistor of the comparator	
		0000	0
		0001	1/16 (CPRLH – CPRLL)
		0010	2/16 (CPRLH – CPRLL)
		0011	3/16 (CPRLH – CPRLL)
		0100	4/16 (CPRLH – CPRLL)
		0101	5/16 (CPRLH – CPRLL)

		0110	6/16 (CPRLH – CPRLL)
		0111	7/16 (CPRLH – CPRLL)
		1000	8/16 (CPRLH – CPRLL)
		1001	9/16 (CPRLH – CPRLL)
		1010	10/16 (CPRLH – CPRLL)
		1011	11/16 (CPRLH – CPRLL)
		1100	12/16 (CPRLH – CPRLL)
		1101	13/16 (CPRLH – CPRLL)
		1110	14/16 (CPRLH – CPRLL)
		1111	15/16 (CPRLH – CPRLL)
Bit[5~4]	CPPS	The positive input end selection of the comparator	
		00	CH1
		01	CH2
		10	CH3
		11	V12
Bit[1~0]	CPNS	The negative input end selection of the comparator	
		00	CH1
		01	CH2
		10	CH3
		11	RLO

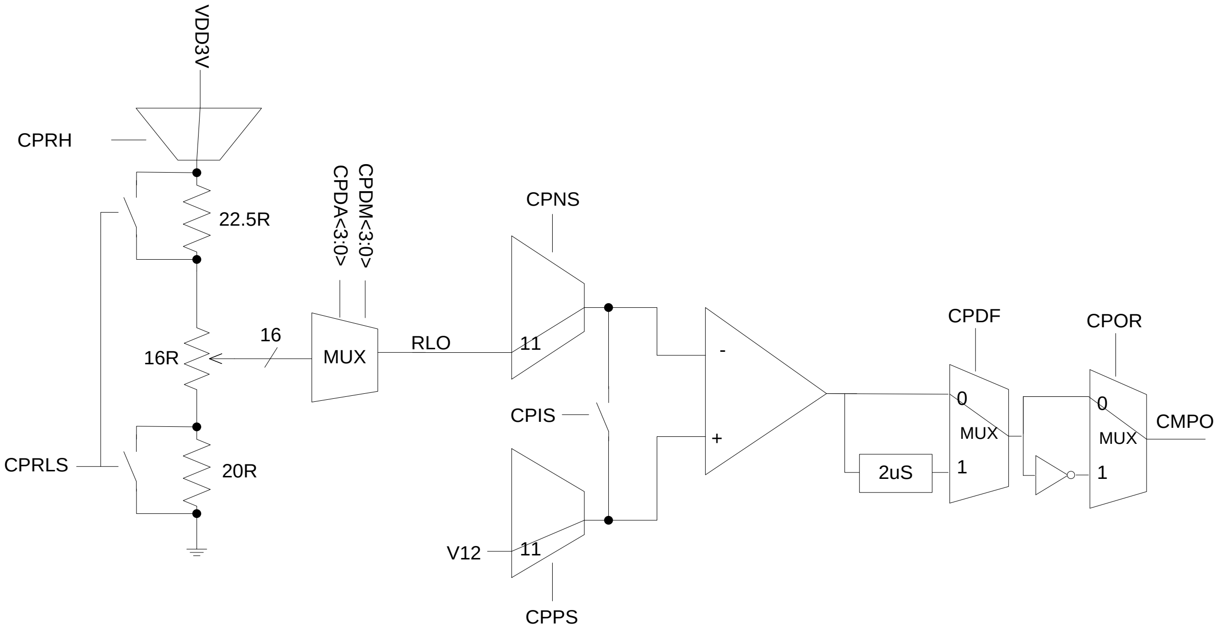
25.3.3. CMP register 2

CMP Base Address + 0x08 (0x41808)					
Symbol	CMPCR2 (CMP Control Register 2)				
Bit	[31:24]	[23:20]		[19:17]	[16]
Name	MASK	-		CPCLS	ENCLIN
RW	R0W-0			RW-0	
Bit	[15:08]	[7:5]	[4]	[3:2]	[1:0]
Name	MASK	-	CPRL	-	CPRH
RW	R0W-0	-	RW-0	-	RW-0

Bit	Name	Description	
Bit[19~17]	CPCLS	The positive input source selection of the touch button function of the comparator	
		000	CL1
		001	CL2
		010	CL3
		011	CL4
		100	CL5
		101	CL6
		110	CL7
		111	CL8
Bit[16]	ENCLIN	The built-in non-overlap function enable control of the comparator. The clock source of the non-overlap controller is TBCLK.	
		0	Disable
		1	Enable, the TBCLK is used as the driving clock source of the non-overlap controller.
Bit[4]	CPRL	The low step short-circuit switch control of the built-in step resistor of the comparator	
		0	The short-circuit switch opens, the low step resistor is not short-circuited.
		1	The short-circuit switch closes, the low step resistor is short-circuited.
Bit[2~0]	CPRH	The voltage source selection of the built-in step resistor of the comparator	
		000	Disable, no voltage supply, under high impedance
		001	CP_I , consistent with the input voltage source of the charge pump
		010	VDD3V chip operating voltage source
		100	VDD18 (1.8V voltage source from the internal LDO voltage regulator of the chip)

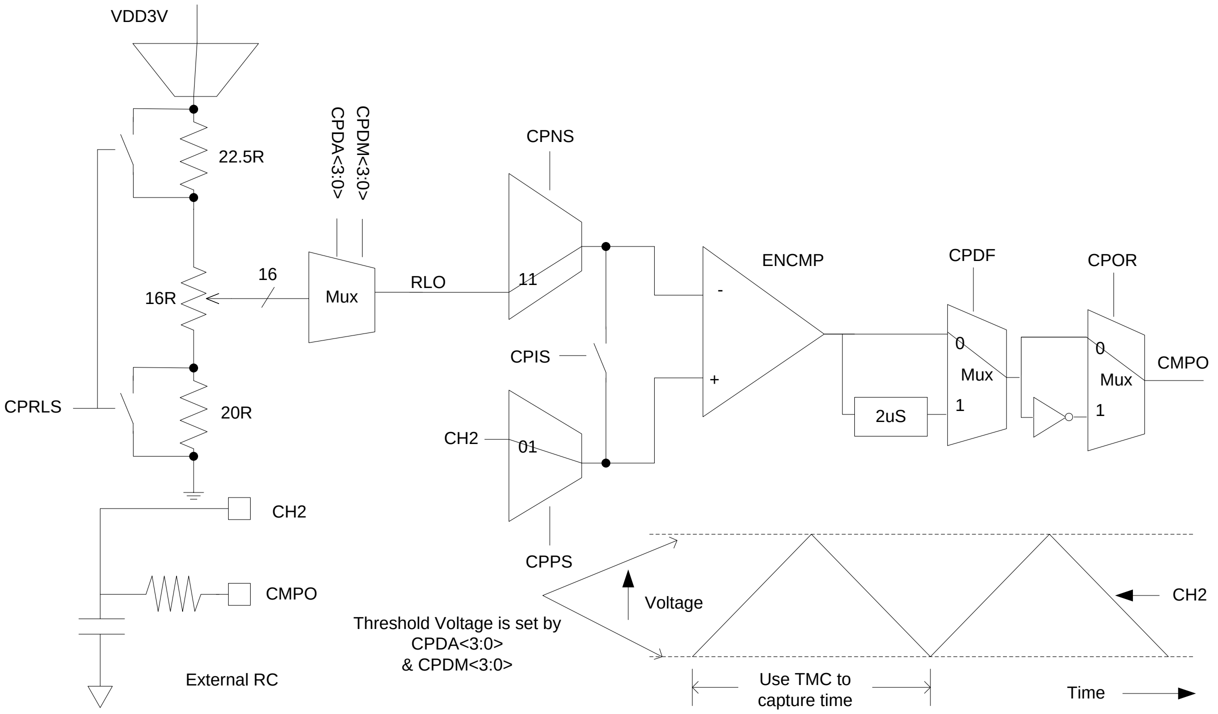
25.4. System example application circuit I

25.4.1. CMP serves as low voltage detector.



25.4.2. System example application circuit II

CMP is used to measure capacitors.



26. SERIAL PERIPHERAL INTERFACE (SPI)

26.1. Overall description

The HY16F19xB has a serial peripheral interface (SPI).

The SPI uses the synchronous serial data communication protocol, and works under the full-duplex mode.

It communicates with the 4-wire bidirectional interface and can work under the master/slave mode.

Under the master mode, it has several configurations to execute different client devices.

Functions:

Full-duplex synchronous transmission.

Support master mode operation or slave mode operation.

Support transmitting MSB first or transmitting LSB first.

The transmission frame is 4~32-bit and can provide programmable bit length setting.

High-speed SPI bus busy-status flag.

Programmable clock pulse rate.

Support high/low potential slave end selection.

Programmable clock polarity and phase

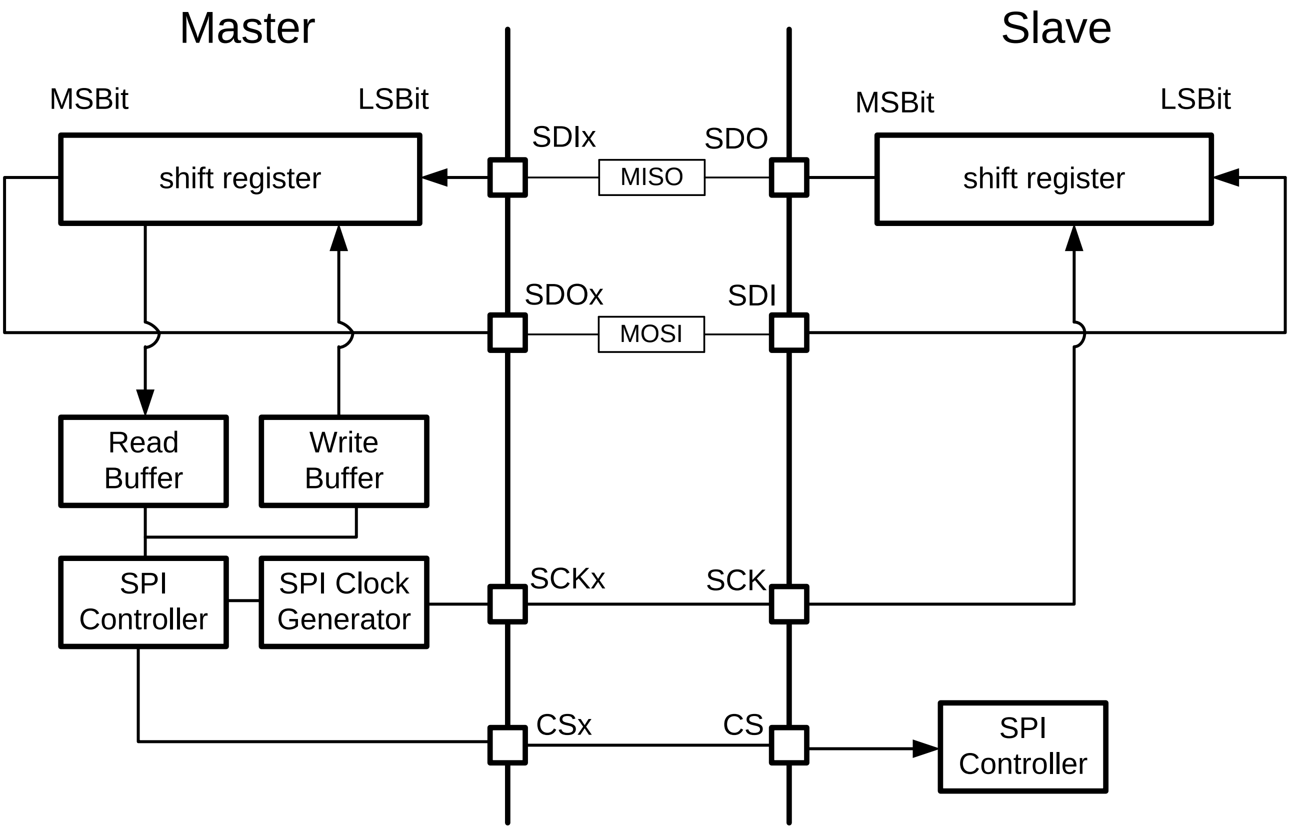


FIG. 26-1 Serial communication SPI structure diagram

The MISO pins are the input of the master device and the output of the slave device. The MOSI pins are the output of the master device and the input of the slave device. The SCK pin is from the serial communication clock output of the master device. The CS pin is from the chip selection of the master device to enable the SPI communication of the slave device. The MOSI/MISO/SCK/CS pins of the master device or the salve device are connected together to execute tasks.

The communication is always enabled by the master device. The master device transmits data to the slave device via the MOSI pins, and the slave device replies to the master device via the MISO pins. So, that is full duplex communication; the data input and output synchronously and use the same clock source.

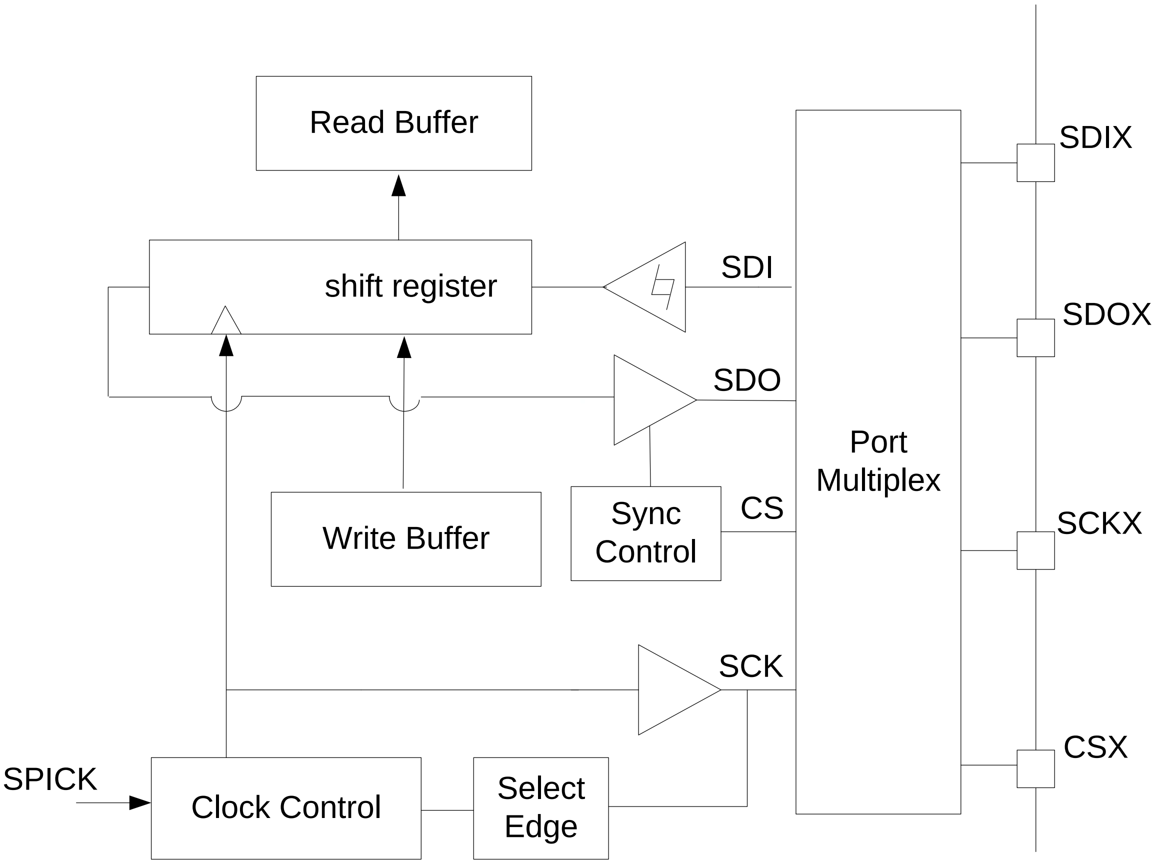


FIG. 26-2 SPI IO pin diagram

Function description: I/O pin setting:

The SPI pins can be programmed for different I/O pins.

Clock phase and clock polarity:

Four different clock types can be formed by software, and controlled by the CPOL and CPHA registers.

The CPOL (clock polarity) is to control the stable status value of the clock without any data transmission.

It can be used in the master mode and the slave mode. If the CPOL is 1(high potential), the SCK is 1 when the SPI is under the idle mode. On the other hand, if the CPOL is 0(low potential), the SCK is 0 when the SPI is under the idle mode (low potential).

The CPHA (clock phase) controls the capturing of the data clock edge of the SCK. If the CPHA is 1(high potential), the second clock edge of the SCK pin (If the CPOL is 1, it is the rising edge; if the CPOL is 0, it is the falling edge.) will capture the data of the MSB.

The data will be locked at the second clock edge of the SCK. On the other hand, if the CPHA is 0 (low potential), the first clock edge of the SCK pin (If the CPOL is 1, it is the falling edge; if the CPOL is 0, it is the rising edge.) will capture the data of the MSB.

The data will be locked at the first clock edge of the SCK. Therefore, the combination of the CPOL and the CPHA can control the data capturing and outputs of the clock edges.

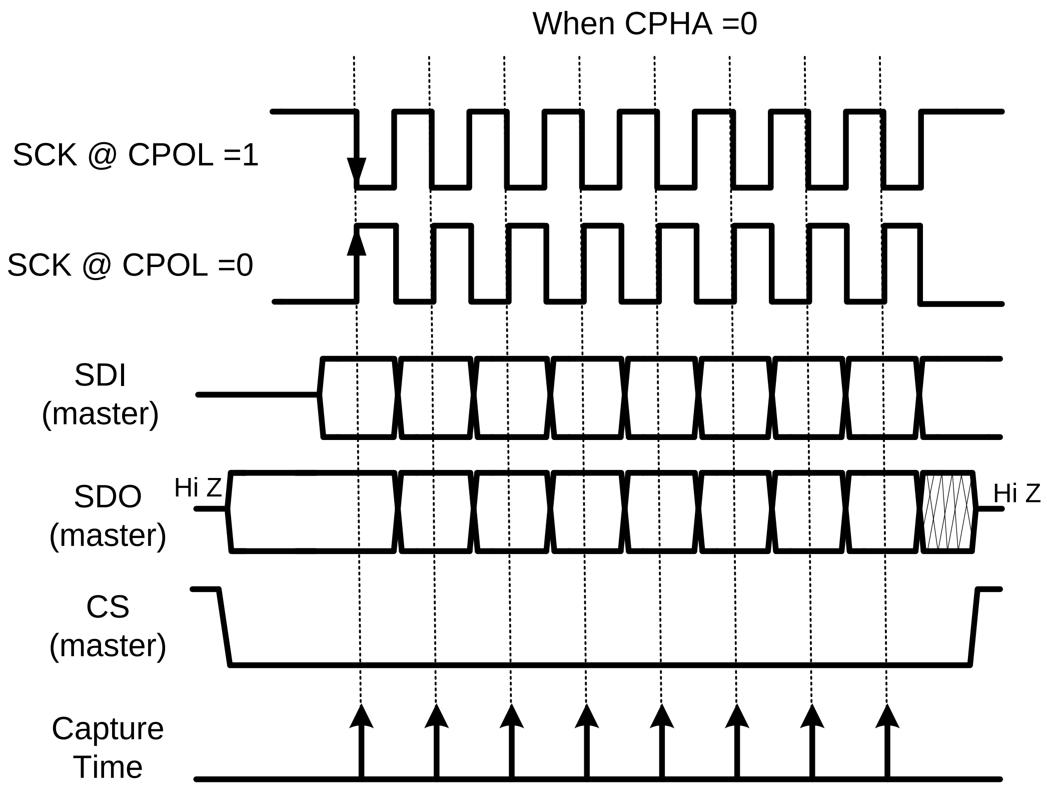


FIG. 26-3 SPI active mode clock diagram (CPHA=0)

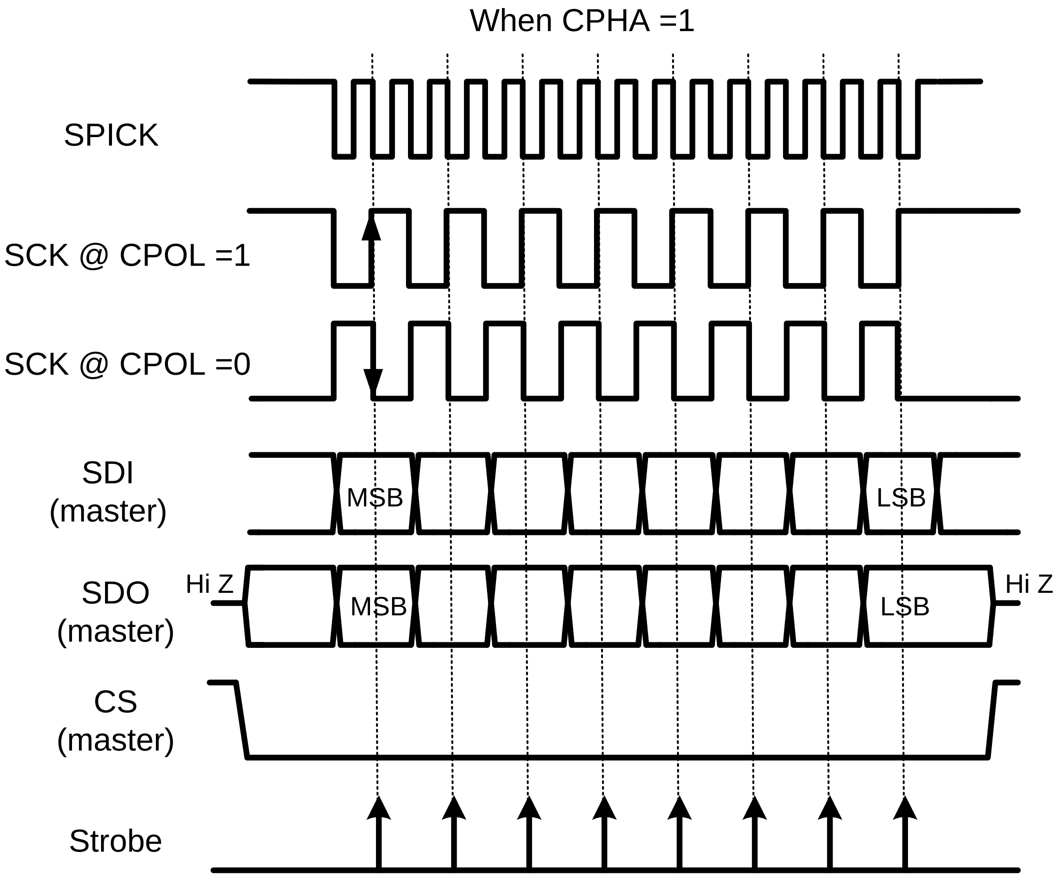


FIG. 26-4 SPI active mode clock diagram (CPHA=1)

Note: SPI Interface when working in Master Mode, SCK operating frequency SPICK / 2.

SPI Control Register 1:

(BL control bit) Data frame format:

The bit length of the transaction word for transmission and reception can be defined in the BL 0x40F04 [4:0]. The lowest bit length is 4 bits, and the highest bit length is 32 bits. The transmission format of the data of the shift register can be to transmit the MSB first or transmit the LSB first, which is defined by the LBF. If the LBF is 0, the data transmission format is to transmit the MSB of the shift register first. Then, the second MSB is transmitted; finally, the LSB is transmitted. If the LBF is 1, the data transmission format is to transmit the LSB of the shift register first.

(CSL control bit) Select the level from the slave device chip:

The CS pin can be defined as 0 or 1 (low potential or high potential) to enable the slave device. That is controlled by the CSL register. If the CSL of the master device is 0, the CS pin will output 0 (low potential) to enable the slave device. On the other hand, if the

CSL is 1, the CS pin will output 1 (high potential) to enable the slave device. If the CSL of the slave device is 0, the slave device will be enabled after receiving the input 0 (low potential) of the CS. On the other hand, if the CSL of the slave device is 1, the slave device will be enabled after receiving the input 1 (high potential) of the CS.

Note: When SPI Interface operates in 4-wire Master mode, CS pin control is a semi-automatic control of the way, For example, when CSL is set to <1>, CS pin will be pulled low, When the SPI Master to write data to the terminal when the SPI Device, CS pin will be automatically pulled to high potential, After the data transfer is complete, will automatically revert to low potential, that is, when Idle Low, Active is High.

(CSO control bit)

This control bit is only 3-wire SPI Slave mode will be used. This pin functions as Chip internal wake-up (CS) signal simulator control. SPI Slave before receiving data first set the CSO = <0> to receive data correctly. When data reception is completed, to read previous RXB Buffer, Must be set CSO = <1>, the received data can be read correctly. After reading the data need to set CSO = <0> ready to receive the next data. When the SPI Slave to return data to SPI Master, Also set CSO = <1>, Then transfers data written TXB Buffer, and then set the CSO = <0>, so that it can transfer data to the Master.

Note: When using a 3-wire SPI transfer if SPI Slave side has to complete initialization, and set CSO = 0, At this point if SPI Master before doing initialization, SPI Slave will cause the possibility of the first data received by mistake. Recommendation initialization process requires Handshake Protocol, confirming the initialization is complete before starting to make data transmission.

SPI Control Register 0**:**

(OVF control bit)

The OVF is the overflow flag of the SPI. When any additional SCK clock edge is inputted during the transmission period, it will be high potential (1). For example, if the bit length of a work is 16 bits and there are 17 clock pulses from the master device before CS changes to high (in this case, CSL is <0>), and when OVF receives the 17th clock edge, its value is 1. That means that errors occur during the transmission. If the 17th clock edge has occurred, it means that the data transmitted first are lost.

(ABF control bit)

The ABF is the interrupt flag of the SPI, which is only used in the slave mode. During the transmission, when the SCK clock edge inputs are insufficient, it will be high potential

(1). For example, if the bit length of a word is 16 bits, there are 15 clock edges from the master device and the CS is changed to high potential (in this case, the CSL is 0), the ABF is 1. That means errors occur during the transmission. The transaction is not finished and the transmitted data are updated to the read register. The transmission is stopped and lost.

(BUF control bit)

The BUF is the busy flag of the SPI. When the SPI is transmitting or receiving data, it is high potential (1). Under the master mode, when the SPI starts to transmit data, it is high potential (1). Once the SPI stops transmitting data or transmission is finished, it will be cleared automatically. Under the slave mode, when the SPI is ready to communicate with the master device, it is 1. Once the SPI stops transmitting data or transmission is finished, it will be cleared automatically.

SPI Interrupt Flag Control bit:

(1)STxIF: the flag STxIF is the transmission interrupt of the SPI. When the write-in register is loaded into the shift register, it is set as 1.

(2)SRxIF: the flag SRxIF is the reception interrupt of the SPI. When the shift register is loaded into the read register, it is set as 1

26.2. Register address

SPI Register Address	31	24	23	16	15	8	7	0
SPI Base Address + 0x00(0x40F00)	SPIC2M		SPIC2		SPIC1M		SPIC1	
SPI Base Address + 0x04(0x40F04)	SPIC0M		SPIC0		-		BL	
SPI Base Address + 0x08(0x40F08)	RXB3		RXB2		RXB1		RXB0	
SPI Base Address + 0x0C(0x40F0C)	TXB3		TXB2		TXB1		TXB0	

26.3. Register function

26.3.1. SPI register 0

SPI Base Address + 0X00 (0X40F00)									
Symbol	SPICR0 (SPI Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	-	RxF	OVF	ABF	BUF	DCF	TxBF	RxBF
RW	R0W-0	-	R-0	RW0-0		R-0			
Bit	[15:08]	[07:04]				[03]	[02]	[01]	[00]
Name	MASK	-				CPHA	CPOL	M/S	En
RW	R0W-0	-				RW-0			

Bit	Name	Description	
Bit[22]	RxF	Reception (Rx) register update flag	
		0	Normal
		1	The reception (RX) register is updated; the reception register cannot be read now.
Bit[21]	OVF	SPI bus data over-length flag	
		0	Normal
		1	The length of the received data length is higher than the set data length BL[4:0]; writing in 0 can clear the OVF flag.
Bit[20]	ABF	SPI bus data insufficient-length flag	
		0	Normal
		1	The length of the received data length is lower than the set data length BL[4:0]; writing in 0 can clear the ABF flag.
Bit[19]	BUF	SPI bus busy flag	
		0	SPI bus interface Idle Standby
		1	SPI bus interface busy status
Bit[18]	DCF	Data lost flag	
		0	Normal
		1	The reception register is full but still keeps receiving data; the old data will be lost and reading the reception register can clear the bit.
Bit[17]	TxBF	TX transmission register full flag	
		0	TX transmission register is empty and can transmit data.
		1	TX transmission register is full and keeping writing data in the register will overwrite old data.
Bit[16]	RxBF	Rx reception register full flag	
		0	RX reception register is empty.
		1	RX reception register is full (reading the reception can clear the bit.)
Bit[3]	CPHA	Clock phase configuration for the SPI bus capturing data	

		0	Capture data at the first clock edge of the SCK.
		1	Capture data at the second clock edge of the SCK.
Bit[2]	CPOL	SPI bus operating frequency polarity control	
		0	SCK low potential is idle.
		1	SCK high potential is idle.
Bit[1]	M/S	SPI operating mode configuration	
		0	Passive mode
		1	Active mode
Bit[0]	EN	SPI function enable control	
		0	Disable
		1	Enable

26.3.2. SPI register 1

SPI Base Address + 0X04 (0X40F04)						
Symbol	SPI CR1(SPI Control Register 1)					
Bit	[31:24]	[23:21]	[20]	[19]	[18]	[17:16]
Name	MASK	-	CSO	CSL	LBF	MD
RW	R0W-0	-	RW-0			
Bit	[15:05]	[04:00]				
Name	-	BL				
RW	-	RW-0				

Bit	Name	Description	
Bit[20]	CSO	Chip internal wake-up (CS) signal simulator control, applicable to the 3-wire mode	
		0	CS signal simulator works.
		1	CS signal simulator stands by.
Bit[19]	CSL	CS signal polarity configuration, for enabling devices, Suitable for 4-wire master end and from the end mode	
		0	Low-potential enablement
		1	High-potential enablement
Bit[18]	LBF	Data transmission order	
		0	Transmit MSB first
		1	Transmit LSB first
Bit[17-16]	MD	SPI interface operating mode configuration	
		00	SPI standard 4-wire communication interface mode
		01	SPI universal 3-wire interface mode
		10	TI mode
		11	TI mode

Bit[4~0]	SPIBL	SPI signal word length transmission configuration	
		00000	8 bits length
		00001	16 bits length
		00010	24 bits length
		00011	4 bits length
		00100	5 bits length
		00101	6 bits length
		00110	7 bits length
		00111	8 bits length
		01000	9 bits length
		01001	10 bits length
		01010	11 bits length
		01011	12 bits length
		01100	13 bits length
		01101	14 bits length
		01110	15 bits length
		01111	16 bits length
		10000	17 bits length
		10001	18 bits length
		10010	19 bits length
		10011	20 bits length
		10100	21 bits length
		10101	22 bits length
		10110	23 bits length
		10111	24 bits length
		11000	27 bits length
		11001	26 bits length
		11010	27 bits length
		11011	28 bits length
		11100	29 bits length
		11101	30 bits length
		11110	31 bits length
		11111	32 bits length

When the MD is set as the 3-wire mode, the original CS pin will become the GPIO mode.

26.3.3. SPI register 2

SPI Base Address + 0x08 (0x40F08)	
Symbol	SPICR2 (SPI Control Register2)
Bit	[31:16]
Name	RXB31-16
RW	R-X
Bit	[15:0]
Name	RXB15-0
RW	RW-X

Bit	Name	Description
Bit[31~0]	SPIRB	SPIRB[31:0] is the 32-bit reception register.

Use the LBF bit to set whether the LSB or MSB is transmitted first.

If the LSB is set to be transmitted first, the position where the data are stored will be influenced, and the RXB effective data will be right-justified.

For example, if the BL is set to be under the 8-bit mode, the received data will be stored at the RXB [7:0]; if the BL is set to be under the 9-bit mode, the received data will be stored at the RXB [8:0], and so on.

If the MSB is set to be transmitted first, the RXB effective data will be left-justified.

For example, if the BL is set to be under the 8-bit mode, the received data will be stored at the RXB [31:24]; if the BL is set to be under the 9-bit mode, the received data will be stored at the RXB [31:23], and so on.

26.3.4. SPI register 3

SPI Base Address + 0x0C (0x40F0C)	
Symbol	SPICR3 (SPI Control Register 3)
Bit	[31:16]
Name	TXB31-16
RW	R-X
Bit	[15:0]
Name	TXB15-0
RW	RW-X

Bit	Name	Description
Bit[31~0]	SPITB	SPITB [31:0] is the 32-bit transmission register.

Use the LBF bit to set whether the LSB or MSB is transmitted first.

If the LSB is set to be transmitted first, the position where the data are stored will be influenced, and the TXB effective data will be right-justified.

For example, if the BL is set to be under the 8-bit mode, the received data will be stored at the TXB [7:0]; if the BL is set to be under the 9-bit mode, the received data will be stored at the TXB [8:0], and so on.

If the MSB is set to be transmitted first, the TXB effective data will be left-justified.

For example, if the BL is set to be under the 8-bit mode, the received data will be stored at the TXB [31:24]; if the BL is set to be under the 9-bit mode, the received data will be stored at the TXB [31:23], and so on.

27. ENHANCED UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMIT (EUART)

27.1. Overall description

For HY16F19xB, there are two groups of asynchronous serial communications-UART and UART2.Those are enhanced EUART (Enhanced Universal Asynchronous Receiver Transmit) The chip has an embedded enhanced universal asynchronous receiver transmit (EUART) management. The peripheral devices of the EUART is so called serial communication interface (SCI). The EUART can be set as the full-duplex asynchronous system, and its peripheral communication devices, including ADC or DAC integrated circuit, serial EEPROM/Flash, etc. The EUART has extra features; including the data frame error detection and automatic address identification. The data frame error detection can determine whether a data frame is valid or passes the frame stop bit. The automatic address identification can compare the address frame content with the single chip address; and the serial interrupt can only be generated when both of them are conformed to each other.

Baud Rate Generator, BRG

Register 0x40E08 [15: 0] is a 16 bit baud rate generator, Support for asynchronous mode EAURT. The following table shows the calculation formula tandem transfer rate. But only apply in Master Mode. And at a given target serial transmission rate and the operating frequency of OSC HAO case, we can calculate the approximate integer values Baud Rate using the formula in the following table, which can determine the serial transmission serial transmission rate error. It recommended after switching the operating frequency. Need to reset serial transmission rate or Use automatic transmission speed serial function, Recalibrate Baud Rate value.

Baud Rate / EUART MODE	serial transmission rate the calculation formula
16 bit / asynchronous	OSC HAO÷[4x(n+1)]
OSC HAO=CPU HAO Operating Frequency; n= 0x40E08 Register value；	

For example: In the asynchronous mode, the operating frequency of OSC HAO (assumed to 4MHz), and the target serial transmission rate of 9600bps, calculate the value of Baud Rate.

Baud Rate= ((OSC_HAO÷Target serial transmission rate) ÷4)-1
= ((4000000÷9600) ÷4)-1
= 103.1667
≈103

According to the above calculation Baud Rate:

Baud Rate = 4000000 ÷ (4x (103 + 1)) = 9615.38; so there is a certain error, the error are calculated as follows:

Error rate = (the actual calculation Baud Rate - the target Baud Rate) / the target Baud Rate
= (9615-9600) / 9600
= 0.16%

Serial automatic transmission rate function (Auto Baud rate detection)

UART modules support automatic detection and corrected serial transmission rate function, it is known as serial automatic transmission rate function. Automatic serial transmission rate must RxEn control bit = 1b and RxABDEn = 1b only valid. After receiving a start condition, that started serial automatic transmission rate detection function (Receiving data need to 0x55), after the automatic detection and correction is completed, the results will be written to Register 0x40E08 [15:0].

UART Auto-Baud rate automatic transmission rate setting process:

1. UART initialization settings: Includes UART TX, RX Port settings. TX and RX correspond to the GPIO pin needs to be set to the corresponding TX and RX Output to Input.
2. Auto Baud rate initialization settings: pre-cleared 0x40E08 [15: 0] register, Close the RX GPIO Input settings, wait RX IRQ (URxIF) generated the interrupt flag, when after receiving RX IRQ (URxIF),Re-set the RX correspond to the GPIO port for the Input. After setting, again cleared UART state flag register and clear the UART RX Data Buffer and RX IRQ (URxIF), to complete the Auto Baud rate initialization settings.
3. Set Auto-baud Enable and Detection: Open Auto baud rate function RxABDEn = 1b, and wait for the 0x55.After receiving 0x55, register 0x40E08 [15: 0] will automatically fill in the target transfer rate, Completion of Auto-baud rate settings. Final recommendations can be done after the Auto-baud rate, increasing the Hand shark process; the purpose is to confirm the auto-baud rate was correct.

UART Auto-baud rate reference app to PT1.5 = RX as an example:

1. Configuration before Auto Baud Rate:

```
ur_08 = 0x0;                          // Clear BRG
pio1_2 = 0x20000000;                  // Disable corresponding GPIO Input PIN of Rx
while(!(int_00 & 0x4));               // Wait for RX IRQ
pio1_2 = 0x20200000;                  // Enable corresponding GPIO Input PIN of Rx
ur_00 = 0xff000000;                   // clear UART status Flags
Delay(1000);                          // Delay about 1000 NOP. OR less
ur_0c;                                // Read UART data buffer
int_00 = 0x00000400;                  // And then, clear UART Rx Interrupt flag
```

2. Auto-Baud rate Enable and Detection :

```
ur_04 =0xff08;                        // Enable Auto-Baud rate detection
while(1)             // break only when Auto Baud Rate and Handshake Complete
{
   while(!(int_00 & 0x4))               // Wait for RX IRQ
   ur_0c;                              // Read UART data buffer
   int_00 = 0x00000400;                   // And then, clear UART Rx Interrupt flag
   if(Handshake())                       // If Handshake process(user defined) was completed
   {
       break;
   }
   ur_04 =0xff08;                        // Retry and Enable Auto-Baud rate detection
}
```

Explanation:

ur_00 to register 0x40e00
ur_04 to register 0x40e04
ur_08 to register 0x40e08
ur_0c to register 0x40e0c
pio1_2 to register 0x40804
int_00 to register 0x40000

Communication IO pins

The EUART communication bus only uses two wires, TX/RX; the chip allocates 8 sets of communication IO pins (each set includes the TX/RX wires) for the EUART module for users to perform designs freely. But this is the IO port multiplexing function, through the GPIO Alternate Function Controller 0x40844's control bits PTUR and PTURE However, the reuse functions of the IO port can be used to conveniently select and enable the communication IO pins of the EUART via the controller PTUR [2:0]/PTURE; accordingly, when using the EUART, the IO communication pins should be enabled, and the corresponding IO pins should be set as the input mode or output mode. The distribution of the EUART communication IO pins is as shown in the following table.

UART

PTUR[2:0]	PTURE	TX	RX	PTUR[2:0]	PTURE	TX	RX
000	1	PT1.0	PT1.1	100	1	PT8.0	PT8.1
001	1	PT1.4	PT1.5	101	1	PT8.4	PT8.5
010	1	PT2.0	PT2.1	110	1	PT9.0	PT9.1
011	1	PT2.4	PT2.5	111	1	PT9.4	PT9.5

UART2

PTUR2[2:0]	PTURE	TX2	RX2	PTUR2[2:0]	PTURE	TX2	RX2
000	1	PT1.2	PT1.3	100	1	PT8.2	PT8.3
001	1	PT1.6	PT1.7	101	1	PT8.6	PT8.7
010	1	PT2.2	PT2.3	110	1	PT9.2	PT9.3
011	1	PT2.6	PT2.7	111	1	PT9.6	PT9.7

Table 27-1 EUART communication IO pin distribution

27.2. Register address

<table>
<tr><td>UART Register Address</td><td>31</td><td>24</td><td>23</td><td>16</td><td>15</td><td>8</td><td>7</td><td>0</td></tr>
<tr><td>UART Base Address + 0x00(0x40E00)</td><td colspan="2">Mask0</td><td colspan="2">REG0</td><td colspan="2">Mask1</td><td colspan="2">REG1</td></tr>
<tr><td>UART Base Address + 0x04(0x40E04)</td><td colspan="2">-</td><td colspan="2">-</td><td colspan="2">Mask2</td><td colspan="2">REG2</td></tr>
<tr><td>UART Base Address + 0x08(0x40E08)</td><td colspan="2">-</td><td colspan="2">-</td><td colspan="4">Baud Rate</td></tr>
<tr><td>UART Base Address + 0x0C(0x40E0C)</td><td colspan="2">-</td><td colspan="2">Tx</td><td colspan="2">-</td><td colspan="2">Rx</td></tr>
</table>

-Reserved

27.3. Register function

27.3.1. UART register 0

UART Base Address + 0x00 (0x40E00)									
Symbol	UARTCR0 (UART Control Register 0)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	Mask	OErr	NErr	FErr	PErr	TxBusy	TxBF	RxBusy	RxBF
RW	R0W-0	RW0-0				R-0			
Bit	[15:08]	[7:6]		[5:4]		[3]	[2]	[1]	[0]
Name	MASK	PLen		DLen		RxIT	RxEn	TxIT	TxEn
RW	R0W-0	RW-1		RW-2		RW-0			

Bit	Name	Description	
Bit[23]	OErr	Rx Buffer over run error flag	
		0	Normal
		1	Over run
Bit[22]	OErr	Rx Noise detected flag	
		0	Normal
		1	Noise detected
Bit[21]	FErr	Rx Frame check error flag	
		0	Normal
		1	Frame check error
Bit[20]	PErr	Rx Parity check error	
		0	Normal
		1	Parity check error
Bit[19]	TxBusy	Tx Busy flag	
		0	Idle
		1	Busy
Bit[18]	TxBF	Tx Buffer Full flag	
		0	Empty
		1	Full
Bit[17]	RxBusy	Rx Busy flag	
		0	Idle
		1	Busy
Bit[16]	RxBF	Rx Buffer Full flag	
		0	Empty
		1	Full
Bit[7~6]	PLen	Tx stop length control	
		0	0.5Bit

		1	1Bit	
		2	1.5Bit	
		3	2 Bit	
Bit[5~4]	DLen	Tx/Rx data length		
			Normal Mode	Parity Check Mode
		0	6 Bit Mode	5 Bit Mode
		1	7 Bit Mode	6 Bit Mode
		2	8 Bit Mode	7 Bit Mode
		3	9 Bit Mode	8 Bit Mode
Bit[3]	RxIT	Rx interrupt method selection		
		0	Send out the interrupt when the Rx Data Buffer has data, and the interrupt disappears after the data are read.	
		1	Send out the interrupt after one piece of data is received by the Rx.	
Bit[2]	RxEn	UART Rx control switch		
		0	Disable	
		1	Enable	
Bit[1]	TxIT	Tx interrupt method selection		
		0	Send out the interrupt when the Tx Data Buffer is idle, and the interrupt disappears after the data are written in.	
		1	Sent out the interrupt after one piece of data is transmitted by the Tx.	
Bit[0]	TxEn	UART Tx control switch		
		0	Disable	
		1	Enable	

27.3.2. UART register 1

UART Base Address + 0X04 (0X40E04)							
Symbol	UARTCR1 (UART Control Register 1)						
Bit	[31:16]						
Name	-						
RW	-						
Bit	[15:08]	[07:05]	[04]	[03]	[02]	[01]	[00]
Name	Mask	-	RxABDF	RxABDEn	RxWUEn	PrtEn	PrtODD
RW	R0W-0	-	RW-0				

Bit	Name	Description	
Bit[4]	RxABDF	Automatic baud rate detection switch	
		0	Normal
		1	Error occur
Bit[3]	RxABDEn	Automatic baud rate detection error flag	
		0	Turn off
		1	Turn on
Bit[2]	RxWUEn	Automatic wake-up mode	
		0	Disable
		1	Enable
Bit[1]	PrtEn	Parity check switch	
		0	Disable
		1	Enable
Bit[0]	PrtODD	Select the odd parity check, even parity check	
		0	Even parity check
		1	Odd parity check

27.3.3. UART register 2

UART Base Address + 0X08 (0X40E08)	
Symbol	UARTCR2 (UART Control Register 2)
Bit	[31:16]
Name	-
RW	-
Bit	[15:0]
Name	Baud Rate
RW	RW-X

Bit	Name	Description
Bit[15~0]	Baud Rate	UART baud rate setting

27.3.4. UART register 3

UART Base Address + 0X0C (0X40E0C)		
Symbol	UARTCR3 (UART Control Register 3)	
Bit	[31:25]	[24:16]
Name	-	Tx Data
RW	-	W-X
Bit	[15:9]	[8:0]
Name	-	Rx Data
RW	-	R-X

Bit	Name	Description
Bit[24~16]	Tx Data	Tx Data Buffer
Bit[8~0]	Rx Data	Rx Data Buffer

27.4. UART using instruction:

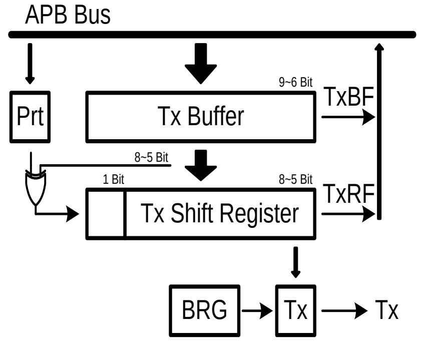
For HY16F19xB, there are two sets UART for users to use, UART and UART2.Use UART serial transmission control process is as follows: First, initialize UART I / O pin setting, the setting section in the initialization of the UART needs to be aware that after you choose TX / RX IO communication pins, you need to turn on IO communication pin action first, and the corresponding IO pins requires GPIO to set as input or output mode. The second point is UART time pulse source selection control: UART time pulse source can choose to use internal oscillator or external oscillator. UART time pulse selection and UART frequency selection determines the speed of transformation. After the above two points have been set to complete the UART transmission protocol, including baud rate settings and transmission bits and other options, the final set in the UART is completed, you need to do a short Delay time setting, this is IO initialization stabilization time, when the IO Initialization to achieve stability, you can do UART enable action to complete the UART initialization action.It is recommended that all the data is received and transmitted at (Interrupt) interrupt events which do treatment, if you are using UART, interrupt processing is done in the INT HW0, if you are using UART2, in INT HW7 do interrupt handling. After completing the initialization of UART and turn on TX and RX UART interrupt enablement, you can start waiting for an interrupt condition fulfilled and do UART serial data transmission.

UART interrupt Description:
The following is URxIF, URxIR, IRxIE use relational instructions.

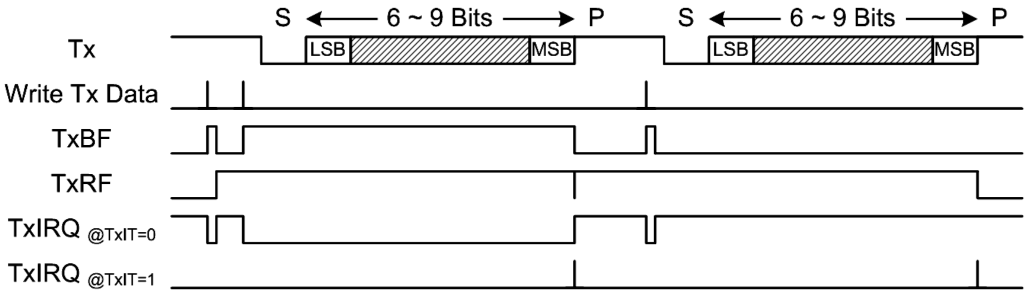
INT Base Address + 0X00 (0X40000)														
Symbol	INTCOM (Interrupt Control Register 0)													
Bit	[31:24]							[23:22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK							-	I2CEIE	I2CIE	UTxIE	**URxIE**	STxIE	SRxIE
RW	R0W-0							-	RW-0					
Bit	[15:14]	[13]	[12]	[11]	[10]	[09]	[08]	[07:06]	[05]	[04]	[03]	[02]	[01]	[00]
Name	MASK							-	I2CEIF	I2CIF	UTxIF	**URxIF**	STxIF	SRxIF
	-	I2CEIR	I2CIR	UTxIR	**URxIR**	STxIR	SRxIR							
RW	R-0							-	RW0-0					

When -URxIE = 0b, UART RX receives an interrupt occurs, URxIR = 0b. URxIF = 1b, but not into the wafer in the interrupt subroutine HW0.
-URxIE = 1b, when the UART RX interrupt reception occurs, URxIR = 1b. URxIF = 1b, the wafer into the interrupt subroutine HW0 in.
- When clearing URxIF = 0b action, while URxIR = 0b.
- Currently the subject of libraries for cleanup actions interrupt flag, is controlled by the operating URxIF.

UART TX Interface Description:



UART Transmit Block Diagram



Action Description:

-TXRF, This is the Tx Shift Register of state.

- When the write data to the TX Data register, TxBF = 1b, on behalf of Tx Buffer is not empty. After the data will be shifted into the Tx shift Register, then Tx Buffer is empty, TxBF = 0b.

- Time when all Tx data has not been sent out, at this time and write data to the TX Data register, the TxBF = 1b, on behalf of Tx Buffer is not empty. Information within Tx shift Register until after all send out, Tx Buffer has shifted to the information within Tx shift Register, the TxBF = 0b.

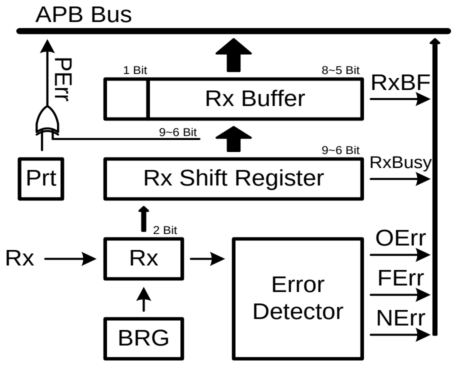
- When the situation under TxBF = 1b, and write data to the TX Data register, then the value will be overwritten with the new information over the past Tx Buffer. Users need to judge, to avoid data being overwritten.

-TxIT Setting will affect the way UTxIF interrupt is generated (Fig described in TxIRQ). When TxIT = 0b, the way it UTxIF generated in HY16F188 products use the same way. In HY16F19xB series product, an increase of TxIT = 1b new feature set.

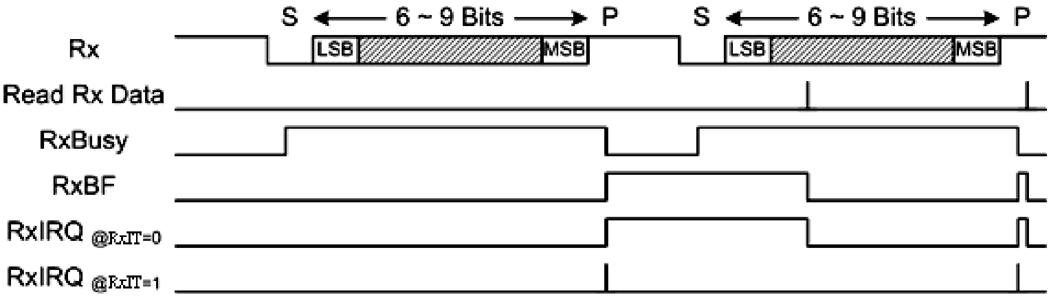
-TxIT = 0b, an interrupt, the interrupt after writing data disappear when the TX Buffer idle; UTxIF TxBF flag and the flag is reversed. Just Tx Buffer is empty, UTxIF = 1b. Therefore, in this state, if the program outset to open UTxIE = 1b, then it will stop entering the interrupt.

- TxIT = 1b, TX when a data transfer completion interrupt is issued; when a data output STOP happen to complete, will generate an interrupt flag UTxIF = 1b. Users can voluntarily remove UTxIF = 0b through instruction. This approach will facilitate the user wants to know when the information after the complete output, interrupt notification occurs.

UART RX Interface Description:



UART Receive Block Diagram



Action Description:

When -UART begins to receive data, when receiving UART CLOCK STAR after half time, RxBusy = 1b, have not received complete information within the RX Buffer, RX Buffer is empty, so RxBF = 0b. When the data reception is complete, the occurrence of STOP, RxBusy = 0b, RX Buffer has information, so RxBF = 1b.

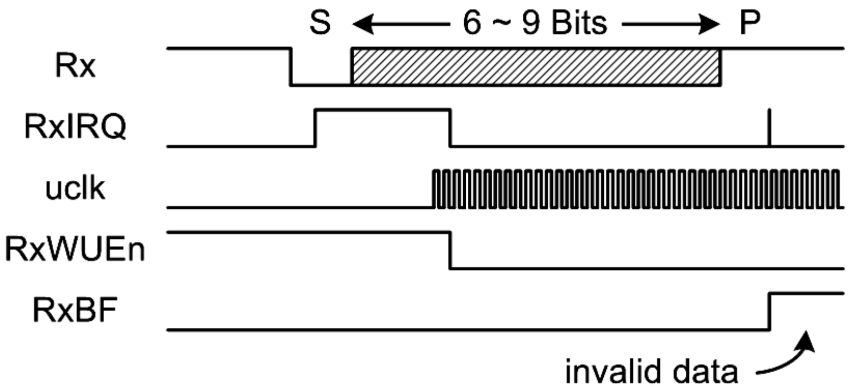
-RxIT Setting will affect the way URxIF interrupt is generated (Fig described in RxIRQ). When RxIT = 0b, the way it URxIF generated in HY16F188 products use the same way. In HY16F19xB series product adds new features set RxIT = 1b.

-RxIT = 0b, an interrupt, the interrupt After reading the information disappears when the RX Buffer Data; when the data reception is complete, RxBusy = 0b, RxBF = 1b, the interrupt flag occurs URxIF = 1b. At this point after reading the Rx Data must register through the action, then the next clear URxIF = 0b action, then we can properly clear the interrupt flag, if no Rx Data register read operation, through the instruction does not clear URxIF state.

-RxIT = 1b, when a data receiving end RX interrupt is issued; when the data reception is complete, RxBusy = 0b, RxBF = 1b, the interrupt flag occurs URxIF = 1b. At this point do not need to read Rx Data register, you can directly through the command to clear URxIF = 0b action.

Use the UART Auto Wake up description:

When the HY16F19xB chip into the power-saving mode (Sleep or Idle Mode), can be designed to use UART's RX pin to do wake-up action. When entering the power-saving mode, RxIRQ received the first data to wake up the chip, the need to avoid using as a related operation. The UART WakeUp setup procedure is described below.



1. UART initialization settings: TX and RX Port settings, TX and RX corresponding to the GPIO pin need to set the corresponding TX for the Output and RX to Input. Note that the RX pin status must be set to the internal Pull High state or the external line to pull the RX pin to the Pull High state.
2. Enable the UART Wake up function by setting the scratchpad 0X40E04 [2] = RxWUEn = 1b and turning on the RX Interrupt and enabling the global interrupt GIE = 1.
3. Set the CPU to enter power saving mode (Sleep or Idle Mode). Note: before entering the power-saving mode, you need to first switch the CPU operating frequency to the internal low-frequency LPO, and then turn off the CPU HAO action, so as to meet the specifications expected to power-saving mode state.
4. Wait for the TX signal from the Host to be sent to the HY16F19xB to wake up the chip. When the HY16F19xB receive the TX signal transmitted by the Host, it will enter the UART interrupt first. After the RxBF Flag, clear the invalid data and the relevant Interrupt Flag, and then re-open the internal HAO high frequency, the CPU operating frequency to switch to HAO, leaving the UART interrupt routine to return to the main program.

Note: The interrupt flag before RxBF is used to wake up the chip, and should be avoided as the related operation. If wake-up from Sleep mode waits at least 64msec (max: <100msec), the chip can begin to operate. Wait for the UART command sent from Host to HY16F19xB to be invalid.

28. ENHANCED UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMIT (EUART2)

28.1. Overall description

Please refer to UART.

28.2. Register address

UART2 Register Address	31	24	23	16	15	8	7	0
UART2 Base Address + 0x00(0x40E10)	Mask0		REG0		Mask1		REG1	
UART2 Base Address + 0x04(0x40E14)	-		-		Mask2		REG2	
UART2 Base Address + 0x08(0x40E18)	-		-		Baud Rate			
UART2 Base Address + 0x0C(0x40E1C)	-		TX2		-		RX2	

-Reserved

28.3. Register function

28.3.1. UART2 register 0

UART2 Base Address + 0x10 (0x40E10)								
Symbol	UART2CR0 (UART2 Control Register 0)							
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	Mask	OErr	NErr	FErr	PErr	TxBusy	TxBF	RxBusy	RxBF
RW	R0W-0	RW0-0				R-0			
Bit	[15:08]	[7:6]		[5:4]		[3]	[2]	[1]	[0]
Name	MASK	PLen		DLen		RxIT	RxEn	TxIT	TxEn
RW	R0W-0	RW-1		RW-2		RW-0			

Bit	Name	Description	
Bit[23]	OErr	Rx Buffer over run error flag	
		0	Normal
		1	Over run
Bit[22]	NErr	Rx Noise detected flag	
		0	Normal
		1	Noise detected
Bit[21]	FErr	Rx Frame check error flag	
		0	Normal
		1	Frame check error
Bit[20]	PErr	Rx Parity check error	
		0	Normal
		1	Parity check error
Bit[19]	TxBusy	Tx Busy flag	

		0	Idle	
		1	Busy	
Bit[18]	TxBF	Tx Buffer Full flag		
		0	Empty	
		1	Full	
Bit[17]	RxBusy	Rx Busy flag		
		0	Idle	
		1	Busy	
Bit[16]	RxBF	Rx Buffer Full flag		
		0	Empty	
		1	Full	
Bit[7~6]	PLen	Tx stop length control		
		0	0.5Bit	
		1	1Bit	
		2	1.5Bit	
		3	2 Bit	
Bit[5~4]	DLen	Tx/Rx data length		
			Normal Mode	Parity Check Mode
		0	6 Bit Mode	5 Bit Mode
		1	7 Bit Mode	6 Bit Mode
		2	8 Bit Mode	7 Bit Mode
		3	9 Bit Mode	8 Bit Mode
Bit[3]	RxIT	Rx interrupt method selection		
		0	Send out the interrupt when the Rx Data Buffer has data, and the interrupt disappears after the data are read.	
		1	Send out the interrupt after one piece of data is received by the Rx.	
Bit[2]	RxEn	UART Rx control switch		
		0	Disable	
		1	Enable	
Bit[1]	TxIT	Tx interrupt method selection		
		0	Send out the interrupt when the TX Data Buffer is idle, and the interrupt disappears after the data are written in.	
		1	Sent out the interrupt after one piece of data is transmitted by the TX.	
Bit[0]	TxEn	UART TX control switch		
		0	Disable	
		1	Enable	

28.3.2. UART2 register 1

UART2 Base Address + 0x14 (0x40E14)							
Symbol	UART2CR1 (UART2 Control Register 1)						
Bit	[31:16]						
Name	-						
RW	-						
Bit	[15:08]	[7:5]	[4]	[3]	[2]	[1]	[0]
Name	Mask	-	RxABDF	RxABDEn	RxWUEn	PrtEn	PrtODD
RW	R0W-0	-	RW-0				

Bit	Name	Description	
Bit[4]	RxABDF	Automatic Baud Rate Detection Error Flag	
		0	Normal
		1	Error occurs
Bit[3]	RxABDEn	Automatic detection of baud rate switch	
		0	Disable
		1	Enable
Bit[2]	RxWUEn	Automatic wake-up mode	
		0	Disable
		1	Enable
Bit[1]	PrtEn	Parity check switch	
		0	Disable
		1	Enable
Bit[0]	PrtODD	Select the odd parity check, even parity check	
		0	Even parity check
		1	Odd parity check

28.3.3. UART2 register 2

UART2 Base Address + 0x18 (0x40E18)	
Symbol	UART2CR2 (UART2 Control Register 2)
Bit	[31:16]
Name	RSV.
RW	R-0
Bit	[15:0]
Name	Baud Rate
RW	RW-X

Bit	Name	Description
Bit[15~0]	Baud Rate	UART baud rate setting

28.3.4. UART2 register 3

UART2 Base Address + 0x1C (0x40E1C)		
Symbol	UART2CR3 (UART2 Control Register 3)	
Bit	[31:25]	[24:16]
Name	-	Tx Data
RW	-	W-X
Bit	[15:9]	[8:0]
Name	-	Rx Data
RW	-	R-X

Bit	Name	Description
Bit[24~16]	Tx Data	Tx Data Buffer
Bit[8~0]	Rx Data	Rx Data Buffer

28.4. UART2 using instruction.

UART2 and UART control position different placese that register with the IO pin configuration different interrupt vector, UART for the INT HW0,UART2 is INT HW7, the same way test the control order, please check UART detailed information.

29. I²C COMMUNICATION INTERFACE

29.1. Overall description

HY16F19xB have one communication interface (I²C), containing the shown (Master) and Slave) Two operating modes are as follows. Host mode can transmit a signal to the I²C Bus I²C packet format combined transfer controller (Transmission Controller, Tx Controller) according to the needs of the system and to determine the transfer rate Clock Generator required. The Slave Controller can receive signals on the I²C Bus to (Slave) mode accepts communications (Master) Host Bus on the demand, combined with a host data transfer controller backhaul needs.

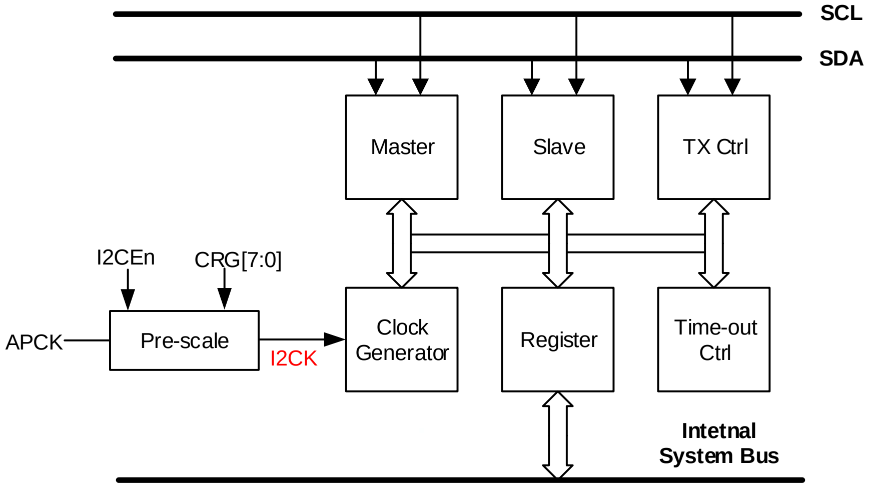


FIG. 29-1 I2C communication structure diagram

29.1.1. Communication I2C interface features:

The standard I²C serial interface includes the SDA and the SCL with two pins.

The pin has the open-type open drain output structure, which needs the external pull-up resistor to ensure the high-level output. The standard I²C serial interface is be set to be under the master mode, slave mode or the master/slave mode. The programmable clock is allowed to adjust the transmission rate of the I²C.

The data are bi-directionally transmitted between the master and the slave.

The I²C allows large operating voltage range. The reference design of the I²C uses a 7-bit length address space but reserves 16 address to deal with a group of buses and the communication between up to 112 (128-16=112) nodes.

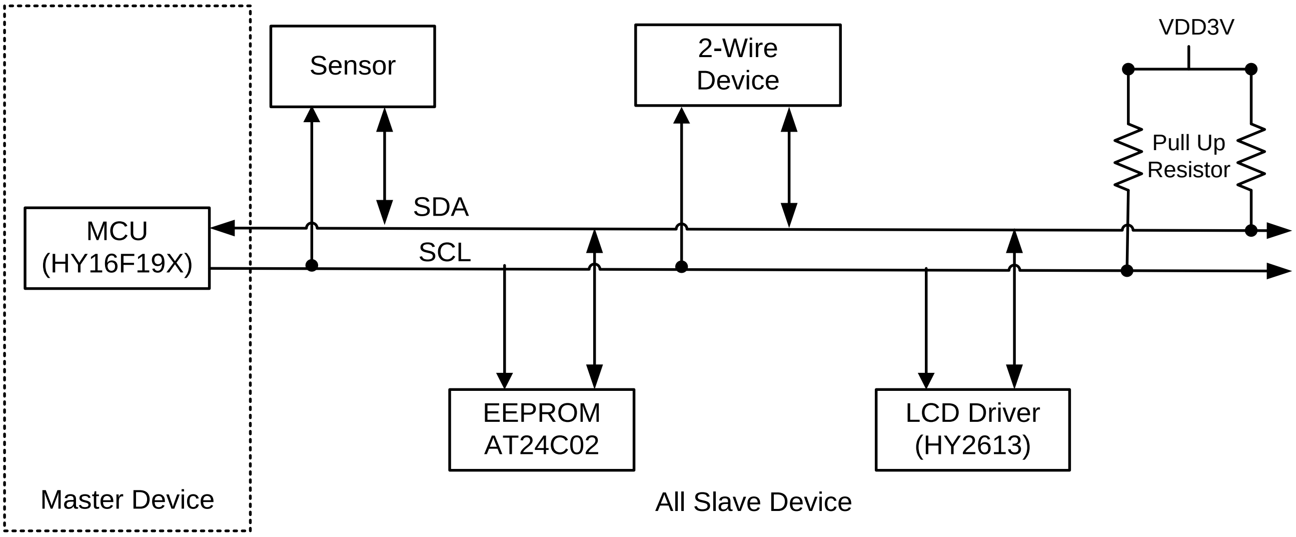


FIG. 29-2 I^{2}C bus device hooking diagram

29.1.2. Communication I2C interface signal

Start signal (START):

Under the master mode, the SCL is high potential. The SDA sent from high potential to low potential to enable the data transmission.

DATA and ADDRESS signals:

I^{2}C serial interface protocol is only needed when the SCL is low potential; The SDA can be changed only according to the data.

Acknowledge signal:

Data reception (Slave) starts right after the initial 8 bits.

Transmitting data to the device (Host) is to send a low potential, which means the data are received.

STOP signal:

Under the master mode, the SCL is high potential. The SDA sent from low potential to high potential to end the data transmission.

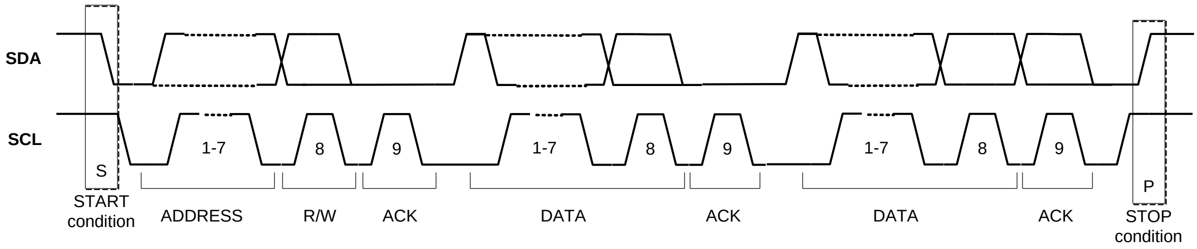


FIG. 29-3 I^{2}C bus clock diagram

Data transmission rate calculation:

I^{2}C internal register CRG [7: 0] can control the main terminal mode data transfer rate. I^{2}C bus SCL connection pin serial data transmission speed depends on the I^{2}C circuit clock source (APCK) and serial transmission rate register CRG [7: 0] value is set, I2C serial data transfer rate can be represented by the following formula to determine:

(I2CK)Data Baud Rate = (APCK) / [4X (CRG + 1)]   (Equation 29-1)

Note:

I^{2}C Master Mode and I^{2}C Slave Mode under, SCL can support a maximum speed of 400kHz.

Time-out function (Time-Out):

Time-out function is to prevent the I^{2}C controller from locking the I^{2}C communication bus when the I^{2}C works in order to provide enough time to deal with the MCU I^{2}C controller. Therefore, the response of the I^{2}C controller to each bit will occur only after the SCL is reduced to low potential; at this time, the master end cannot receive the next clock signal; in other word, a clock stretching takes place. However, when the MCU is too busy or cannot reply to the I^{2}C controller for other reasons, the SCL I^{2}C communication bus may be locked under the low potential.

In order to prevent from the above situation, Time-out controller can be set according to programmers Time-out conditions. The machine detects SCL is pulled Low time when the Time-out is determined to establish, I^{2}C controller will release the SCL and generates an interrupt to the MCU.

Time-out control circuit is I^{2}C clock source (I2CK) to count condition,I2CK to TOPS set of values can be up to 128 Pre-scale, According to TOPS and TOLimit register to determine the Time-out time the machine is set to Low .SCL time. (Referred to herein as SCLo) If has not been reached SCLo in Time-out time is released as High, the Time-out controller internal counter will be reset, and in the next SCLo counted again when pulled Low; If SCLo exceeds the Time-out time is still pulled Low, the Time-out flag (TOFlag) will be set, , and interrupt request MCU processing.

After the Time-out flag is set up, and later in the course of transmission will be in response to NACK I^{2}C Bus. To clear the MCU after the Time-out flag, I^{2}C control circuit to normal use, time-out of the flag must be cleared TOWn off and back on, so Time-out control circuit reply initial state.

I²C communication pin

The I²C bus only has two wires, but the chip allocates 8 sets of communication IO pins for the I²C module (Each set of IO pins includes SCL/SDA), which is for the reuse functions of the IO port. In this way, users can conveniently select different communication pins. The corresponding communication pins can be selected and enabled via the controllers I2CPTS0x40844 [19:17], I2CPTEn 0x40844[16]. When using the functions of the I²C, the communication IO pins should be enabled, and the corresponding IO pin should be set under the input mode or output mode. The following table is the communication pin distribution table.

I2CPTS[2:0]	I2CPTEN	SCL	SDA	I2CPTS[2:0]	I2CPTEN	SCL	SDA
000	1	PT1.0	PT1.1	100	1	PT2.0	PT2.1
001	1	PT1.2	PT1.3	101	1	PT2.2	PT2.3
010	1	PT1.4	PT1.5	110	1	PT2.4	PT2.5
011	1	PT1.6	PT1.7	111	1	PT2.6	PT2.7

Table 29-1 I²C communication IO pin distribution

Note: HY16F19xB portfolio I²C application, initialize GPIO pin function as input or output mode is selectively set, a user on the I²C initialization process can omit this step of the process.

29.1.3. Communication I2C interface flow

I²C serial interface terms

(SPIA): It means Action Register (ACT) giving instructions to the Action control register, where S is the Start instruction, and P is the Stop instruction, I am the interrupt flag and A is the Acknowledge instruction.

SPIA: It means Action Register (ACT) reading the value of the Action control register, which can be used to determine the interrupt flag or other instructions are finished or not.
STA: It means reading the value of the Status register, which is used to show the current operating status of the I²C circuit.

The following flow chart will respectively express the statuses of the I²C interface by (circular frame with gray background), (circular frame with white background) and (white rectangular frame):

● Status with IRQ
○ Status without IRQ
□ Action

Circular frame with gray background: it means the I^{2}C status that the interrupt flag is established.
Circular frame with white background: it means the I^{2}C status that the interrupt flag is not established and needs to be read actively by the MCU.
White rectangular frame: it means the instructions to the I^{2}C should be given by the MCU.

29.1.4. I2C Master TX flow

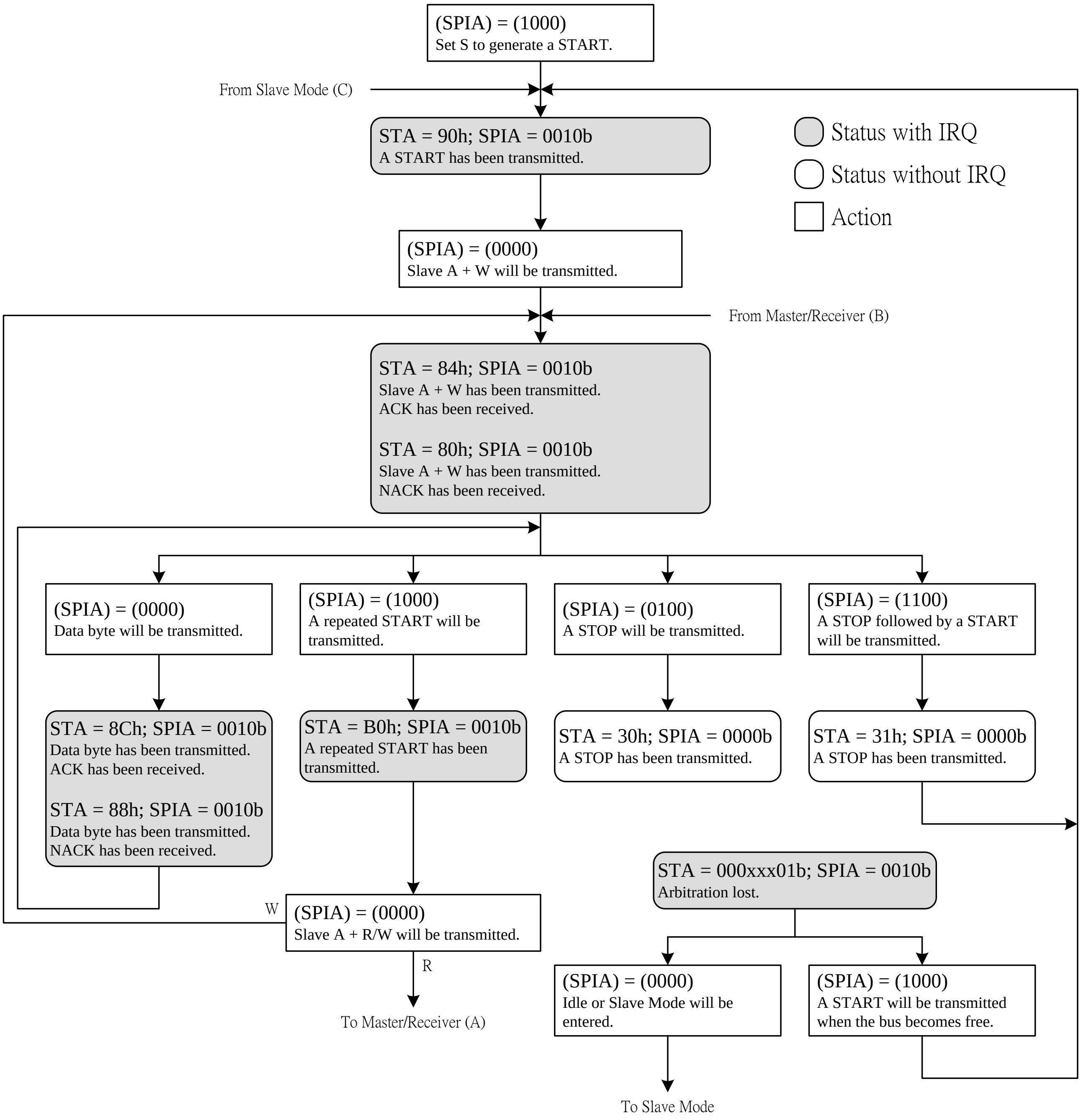

(SPIA) = (1000)
Set S to generate a START.
From Slave Mode (C)
STA = 90h; SPIA = 0010b
A START has been transmitted.
Status with IRQ
Status without IRQ
Action
(SPIA) = (0000)
Slave A + W will be transmitted.
From Master/Receiver (B)
STA = 84h; SPIA = 0010b
Slave A + W has been transmitted.
ACK has been received.
STA = 80h; SPIA = 0010b
Slave A + W has been transmitted.
NACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
(SPIA) = (1000)
A repeated START will be transmitted.
(SPIA) = (0100)
A STOP will be transmitted.
(SPIA) = (1100)
A STOP followed by a START will be transmitted.
STA = 8Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
STA = 88h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = B0h; SPIA = 0010b
A repeated START has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
STA = 31h; SPIA = 0000b
A STOP has been transmitted.
STA = 000xxx01b; SPIA = 0010b
Arbitration lost.
W
(SPIA) = (0000)
Slave A + R/W will be transmitted.
R
To Master/Receiver (A)
(SPIA) = (0000)
Idle or Slave Mode will be entered.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
To Slave Mode

FIG. 29-4 Master Transmitter Mode

29.1.5. I2C Master RX flow

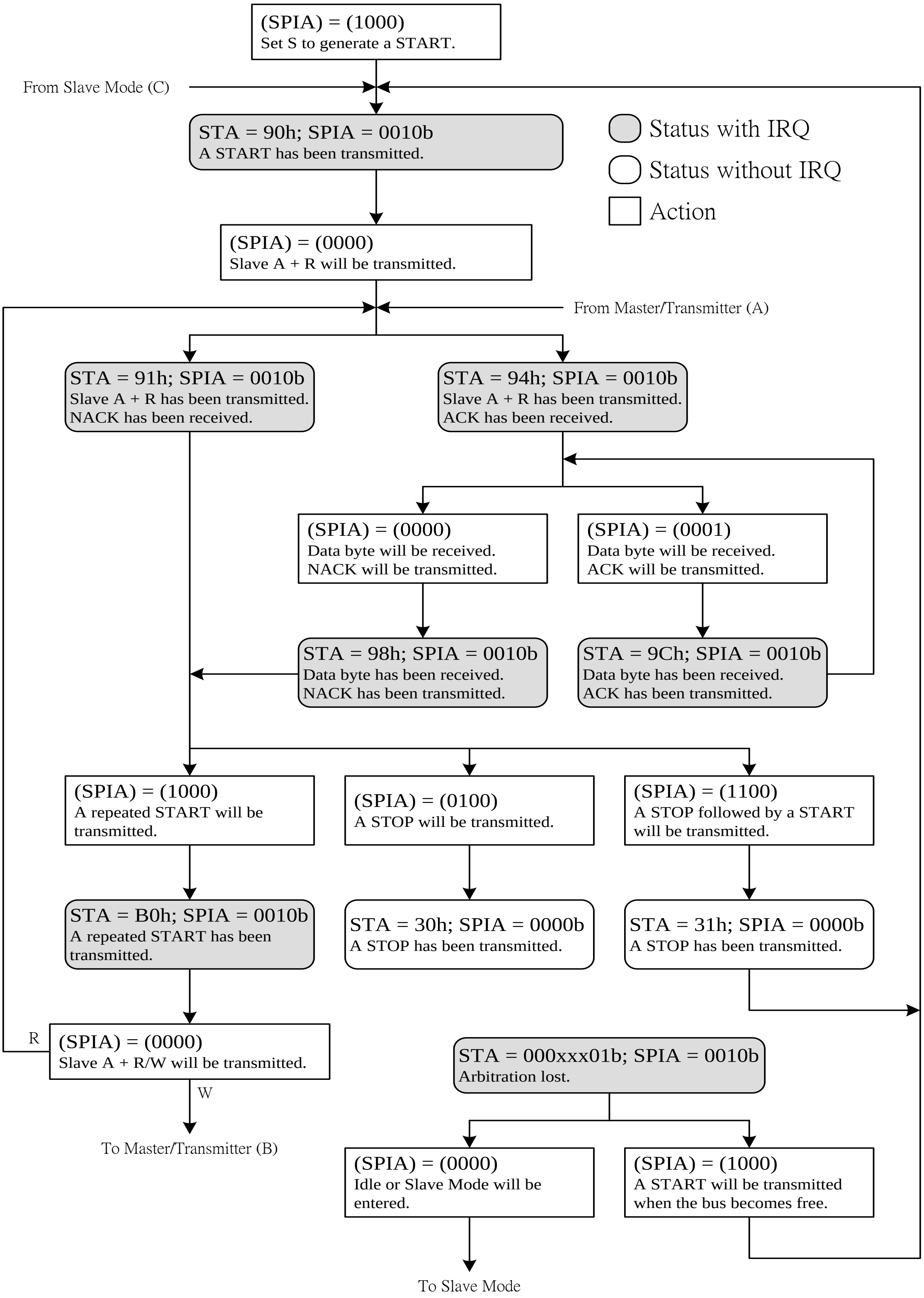

(SPIA) = (1000)
Set S to generate a START.
From Slave Mode (C)
STA = 90h; SPIA = 0010b
A START has been transmitted.
Status with IRQ
Status without IRQ
Action
(SPIA) = (0000)
Slave A + R will be transmitted.
From Master/Transmitter (A)
STA = 91h; SPIA = 0010b
Slave A + R has been transmitted.
NACK has been received.
STA = 94h; SPIA = 0010b
Slave A + R has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 98h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 9Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
(SPIA) = (1000)
A repeated START will be transmitted.
(SPIA) = (0100)
A STOP will be transmitted.
(SPIA) = (1100)
A STOP followed by a START will be transmitted.
STA = B0h; SPIA = 0010b
A repeated START has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
STA = 31h; SPIA = 0000b
A STOP has been transmitted.
R
(SPIA) = (0000)
Slave A + R/W will be transmitted.
W
To Master/Transmitter (B)
STA = 000xxx01b; SPIA = 0010b
Arbitration lost.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
To Slave Mode

FIG. 29-5 Master Receiver Mode

29.1.6. I2C Slaver TX flow

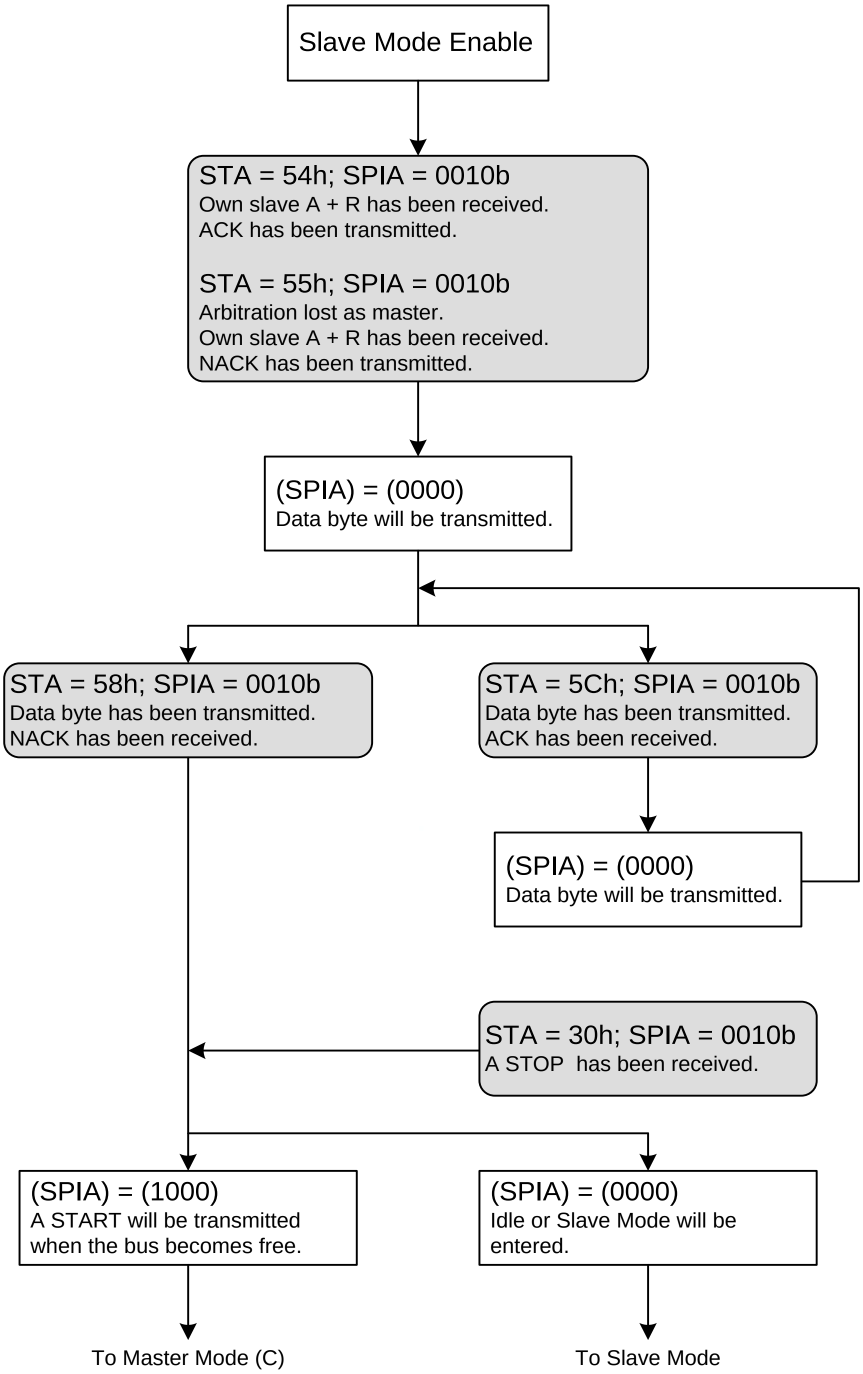

Slave Mode Enable
STA = 54h; SPIA = 0010b
Own slave A + R has been received.
ACK has been transmitted.
STA = 55h; SPIA = 0010b
Arbitration lost as master.
Own slave A + R has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be transmitted.
STA = 58h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = 5Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
STA = 30h; SPIA = 0010b
A STOP has been received.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

FIG. 29-6 Slave Transmitter Mode

29.1.7. I2C Slaver RX flow

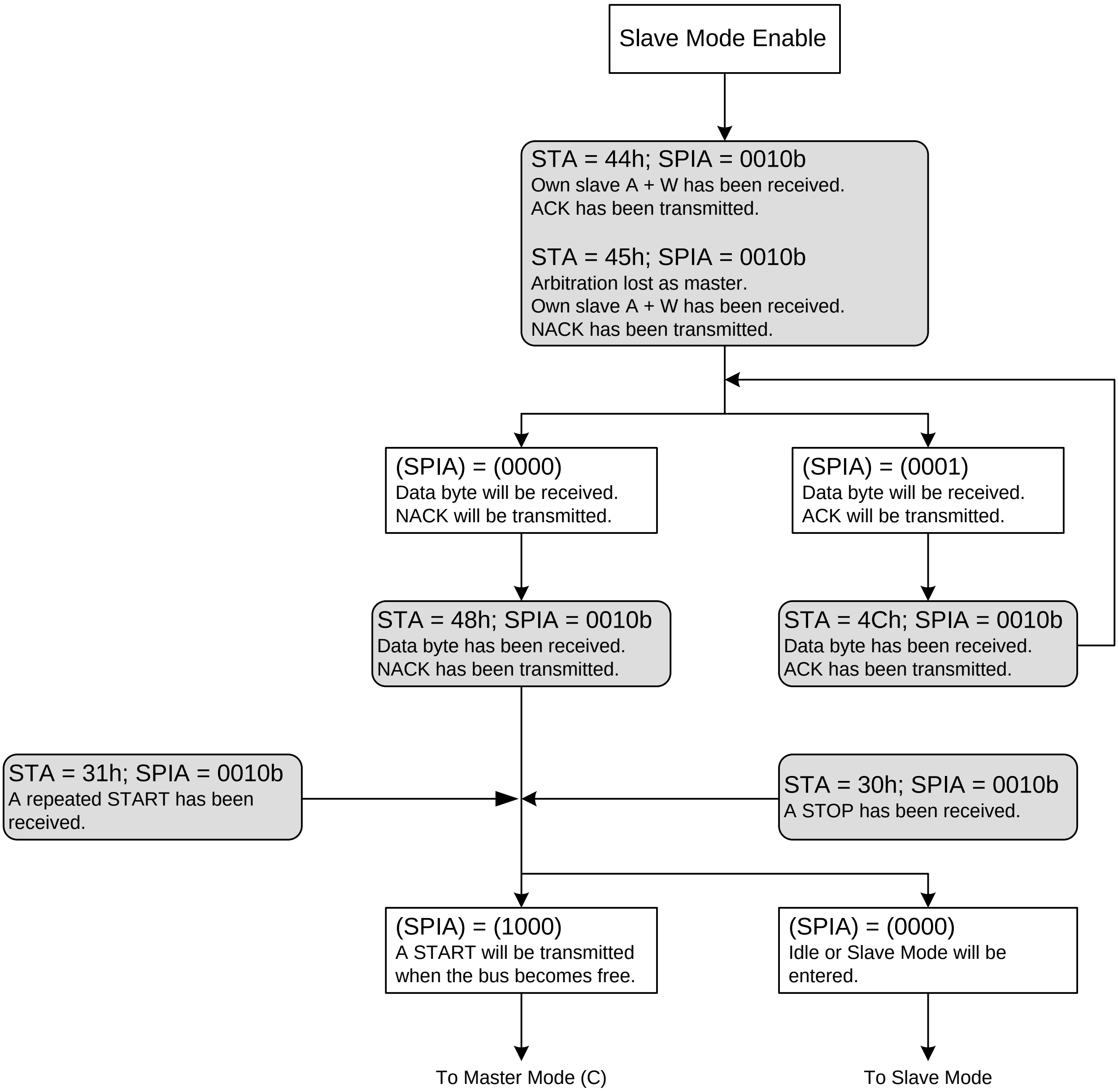

Slave Mode Enable
STA = 44h; SPIA = 0010b
Own slave A + W has been received.
ACK has been transmitted.
STA = 45h; SPIA = 0010b
Arbitration lost as master.
Own slave A + W has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 48h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 31h; SPIA = 0010b
A repeated START has been received.
STA = 30h; SPIA = 0010b
A STOP has been received.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

FIG. 29-7 Slave Receiver Mode

29.1.8. I2C General Call flow

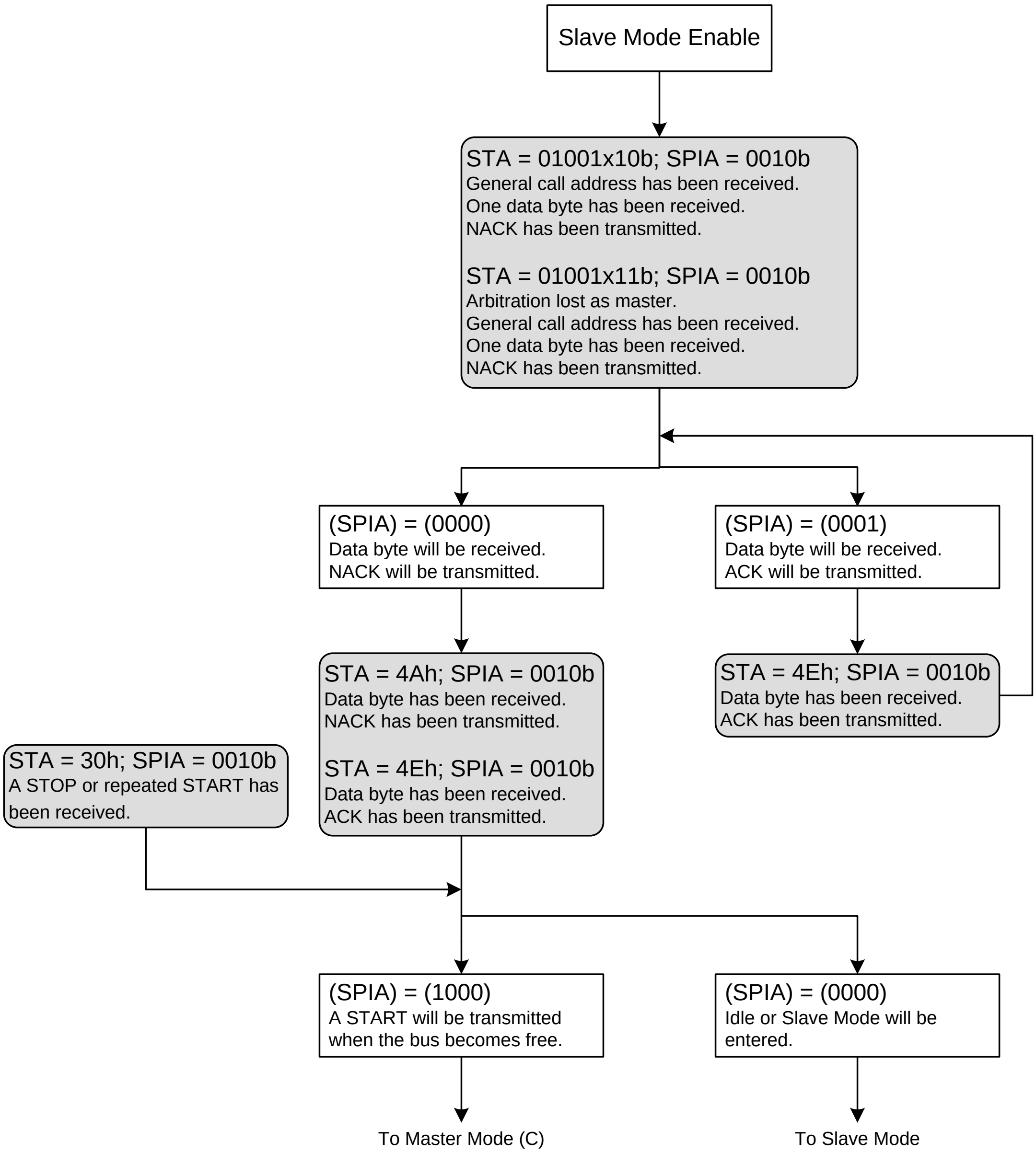

Slave Mode Enable
STA = 01001x10b; SPIA = 0010b
General call address has been received.
One data byte has been received.
NACK has been transmitted.
STA = 01001x11b; SPIA = 0010b
Arbitration lost as master.
General call address has been received.
One data byte has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 4Ah; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Eh; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 4Eh; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 30h; SPIA = 0010b
A STOP or repeated START has been received.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

FIG. 29-8 General Call Mode

29.2. Register address

I2C Register Address	31	24	23	16	15	8	7	0
I^{2}C Base Address + 0X00 (0X41000)	-		-		MASK0		I2C_CON0	
I^{2}C Base Address + 0X04 (0X41004)	MASK1		I2C_CON1		MASK2		I2C_CON2	
I^{2}C Base Address + 0X08 (0X41008)	MASK3		I2C_CON3		MASK4		I2C_CON4	
I^{2}C Base Address + 0X0C (0X4100C)	MASK5		MASK6		I2C_CON5		I2C_CON6	
I^{2}C Base Address + 0X10 (0X41010)	-		-		-		I2C_CON7	
I^{2}C Base Address + 0X14 (0X41014)	-		-		-		I2C_CON8	

- Reserved

29.3. Register function

29.3.1. I2C register 0

I^{2}C Base Address + 0x00 (0x41000)					
Symbol	I2CCR0 (I2C Control Register 0)				
Bit	[31:16]				
Name	RSV.				
RW	R-0				
Bit	[15:8]	[7:3]	[2]	[1]	[0]
Name	MASK	-	GCRst	TOEn	I2CEn
RW	R0W-0	-	RW-0		

Configuration Register (CFG)

Bit	Name	Description	
Bit[02]	GCRst	General calling reset enable control	
		0	Disable
		1	Enable
Bit[01]	TOEn	Time-out reset function enable control	
		0	Disable
		1	Enable
Bit[00]	I2CEn	I2C function enable control	
		0	Disable
		1	Enable

Note: When I2CEn is off, it will shut off the I^{2}C internal clock, in addition to Configuration Register can write, the rest of the register will not be able to write data.

29.3.2. I2C register 1

Action Register (ACT)

I^{2}C Base Address + 0x04 (0x41004)									
Symbol	I2CCR1 (I2C Control Register 1)								
Bit	[31:24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
Name	MASK	MAct	SAct	Rx P/Sr	R/W	DF	A/NA	GC	ARB
RW	R0W-0	R-0							
Bit	[15:08]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name	MASK	SEn	10bEn	3BEn	EIRQFlag	START	STOP	IRQFlag	A/NA
RW	R0W-0	RW-0							

Bit	Name	Description	
Bit[23]	MAct	Master mode enable flag	
		0	Disable
		1	Enable
Bit[22]	SAct	Slave mode enable flag	
		0	Disable
		1	Enable
Bit[21]	Rx P/Sr	Reception stop or restart flag	
		0	Normal
		1	The reception stop or restart flag has been sent or received
Bit[20]	R/W	Read/write status flag	
		0	The writing instruction has been sent or received.
		1	The reading instruction has been sent or received.
Bit[19]	DF	Data flag	
		0	Normal
		1	The I^{2}C data have been sent or received.
Bit[18]	A/NA	Response signal (ACK) status flag	
		0	The response signal (ACK) has yet to be sent or received.
		1	The response signal (ACK) has been sent or received.
Bit[17]	GC	General calling status flag	
		0	Normal
		1	The general calling operation is performed now.
Bit[16]	ARB	Arbitration loss flag	
		0	Normal
		1	Arbitration loss
Bit[07]	SEn	(I^{2}C Slave)Slave mode enable control	
		0	Disable
		1	Enable
Bit[06]	10bEn	(I^{2}C Slave)Slave 10-bit address code mode enable control	
		0	Disable
		1	Enable the 10-bit address code mode
Bit[05]	3BEn	(I^{2}C Slave)Slave sending 3 pieces of data function enable control	
		0	Disable

		1	Enable the slave continuously sending 3 pieces of data function.
Bit[04]	EIRQFlag	Error flag, related to the error interrupt flag I2CEIF; the I2CEIF can be cleared after the bit is cleared.	
		0	Normal
		1	The time-out or receiving accidental start (stop) signal or arbitration failure takes place.
Bit[03]	START(S)	(S) Start signal control bit	
		0	Normal
		1	Generate the start signal from the I^2C bus.
Bit[02]	STOP(P)	(P) Stop signal control bit	
		0	Normal
		1	Generate the stop signal from the I^2C bus.
Bit[01]	IRQFlag(I)	(I) Interrupt control bit	
		0	Normal;
		1	Reply to the interrupt; the device will reply to the interrupt after receiving 9 clocks, and then draw the SCL to low potential until the bit is cleared and release the SCL signal wire; writing in 0 will clear the device status control bit and make the I^2C proceed to the next status.
Bit[00]	A/NA(A)	(A) Response signal replying control bit	
		0	No reply ACK or Reply NACK
		1	The response signal (ACK) has been replied.

29.3.3. I2C register 2

I^2C Base Address + 0x08 (0x41008)				
Symbol	I2CCR2 (I2C Control Register 2)			
Bit	[31:24]	[23:16]		
Name	MASK	CRG		
RW	R0W-0	RW-0		
Bit	[15:08]	[7]	[6:4]	[3:0]
Name	MASK	TOFlag	TOPS	TOLimit
RW	R0W-0	R-0	RW-0	

Bit	Name	Description	
Bit[23~16]	CRG	I^2C bus data serial transmission rate control register	
		0	Set 0
		1	Set 1

The data serial transmission rate of the I^2C bus is determined by the values of the clock source of the I^2C and the serial transmission rate control register CRG; the data serial transmission rate of the I^2C bus can be calculated according to the following equation:

I2CK = (APCK) / [4 x (CRG + 1)]

Note:

I2CK recommend less than 400K Hz

Bit	Name	Description	
Bit[7]	TOFlag	Time-out flag	
		0	Normal
		1	The I^{2}C bus clock wire high/low potential control is overtimed.
Bit[6~4]	TOPS	Time-out clock frequency divider configuration	
		111	CLKPS = I2CK / 128
		110	CLKPS = I2CK / 64
		101	CLKPS = I2CK / 32
		100	CLKPS = I2CK / 16
		011	CLKPS = I2CK / 8
		010	CLKPS = I2CK / 4
		001	CLKPS = I2CK / 2
		000	CLKPS = I2CK / 1
Bit[3~0]	TOLimit	Time-out upper limit configuration	
		1111	16x CLKPS Cycle
		1110	15x CLKPS Cycle
		1101	14x CLKPS Cycle
		1100	13x CLKPS Cycle
		1011	12x CLKPS Cycle
		1010	11x CLKPS Cycle
		1001	10x CLKPS Cycle
		1000	9x CLKPS Cycle
		0111	8x CLKPS Cycle
		0110	7x CLKPS Cycle
		0101	6x CLKPS Cycle
		0100	5x CLKPS Cycle
		0011	4x CLKPS Cycle
		0010	3x CLKPS Cycle
		0001	2x CLKPS Cycle
		0000	1x CLKPS Cycle

29.3.4. I2C register

Slave ID0 (SID0)

I^{2}C Base Address + 0x0C (0x4100C)				
Symbol	I2CCR3 (I2C Control Register 3)			
Bit	[31:24]		[23:16]	
Name	SID1 MASK		SID0 MASK	
RW	R0W-0		R0W-0	
Bit	[15:9]	[8]	[7:1]	[0]
Name	SID1	VD1	SID0	VD0
RW	RW-0		RW-0	

Bit	Name	Description	
Bit[31~24]	SID1 MASK	SID1 MASK	
		0	Disable
		1	Enable
Bit[23~16]	SID0 MASK	SID0 MASK	
		0	Disable
		1	Enable
Bit[15~9]	SID1	SID1 slave address code configuration	
		0	Set 0
		1	Set 1
Bit[08]	VD1	Slave address code valid control bit, the bit should be 1 when writing in the address code.	
		0	The slave address code is invalid.
		1	The slave address code is valid.
Bit[7~1]	SID0	SID0 slave address code configuration	
		0	Set 0
		1	Set 1
Bit[00]	VD0	Slave address code valid control bit, the bit should be 1 when writing in the address code.	
		0	The slave address code is invalid.
		1	The slave address code is valid.

Note:

When operating in I^{2}C slave mode, provides two Slave ID register, it can be applied at the same time the existence of two sets of slave mode. For example: VD0 set to 0x30, VD1 is set to 0x32. You can also use a combination of the two Slave ID 10 yuan slave addressing mode.

Built-in I^{2}C Slave ID Comparator for comparing the received on I^{2}C Bus Slave ID whether on Slave ID Register set ID consistency. When comparing the results were consistent demand Slave Mode will send an interrupt signal to the host system is ready for service, and when the I^{2}C Bus clock signal SCL pulled Low Wait for the machine so that the host system has responded. The Slave Controller will wait for a response until the system has surrendered control over the SCL host. Therefore, in order to avoid system anomalies native prolonged occupation I^{2}C Bus, users must set the appropriate timeout controller (Time-out Controller), so that for too long when no response to the system by releasing SCL Slave Controller of self-control, and send the wrong status interrupt signal.

29.3.5. I2C register

I²C Base Address + 0x10 (0x41010)			
Symbol	I2CCR4 (I2C Control Register 4)		
Bit	[31:16]		
Name	RSV.		
RW	R-0		
Bit	[15:8]	[7:1]	[0]
Name	-	Rx A7-1/D7-1	RW/D0
RW	-	R-X	

Receiver Data Buffer (RxAD)

Bit	Name	Description	
Bit[7~1]	Rx A7-1/D7-1	Register RX[7:0] for receiving address or data	
		0	Set 0
		1	Set 1
Bit[0]	RW/D0	The received data is the 0th value of the read/write instruction or data.	
		0	Set 0
		1	Set 1

Transmitter Data Buffer 0 (TXAD)

I²C Base Address + 0x14 (0x41014)			
Symbol	I2C 5 (I2C Control Register 5)		
Bit	[31:24]	[23:17]	[16]
Name	RSV.	TX2 A7-1/D7-1	Flag/D0
RW	R-0	RW-X	
Bit	[15:08]	[7:1]	[0]
Name	TX1 A7-0/D7-0	TX0 A7-1/D7-1	RW/D0
RW	RW-X		

Bit	Name	Description	
Bit[23~17]	TX2 A7-1/D7-1	Transmission register 2 for transmitting the address or the value of the data[7:1]	
		0	Set 0
		1	Set 1
Bit[16]	Flag/D0	Transmission register 2 for transmitting the read/write instruction or the value of the data[0]	
		0	Set 0

		1	Set 1
Bit[15~8]	TX1 A7-0/D7-0	Transmission register 1 for transmitting the address or the value of the data[7:0]	
		0	Set 0
		1	Set 1
Bit[7~1]	TX0 A7-1/D7-1	Transmission register 0 for transmitting the address or the value of the data[7:1]	
		0	Set 0
		1	Set 1
Bit[00]	RW/D0	Transmission register 0 for transmitting the read/write instruction or the value of the data[0]	
		0	Set 0
		1	Set 1

Note:

In the communication process, to be transmitted when there is no data transmission register must be set to 0XFF.

Because the lowest bit, long pull Low easy for SDA bus lock in Low (0).

When operating in I2C slave mode (Slave), 1 byte if the work in standard mode, using TX0 A7-1 / D7-1 be single data transfer, if the work of 3 bytes mode, using TX0 A7-1 / D7- 1, TX1 A7-0 / D7-0, TX2 A7-1 / D7-1 perform data transfer.

TX1 A7-0 / D7-0: Transmitter 2nd Data Buffer Bit7-0 for 3 Byte Mode Only

After 3 byte mode data transfer is complete, this register is automatically set to FFh.

TX2 A7-0 / D7-0: Transmitter 3rd Data Buffer bit7-1 for 3 Byte modes only.

Flag / D0: Transmitter 3rd Flag or Data buffer bit 0 for 3 byte mode only

After 3 byte mode data transfer is complete; this register is automatically set to FFh.

29.4. Model program function

29.4.1. I2C initialize instruction

I^{2}C Master Mode Initial

- Lead: Configuration I2CK, configured as I2C GPIO mode.
- Configure I2CEn-enable I^{2}C and I^{2}C internal clock circuit.
- Configure CRG enable I^{2}C operating in the required transmission rate.
- Configure TOPS Clock Stretching and TOLimit set time limit.
- Configuration I2CEn and TOEn-enable I2C and Time-out control circuit.
- I^{2}C Master Mode Initial complete, the user can make use of ACT, STA, RxAD and TxAD registers and other information required to complete the transfer.

I^{2}C Slave Mode Initial

- Lead: Configuration I2CK, configured as I^{2}C GPIO mode.
- Configure I2CEn-enable I^{2}C and I^{2}C internal clock circuit.
- Configure TOPS Clock Stretching and TOLimit set time limit.
- Configure Slave IDx and Slave IDx Mask as a comparison of the Slave Mode ID.
- Configure SEn-enable Slave Mode circuitry.
- Configuration I2CEn and TOEn-enable I^{2}C and Time-out control circuit.
- I2C Slave Mode Initial complete, the user can make use of ACT, STA, RxAD and TxAD registers and other information required to complete the transfer.

10 Bit Addressing I^{2}C Slave Mode Initial

- Pre-Assignment: Configuring I2CK, the configuration of I^{2}C GPIO mode.
- Configure I2CEn-enable I^{2}C and I^{2}C internal clock circuit.
- Configure TOPS Clock Stretching and TOLimit set time limit.
- Configure Slave IDx and Slave IDx Mask as a comparison of the Slave Mode ID.
- Configuration SEn and 10bEn-enable Slave Mode circuit 10 Bit Addressing circuit.
- Configuration I2CEn and TOEn-enable I^{2}C and Time-out control circuit.
- I^{2}C Slave Mode Initial complete, the user can make use of ACT, STA, RxAD and TxAD registers and other information required to complete the transfer.

3 Byte I^{2}C Slave Mode Initial

- Lead: Configuration I2CK, configured as I^{2}C GPIO mode.
- Configure I2CEn-enable I^{2}C and I^{2}C internal clock circuit.
- Configure TOPS Clock Stretching and TOLimit set time limit.
- Configure Slave IDx and Slave IDx Mask as a comparison of the Slave Mode ID.
- Configuration SEn and 3BEn-enable Slave Mode 3 Byte TX circuit and a control circuit.
- Configuration I2CEn and TOEn-enable I^{2}C and Time-out control circuit.
- I^{2}C Slave Mode Initial complete, the user can make use of ACT, STA, RxAD, TX0, TX1, TX2 and other information required to complete the transfer register.

29.4.2. I2C operating process instruction

Below is a common I^{2}C EEPROM (24C02) control data format.

The following provides I^{2}C Master Write & Read to read and write EEPROM allows the user to control commentary HY16F19xB awareness of I^{2}C operational processes.

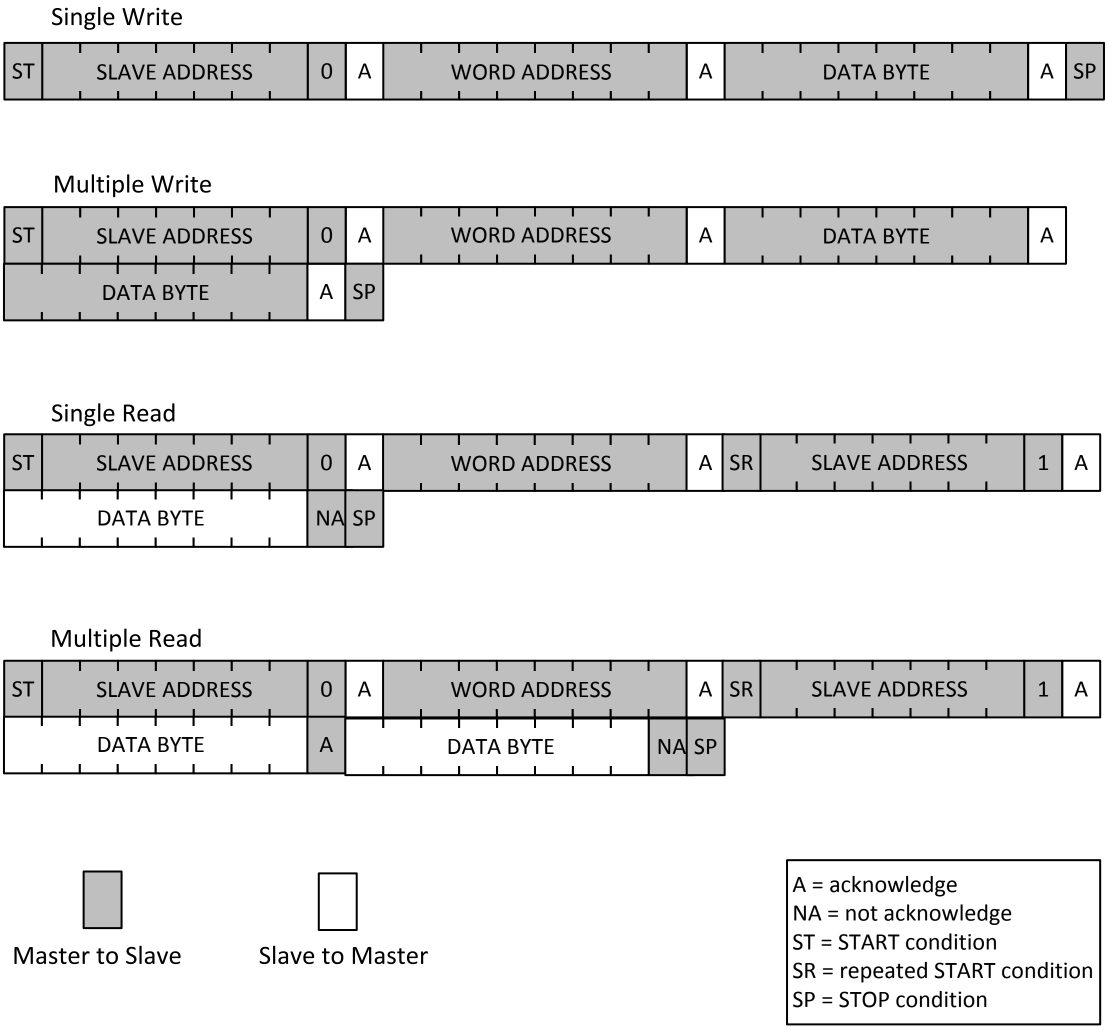
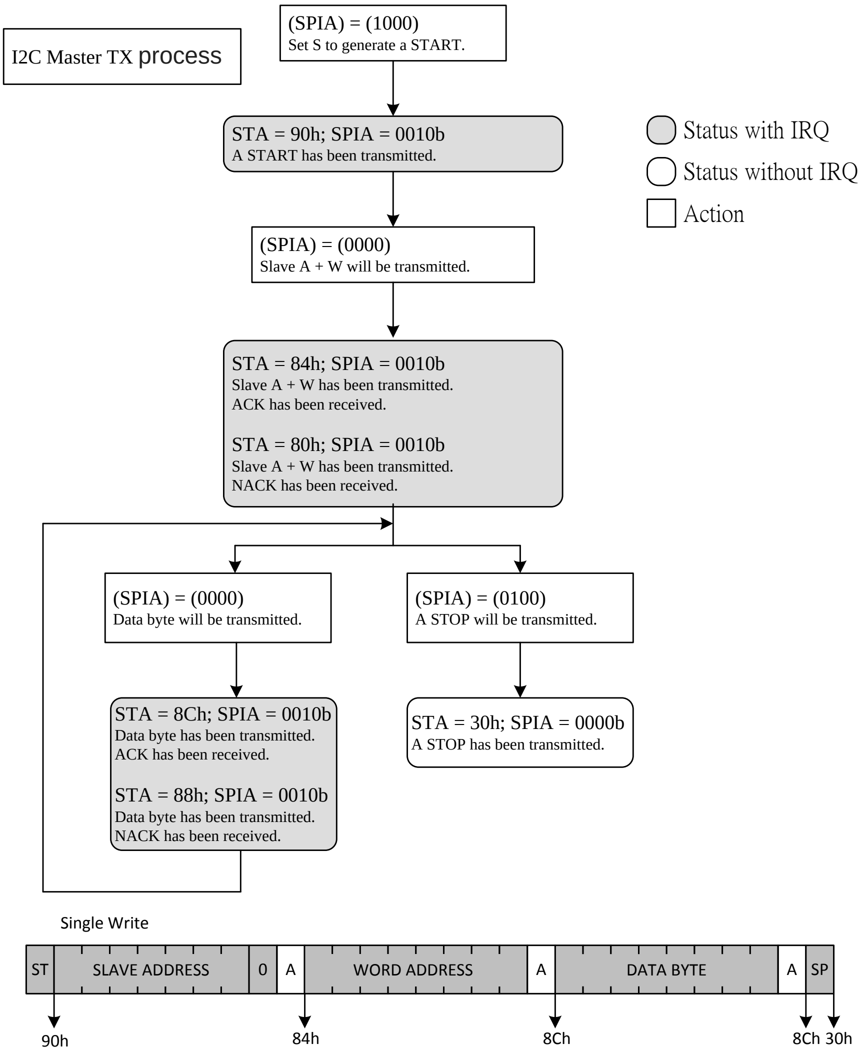


Figure 29-9 I^{2}C EEPROM data control format

I2C Master TX side process description:

Readers can refer to the following text, clearly understand I^{2}C Master TX end I^{2}C Single Write operation flow.



Master TX state flowchart of FIG 29-10 I^{2}C

1. First use HYCON C library instruction DrvI2C_Ctrl (1,0,0,0) setting (SPIA) = (1000) to complete the START condition, when executing the instruction DrvI2C_Ctrl (1,0,0,0), you can use oscilloscope from SCL and SDA pin observe the I2C START signal waveform has been sent.
2. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 90h, if 90h, 90h entering state. Use instruction within 90h state DrvI2C_WriteData fill SLAVE ADDRESS and using instructions DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), when executing the DrvI2C_Ctrl within 90h state (0,0,0,0) You can use an oscilloscope I^{2}C SLAVE ADDRESS waveform signal has been sent from the SCL and SDA pins.

3. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 84h, if 84h, on behalf of Slave has returned ACK, enter 84h state. Use instruction in this state DrvI2C_WriteData fill WORD ADDRESS and using instructions DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), when executing the DrvI2C_Ctrl within 84h state (0,0,0,0) It can be observed from SCL and SDA pin I2CWORD ADDRESS waveform signal has been sent.
4. Use HYCON C library instruction Dev I^{2}C GetStatus Flag status is confirmed STA 8Ch, if 8Ch, on behalf of Slave has returned ACK, enter 8Ch state. Within this state, use instruction DrvI2C_WriteData fill DATA BYTE and using instructions DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), when executing the DrvI2C_Ctrl within 8Ch state (0,0,0,0), you can observe the I2C DATA BYTE signal has been sent from the waveform SCL and SDA pin.
5. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 8Ch, if 8Ch, on behalf of Slave gone back ACK. At this point, or enter 8Ch state, in this state, as it has been without any data transfer, so you can use the command DrvI2C_Ctrl (0,1,0,0) setting (SPIA) = (0100), when executed within 8Ch state Ends DrvI2C_Ctrl (0,1,0,0), can be observed I2C STOP waveform signal has been generated from the SCL and SDA pins.
6. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 30h, if 30h, representing the first document data transfer has been completed.

I²C Master TX & RX side process description:

Readers can refer to the following text; clearly understand I²C Master TX & RX and I2C Single Read operation flow.

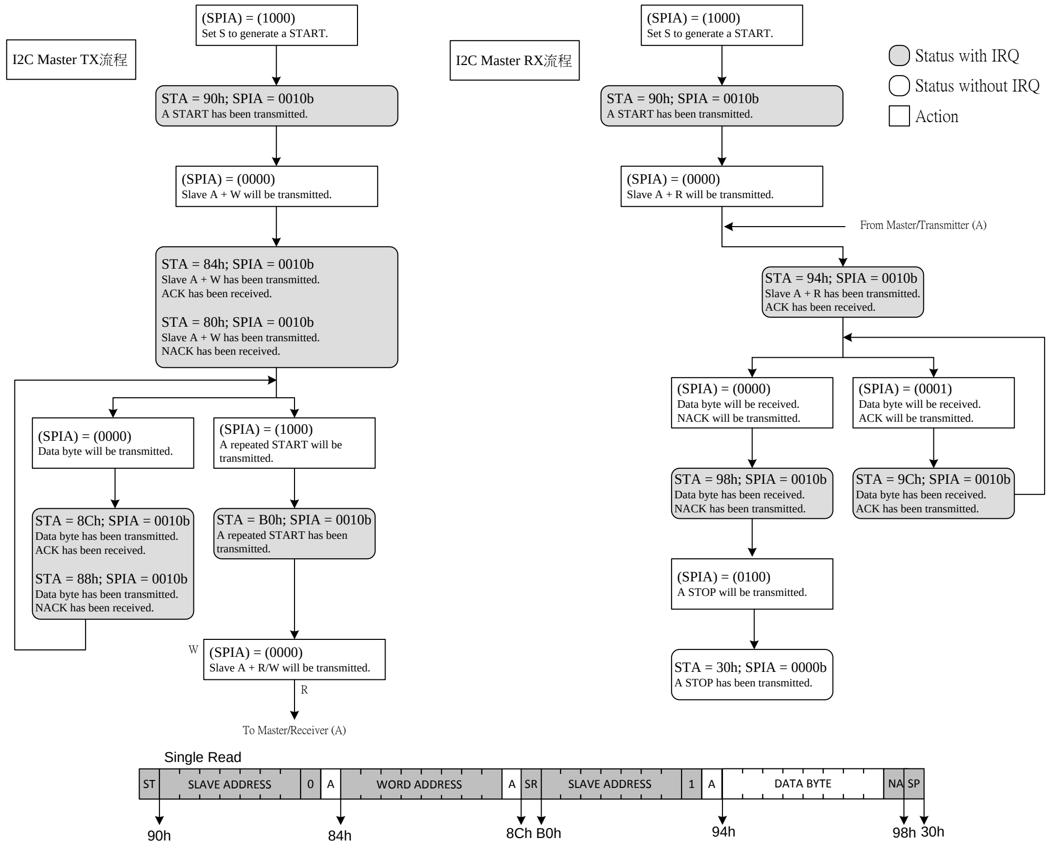

I2C Master TX流程
(SPIA) = (1000)
Set S to generate a START.
STA = 90h; SPIA = 0010b
A START has been transmitted.
(SPIA) = (0000)
Slave A + W will be transmitted.
STA = 84h; SPIA = 0010b
Slave A + W has been transmitted.
ACK has been received.
STA = 80h; SPIA = 0010b
Slave A + W has been transmitted.
NACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
(SPIA) = (1000)
A repeated START will be transmitted.
STA = 8Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
STA = 88h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = B0h; SPIA = 0010b
A repeated START has been transmitted.
W
(SPIA) = (0000)
Slave A + R/W will be transmitted.
R
To Master/Receiver (A)
I2C Master RX流程
(SPIA) = (1000)
Set S to generate a START.
STA = 90h; SPIA = 0010b
A START has been transmitted.
(SPIA) = (0000)
Slave A + R will be transmitted.
From Master/Transmitter (A)
STA = 94h; SPIA = 0010b
Slave A + R has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 98h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 9Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
(SPIA) = (0100)
A STOP will be transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
Status with IRQ
Status without IRQ
Action
Single Read
ST
SLAVE ADDRESS
0
A
WORD ADDRESS
A
SR
SLAVE ADDRESS
1
A
DATA BYTE
NA
SP
90h
84h
8Ch B0h
94h
98h 30h

Figure 29-11 flowchart state of I²C Master TX RX

1. Before performing I²C Master RX process or the first to do I²C Master TX processes. First use HYCON C library instruction DrvI2C_Ctrl (1,0,0,0) setting (SPIA) = (1000) START condition, when executing the instruction DrvI2C_Ctrl (1,0,0,0), from SCL and SDA Pin observe the I²C START signal waveform has been sent.
2. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 90h, if 90h, 90h entering state. Use instruction within 90h state DrvI2C_WriteData fill SLAVE ADDRESS and using instructions DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), when executing the DrvI2C_Ctrl within 90h state (0,0,0,0) It can be observed from SCL and SDA pin I2C SLAVE ADDRESS waveform signal has been sent.
3. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 84h, if 84h, on behalf of Slave has returned ACK, enter 84h state, using the instruction in this state DrvI2C_WriteData fill WORD ADDRESS and using instructions DrvI2C_Ctrl (0, 0,0,0) Setting (SPIA) = (0000), when executing the DrvI2C_Ctrl within 84h state (0,0,0,0)

can be observed I^{2}C WORD ADDRESS waveform signal from SCL and SDA pin has been sent.

4. Use HYCON C library instruction Drv I2C GetStatus Flag status is confirmed STA 8Ch, if 8Ch, on behalf of Slave has returned ACK, enter 8Ch state. Use instruction DrvI2C_Ctrl in this state (1,0,0,0) setting (SPIA) = (1000), when executing the DrvI2C_Ctrl within 8Ch state (1,0,0,0), from SCL and SDA pin Observation I^{2}C repeated start waveform signal has been sent.
5. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is B0h, if B0h, on behalf of repeated start signal has been sent. Use instruction DrvI2C_WriteData fill SLAVE ADDRESS + 1 and using instructions DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), when executed within B0h state Ends DrvI2C_Ctrl (0,0,0,0), can I2C SLAVE ADDRESS observed from SCL and SDA pin + 1 waveform signal has been sent, this time into the I^{2}C Master RX processes.
6. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 94h, if 94h, on behalf of the Master side has received the return of SLAVE ADDRESS + 1 ACK, enter 94h state, use instruction DrvI2C_Ctrl (0,0,0 , 0) is set (SPIA) = (0000), when executing the DrvI2C_Ctrl within 94h state (0,0,0,0) can be observed I^{2}C Slave DATA BYTE waveform signal from SCL and SDA pin has been sent.
7. Use HYCON C library instruction Dev I^{2}C GetStatus Flag confirm STA status is 98h, if it is 98h, the representative has received DATA BYTE Master end Slave end of the back data and Master has sent NACK signal to the Slave side. DrvI2C_ReadData read back using the command DATA BYTE data sent by Slave end and set (SPIA) = (0100), Stop sending end signal.
8. Use HYCON C library instruction DrvI2C_GetStatusFlag confirm STA status is 30h, if 30h, representing the first document data read has been completed.

I²C Slave RX side process description:

Readers can refer to the following text; clearly understand I²C Slave RX end and operational processes of I²C Single Write.

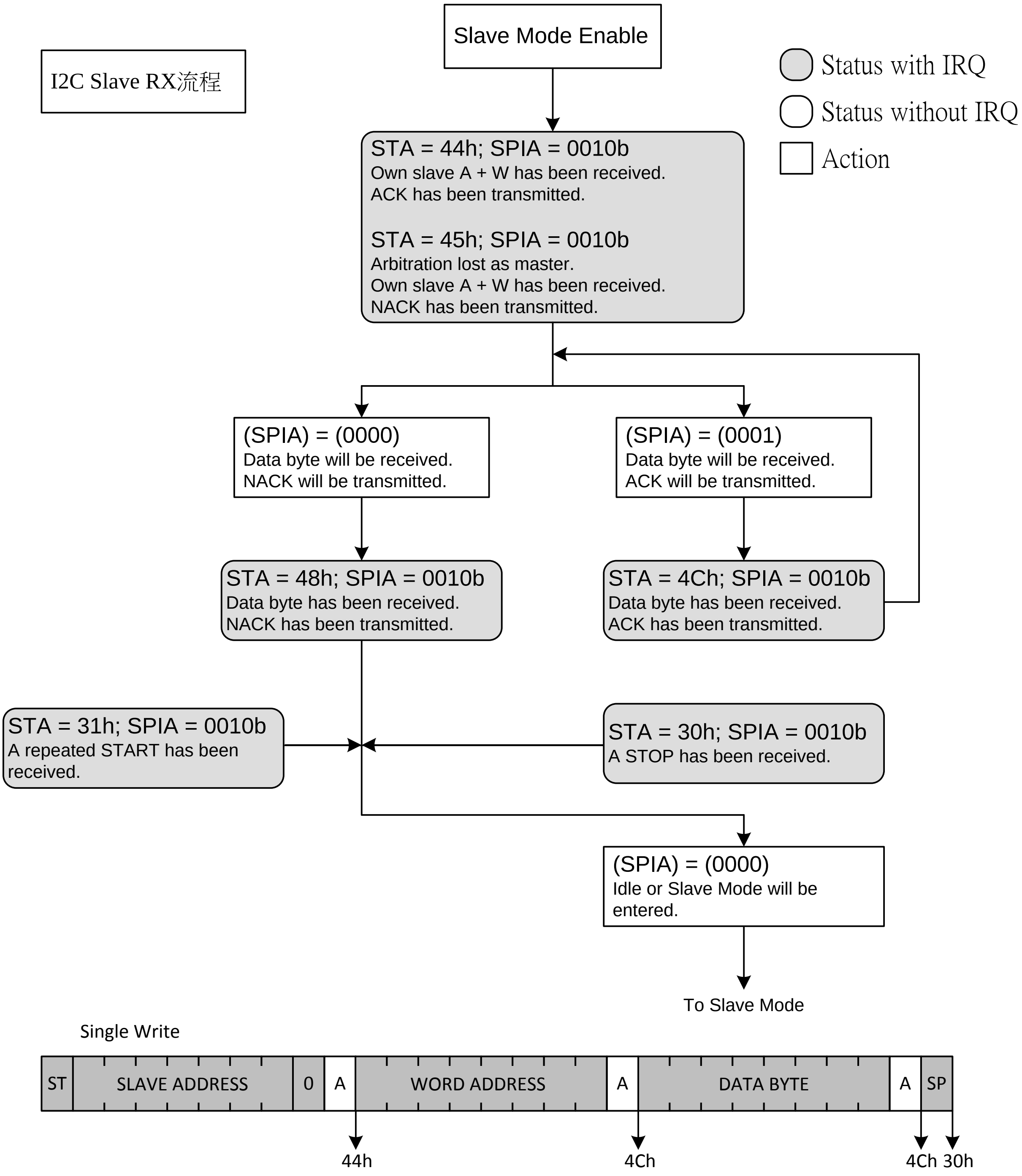

I2C Slave RX流程
Slave Mode Enable
Status with IRQ
Status without IRQ
Action
STA = 44h; SPIA = 0010b
Own slave A + W has been received.
ACK has been transmitted.
STA = 45h; SPIA = 0010b
Arbitration lost as master.
Own slave A + W has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 48h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 31h; SPIA = 0010b
A repeated START has been received.
STA = 30h; SPIA = 0010b
A STOP has been received.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Slave Mode
Single Write
ST
SLAVE ADDRESS
0
A
WORD ADDRESS
A
DATA BYTE
A
SP
44h
4Ch
4Ch 30h

FIG. 29-12 I²C Slave RX state flowchart

1. After completion of I²C Slave initialization, the first end of the first I²C Master SLAVE ADDRESS send information to the I²C Slave end, if there is the right-side I²C Slave back ACK, this time I²C Slave side will enter the state 0x44, and this time if you use an

oscilloscope SCL and SDA pin state, it can be seen I^{2}C Master has sent information to the SLAVE ADDRESS I^{2}C Slave, and I^{2}C Slave side has done a first ACK signal response.

2. When the I^{2}C Slave end into the 0x44 state, this time on behalf of I^{2}C Slave has done a first ACK reply, this time I^{2}C Master end WORD ADDRESS if sent information to the I^{2}C Slave side, when I^{2}C Slave end use instruction DrvI2C_Ctrl (0 , 0,0,1) setting (SPIA) = (0001), when the implementation of End (SPIA) = (0001) control state, on behalf of I^{2}C slave terminal has received the information and do WORD ADDRESS ACK reply, this time if you use oscilloscope SCL and SDA pin state, it can be seen I^{2}C Slave has done a second signal of ACK reply, I^{2}C Slave enter 0x4C end state.
3. I^{2}C Slave enter 0x4C end state, then I^{2}C Slave end use instruction DrvI2C_ReadData accept WORD ADDRESS data, this time I^{2}C Master will then send DATA BYTE end data to I2C Slave side, when I2C Slave end use instruction DrvI2C_Ctrl (0, 0,0,1) Setting (SPIA) = (0001), when the implementation of End (SPIA) = (0001) control state, on behalf of I^{2}C Slave DATA BYTE data has been received and made ACK reply, and this time if you use an oscilloscope SCL and SDA pin state, it can be seen I^{2}C Slave side has done a third of the ACK signal reply, I^{2}C Slave side or enter 0x4C state.
4. I^{2}C Slave enter 0x4C end state, then I^{2}C Slave end use instruction DrvI2C_ReadData accept DATA BYTE data, when the I^{2}C Slave end use instruction DrvI2C_Ctrl (0,0,0,1) setting (SPIA) = (0001) after, I^{2}C Master terminal will send signals to the I^{2}C Slave STOP end, this time if you use an oscilloscope to observe the state of SCL and SDA pin, you can see I^{2}C Master terminal sends signals to the I^{2}C Slave STOP end, I^{2}C Slave end into the 0x30 state.
5. I^{2}C Slave end into the 0x30 state, on behalf of I^{2}C Slave end has received I^{2}C Master STOP signal sent by the end, this time using the command DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), let I^{2}C Slave end re-enter the initial state, waiting for the next I^{2}C Master end signal sent signals.

I²C Slave TX & RX Flow Description:

Readers can refer to the following text; clearly understand I²C Slave TX & RX and I2C Single Read operation flow.

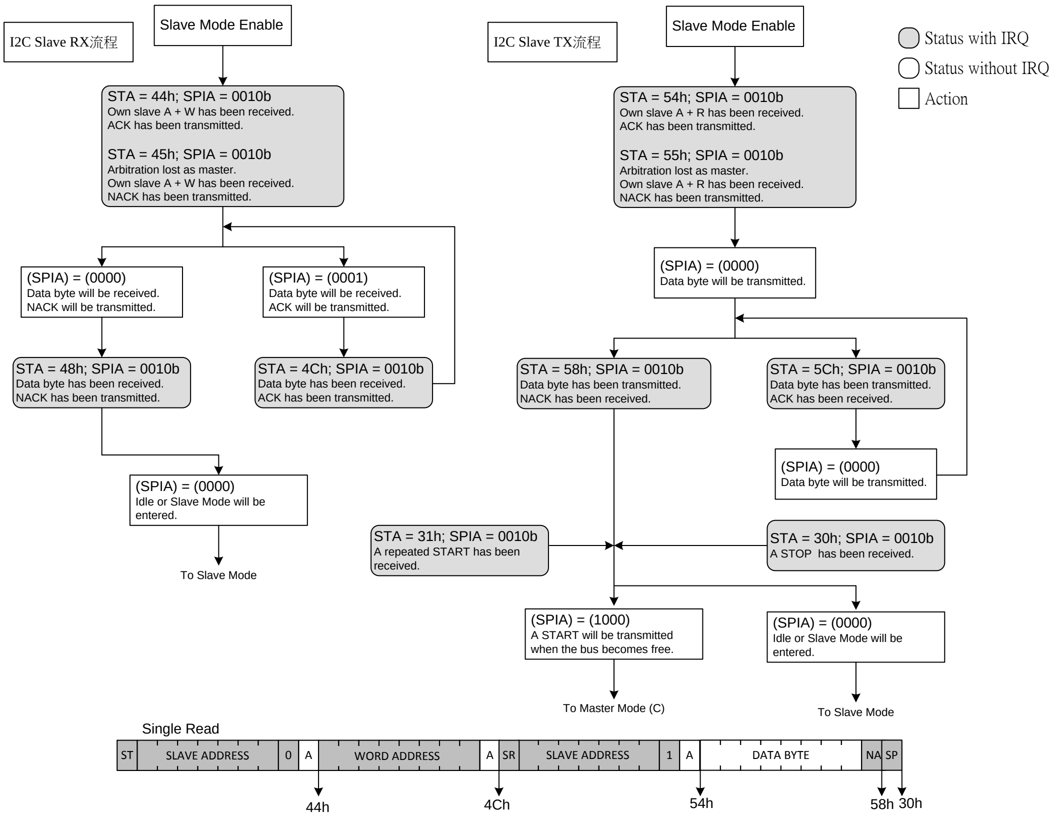

I2C Slave RX流程
Slave Mode Enable
STA = 44h; SPIA = 0010b
Own slave A + W has been received.
ACK has been transmitted.
STA = 45h; SPIA = 0010b
Arbitration lost as master.
Own slave A + W has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 48h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Slave Mode
I2C Slave TX流程
Slave Mode Enable
STA = 54h; SPIA = 0010b
Own slave A + R has been received.
ACK has been transmitted.
STA = 55h; SPIA = 0010b
Arbitration lost as master.
Own slave A + R has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be transmitted.
STA = 58h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = 5Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
STA = 31h; SPIA = 0010b
A repeated START has been received.
STA = 30h; SPIA = 0010b
A STOP has been received.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode
○ Status with IRQ
○ Status without IRQ
□ Action
Single Read
ST
SLAVE ADDRESS
0
A
WORD ADDRESS
A
SR
SLAVE ADDRESS
1
A
DATA BYTE
NA
SP
44h
4Ch
54h
58h
30h

FIG.29-13 I²C Slave TX RX state flowchart

1. After completion of I²C Slave initialization, the first end of the first I²C Master SLAVE ADDRESS send information to the I²C Slave end, if there is the right-side I²C Slave back ACK, this time I²C Slave side will enter the state 0x44, and this time if you use an oscilloscope SCL and SDA pin state, it can be seen I²C Master has sent information to the SLAVE ADDRESS I²C Slave, and I²C Slave side has done a first ACK signal response.
2. When the I²C Slave end into the 0x44 state, this time on behalf of I²C Slave has done a first ACK reply, this time I²C Master end WORD ADDRESS if sent information to the I²C Slave side, when I2C Slave end use instruction DrvI2C_Ctrl (0 , 0,0,1) setting (SPIA) = (0001), when the implementation of End (SPIA) = (0001) control state, on behalf of I2C slave terminal has received the information and do WORD ADDRESS ACK reply, this time if you use oscilloscope SCL and SDA pin

state, it can be seen I^{2}C Slave has done a second signal of ACK reply, I^{2}C Slave enter 0x4C end state.

3. I^{2}C Slave enter 0x4C end state, then I^{2}C Slave end use instruction DrvI2C_ReadData accept WORD ADDRESS information, and this time end I^{2}C Master SLAVE ADDRESS + 1 will be sent information to the I^{2}C Slave side, when I2C Slave end use instruction DrvI2C_Ctrl (0, 0,0,1) Setting (SPIA) = (0001), when the implementation of End (SPIA) = (0001) control state, on behalf of I2C Slave SLAVE ADDRESS + 1 has received information and made ACK reply, this time if you use oscilloscope SCL and SDA pin state, it can be seen I2C Slave side has done a third of the ACK signal reply, I^{2}C Slave end into the 0x54 state.
4. I^{2}C Slave end into the 0x54 state where the I^{2}C Slave end use instruction DrvI2C_WriteData fill DATA BYTE data you want to back to the I^{2}C Master side, when I^{2}C Slave end use instruction DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000) after this time if you use an oscilloscope to observe the state of SCL and SDA pin, you can see I^{2}C Master I^{2}C Slave side end has received DATA BYTE sent signals, and I^{2}C Master has sent NACK signal to the I^{2}C Slave end, I^{2}C Slave end into the 0x58 state.
5. I^{2}C Slave end into the 0x58 state, on behalf of I^{2}C Slave I^{2}C Master terminal has received NACK signal sent by the end, when I^{2}C Slave end use instruction DrvI2C_WriteData the highest bit MSB is set to High, and using instructions DrvI2C_Ctrl (0,0,0 , 0) is set (SPIA) = (0000), the implementation of End (SPIA) = (0000) after this time if you use an oscilloscope to observe the state of SCL and SDA pin, you can see I^{2}C Master sends STOP signal end, I^{2}C Slave end enter 0x30 state.
6. I^{2}C Slave end into the 0x30 state, on behalf of I^{2}C Slave end has received I^{2}C Master STOP signal sent by the end, this time using the command DrvI2C_Ctrl (0,0,0,0) setting (SPIA) = (0000), let I2C Slave end re-enter the initial state, waiting for the next I2C Master end signal sent signals.

29.5. I²C General Call Mode

HY16F198B the I²C Slave support I²C General Call Function.

When the General Call Function I²C Slave mode is activated, then I²C Master Mode can be used to do a broadcast control I²C Slave Controller, I²C Slave work in broadcast mode when Slave Address ID is 00h, I²C Slave of General Call Function key differentiator General Call and General Call Reset two kinds.

General Call:

When the unit of General Call is a call, I²C Slave Controller will post more information will be waiting to receive an interrupt signal, rather than a general agreement received Slave ID will be issued immediately interrupt signal, and in response to the position of the handle , will be issued a "NACK" signal and an interrupt signal to the machine after the processor receives I²C Slave Controller when slave mode is on General Call ID automatically issued to all "ACK" signal to the host when 00h and continue to automatically receive the next piece of data . The following figure shows General Call control commands, when SEn and GCRst function can simultaneously cause (Enable) when you can use General Call to make control of the I²C Slave device. I²C Master control process received its first information sent by General Call "00h" to end when the I²C Slave reply ACK NACK control command reply will be issued when the first pen interrupt signal, reads the STA is 4Ah or 4Eh representatives I²C Slave has been working in General Call mode, and the first document data has been stored in the RX Receive data register, I²C Slave RX receiving end can read data register and make a judgment as to what command and the corresponding action.

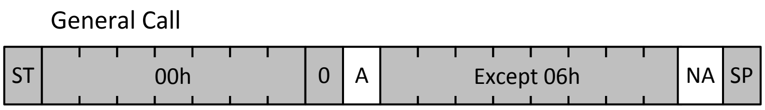


Figure 29-14 I²C General Call

General Call Reset:

I²C Slave Controller also supports General Call Reset function, when SEn and GCRst function is turned on at the same time, if the I²C Controller General call ID 00h reception and the first document data is "06h" is the General Call Reset condition is satisfied, then the original will be sent to native processor interrupt signal (Interrupt) will be reset signal (Reset) substituted, provided the external host can reset the machine functions wafer via I²C Bus.

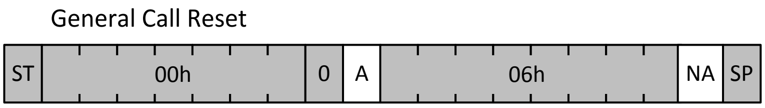


Figure 29-15 I²C General Call Reset

29.6. 10 Bit Addressing Mode

29.6.1. I2C 10 Bit Addressing Mode instruction

10 bit addressing (10-bit addressing) mode I^{2}C original seven yuan addressing extensions, 10 yuan and 7 yuan addressing mode can coexist on the I^{2}C existing infrastructure. 10 yuan addressing mode is the top two byte transfer in slave address after START, I^{2}C Bus standard for the format of the machine is also made here defined as follows 29-16, the first byte must 11110xx0b, which is the first byte is necessarily a "write" command, which Bit2 ~ 1 from the machine address of Bit9 ~ 8, compared with the second group of bits of the slave address Bit7 ~ 0 start is the third part of the information bytes. So when the first byte of the host transmission, it may also issue a response to several slaves. When the first two bytes are transmitted hosts receive a response, only representatives from the machine is ready to communicate with it by a third byte begin transmitting data.

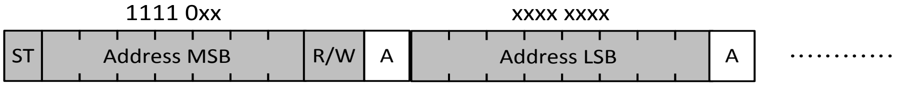


Figure 29-16 I^{2}C 10 Bit Addressing Mode

Master Transmitter:

Host for the transmission of information of the operation, using the I^{2}C host controller and seven yuan addressing mode there is no difference, only the first two bytes transferred from the machine address and write command, you can start transmitting data, but I^{2}C controller will be used in (Slave) Slave mode needs to be part of the register set. You must first set SID0 to Address MSB, SID1 to Address LSB, as shown in Figure 29-17. And the SID0 VD0 bit is set to High-enable address comparator circuit, in addition to the scratchpad and then ACT within SEn and 10bEn bit is set to High, opened 10 yuan slave mode and addressing mode will cause the I2C controller operating in 10 yuan slave mode.

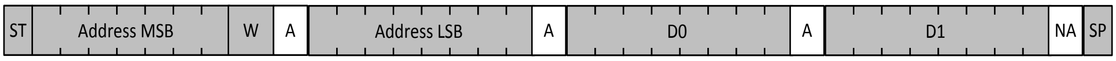


Figure 29-17 Master Transmitter

Master Receiver:

Under the operation of data read by the host, the host must first to transmit "write" and the slave address can cause a manner corresponding to the slave, then through the Repeat Start switch to "read" instruction, of course, the host sends Repeat Start before data can still be written in part caused by the reading mode and then switch to read the required information. And the slave after the slave address after START accord will be able to write mode is activated later on here than just a byte of the first in line to receive Repeat Start If that is representative of the host is still present slave communication, i.e., Address MSB and Repeat Start after following figure within 29-18 after START must be the same in order to make the slave enters read mode, if different from the machine will exit the newsletter rather address MSB Repeat Start after other seven yuan will be here from the machine finds compared with responses to another communications start.

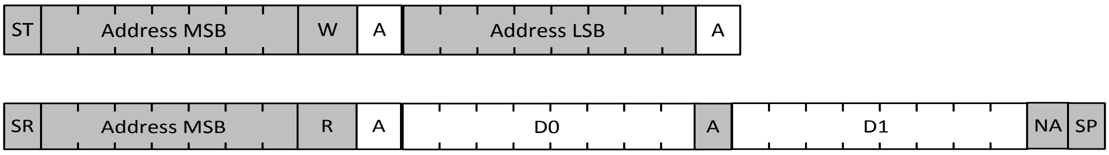


Figure 29-18 Master Receiver

Slave ID Mask:

Slave address (Slave ID, SID) provides native operating in slave mode through the SID register is set, the machine can be set Slave ID and through the native processor, will be applied to a wide range of wafer with I^{2}C interface for the transmission of applications. After the mask and the slave address (Slave ID Mask, SIDM) may further increase the application example Slave mode, SID Mask individual bit is set to High, Slave Address Slave ID and I^{2}C Bus on its corresponding bit yuan will all be as "consistent" and therefore SID SID Mask will allow expansion from a single address range to range-type comparison.

29.6.2. 10 Bit Addressing data write in process instruction

10 bit addressing mode data write process:

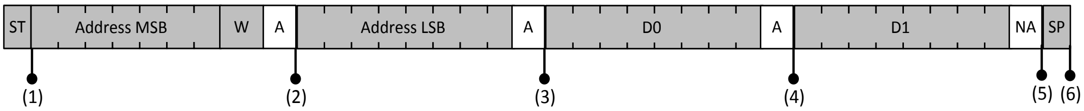


Figure 29-19 10 bit addressing mode to write data flow

- - 10 Bit Address Slave data is written to the process (which can refer I^{2}C Slave Receiver Mode flow chart).

1. to set up the initialization phase SEn and 10bEn-enable 10 Bit Addressing Slave mode and waits for an interrupt signal. (3)
2. Read the STA as Slave 44h Bus on behalf of the ID-based machine ID, and the situation is being written.
3. Set the A / NA register is High, data transmission is complete after this pen transmits an ACK to Master, on behalf of the unit will be able to continue to receive the next piece of data, clear the interrupt flag IRQFlag trigger procedures to be written, and waiting for the next one interrupt signal. (4)
4. Read the STA is 4Ch acknowledgment ACK has been transmitted, if the Slave can still receive data write, then repeat the previous step, if not then proceed to the next step. (Above two steps can be ignored in the case of only a single piece of data is written in)

5. Set the A / NA register to Low, data transmission is complete after this pen transmits a NACK to Master, on behalf of the unit will not receive the next piece of data, clear the interrupt flag IRQFlag trigger procedures to be written, and waits for the next interrupt signal. (5)
6. Read the STA to confirm NACK has been transmitted 48h, and ready to end the program is written.
7. Clear the interrupt flag IRQFlag wait STOP signal issued by the host, and waits for an interrupt signal. (6)
8. Read the STA to 30h on behalf of the host has finalized procedures.
9. Clear the interrupt flag IRQFlag has entered the next program, you can set up START scratchpad to enter host mode trying to get control of the Bus, or just clear the interrupt flag IRQFlag continue to maintain slave mode.

29.6.3. 10 Bit Addressing data readout Flow Description

10 bit addressing mode data read-out process:

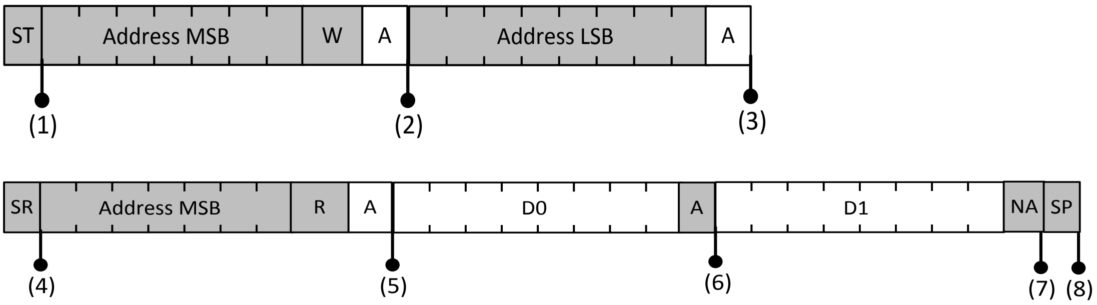


Figure 29-20 10 yuan addressing mode data read-out process

- 10 Bit Addressing Master data read process (can be flow refer I^{2}C Master Transmitter Referring I^{2}C Master Receiver Mode flow chart)
1. Establishment of START scratchpad issue Start Bit, and waiting for an interrupt signal. (1)
2. Start reading the STA to 90h on behalf of the successful issue and obtain Bus control.
3. The address of the target Slave Address MSB of fill in the TxAD register.
4. Clear the interrupt flag IRQFlag trigger bit only transfer program, and waiting for an interrupt signal. (2)
5. Read the STA is on 80h no such representatives Bus Slave or Slave unable to respond (may be too busy or have already crashed). Such as reading STA to 84h Slave has been issued on behalf of a response. There may be a number of 10 Bit Addressing Slave mode at the same time to respond to Address MSB.
6. Slave address Address MSB goal of fill in the TxAD register.
7. Clear the interrupt flag IRQFlag trigger bit only transfer program, and waiting for an interrupt signal. (3)

8. Read SATA Bus no such representatives of the 88h Slave Slave or unable to respond (may be too busy or have already crashed). Such as reading STA is 8CH been issued on behalf of Slave response and wait for data to write.
(If your host has information or commands to be sent to the machine, you can execute the process of sending data after this step)
9. Establishment START register, and clears the interrupt flag IRQFlag trigger Repeat Start Bit transfer program, and wait for an interrupt signal to perform for 10 Bit Addressing Slave data read. (4)
10. Read B0h STA is issued on behalf Repeat Start to success and achieved Bus control.
11. Slave address Address MSB goal of fill in the TxAD register.
12. Clear the interrupt flag IRQFlag trigger transfer the program here, and waiting for an interrupt signal. (5)
13. STA read as 91h on the Bus no such representatives Slave Slave or unable to respond (may be too busy or have already crashed). Such as reading STA to 94h has been issued on behalf of Slave response and wait for data read.
14. Set A / NA register transfer ACK High thereto pen after the completion of data transmission to the information on behalf of the follow-up remains to be transferred Slave. Clear the interrupt flag IRQFlag trigger data reading program, and waits for the next interrupt signal. (6)
15. Slave reads return within RxAD of the information and read STA is 9Ch acknowledgment ACK has been transmitted, if the data is read again by the Slave still need to read the data Repeat the previous step, if not the next a step. (Only in the case of the above two steps to read a single piece of data can be ignored)
16. Setting A / NA register to Low thereto pen data transmission completion of the transfer to the representative NACK transmission process coming to an end, clearing the interrupt flag IRQFlag trigger data reading program, and waits for the next interrupt signal Slave. (7)
17. backhaul read the information from the Slave within RxAD; read the STA as the representative NACK 98h to spread.
18. The establishment of STOP scratchpad, and clears the interrupt flag IRQFlag trigger STOP Bit transfer program has ended this data transmission.

29.6.4. 10 Bit Addressing data read out process instruction

- 10 Bit Addressing Slave data is read process (please refer I^{2}C Slave Receiver Process Referring I^{2}C Slave Transmitter Mode Process)
1. To set up the initialization phase SEn and 10bEn-enable 10 Bit Addressing Slave mode and waits for an interrupt signal. (3)
2. Read the STA as Slave 44h Bus on behalf of the ID-based machine ID, and the situation is being written.

3. Set the A / NA register as High, after the completion of this document data transmission to transmit Ack Master, on behalf of the unit will be able to continue to receive the next piece of data, clear the interrupt flag IRQFlag trigger procedures to be written, and waiting for the next interrupt signal.
4. Read the STA if it is still in the 4Ch behalf of Master of Slave write data or command of the trip. At this time the interrupt signal is 10 yuan addressing mode data write process No. (4) interrupted and asked to change the process to continue.
5. If the Master has been issued Repeat Start Bit, at this time of the interrupt-based resolution of the case (4) interrupts, read STA is 70h, because the continuous re-transmission of the new Master Slave ID, STA exists only for the 70h Bit7 of Before SCL negative edge, negative edge after the read of the STA may be 50h or 54h, clears the interrupt flag IRQFlag end of the previous writers, and waits for the next interrupt signal. (5)
6. Read the STA to 54h on behalf of Slave Bus ID of this machine ID, and the situation is being read. Please note that if the unit one step too late to deal interrupt signal, will likely be the interruption of the step coverage.
7. To read the data stored within TxAD, clears the interrupt flag IRQFlag trigger is read by the program, and waits for an interrupt signal. (6) (7)
8. Read STA 58h if reading program on behalf of the host coming to an end, if it is 5Ch said host data can be read, Slave prepare the relevant information and repeat the previous step.
9. Clear the interrupt flag IRQFlag wait Stop Bit signal issued by the host, and waits for an interrupt signal. (8)
10. Read the STA to 30h on behalf of the host to read the program has ended.
11. Clear the interrupt flag IRQFlag has entered the next program, you can set up START scratchpad has entered the host mode, trying to get control of the Bus, or just clear the interrupt flag IRQFlag continue to maintain slave mode.

29.7. 3 Byte Data Mode

29.7.1. I2C 3 Byte Date Mode instruction

The three-tuple data transfer mode (3 Byte Data Mode) provides continuous data transfer of three bytes, the unit processor can be three bytes of data simultaneously fill TX0, TX1 and TX2 three registers inside, and start the three-tuple transfer mode, I2C Controller will automatically be transferred after three bytes of data will be issued interrupt notification native processor to the next control. This feature is designed to reduce the frequency of I^{2}C Controller interrupt, and reduce the number of service interruptions native processor need to call to improve the overall operational efficiency of the wafer.

After 3Byte Data Mode has been started, the native processor has filled in data transfer register, at the same time issued data transfer instruction will 3BEn bit set up so that I^{2}C

Controller started three bytes of data transmission. Using this mode requires special attention, only the machine at the slave mode and can be opened 3BEn bit when you need to transfer data, when the end of the data transfer stroke must be closed 3BEn bit, I^{2}C Controller avoid abnormal movement, in addition to note is 3Byte Data Mode can be used only once every stroke transmission, which means that every time when Start Bit of the machine when the machine from data transmission mode, only use three-tuple mode automatically sent three items of information, after The data transmission can only be a single general data transfer mode.

29.7.2. I2C 3 Byte Date continuous reading Flow Description

I^{2}C 3 Byte Date continuous reading Flow Description

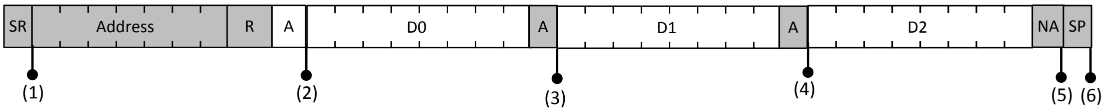


Figure 29-21 3 Byte Data continuous read mode

29.7.3. 3 Byte Master data read process (see I2C Master Receiver flowchart)

1. Master in this mode is the general reading processes, the associated interrupt signal (1-5) No. interrupt.

I^{2}C 3 Byte Data is read Flow Description

3 Byte Slave Process data is read (see I^{2}C Slave Transmitter flow diagram)

1. to set up the initialization phase SEn-enable Slave mode and waits for an interrupt signal. (2)

2. Read the STA as Slave 54h Bus on behalf of the ID-based machine ID, and the situation is being read.

3. For the information will be read the stored TX0, TX1, within TX2, establish 3BEn and clears the interrupt flag IRQFlag trigger is read by the program, and waiting for an interrupt signal. (5)

4. Read the STA is coming to an end 58h to read the program on behalf of the host.

5. Clear 3BEn interrupt flag IRQFlag wait Stop Big signal issued by the host, and waits for an interrupt signal. (4)

6. Read the STA to 30h on behalf of the host to read the program has ended.

7. Clear the interrupt flag IRQFlag to go to the next program, you can set up START scratchpad has entered the host mode trying to get control of the Bus, or just clear the interrupt flag IRQFlag continue to maintain slave mode.

30. HARDWARE REAL TIME CLOCK (HW RTC)

30.1. Overall description

The real time clock controller provides the real time clock and calendar.

The clock source of the RTC is from the external 32.768 KHz crystal connected to the I/O port or the internal 35 kHz LPO oscillator.

The RTC controller shows the time information about hour/minute/second by binary coded decimal (BDC) and the calendar information about year/month/day.

The controller has a programmable alert interrupt program and a periodically programmable wake-up interrupt program, such that the system can be automatically wakened to deal with the low power mode. The controller further has a 6-bit digital timing crystal oscillator offset compensation mechanism.

Function: The time information (hour/minute/second) and the date information (year/month/day) are stored in the register.

Alert register (year/month/day/hour/minute/second)

All time and date information are shown by the BCD format.

Leap automatic compensation (years: 2012~2099)

Week counter

6-bit digital timing crystal oscillator offset compensation

Support periodically wake up the CPU from the idle mode.

Support 8 periodical wake-up period options: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, and 1

Support two time modes, 12/24 systems.

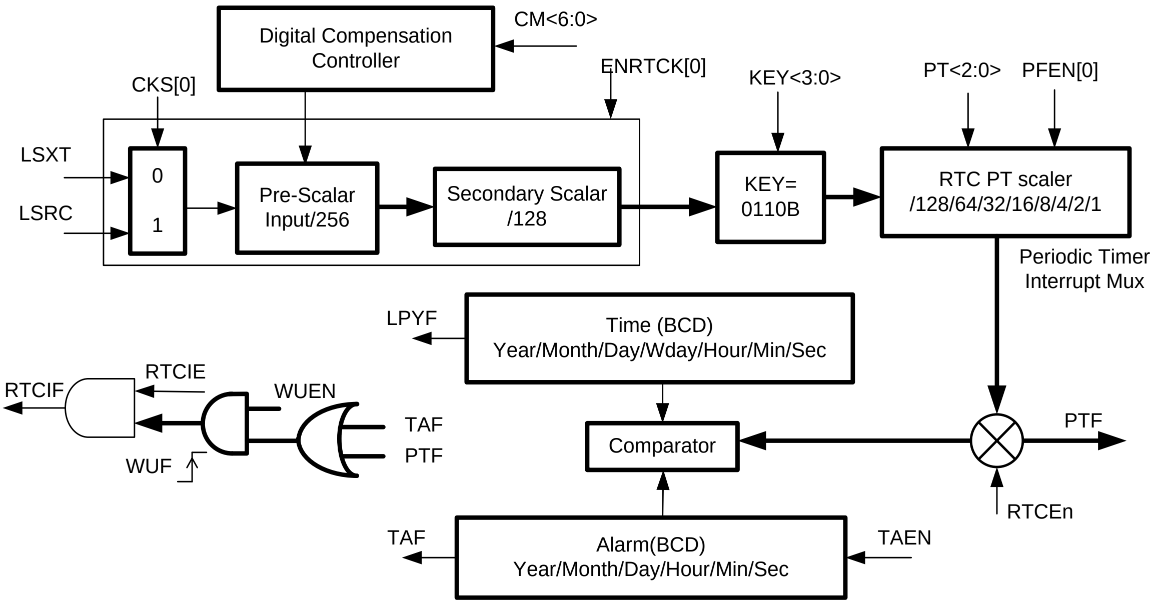

Digital Compensation Controller
CM<6:0>
ENRTCK[0]
CKS[0]
KEY<3:0>
PT<2:0>
PFEN[0]
LSXT
LSRC
0
1
Pre-Scalar Input/256
Secondary Scalar /128
KEY= 0110B
RTC PT scaler /128/64/32/16/8/4/2/1
Periodic Timer Interrupt Mux
LPYF
Time (BCD) Year/Month/Day/Wday/Hour/Min/Sec
RTCIE
RTCIF
WUEN
TAF
PTF
WUF
Comparator
PTF
RTCEn
TAF
Alarm(BCD) Year/Month/Day/Hour/Min/Sec
TAEN

FIG. 30-1 RTC structure diagram

Access the RTC register:

The frequency of the RTC clock is different from that of the system clock; thus the register will be updated after two RTC clock pulses if the user has written new data in the register. The RTC data should be updated frequently.

A protection button for writing data in the register is provided.

When writing data in the RTC register, the RTKEY button should be set as <0110>, and other values of the RTKEY button will not allow any data to be written in the RTC register. Please note that the RTC will not check the data format written in the register; thus, the user should be extremely careful with the write-in operation.

Enable the RTC:

It is necessary to write <0110> in the KEY0x41A00 [23:20] before writing data into the RTC register.

If the user wants to enable the RTC, the user should check whether the LSXT or LSRC can be used first. Then, set the RTCEN 0x41A00 [0] as <1>.

Frequency compensation:

The RTC allows the digital compensation for the clock input. The central frequency of the RTC is 32768Hz.

Any imperfect operations may result in the frequency offset. The digital compensation can be used to reduce the frequency offset.

The compensation method is to execute +/-2ppm at each step; the permissible maximal frequency change is +126ppm, and the permissible minimal frequency change is -126ppm. The maximal input frequency is 32772Hz, and the minimal input frequency is 32763Hz.

The maximal reference frequency that the user can input is 16MHz to measure the RTC clock during the manufacturing period.

The measurement value is calculated to obtain the compensation value. Then, the compensation value will be stored in the flash memory.

Once the system starts up, the compensation value will be loaded into the CM 0x41A04 [22:16].

Time information:

The time information is stored in the 0x41A08 and 0x41A0C registers, which use BCD format.

The user can set the time as the 24 hour system or 12 hour (AM/PM) system.

The time default value is 00:00:00 (hour/minute/second), and it is 24 hour system.

Calendar information:

The calendar information is stored in the 0x41A10 and 0x41A14 registers, which use BCD format.   The algorithm for leap year is performed by the hardware.

The effective year period is between 2012~2099.   If the LPYF0x41A00 [19] is <1>, it is the leap year.

The year is expressed by two digits, which stands for 20xx year.   The default date after the system is reset is 12/1/1 Sunday (January 2, 2012).

The maximal year is 99; and the year will become 00/1/1 after 99/12/31; but the leap year compensation will fail if the above condition takes place.

Week counter:

The RTC controller provides the information about one week.   The WDA0x41A14 [2:0] value is defined from 0 to 6, which stands for Sunday to Saturday respectively.

TAF Clock Alert interrupts:

If the 0x41A08/0x41A0C/0x41A10/0x41A14registers, conform to the registers, 0x41A18/0x41A1C and the TAEN 0x41A00 [03] is 1, the TAF0x41A00 [16] interrupt flag will be set as <1> to MCU.

PTF Periodic timer interrupts:

The periodic timer has 8 periodic options for interrupt: 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second.

Set the PTF 0x41A00 [18] as <1> to enable the periodic timer interrupt.   These periodic options are controlled by the PT0x41A04 [2:0].

WUF System wake-up interrupt:

When the MCU enters the idle mode, it can be wakened by the system wake-up interrupt program.

There are two sources able to wake up the MCU: the periodic timer interrupt and alert interrupt.   Set the WUF 0x41A00 [17] as <1> to enable the interrupt program.

30.2. Register address

RTC Register Address	31	24	23	16	15	8	7	0
RTC Base Address + 0x00 (0x41A00)	RTKEY		RTCC1		RTCC0M		RTCC0	
RTC Base Address + 0x04 (0x41A04)	RTCOM		RTCO		RTCPTM		RTPT	
RTC Base Address + 0x08 (0x41A08)	-		-		RTHRM		RTHR	
RTC Base Address + 0x0C (0x41A0C)	RTMIM		RTMI		RTSEM		RTSE	
RTC Base Address + 0x10 (0x41A10)	RTYEM		RTYE		RTMOM		RTMO	
RTC Base Address + 0x14 (0x41A14)	RTDAM		RTDA		RTWDM		RTWDA	

RTC Base Address + 0x18 (0x41A18)	-	RCHR	RCMI	RCSE
RTC Base Address + 0x1C (0x41A1C)	-	RCYE	RCMO	RCDA

-Reserved

30.3. Register function

30.3.1. Hardware RTC register RTCCR0

<table>
<tr><td colspan="10">RTC Base Address + 0x00 (0x41A00)</td></tr>
<tr><td>Symbol</td><td colspan="9">RTCCR0 (RTC Control Register 0)</td></tr>
<tr><td>Bit</td><td>[31:28]</td><td>[27:24]</td><td colspan="3">[23:20]</td><td>[19]</td><td>[18]</td><td>[17]</td><td>[16]</td></tr>
<tr><td>Name</td><td>-</td><td>MASK</td><td colspan="3">KEY</td><td>LPYF</td><td>PTF</td><td>WUF</td><td>TAF</td></tr>
<tr><td>RW</td><td>-</td><td>R0W-0</td><td colspan="3">RW-0</td><td>R-0</td><td>RW0-0</td><td>R-0</td><td>RW0-0</td></tr>
<tr><td>Bit</td><td>[15:14]</td><td>[13:8]</td><td>[7:6]</td><td>[5]</td><td>[4]</td><td>[3]</td><td>[2]</td><td>[1]</td><td>[0]</td></tr>
<tr><td>Name</td><td>-</td><td>MASK</td><td>-</td><td>PTEN</td><td>WUEN</td><td>TAEN</td><td>HRF</td><td>CKS</td><td>RTCEN</td></tr>
<tr><td>RW</td><td>-</td><td>R0W-0</td><td>-</td><td colspan="6">RW-0</td></tr>
</table>

<table>
<tr><td>Bit</td><td>Name</td><td colspan="2">Description</td></tr>
<tr><td rowspan="3">Bit[23-20]</td><td rowspan="3">KEY</td><td colspan="2">The secret key of the RTC register;
it can be locked to protect the register
to prevent data from being written in the register.</td></tr>
<tr><td>0110</td><td>The write-in secret key; it can lift the protection of the register,
and data can be written in the register only after it is unlocked.</td></tr>
<tr><td>Others</td><td>Lock the register to protect it;
no data can be written in the register.</td></tr>
<tr><td rowspan="3">Bit[19]</td><td rowspan="3">LPYF</td><td colspan="2">Leap year flag</td></tr>
<tr><td>0</td><td>The current year is not a leap year.</td></tr>
<tr><td>1</td><td>The current year is a leap year.</td></tr>
<tr><td rowspan="3">Bit[18]</td><td rowspan="3">PTF</td><td colspan="2">Timer wake-up interrupt flag</td></tr>
<tr><td>0</td><td>Normal</td></tr>
<tr><td>1</td><td>Timer wake-up is triggered.</td></tr>
<tr><td rowspan="3">Bit[17]</td><td rowspan="3">WUF</td><td colspan="2">Wake-up interrupt flag</td></tr>
<tr><td>0</td><td>Normal</td></tr>
<tr><td>1</td><td>The wake-up interrupt is triggered.</td></tr>
<tr><td rowspan="3">Bit[16]</td><td rowspan="3">TAF</td><td colspan="2">Alarm clock status flag</td></tr>
<tr><td>0</td><td>Normal</td></tr>
<tr><td>1</td><td>The alarm interrupt is triggered.</td></tr>
<tr><td>Bit[05]</td><td>PTEn</td><td colspan="2">RTC timer timing function enable control</td></tr>
</table>

		0	Disable
		1	Enable
Bit[04]	WUEn	RTC wake-up function enable control	
		0	Disable
		1	Enable
Bit[03]	TAEn	RTC alarm clock function enable control	
		0	Disable
		1	Enable
Bit[02]	HRF	RTC hour format configuration (24/12)	
		0	24 hour system
		1	12 hour system (PM/AM)
Bit[01]	CKS	RTC clock source input selection	
		0	External low-speed crystal oscillator source
		1	Internal low-speed crystal oscillator source
Bit[00]	RTCEn	RTC function enable control	
		0	Disable the function of the RTC.
		1	Enable the function of the RTC

Precautions:

(1) RTC Clock Source Selection "CKS" has a foolproof protection under LSXT but if CKS select Enable LSXT not the case, the circuit will automatically switch to LSRC as clock source.

(2) When the RTC is set to work in 24-hour time, RTC hours (Hour) units ranges from 0 to 23 counts cycle count, when the RTC is set to work in 12-hour time, RTC hours (Hour) unit count range loop count is 0 to 11

(3) When the HRF control bit is set to <1> of time, that is working in 12-hour, then if you want to do a write operation to the RTC time, in hours (Hour) units, if more than the number 12 will cause RTC invalid write operation.

(4) RTC register data is written should be noted that, if set to <0> control bits in the HRF time, that is working in 24-hour format, writing in time if it is greater than 12 hours, information can be normal write into the RTC register. And then if then HRF control bit is set to <1> when the RTC registers will result in the hours count-up unit constantly up, this time, even set to work in 12-hour, hour units do not count will be the loop count from 0 to 11, there will be an exception condition occurs.

30.3.2. Hardware RTC register RTCCR1

RTC Base Address + 0x04 (0x41A04)						
Symbol	RTCCR1 (RTC Control Register 1)					
Bit	[31]	[30:24]	[23]	[22:16]		
Name	-	MASK	-	CM		
RW	-	R0W-0	-	RW-0		
Bit	[15:08]		[7:5]	[4]	[3]	[2:0]
Name	MASK		-	CKH	-	PT
RW	R0W-0		-	RW-0	-	RW-0

Bit	Name	Description	
Bit[22~16]	CM	RTC clock frequency compensation value configuration	
		0111111	+126 PPM crystal oscillator frequency compensation(maximum)
		0111110	+124 PPM crystal oscillator frequency compensation
		…	Incremental step: +2 PPM crystal oscillator frequency compensation
		0000001	+2 PPM crystal oscillator frequency compensation
		0000000	0 PPM crystal oscillator frequency compensation
		1000000	0 PPM crystal oscillator frequency compensation
		1000001	-2 PPM crystal oscillator frequency compensation
		…	Decremental step: -2 PPM crystal oscillator frequency compensation
		1111110	-124 PPM crystal oscillator frequency compensation
		1111111	-126 PPM crystal oscillator frequency compensation (minimum)
Bit[4]	CKH	RTC high-speed clock source enable control; it is suggested that the function be enabled under the test mode.	
		0	Disable; the low-speed clock source is used.
		1	Enable; the high-speed clock source is used.
Bit[2~0]	PT	Timer timing wake-up time configuration	
		000	1/128 s
		001	1/64 s
		010	1/32 s
		011	1/16 s
		100	1/8 s
		101	1/4 s
		110	1/2 s
		111	1 s

30.3.3. Hardware RTC register RTCHRC

RTC Base Address + 0x08 (0x41A08)					
Symbol	RTCHRC (RTC Hour Control Register For calendar)				
Bit	[31:15]				
Name	RSV				
RW	R-0				
Bit	[14:08]	[7]	[6]	[5:4]	[3:0]
Name	MASK	-	PM	10HR	1HR
RW	R0W-0	-	RW-0		

Bit	Name	Description	
Bit[6]	PM	Real time clock hour format am/pm	
		0	AM or 24 hour system
		1	PM (it should be set as 1 if, the HRF is set as 1.)
Bit[5~4]	10HR	The tens place of the hour (BCD code format)	
		00	0
		01	1
		10	When 2 (HRF=1) / HRF=0, it is invalid.
		11	Invalid
Bit[3~0]	1HR	The one’s place of the hour (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid

30.3.4. Hardware RTC register RTCCR3

RTC Base Address + 0x0C (0x41A0C)				
Symbol	RTCSMC (RTC seconds and min Control Register For calendar)			
Bit	[31:24]	[23]	[22:20]	[19:16]
Name	MASK	-	10MIN	1MIN
RW	R0W-0	-	RW-0	RW-0
Bit	[15:08]	[07]	[06:04]	[03:00]
Name	MASK	-	10SEC	1SEC
RW	R0W-0	-	RW-0	RW-0

Bit	Name	Description	
Bit[22~20]	10MIN	The tens place of the minute (BCD code format)	
		000	0
		001	1
		010	2
		011	3
		100	4
		101	5
		110	6
		111	Invalid
Bit[19~16]	1MIN	The one's place of the minute (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[6~4]	10SEC	The tens place of the second (BCD code format)	
		000	0
		001	1
		010	2
		011	3
		100	4

		101	5
		110	6
		111	Invalid
Bit[3~0]	1SEC	The one's place of the second (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid

30.3.5. Hardware RTC register RTCCR4

RTC Base Address + 0x10 (0x41A10)				
Symbol	RTCYMC (RTC Year and Month Control Register For Calendar)			
Bit	[31:24]	[23:20]		[19:16]
Name	MASK	10YEAR		1YEAR
RW	R0W-0	RW-1		RW-2
Bit	[15:08]	[07:05]	[04]	[03:00]
Name	MASK	-	10MO	1MO
RW	R0W-0	-	RW-0	RW-1

Bit	Name	Description	
Bit[23~20]	10YEAR	The tens place of the year (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9

		Other values	Invalid
Bit[19~16]	1YEAR	The one's place of the year (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[4]	10MO	The tens place of the month (BCD code format)	
		0	0
		1	1
Bit[3~0]	1MO	The one's place of the month(BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid

30.3.6. Hardware RTC register RTCCR5

RTC Base Address + 0x14 (0x41A14)				
Symbol	RTCDWC (RTC Date and week Control Register For calendar)			
Bit	[31:24]	[23:22]	[21:20]	[19:16]
Name	MASK	-	10DAT	1DAT
RW	R0W-0	-	RW-0	RW-1
Bit	[15:08]	[07:03]		[02:00]
Name	MASK	-		WDA
RW	R0W-0	-		RW-0

Bit	Name	Description	
Bit[21~20]	10DAT	The tens place of the date (BCD code format)	
		00	0
		01	1
		10	2
		11	3
Bit[19~16]	1DAT	The one's place of the month (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[2~0]	WDA	The value of the Week (BCD code format)	
		000	Sunday
		001	Monday
		010	Tuesday
		011	Wednesday
		100	Thursday
		101	Friday
		110	Saturday
		111	Invalid

30.3.7. Hardware RTC register RTCCR6

RTC Base Address + 0x18(0x41A18)							
Symbol	RTCHRA (RTC Hour and min and seconds Control Register for alarm)						
Bit	[31:24]			[23]	[22]	[21:20]	[19:16]
Name	RSV			-	CPM	10CHR	1CHR
RW	R-0			-	RW-0		
Bit	[15]	[14:12]	[11:8]	[7]	[6:4]	[3:0]	
Name	-	10CMI	1CMI	-	10CSE	1CSE	
RW	-	RW-0		-	RW-0		

Bit	Name	Description	
Bit[22]	CPM	The format of the alarm clock is am/pm.	
		0	AM or 24 hour system
		1	PM (when HRF=1, the bit should be set as 1.)
Bit[21~20]	10CHR	The tens place of the hour under the alarm clock mode (BCD code format)	
		00	0
		01	1
		10	When 2 (HRF=1) / HRF=0, it is invalid.
		11	invalid
Bit[19~16]	1CHR	The one’s place of the hour under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[14~12]	10CMI	The tens place of the minute under the alarm clock mode (BCD code format)	
		000	0
		001	1

		010	2
		011	3
		100	4
		101	5
		110	6
		111	Invalid
Bit[11~8]	1CMI	The one's place of the minute under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[6~4]	10CSE	The tens place of the second under the alarm clock mode (BCD code format)	
		000	0
		001	1
		010	2
		011	3
		100	4
		101	5
		110	6
		111	Invalid
Bit[3~0]	1CSE	The one's place of the second under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4

		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid

30.3.8. Hardware RTC register RTCCR7

RTC Base Address + 0x1C(0x41A1C)						
Symbol	RTCYMDA (RTC Year /month/date Control Register For alarm)					
Bit	[31:24]			[23:20]		[19:16]
Name	RSV			10CYE		1CYE
RW	R-0			RW-1		RW-2
Bit	[15:13]	[12]	[11:8]	[07:06]	[05:04]	[03:00]
Name	-	10CMO	1CMO	-	10CDAT	1CDAT
RW	-	RW-0	RW-1	-	RW-0	RW-1

Bit	Name	Description	
Bit[23~20]	10CYE	The tens place of the year under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[19~16]	1CYE	The one's place of the year under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2

		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid
Bit[12]	10CMO	The tens place of the month under the alarm clock mode (BCD code format)	
		0	0
		1	1

Bit	Name	Description	
Bit[11~8]	1CMO	The one's place of the month under the alarm clock mode (BCD code format)	
		0000	0
		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Others	Invalid
Bit[5~4]	10CDAT	The tens place of the date under the alarm clock mode (BCD code format)	
		00	0
		01	1
		10	2
		11	3
Bit[3~0]	1CDAT	The one's place of the date under the alarm clock mode (BCD code format)	
		0000	0

		0001	1
		0010	2
		0011	3
		0100	4
		0101	5
		0110	6
		0111	7
		1000	8
		1001	9
		Other values	Invalid

31. POWER-SAVING MODE INTRODUCTION

31.1. Overall description

The paragraph will describe different power modes and their corresponding function modules.

Under the active mode, all peripheral circuits can be enabled, and the clock of the MCU is HS_CK or LS_CK clock; under the mode, the system can freely switch to other modes and have shortest response time.

Under the low-power mode, all analog circuits can be enabled and the clock of the MCU is LS_CK clock; under the mode, the MCU works under the lowest frequency and the system can switch to other modes by executing instructions.

There are three power-saving modes, Including Sleep Mode, Idle Mode, Wait mode, allows the MCU to stop executing instructions.

These modes can be disabled by the interrupt. Once the interrupt is triggered, The MCU will leave the power saving mode. Before entering power-saving mode, the corresponding interrupt vectors must be enabled to wake up the chip. Otherwise, the chip can't achieve power saving effect. For example, in the sleep mode, the timer interrupt is invalid, and the chip only can be wakened up by the communication interrupt or IO port external interrupt or reset. In details, refer to the table below, the table lists the wake-up interrupt vector for each power-saving mode. It should be noted in different power-saving mode, only the number of functional modules can be enabled, and only some of the interrupt functions can wake up the MCU from power-saving mode.

31.2. Interrupt point configuration

When the CPU is under different operating modes, the interrupt-triggered items supported by the CPU are also different; the following table shows the interrupt and wake-up levels supported by each function. Similarly, different modes have different current consumption; the current consumption from high to low is: active mode > wait mode > idle mode > sleep mode.

Note: If you want to enter the power saving settings, should be performed before entering the power saving mode, the CPU operating frequency of the low frequency after the first change to LPO, then turn off the high frequency HAO. It has turned the analogy power output is also required to make the corresponding closing action, after such power saving mode can be achieved with specification (Datasheet) as current consumption.

Wake-up time: Sleep Mode (sleep mode)> Idle Mode (standby mode)> Wait Mode (standby mode) Sleep Mode and Idle Mode, although many are still saving ratio Wait Mode, but through interrupt wake-up time is relatively long .

Wake-up Interrupt level: Such as I2C TX pin interrupt function can only support Idle Mode, Wait Mode and Active Mode, This means that when the chip enters Sleep mode, the chip cannot be woken up by the I2C TX pin so that the chip can enter the break point. For example, after the chip enters the sleep mode, only following actions and interrupts can make the chip leave the sleep mode: Power On Reset, Reset PIN, I2C RX IRQ, UART1/2 RX IRQ, SPI RX IRQ, CMP IRQ, PT1 IRQ and PT2 IRQ, etc.

Interrupt/Reset Mode	Sleep Mode		Idle Mode		Wait Mode		Active Mode		Note
	Enter	leave	Enter	leave	Enter	leave	Enter	leave	
Power On Reset		V		V		V	V	V	Chip Reset
Reset PIN		V		V		V	V	V	Chip Reset
WDT Reset				V		V	V	V	WDT Reset Type
I2C TX IRQ			V	V	V	V	V	V	I2CIE
I2C RX IRQ	V	V	V	V	V	V	V	V	I2CIE
I2C Error IRQ						V	V	V	I2CEIE
UART1/2 TX IRQ			V	V	V	V	V	V	UTXIE
UART1/2 RX IRQ	V	V	V	V	V	V	V	V	URXIE
SPI TX IRQ			V	V	V	V	V	V	STXIE
SPI RX IRQ	V	V	V	V	V	V	V	V	SRXIE
RTC IRQ			V	V	V	V	V	V	RTCIE
WDog IRQ			V	V	V	V	V	V	WDTIE
TMA IRQ			V	V	V	V	V	V	TMAIE
TMB IRQ			V	V	V	V	V	V	TMBIE
TMC IRQ			V	V	V	V	V	V	TMCIE
ADC IRQ			V	V	V	V	V	V	ADCIE
CMP IRQ	V	V	V	V	V	V	V	V	CPIE
OPAMP IRQ					V	V	V	V	OPOIE
PT1 IRQ	V	V	V	V	V	V	V	V	PT1IE
PT2 IRQ	V	V	V	V	V	V	V	V	PT2IE
Debug Exception						V	V	V	EDM

Mode	Setting	Description

Wait Mode	sys_04=0xFF10;
	asm("syscall 13");
		//Wait Set
		//Wait Mode

Idle Mode	sys_04=0xFF10;
	asm("syscall 11");
		//Idle Set
		//Idle Mode

Sleep Mode	sys_04=0xFF00;
	asm("syscall 12");
		//Sleep Set
		//Sleep Mode

SYS_04 represents register address 0x40104, refer to Chapter IV

Use CPU instruction asm ("syscall 10") to entere Wait Mode
Use CPU instruction asm ("syscall 11") to enter the Idle Mode
Before entering Wait Mode or Idle Mode, you should set 0x40104 [4] = <1> first

Use CPU instruction asm ("syscall 12") to enter Sleep Mode
Before entering Sleep Mode, you should first set 0x40104 [4] = <0>

The state of 0x40400 [0] will affect the Sleep Mode power consumption. The details are as follows:

- 0x40400 [0] = 0b -> When the sleep mode wake-up, this bit should be set to 0, the LDO into the normal mode.
- 0x40400 [0] = 1b -> Before entering the sleep mode, set this bit to 1 to make the LDO enters a low-power mode.

At sleep mode power consumption -> 0x40400 [0] = 0b --------- 3.5uA
At sleep mode power consumption -> 0x40400 [0] = 1b --------- 2.5uA

32. LCD DRIVER

32.1. Overall description

The LCD driver circuit is for the TN-LCD and STN-LCD, and it has the following features:

Built-in voltage regulating circuit (Regulated charge pump)
4-stage adjustable driving voltage levels
Support four kinds of LCD waveform operation
  1/3 Duty, 1/3 bias. (3-mux,1/3bias)
  1/4 Duty, 1/3 bias. (4-mux,1/3bias)
  1/5 Duty, 1/3 bias. (5-mux,1/3bias)
  1/6 Duty, 1/3 bias. (6-mux,1/3bias)
Selectable input clock sources and programmable output frequency
With blinking capability

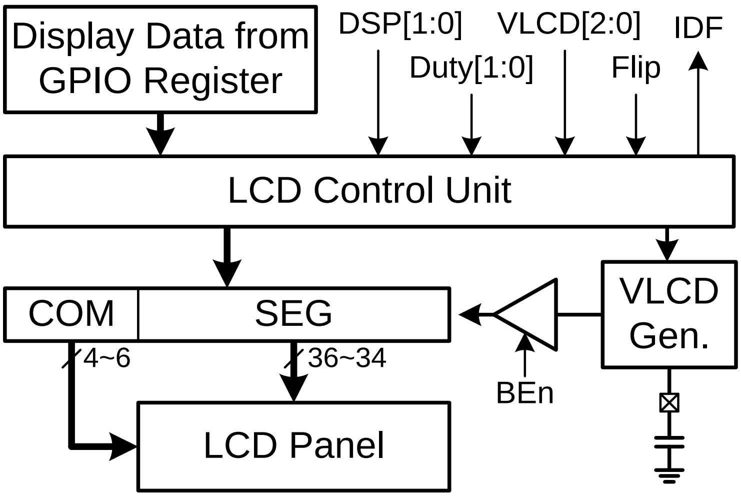
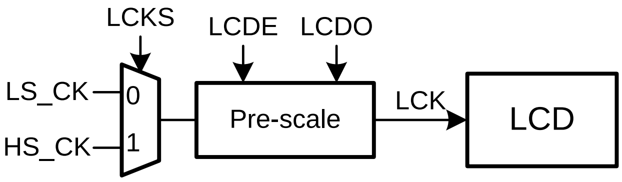


FIG. 32-1 LCD structure diagram

LCD initializes setting:

(1)Operating frequency and output frequency setting: LCD operating frequency can be selected by the control bit LCKS 0x40310 [0] of LS_CK or HS_CK. The LCD control bit LCDE 0x40310 [3: 1] and the control bit LCDO 0x40310 [6: 4] Source 1 and 2-stage divider settings, to provide the appropriate operating frequency to the LCD output amplitude frequency.



(2)Charge Pump voltage supply and the LCD operating voltage’s voltage source setting is VLCD, which produced two methods: To provide VLCD voltage externally, the VLCD 0x41B00 [2: 0] of the scratchpad must be set to <001> VLCD R-Type, then the external VLCD pin is used to determine the LCD operating voltage. When pushing the bigger size loader or larger LCD monitor, the LCD output buffer could set as BEn 0x41B00 [3] <1>, enable buffer function to increase the LCD drive capability. Set the Charge Pump circuit controller VLCD 0x41B00 [2: 0] in the range 011b ~ 101b, can produce the different VLCD voltage source to supply LCD. VLCD 0x41B00 [2: 0] can be set to 4 different operating voltages and must be enabled when the Charge Pump circuit is enabled.
(3)register control bits Duty 0x41B00 [5: 4] can set LCD operating waveform, frequency and amplitude of the waveform operation must be set correctly according to the external LCD monitor specifications, otherwise the LCD display will appear blur or byte display abnormal phenomenon.
(4) Set the LCD multiple function IO port PT6 ~ PT10 and COM5 / COM4 operating modes, which means set register 0x41B04 ~ 0x41B08.
(5) Write data to the LCD data register LCD0 ~ LCD17, make LCD information display.

32.2. Register address

LCD Register Address	31	24	23	16	15	8	7	0
LCD Base Address + 0X00 (0X41B00)	Mask0		REG0		Mask1		REG1	
LCD Base Address + 0X04 (0X41B04)	PT9LEn		PT8LEn		PT7LEn		PT6LEn	
LCD Base Address + 0X08 (0X41B08)	-		-		-		REG2	

-Reserved

32.3. Register function

32.3.1. Register LCDCR0

<table>
<tr><td colspan="9">LCD Base Address + 0x00 (0x41B00)</td></tr>
<tr><td>Symbol</td><td colspan="8">LCDCR0 (LCD Control Register 0)</td></tr>
<tr><td>Bit</td><td>[31:24]</td><td colspan="2">[23:21]</td><td colspan="2">[20]</td><td colspan="2">[19:18]</td><td>[17:16]</td></tr>
<tr><td>Name</td><td>MASK</td><td colspan="2">-</td><td colspan="2">IDF</td><td colspan="2">-</td><td>DSP</td></tr>
<tr><td>RW</td><td>R0W-0</td><td colspan="2">-</td><td colspan="2">R-1</td><td colspan="2">-</td><td>RW-0</td></tr>
<tr><td>Bit</td><td>[15:8]</td><td>[7]</td><td colspan="2">[6]</td><td colspan="2">[5:4]</td><td>[3]</td><td>[2:0]</td></tr>
<tr><td>Name</td><td>MASK</td><td>-</td><td colspan="2">Flip</td><td colspan="2">Duty</td><td>BEn</td><td>VLCD</td></tr>
<tr><td>RW</td><td>R0W-0</td><td>-</td><td colspan="2">RW-0</td><td colspan="3">RW-1</td><td>RW-0</td></tr>
</table>

Bit	Name	Description	
Bit[20]	IDF	LCD Idle control flag	
		0	Active
		1	Idle
Bit[17~16]	DSP	LCD display mode	
		00	Normal mode
		01	The LCD is turned on no matter what the input is.
		10	The LCD is turned off no matter what the input is.
		11	Normal mode
Bit[6]	Flip	Reverse the order between COM and SEG.	
		0	Normal
		1	Inversed
Bit[5~4]	Duty	LCD operating period selection	
		00	1/3 Duty
		01	1/4 Duty
		10	1/5 Duty
		11	1/6 Duty
Bit[3]	BEn	VLCD buffer control	
		0	Disable
		1	Enable(it should be enabled and the functions of the LCD can be used normally.)
Bit[2~0]	VLCD	VLCD mode	
		000	Disable(Charge Pump is disabled, VLCD Ris disabled, VLCD buffer is disabled)
		001	R-Type(Charge Pump is disabled, VLCD R is enabled)
		010	3.3V(Charge Pump is enabled, VLCD R is disabled)

		011	3.0V (Charge Pump is enabled, VLCD R is disabled)
		100	2.7V (Charge Pump is enabled, VLCD R is disabled)
		101	2.4V (Charge Pump is enabled, VLCD R is disabled)
		110	Disable (Charge Pump is disabled, VLCD R is disabled, VLCD buffer is disabled)
		111	Disable (Charge Pump is disabled, VLCD R is disabled, VLCD buffer is disabled)

32.3.2. LCD register LCDCR1

LCD Base Address + 0x04 (0x41B04)		
Symbol	LCDCR1 (LCD Control Register 1)	
Bit	[31:24]	[23:16]
Name	PT9LEn	PT8LEn
RW	RX-0	
Bit	[15:8]	[7:0]
Name	PT7LEn	PT6LEn
RW	RX-0	

Bit	Name	Description	
Bit[31~24]	PT9LEn	PT9.x mode selection	
		0	I/O mode
		1	LCD mode
Bit[23~16]	PT8LEn	PT8.x mode selection	
		0	I/O mode
		1	LCD mode
Bit[15~8]	PT7LEn	PT7.x mode selection	
		0	I/O mode
		1	LCD mode
Bit[7~0]	PT6LEn	PT6.x mode selection	
		0	I/O mode
		1	LCD mode

32.3.3. LCD register LCDCR2

LCD Base Address + 0x08 (0x41B08)			
Symbol	LCDCR2 (LCD Control Register 2)		
Bit	[31:16]		
Name	-		
RW	-		
Bit	[15:4]	[3:2]	[1:0]
Name	-	COMLEn	PT10LEn
RW	-	RX-0	RX-0

Bit	Name	Description	
Bit[3~2]	COMLEn	COM5/COM4 mode selection	
		0	I/O mode
		1	LCD mode
Bit[1~0]	PT10LEn	PT10.x mode selection	
		0	I/O mode
		1	LCD mode

32.3.4. LCD register LCDCR3

LCD Mode Base Address + 0X24 (0X41F24)			
Symbol	LCDCR0 (LCD Control Register 0)		
Bit	[31:24]	[23:16]	
Name	MASK	-	
RW	R0W-0	-	
Bit	[15:8]	[7:2]	[1:0]
Name	MASK	-	EN_RShift
RW	R0W-0	-	RW-0

Bit	Name	Description	
Bit[1:0]	EN_RShift	EN_RShift Bit	
		1	Set 1
		0	Set 0

Register address 0X41F24 Need to control MASK corresponding BIT1 ~ BIT0

VLCD All Mode View:

Add	0X41F24		0X41B00			MODE
Bit	1	0	2	1	0	
Name	EN_Rshift	EN_Rshift0	VLCD0	EN_Rshift1	EN_Rshift0	V
[01]	0	0	0	1	1	VLCD=3.43V
[02]	0	0	1	0	0	VLCD=3.16V
[03]	0	0	1	0	1	VLCD=2.93V
[04]	1	1	1	0	1	VLCD=2.73V
[05]	0	1	1	0	1	VLCD=2.55V

Note: VLCD voltage is about +/- 10% margin of error after manufactured, if you want a more accurate VLCD voltage, you can use VLCD voltage calibration function.In VLCD voltage calibration part, you can use HYCON C library DrvLCD_VLCDTrim this function, you can select different segments VLCD voltage, and through this setting function, you can control the voltage error range is within +/- 5%. Detail VLCD upper and lower limits specification,please refere to page51 description of DS-HY16F198B_EN file. Function can refer to the following instructions or file APD-HY16IDE007_EN:

-function

unsigned char DrvLCD_VLCDTrim (short Umode)

- Function

Conduct calibration of Chip VLCD according to VLCD calibration parameters after manufactory; setting register is 0x41B00 [2: 0]

-Input parameters

Umode [in] to be corrected VLCD voltage mode selection;

1: VLCD ~ 3.43V; 2: VLCD ~ 3.16V

3: VLCD ~ 2.93V; 4: VLCD ~ 2.73V

5: VLCD ~ 2.55

32.4. LCD RAM function

LCD Register Address 0X41B04 and 0X41B08 may decide to PT 6 ~ PT 10 is set to GPIO Mode or LCD Mode. When set LCD Mode, you can register as a PT6 ~ PT10 IO LCD RAM usage control LCD display.

LCD Mode Address	Bit[31:24]	Bit[23:16]	Bit[15:08]	Bit[07:00]
0X40850	MASK	SEG3	MASK	SEG2
0X40854	MASK	SEG5	MASK	SEG4
0X40858	MASK	SEG7	MASK	SEG6
0X4085C	MASK	SEG9	MASK	SEG8
0X40860	MASK	SEG11	MASK	SEG10
0X40864	MASK	SEG13	MASK	SEG12
0X40868	MASK	SEG15	MASK	SEG14
0X4086C	MASK	SEG17	MASK	SEG16
0X40870	MASK	SEG19	MASK	SEG18
0X40874	MASK	SEG21	MASK	SEG20
0X40878	MASK	SEG23	MASK	SEG22
0X4087C	MASK	SEG25	MASK	SEG24
0X40880	MASK	SEG27	MASK	SEG26
0X40884	MASK	SEG29	MASK	SEG28
0X40888	MASK	SEG31	MASK	SEG30
0X4088C	MASK	SEG33	MASK	SEG32
0X40890	MASK	SEG35	MASK	SEG34
0X40894	MASK	SEG1	MASK	SEG0

32.5. LCD power saving features

When HY16F19xB enter power-saving mode, LCD settings also need your attention. Before entering the power saving mode, if not discharged first, LCD might happen Blur. When entering the power saving mode, you can refer to the following settings, By that, you can ensure that LCD is discharge before entering the power saving Mode.

```
DrvLCD_DisplayMode (2); // 2: whether to enter any value, LCD is all off mode
DrvLCD_VLCDMode (E_VLCD_DISABLE);
while ((inw (0x41B00) & (1 << IDF)) == 0); // Wait LCD Idle, IDF = 20
asm ("syscall 12"); // wait = 10; idle = 11; sleep = 12
```

Note: The above content usage to HYCON C library, you can refer to the file "APD-HY16IDE007_EN".

32.6. LCD frequency(LCK) setting

Avoid to occur the LCD flicker, the LCD frame Frequency should be near 60Hz.
EX : 4 com LCD, the LCD frequency(LCK) have to set 240Hz(4*60Hz) at least. But the LCK of HY16F198B is (1/2T), so LCK have to set 480Hz(240Hz*2) not 240Hz.

33. Revisions

The following describes the major changes made to the document, excluding the punctuation and font changes.

Version	Page	Revision Summary	Date
V01	ALL	First edition	2017/02/13
V02	ALL	1. Remove the GPIO multiplex function of the PT3.2 / PT3.3 pin, which only retains the AIO4 / AIO5 analog function. 2. Modify the PWMA to PWMG description to PWM Duty Cycle = (PWM Duty) * (PWM Period). 3. Modify the TPS initialization settings and calculation methods. 4. Added ADC input impedance (RADC) description, ADC network diagram (ADCLK renamed ADCK). 5. Modify R_{PU}=85k (Internal pull high resistor) description. 6. Modify 8-bit Resistance Ladder figure 24-2 block diagram. Modify the application circuit system 01.	2017/10/12
V03	ALL	1. Recover PT3.2 & PT3.3 Multiplexing pin function. 2. Added PTUR2E description in register 0x4084C 3. Correction the TPS calculation methods. 4. Correction ADC reference Voltage input range in table 22.4 5. Correction the VLCD description in register 0x41B00[2:0] 6. Correction the PTPW[2:0] description in Table 10.1	2018/06/19
V04	ALL	1. Added WDT Reset description 2. Added 32.6 LCD frequency(LCK) setting	2020/03/09
V05	ALL	1. Added memory structures description, support (Timeout Entry)4-WIRE and (CheckPin Entry) 5-WIRE UART interface ROM ISP Bootloader function. 2. Revise Timer B block diagram (TBCLK correct to TBCK. TMBC0/TMBC1/TMBC2 correct to TBC0/TBC1/TBC2, ENTMB correct to TBEN) Revise TMB counting waveform Revise PWMA~G counting waveform Revise PWMA~G duty cycle and formula 3. Revise Timer B2 block diagram (TB2CLK correct to TB2CK, TMB2C0/TMB2C1/TMB2C2 correct to TB2C0/TB2C1/TB2C2, ENTMB2 correct to TB2EN) 4. Revise Timer C block diagram (ENTMC correct to TCEN)	2021/05/11

		5. Added ADCMP register(0x41104[28]). Revise CMP block diagram(Added ADCMP) . 6. Revise I2C block diagram(I2CCK correct to I2CK).	