HY15S41

User's Guide

Mixed Signal Microcontroller

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. Terms and Definition

1.1. Index of frequently-used terms

1MW	1MegaWord
1KB	1KiloByte
ADC	Analog to Digital Converter
Bit	bit
BOR	Brown-Out Reset
BSR	Bank Select Register
Byte	Byte
CCP	Capture and Compare
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DM	Data Memory
ECAP	Enhance Comparator
FSR	File Select Register
GPR	General Purpose Register
HAO	High Accuracy Oscillator
LNOP	Low Noise OP AMP
LPO	Low Power Oscillator
LSB	Least Significant Bit
MEM	Memory
MPM	Main Program Memory
MSB	Most Significant Bit
OTP	One Time Program-EPROM
PC	Program Counter
PPF	PWM and PFD
ΣΔADC	Sigma-Delta ADC
SR	Special Register
SRAM	Static Random Access Memory
STK	Stack
WDT	Watch Dog Timer
WREG	Work Register

1.2. Terms related to register

[ ]	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

rw	Read/Write
r	Read only
r0	Read as 0
r1	Read as 1
w	Write only
w0	Write as 0
w1	Write as 1
h0	cleared by Hardware
h1	set by Hardware
u0	cleared by User
u1	set by User
-	Not use
!	users are forbidden to change
u	unchanged
x	unknown
d	depends on condition

2. CPU

2.1. CPU Core

In order to make the CPU core (H08) of the CPU have higher execution efficiency, we adopt the concept of Harvard architecture to separate the program memory from the data memory, so the program memory allows the users to conveniently write the programs.

The characteristics of the CPU include:

- The design structure which separates the program memory from the data memory increases the execution speed of the instructions, and betters the efficiency of the CPU.
- Up to 66 operation instructions are provided, including the 16-bit look-up table, 8x8 hardware multiplier, data memory block switching and stack control.
- The data transfer from Register A to Register B can be finished by one instruction without changing the data of the work register.
- The data transfer of up to 16-bit FSR register can be finished by one instruction, and the table look-up instruction can address a program memory up to 1MW.
- The operations of the data memory include the data transfer of the program counters (PC), the status registers and the stack registers.
- The CPU core is the simplified H08C (Compatible H08A) core.

2.2. Memory

The composition of the memory can be classified into two types; one is the program memory, which is composed of the OTP; the other one is the data memory, which is composed of the SRAM. The products with different models have different memory sizes; thus, it is necessary to pay more attention to the specification description of a product when reading the user's guide of the product.

Program memory:

Main program memory (MPM)

Program Counter (PC)

Stack (STK)

Data memory:

Special Register (SR)

General Purpose Register (GPR)

Summary of registers related to memory: (x means that it is composed of several registers)

PC[11:0]	PCHSR[2:0],PCLATH[3:0],PCLATL[7:0]
TOS[10:0]	TOSH[2:0],TOSL[7:0]
FSRx[8:0]	FSRxH[8],FSRxL[7:0]
INDFx	INDF0[7:0],INDF1[7:0]
POINCx	POINC0[7:0], POINC1[7:0]
PODECx	PODEC0[7:0], PODEC1[7:0]
PRINCx	PRINC0[7:0], PRINC1[7:0]
PLUSWx	PLUSW0[7:0], PLUSW1[7:0]
STKCN	STKFL[0],STKOV[0],STKUN[0],SKPRT[2:0]
PSTAT	SKERR[0]
BSRCN	BSR[0]

2.2.1. Program Memory

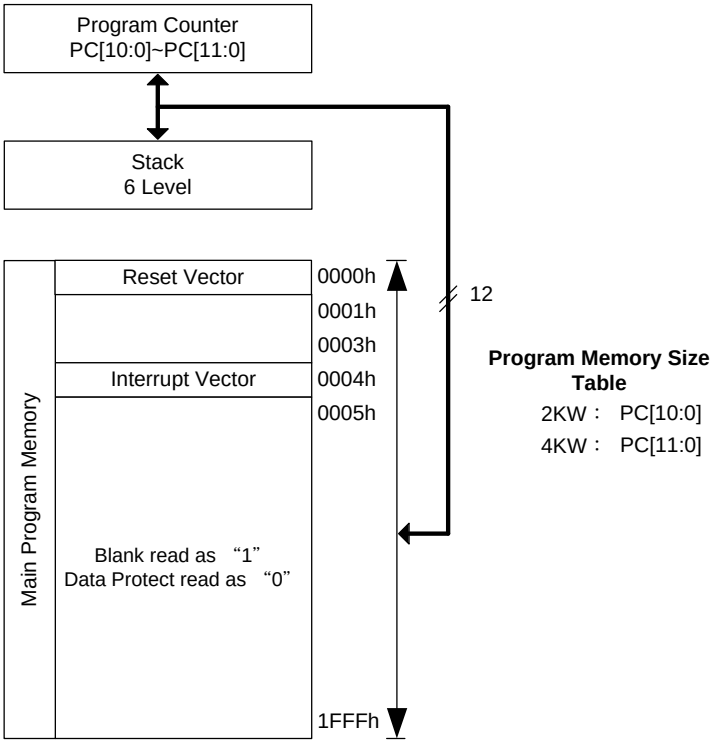


Figure 2-1 Structure diagram of program memory

2.2.1.1. Main program memory (MPM)

The structure of the main program memory is as follows:

- Interrupt Vector
- Reset Vector

Its addressing ability is from 0x0000h to 0x1FFFh, and the total capacity is 8192 bits; the products with different models may have different capacities.

When the chip has no program written in, the data type of all addresses is 1; after a program is written in the chip, each address is 1 or 0 according to the data type written in. Please note that if the assembling options of the simulation software (HYIDE) provide the burning protection function when developing the program, the data type read from all addresses of the chip during the burning process is 0.

2.2.1.2. Program counter (PC)

The program counter PC is composed of the transmit shift register PCSR and the buffer register PCLAT, as shown in figure 2-2.

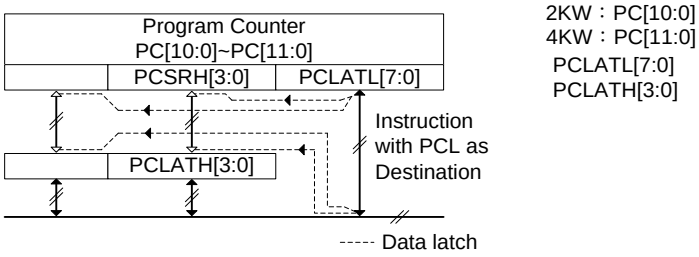


Figure 2-2 Structure diagram of program diagram

The chip of the development tools used by PC[11:0] has a data length of 12 bits, which is composed of two specially registers PCSRH [3:0] and PCLATL [7:0]. PCLATL[7:0] and PCLATH[3:0] can be directly read/write, but PCSRH [3:0] cannot be directly read/write; it is necessary to indirectly read/write it via the buffer register PCLATH[3:0].

- When reading PC[11:0], it is necessary to read PCLATL[7:0], and then read PCLATH[3:0]; if the order is incorrect, the correct data cannot be read.
- When writing the data in PC[11:0], it is necessary to write the data in PCLATH[3:0] and then write the data in PCLATL[7:0]; if the order is incorrect, the correct data cannot be written in.

2.2.1.3. Stack (STK)

The stack STK is mainly composed of the stack pointer control register STKCN, the stack error flag SKEER and the stack error reset controller SKRST.

When a stack overflows or underflows, the execution result of the program may be unpredictable; the chip should be reset if necessary. During the program development process, the stack reset control bit SKRST[1] can be set to be <1> via software; when the stack underflows or overflows, a reset signal occurs to set SKERR to be <1> and then reset the chip.

- Full flow: set STKFL as <1>, and PC[11:0] remains unchanged.
- Underflow: set STKUN as <1>，move PC[11:0] to 0x00000h, and point the position stack pointer SKPRT to 0 Level. If SKRST is set as <1>, a reset signal occurs after underflow, and SKERR is set as <1>; STKUN is set as <0> after reset.
- Overflow: set STKOV to be <1>, and PC[11:0] remains unchanged, but SKPRT still stays at the last layer and new values are inputted; in other words, the data inputted last are saved after full flow. If SKRST is set as <1>, a reset signal occurs after overflow and SKERR is set as <1>; STKOV is set as <0> after reset.
- Error: set SKERR as <1>, and the chip has suffered a stack error. If SKRST is set as <1>, a reset signal occurs after overflow and SKERR is set as <1>; STKUN and STKOV are set as <0> after reset.

[1] SKRST will give controlling bits of recovery signal for stack error. It cannot be read/edited but only be set in software at the stage of developing program. It has to be decided whether the recovery signal will be given on stack error. If recovery is decided, the bit will be set at <1> or cleared as <0> on the contrary after the chip is given power.

2.2.1.4. Register description-Program memory controller

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1											
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition											
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**R/W**
018h	SKCN	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. .000	u$$. .$$$
01Ah	PCLATH	-	-	-	-	PC[11]	PC[10]	PC[9]	PC[8]	.... .0000	 0000
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
02Ch	PSTATUS	POR	PD	TO	IDL	RST	SKERR	MCO	-	$000 $00.	uu$u u$u.

Table 2-1 Program memory control register

STKCN: Stack controller

Bit	Name	Description
Bit7	STKFL	Stack full flow flag <0> Not occur. <1> Occur.
Bit6	STKUN	Stack underflow flag <0> Not occur. <1> Occur.
Bit5	STKOV	Stack overflow flag <0> Not occur. <1> Occur.
Bit3~0	SKPRT[3:0]	Stack pointer register <000> 0^{th} layer，TOS[10:0]=0x0000h ⋮ <110> 6^{th} layer <111> 7^{th} layer

PCLATH: high byte of program counter, PC[11:8]

PCLATL: low byte of program counter, PC[7:0]

PSTAT: Status register

Bit	Name	Description
Bit2	SKERR	Stack error reset occurrence flag <0> Not occur. <1> Occur.

2.2.2. Data memory (DM)

The data memory DM is composed of the specially register SR and the general-purpose register GPR, and each block is 256byte. The 128-byte specially register and the 128-byte general purpose register are as shown in Figure 2-3.

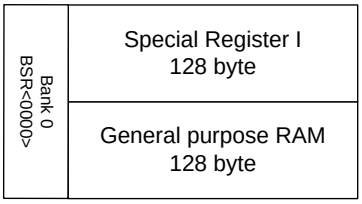


Figure 2-3 Structure diagram of data memory

2.2.2.1. Memory and instructions

H08 instruction set can be classified into A version and B version; there are a lot of differences between the two versions, such as addressing ability, hardware multiplier, table look-up instruction, support functions and parameter definitions; the chapter only discusses the definitions of the instruction memory parameters. Please refer to the chapter "Instruction set" for more detailed description of the instruction parameters.

In the instruction set, the instructions with the address operation function have at most three parameters, including "f", "d" and "a".

"f" stands for the address of the data or the data memory,

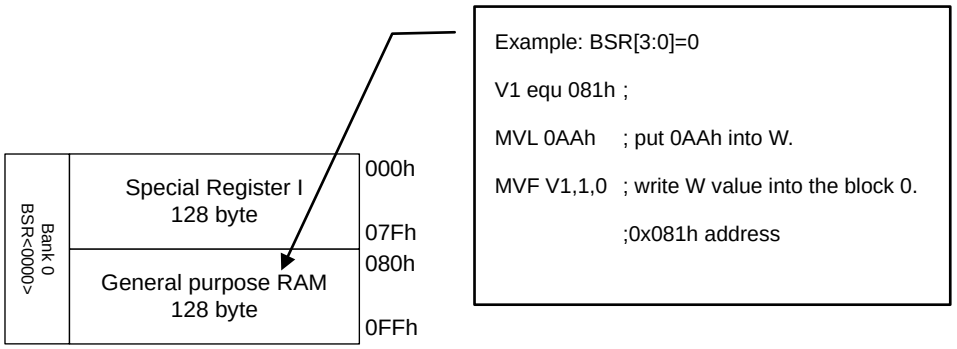
"d" stands for the place saving the data after operation. d=0 is saved in the WREG register, and d=1 is saved in the data memory register.

"a" stands for the block where the designated memory operates; a=0 operates in the block 0, and a=1 operates in the designated block BSR[3:0].

2.2.2.2. Block selection control register

Data memory is arranged to make each 256 byte is a block; in other words, 000h～0FFh is one block; If it is necessary to read/write the data memories after the address 0FFh, the block control register BSR[3:0] and the parameter "a" of the instruction should be correctly set, as follows:

- When a = 0, the instruction will only read/write the block 0 of the data memory, no matter which block is designated by BSR[0].
- When a = 1, the instruction of H08C CPU Core will read/write the block of the data memory designated by BSR[0].



Example 2-4 Example program and data memory relation of block selector

2.2.2.3. Specially register

The specially registers include the registers related to the CPU core and its peripheral functions, which mainly include the control function register and the data return register. The data read from the non-defined addresses or the bits for addresses of the data register will be 0.

The specially registers also include several registers designed for using together with the instructions; however, the chapter inly introduces two most frequently-used registers; one is the work register and the other one is the file select register FSR. The other specially registers not introduced here will be clearly described in other different chapters.

2.2.2.3.1.1. Work register (WREG)

The abbreviation of the work register is W, which is the register most frequently used together with the instructions, such as data transfer, operation and determination, etc.

2.2.2.3.1.2. File select register (FSR and INDF)

The file select register (FSR) is composed of the pointer registers FSR0[7:0], FSR1[7:0], FSR2[7:0], and the index registers INDF0[7:0], INDF1[7:0], INDF2[7:0]; as their functions are similar to one another, the chapter only discusses FSR0.

FSR0[8:0] can be divided into two registers, FSR0H[1:0] and FSR0L[7:0], which can address different blocks without setting BSR[4:0]; in addition, by means of special instructions, 16-bit data can be directly written in the register by only one instruction.

INDF0[7:0] is the index register, which can read the data of the data memory address pointed by FSR0[8:0].

It is worthy to note that H08C instruction set supports enhanced index register, and the description of its functions is as follows:

- POINC0[7:0]: the following events will occur when reading and writing the register POINC0[7:0] via the instruction.
  - Return the content of the address currently pointed by FSR0[8:0] first.
  - Then, add 1 to the value of the pointer register FSR0[8:0] and point to the next address.
- PODEC0[7:0]: the following events will occur when reading and writing the register PODEC0[7:0] via the instruction.
  - Return the content of the address currently pointed by FSR0[9:0] first.
  - Then, deduct 1 from the value of the pointer register FSR0[9:0] and point to the previous address.
- PRINC0[7:0]: the following events will occur when reading and writing the register PRINC0[7:0] via the instruction.
  - Add 1 to the value of the pointer register FSR0[8:0] and point to the next address first.
  - The, return the content of the address currently pointed by FSR0[8:0].
- PLUSW0 [7:0]: the following events will occur when reading and writing the register PLUSW0 [7:0] via the instruction.
  - Add the content of the work register W to the value of the pointer register FSR0[8:0] first.
  - Return the content of the address pointed by FSR0[9:0]. Besides, the content of W is the value with the sign bit, ±128d.

2.2.2.3.1.3. General Purpose Register

The general-purpose regulator GPR is a zone for the user to freely arrange; the user can save data, perform operation, or set flag, etc., in the register.

2.2.2.4. Register description-Data memory controller

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1											
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition											
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**R/W**
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	*,*,*,*,*,*,*,*
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	*,*,*,*,*,*,*,*
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	*,*,*,*,*,*,*,*
005h	INDF1	Contents of FSR1 to address data memoryvalue of FSR0 not changed								xxxx xxxx	*,*,*,*,*,*,*,*
006h	POINC1	Contents of FSR1 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
007h	PODEC1	Contents of FSR1 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	*,*,*,*,*,*,*,*
008h	PRINC1	Contents of FSR0 to address data memoryvalue of FSR1 pre-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
009h	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	*,*,*,*,*,*,*,*
00Ah	INDF2	Contents of FSR2 to address data memoryvalue of FSR2 not changed								xxxx xxxx	uuuu uuuu
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-incremented								xxxx xxxx	uuuu uuuu
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								xxxx xxxx	uuuu uuuu
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								xxxx xxxx	uuuu uuuu
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	uuuu uuuu
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	*,*,*,*,*,*,*,*
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	*,*,*,*,*,*,*,*
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu

Table 2-2 Data memory control registers

INDFx/POINCx/PODECx/PRINCx/PLUSWx: Index registers with different functions (x=0、1、2)

INDFx[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

POINCx[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

PODECx[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

PRINCx[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

PLUSWx[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

FSRx: File select register (x=0、1、2)

FSRxL[7:0]: Please refer to 2.2.2.3.2 for more information about the file select register, FSR and INDF.

WREG: Work register

WREG[7:0]: Please refer to 2.2.2.3.1 for more information about the work register, WREG.

BSRCN: Memory block read/write control register

Bit	Name	Description
Bit0	BSR[0]	Memory read/write block pointer register <0> Block 0, address 0x000h~0x0FFh <1> Block 1, address 0x100h~0x1FFh

2.2.1. Register list-Data memory

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
005h	INDF1	Contents of FSR1 to address data memoryvalue of FSR1 not changed								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
006h	POINC1	Contents of FSR1 to address data memoryvalue of FSR1 post-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
007h	PODEC1	Contents of FSR1 to address data memoryvalue of FSR1 post-decremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
008h	PRINC1	Contents of FSR1 to address data memoryvalue of FSR1 pre-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
009h	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR1 offset by W								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
00Ah	INDF2	Contents of FSR2 to address data memoryvalue of FSR2 not changed								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
016h	TOSH	-	-	-	-	TOS[11:8]				..xx xxxx	..uu uuuu	-,-,*,*,*,*,*,*
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
018h	SKCN	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. 0000	u$$. $$$$	rw 0,rw 0,rw 0,- *,*,*,*
01Ah	PCLATH	-	-	-	-	PC[11:8]				..00 0000	..00 0000	*,*,*,*,*,*,*,*
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	*,*,*,*,*,*,*,*
01Dh	TBLPTRH	-	-	-	-	TBLPTR[11:8]				..xx xxxx	..uu uuuu	-,-,*,*,*,*,*,*
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE			E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024h	INTE1	TA1IE		TXIE	RCIE	I2CERIE	I2CIE			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF			E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
027h	INTF1	TA1IF		TXIF	RCIF	I2CERIF	I2CIF			0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
02Bh	MSTAT	-	-	-	C	DC	N	OV	Z	...x xxxx	...u uuuu	-,-,-,*,*,*,*,*
02Ch	PSTAT	POR	PD	TO	IDL	-	SKERR	-	-	$000 $00.	uu$u u$u.	rw0,rw0,rw0,rw0 rw0,rw0,rw0,-
02Eh	BIECN	1	-	-	-	-	-	BIEWR	BIERD	1... $000	1... $uuu	r1,-,-,- r,*,*,*
030h	BIEARL	-	-	BIE Address Register as BIEAL[5:0]						xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
033h	PWRCN	ENBGR	LDOC[2:0]			LDOM[0]	LDOM	ENLDO	CSFON	0000 0000	uuuu u00u	*,*,*,*,*,w r0,w r0,*
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
035h	OSCCN1	CCOPT		DADC[1:0]		DTMB[1:0]		TMBS	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,-
036h	OSCCN2						HAOM[1:0]		ENHAO	0000 0001	uuuu uu01	*,*,*,*,*,*,*,*
037h	WDTCN					ENWDT	DWDT[2:0]			0000 0000	uuuu $000	-,*,*,* rw 1,*,*,*
03Ah	AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu	-,-,*,*,*,*,*,*
03Bh	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
03Ch	AD1L	ADC1 conversion low byte data register						0	0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
03Dh	AD1CN0	ENAD1	-	-	OSR[3:0]				CMFR	000. 0000	uuu. uuuu	*,*,*,*,*,*,*,*
03Eh	AD1CN1	-	-	VREGN		ADGN[2:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03Fh	AD1CN2	-	BIAS[2:0]			DCSET[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
040h	AD1CN3	INP[3:0]				INN[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
041h	AD1CN4	VRH[1:0]		VRL[1:0]		INX[1:0]		VRIS	INIS	0010 0000	uuuu uuuu	*,*,*,*,*,*,*,*
042h	AD1CN5	ENACM	ENV12	VCMS	LDOPL	ENBS	-	ENTPS	TPSCH	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
043h	CSFCN0	SKRST	HAOTR[6:0]							.1..	....	-,-,-,- -,-,-,-
044h	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]			-	-	0000 00..	u0uu uu..	*,rw 1,*,*,*,-,-
045h	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
046h	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0

Table 2-3 Data memory list

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1

"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
047h	AIXM1	APDR3[1:0]		-	-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
048h	AIXM2	-	-	-	-	-	-	APDR4[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04Eh	TB1Flag	-	-	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu	-,-,r,r r,r,r,r
04Fh	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	-	-	0000 0000	uuuu u0uu	*,*,*,*,*,rw 1,*,*
050h	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
051h	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
052h	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
053h	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
054h	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
055h	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
056h	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
057h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
058h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
059h	TCCN0	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
061h	CFG	-	-	-	-	-	GCRST	ENI2CT	ENI2C	0000 0000	.... .uuu	-,-,-,- -,*,*
062h	ACT	SLAVE	ADR10	SLAVE24	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
063h	STA	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*,*,*,*,*,*,*,*
064h	CRG	CRG[7:0]								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
065h	TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
066h	RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
067h	TDB0	TDB0[7:1]							TDB0[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
068h	SID0	SID0[7:1],The corresponding address of the 7-bit mode							SID0V[0]	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
069h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	*,*,*,*,-,-,*
06Ah	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r r,r,r,rw 0
06Bh	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD	 0000	.... uuuu	-,-,-,- *,*,*,*
06Ch	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-,-,-,*,*,*,*,*
06Dh	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06Eh	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06Fh	RCREG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
070h	PT1								PT1.0	xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
074h	PT1M1	-	-	-	-	-		INTEG0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
075h	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3				xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
076h	TRISC4	TC4.7	TC4.6	TC4.5	TC4.4	TC4.3				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
077h	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
078h	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
079h	PT4PD1	PDR4.3[1:0]		-	-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Ah	PT4PD2	PDR4.7[1:0]		PDR4.6[1:0]		PDR4.5[1:0]		PDR4.4[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Bh	PT4INT	INTG4.7	INTG4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Ch	PT4INTE	INTE4.7	INTE4.6							0000 0000	uuuu uuuu	
07Dh	PT4INTF	INTF4.7	INTF4.6							0000 0000	uuuu uuuu	
07Eh	PT4M2	-	PM4.7[0]	-	PM4.6[0]	-	-	-	-	0000 0000	uuuu uuuu	
080h ~ 0FFh		SRAM as 128Byte								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*

Table 2-4 Data memory list (cont'd)

3. Oscillator, clock source and power consumption management

HY15P41 series has two clock sources, HAO and LPO, as shown in Table 3-1. By means of the configuration of the clock control register, the CPU and the working frequencies of its peripheral circuits can be flexibly distributed and managed, which can more properly adjust the power consumption of the chip in order to save more energy.

Summary of clock control register:

OSCCN0 OSCS[1:0], DHS[1:0], DMS[2:0], CUPS

OSCCN1 DADC[1:0], DTMB[1:0], TMBS, LCDS

OSCCN2 HAOM[1:0], ENHAO

Symbol	Frequency	Configuration of frequency source controller CLKCN[7:0]		Instruction implement state	
		ENHAO	HAOM[1:0]	SLP	IDLE
HAO	2MHz	1	00	Stop	Oscillate
	4MHz	1	01	Stop	Oscillate
	-	1	10	Stop	Oscillate
	8MHz	1	11	Stop	Oscillate
LPO	14KHz	Oscillate after the chip is powered.		Stop	Oscillate

Table 3-1 Parameters of internal RC oscillator, configuration of frequency controller and instruction status

3.1. Oscillator

3.1.1. HAO oscillator

HAO is the internal high-speed RC oscillator, and its typical output frequency is 2.0~8.0MHz.

When HY15P41 series products use other oscillators as their working clock sources, the HAO oscillator can be turned off via set ENHAO as <0>.

3.1.2. LPO oscillator

LPO is the internal low-speed RC oscillator, and its typical output frequency is 14KHz. The oscillator is mainly applied to serve as the clock source of the low-speed and energy-saving CPU operation modes.

After HY15P41 series products execute the instruction "Sleep", the LPO oscillator is turned off; when the chip is awaked, the LPO oscillator is automatically turned on.

3.2. CPU and clock sources of peripheral circuits

3.2.1. Distribution of clock sources

The outputs (HS_CK、LS_CK) of the two oscillators will be enabled/disabled, switched and processed by frequency elimination in advance by the front working clock distributor, and then inputted into the CPU and all peripheral circuits of the chip, as shown in Figure 3-1.

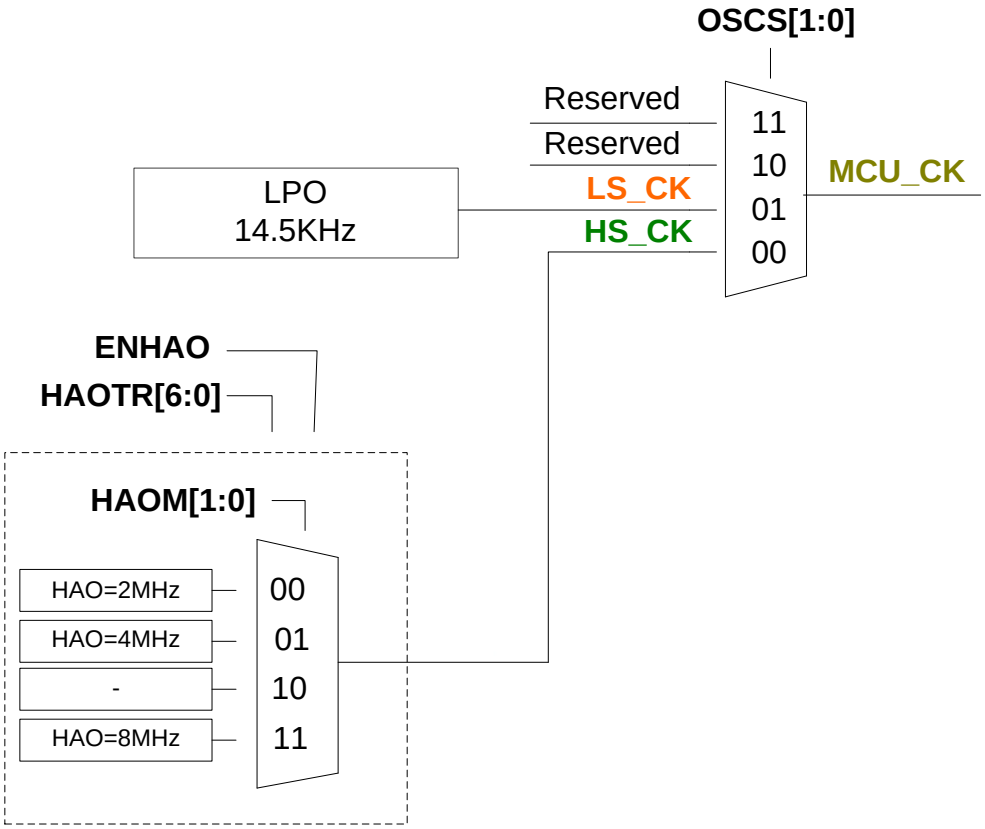


Figure 3-1 Front working clock distributor

3.2.2. CPU clock sources

The CPU has many working frequencies to be choose from; by means of CPUS, the working frequency from HS_CK or DHS_CK can be selected.

The instruction working frequency adopts 1/4 CPU_CK, and the frequency source of INTR_CK can be obtained via frequency division.

- When operating ΣΔADC, it is suggested to process HS_CK by frequency elimination to serve as the working frequency so as to achieve better performance.
- The frequency and the instruction execution period of CPU_CK is as shown in Figure 3-2; Table 3-2 briefly lists the working frequencies of the CPU and the relation between which and the instruction period.

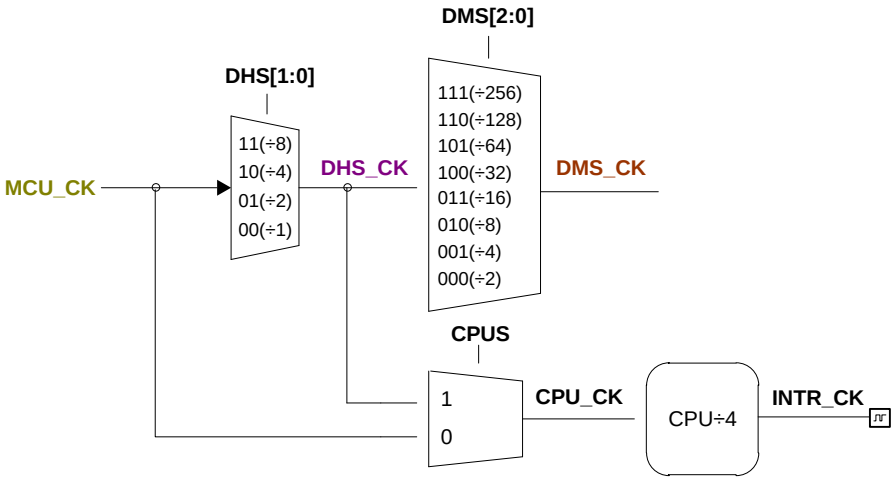


Figure 3-2 CPU and working frequencies of peripheral circuits

Working frequency / CPU_CK	CPU	Instruction	
	Frequency	Frequency	Period
8MHZ	8MHZ	2MHz	0.5us
4MHZ	4MHZ	1MHz	1us
2MHZ	2MHZ	0.5MHz	2us
14.5KHz	14.5KHz	3.625KHz	275.86us

Table 3-2 Working frequencies and instruction execution periods of CPU

3.2.3. Clock sources of peripheral circuits of CPU

The working clocks of the peripheral circuits of HY15P41 series are configured by different distribution controllers and pre-frequency eliminators; the configurations will be detailedly described in the sub-chapter of each of the peripheral circuits; the chapter only provides the working frequency configuration diagram of peripheral circuits, as shown in Figure 3-3.

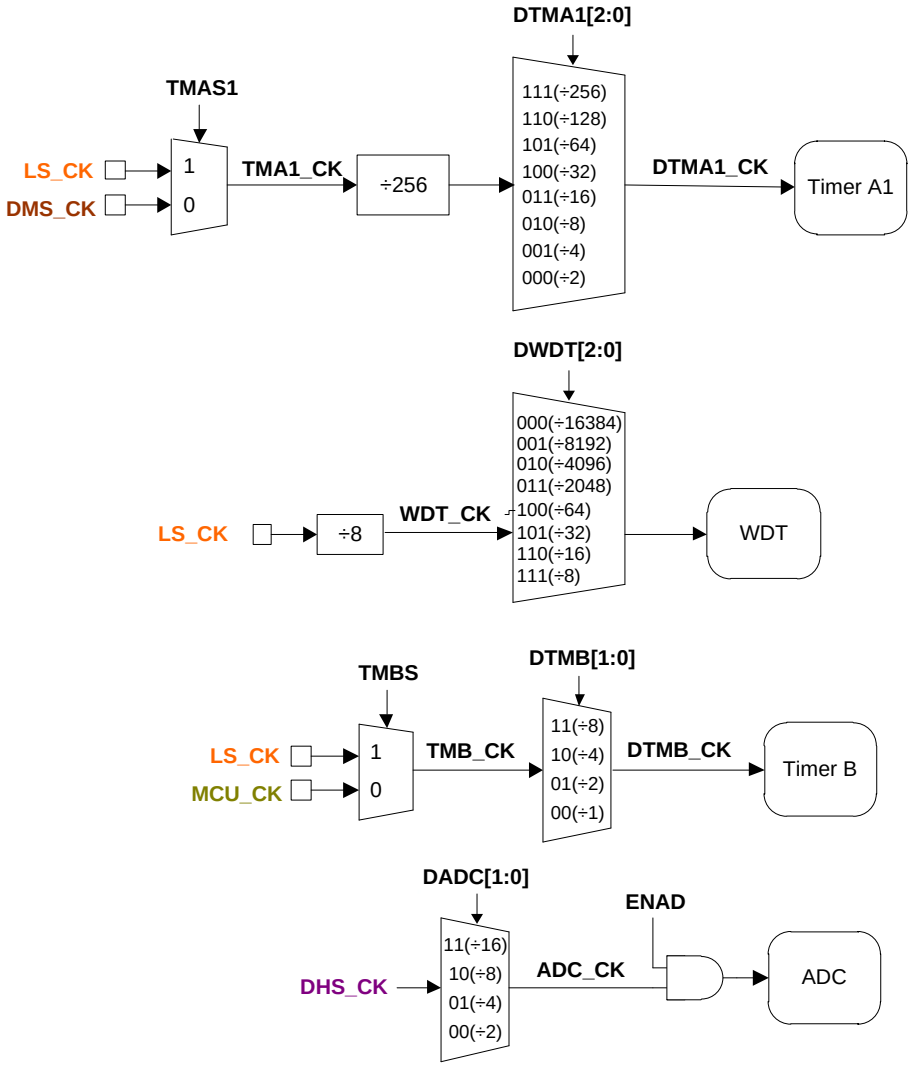


Figure 3-3 Working frequency configuration diagram of peripheral circuits

3.3. Register description-Working clock source controller

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
033h	PWRCN								CSFON	0000 0000	uuuu u00u	*,*,*,* *,w r0,w r0,*
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
035h	OSCCN1	-	-	DADC[1:0]		DTMB[1:0]		TMBS	-	0000 0000	uuuu uuu.	*,*,*,*,*,*,*,-
036h	OSCCN2	-	-	-	-	-	HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*,*,*,*,*,*,*,r
043h	CSFCN0		HAOTR[6:0]							0	....u	-,-,-,- -,-,-,*

Table 3-3 Working clock source control registers

OSCCN0[7:0]: Working frequency control register of chip

Bit	Name	Description
Bit7~6	OSCS[1:0]	Frequency selector of HS_CK <00>HS_CK <01>LS_CK <10>Reserved <11>Reserved
Bit5~4	DHS[1:0]	Frequency distribution selector of DHS_CK <00>MCU_CK ÷ 1 <01>MCU_CK ÷ 2 <10>MCU_CK ÷4 <11>MCU_CK ÷ 8
Bit3~1	DMS[2:0]	Frequency distribution selector of DMS_CK
Bit0	CUPS	Frequency selector of CPU_CK <0>MCU_CK <1>DHS_CK

DMS[2:0]	DMS_CK	DMS[2:0]	DMS_CK
000	DHS_CK ÷ 2	100	DHS_CK ÷ 32
001	DHS_CK ÷ 4	101	DHS_CK ÷ 64
010	DHS_CK ÷ 8	110	DHS_CK ÷ 128
011	DHS_CK ÷ 16	111	DHS_CK ÷ 256

OSCCN1[7:0]: Working frequency control register of chip

Bit	Name	Description
Bit5~4	DADC[1:0]	Frequency distribution selector of ADC_CK

DADC[1:0]	Pre-scale
00	DHS_CK ÷ 2
01	DHS_CK ÷ 4
10	DHS_CK ÷ 8
11	DHS_CK ÷ 16

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit3~2</td><td>DTMB[1:0]</td><td>Frequency distribution selector of DTMB_CK
<table>
<tr><th>DTMB[1:0]</th><th>Pre-scale</th></tr>
<tr><td>00</td><td>TMB_CK ÷ 1</td></tr>
<tr><td>01</td><td>TMB_CK ÷ 2</td></tr>
<tr><td>10</td><td>TMB_CK ÷ 4</td></tr>
<tr><td>11</td><td>TMB_CK ÷ 8</td></tr>
</table></td></tr>
<tr><td>Bit1</td><td>TMBS</td><td>Frequency selector of TMB_CK
<0>HS_CK
<1>LS_CK</td></tr>
</table>

OSCCN2[7:0]: Working frequency control register of chip

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit2~1</td><td>HAOM[1:0]</td><td>Oscillating frequency selector of internal oscillator HAO
<table>
<tr><th>HAOM[1:0]</th><th>HAO oscillating frequency</th></tr>
<tr><td>00</td><td>2MHz</td></tr>
<tr><td>01</td><td>4MHz</td></tr>
<tr><td>10</td><td>The setting is disabled.</td></tr>
<tr><td>11</td><td>8MHz</td></tr>
</table></td></tr>
<tr><td>Bit0</td><td>ENHAO</td><td>Enable control bit of internal HAO
<0> Disable
<1> Enable</td></tr>
</table>

PWRCN[7:0] Linear voltage regulator and analog common ground control register

Bit	Name	Description
Bit0	CSFON	CSF(Chip Special Function) enable written-in controller <0> Not enable the CSF function. <1> Enable the written-in function of CSF; when the user should configure the control registers of the zone, the user should set CFSON[0] as <1> before writing data into CSFCN0[7:0].

CSFCN0[7:0] Specially control bit register

Bit	Name	Description
Bit6~0	HAOTR	Frequency center adjustment controller of HAO <0000000>Adjust 25.00% (maximum) . <1000000>Central point 0.00% . <1111111>Adjust -25%(minimum)

4. RESET

The reset circuit of HY15P41 series includes the following events to trigger the reset signal, and the reset block diagram is as shown in Figure 4-1.

- **BOR** Brown-Out Reset.
- **WDT** Watch Dog Timer reset.
- **SKERR** Stack Error Reset. (determined by User)

Summary of operation status register:

PSTAT POR[0],PD[0],TO[0],IDL[0],SKERR[0]

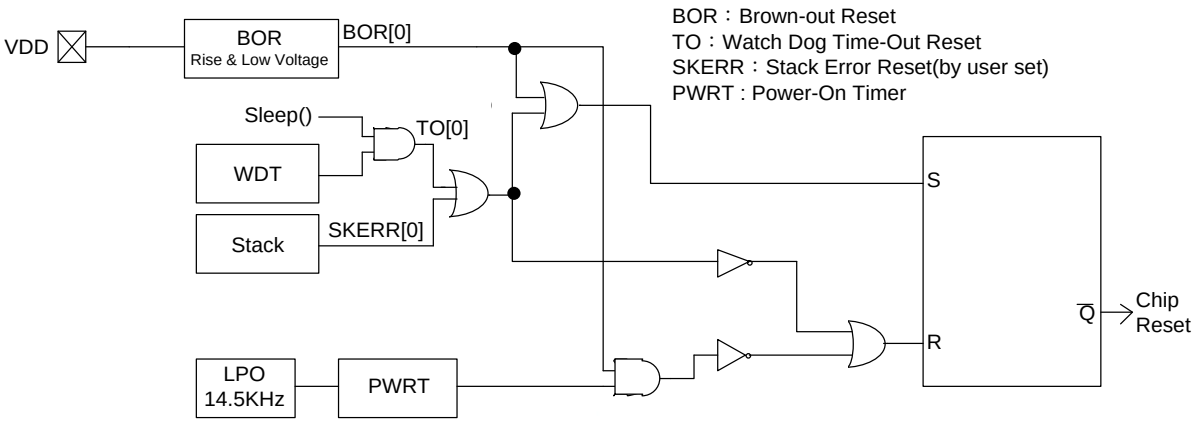


Figure 4-1 Reset block diagram

These reset events can be classified into the hardware reset and the software reset, as shown in Table 4-1. After being reset, the CPU can be enabled by 0x0000h via the reset program.

Reset type	Event	Symbol	Description
Hardware reset	BOR	A-RESET	The CPU is reset, and returns to the normal working state after the internal oscillator enables counting.
Low-level reset	SKERR	I-RESET	Only some registers are cleared, and the CPU can swiftly return to the normal working status.

Table 4-1 Reset level table

4.1. Reset event description

4.1.1. Brown-Out Reset (BOR)

When the CPU is being powered or the power source is interfered by external factors, the CPU will enter the normal working voltage from the abnormal low working voltage. Thus, if the CPU cannot enter the reset state when under low working voltage, the CPU will crash, so its peripheral circuits will work normally. For the reason, with the function of the BOR circuit, when detecting that the working voltage is interfered and the voltage level is lower than the designed value, the BOR circuit generates the reset signal to make the chip enter the reset state; then, the chip can enter the normal working

mode after returns to the normal working voltage and the reset signal is released.

When the BOR reset occurs, the BOR flag of the register PSTAT[7:0] is set as <1> to record the event which occurs.

The BOR circuit of HY15P41 series will generate the current consumption of about 0.6uA, which cannot be disabled via programs or other setting methods.

4.1.2. Stack error reset (SKERR)

When the program suffers the stack overflow or underflow, a reset signal is generated to make the chip enter the swift enable status. When the SKERR stack error reset occurs, the SKERR flag of the register PSTAT[7:0] is set as <1> to record the event which occurs. Please refer to the chapter "***Memory***" for more information about the detailed operation description.

4.2. Status register

The operation state of the chip is displayed on the reset register PSTAT[7:0], and the mutual relation is as shown in Table 4-2.

"0": Not occur, "1": Occur, "u": Remain unchanged, "-": Not use

Name/State	Address	7	6	5	4	3	2	1	0
PSTAT	02CH	BOR	PD	TO	IDL	-	SKERR	-	-
Hardware reset (A-RESET)	BOR	1	0	0	0	-	0	-	-
Software reset (I-RESET)	WDT	u	u	1	u	-	u	-	-
	SKERR	u	u	u	u	-	1	-	-

Table 4-2 Reset state flag relation table

4.2.1. Timing diagram of reset state

After the hardware reset signal occurs, the chip enters the timing diagram of the operation state, as shown in Figure 4-2, which shows the time intervals between different reset signals occurring and the chip entering the operation state.

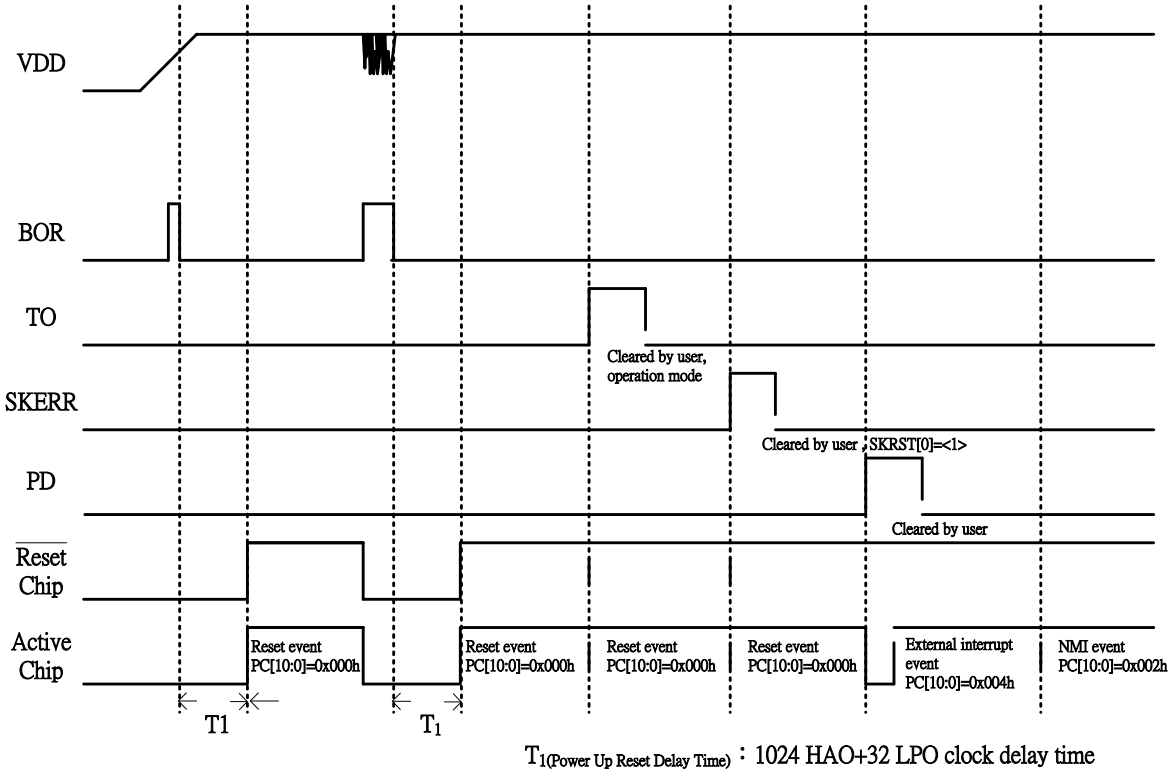


Figure 4-2 Reset and operation modes and states flag timing diagram.

"-": Not defined

Reset signal	Delay time			Operation state		
	Symbol	T1	T2	operation	Standby	Sleep
BOR	t_{RST}	T1＋T2		Effective	Effective	Effective
SKERR	-	-		Effective	Not effective	Not effective

Table 4-3 Delay time of reset states and operation state relation table

4.3. Register description-Reset state

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
02Ch	PSTAT	POR	PD	TO	IDL	-	SKERR	-	-	$000 $00.	uu$u u$u.	rw0,rw0,rw0,rw0 rw0,rw0,rw0,-
043h	CSFCN0	SKRST								0	....u	-,-,-,- -,-,-,*

Table 4-4 Reset registers

PSTAT: Status register

Bit	Name	Description
Bit7	BOR	Brown-Out Reset (BOR) flag. <0> Use an instruction to clear the bit. <1> Set the bit as <1> when BOR functions.
Bit6	PD	Sleep status flag <0> The bit should be cleared via BOR, RST or the instruction. <1> Set the bit as <1> when the SLEEP instruction is executed.
Bit5	TO	Watch Dog Timer(WDT) flag <0> The bit should be cleared via BOR, RST or the instruction. <1> Set the bit as <1> when the counting of the watch dog ends.
Bit4	IDL	Standby state flag <0> The bit should be cleared via BOR, RST or the instruction. <1> Set the bit as <1> when the IDLE instruction is executed.
Bit2	SKERR	Stack error reset flag <0> The bit should be cleared via BOR, RST or the instruction. <1> Set the bit as <1> when the stack error occurs.

CSFCN0: Specially control bit register

Bit	Name	Description
Bit7	SKRST	Stack error reset controller <0> Not enable the error reset chip. <1> Enable the error reset chip.

5. Interrupt

The interrupt is composed of the interrupt enable controller INTE and the interrupt event flag INTF. If an interrupt event occurs during occurrence of the interrupt service, the program counter PC jumps to the interrupt vector address 0x0004h of the program memory to execute the interrupt service program.

Summary of interrupt control register:

INTE0 GIE, TA1CIE, ADIE, WDTIE, TB1IE, E0IE

INTE1 TMA1IE, TXIE, RCIE, I2CERIE, I2CIE

INTF0 TA1CIF, ADIF, WDTIF, TB1IF, E0IF

INTF1 TMA1IF, TXIF, RCIF, I2CERIF, I2CIF

PT4INTE INTE4.7, INTE4.6

PT4INTF INTF4.7, INTF4.6

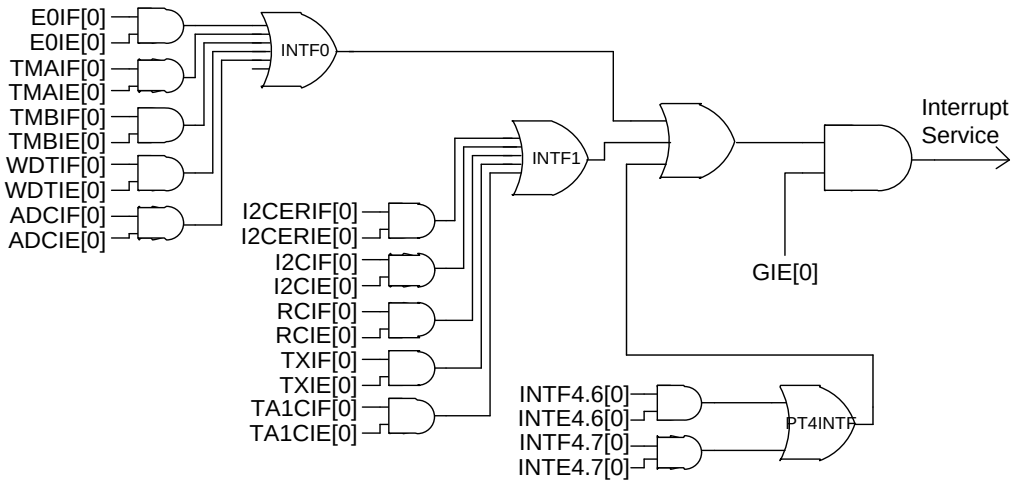


Figure 5-3 Interrupt vector block diagram

The interrupt service event has two controlling layers; the major layer is the interrupt service controller GIE, and the minor layer is the enable control bit of the interrupt event.

- The interrupt event can be enabled by just set the controller of the corresponding interrupt event enable register INTEx[7:0] as <1>; on the contrary, the interrupt event is disabled if the controller is set as <0>.
- The interrupt service can be enabled by just set the interrupt service controller GIE of the corresponding interrupt control register INTE0[7:0] as <1>; on the contrary, the interrupt service is disabled if the controller is set as <0>.

When entering the interrupt service vector, GIE is automatically set as <0>; after the execution of the interrupt service program is finished and is going to return to the interrupt occurrence address, the interrupt return instruction RETI can be directly executed; meanwhile, GIE is automatically set as <1>; alternatively, the return instruction RET can be executed; meanwhile, the state of GIE remains 0.

5.1. Register description-Interrupt

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	-	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024h	INTE1	TA1IE	-	TXIE	RCIE	I2CERIE	I2CIE	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	-	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
027h	INTF1	TA1IF	-	TXIF	RCIF	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
07Bh	PT4INT	INTG4.7	INTG4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Ch	PT4INTE	INTE4.7	INTE4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Dh	PT4INTF	INTF4.7	INTF4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 5-1 Interrupt register

INTE0: Interrupt enable control register 0

Bit	Name	Description
Bit7	GIE	Interrupt service controller <0> OFF. <1> ON.
Bit6	TA1CIE	Timer-A1 comparison event enable controller <0> OFF. <1> ON. (Comparison event/timer A1)
Bit5	ADIE	ADC interrupt event enable controller <0> OFF. <1> ON. (Analog to digital converter, ΣΔADC)
Bit4	WDTIE	Watch Dog interrupt event enable controller <0> OFF. <1> ON. (Watch dog, WDT)
Bit3	TMBIE	Timer-B interrupt event enable controller <0> OFF. <1> ON. (Clocking/timer B,TMB)
Bit0	E0IE	Input pin 0 interrupt event enable controller <0> OFF. <1> ON. (External input pin, PT1.0)

INTE1: Interrupt enable control register 1

Bit	Name	Description
Bit7	TMA1IE	Timer-A1 interrupt event enable controller <0> OFF. <1> ON. (Clocking/timer A, TMA1)
Bit5	TXIE	TX interrupt event enable controller <0> OFF. <1> ON. (Communication interface, EUART)

Bit	Name	Description
Bit4	RCIE	RC interrupt event enable controller <0> OFF. <1> ON. (Communication interface, EUART)
Bit3	I2CERIE	Peripheral I^2C error interrupt vector service controller <0> Disable I^2C interrupt vector service <1> Enable I^2C interrupt vector service
Bit2	I2CIE	Peripheral I^2C error interrupt vector service controller <1>Enable I^2C interrupt vector service <0>Disable I^2C interrupt vector service

INTF0: Interrupt event flag register 0

Bit	Name	Description
Bit6	TA1CIF	Timer-A1 comparison event flag <0> OFF. <1> ON. (Comparison event/timer A1)
Bit5	ADIF	ADC interrupt event flag <0> Not occur. <1> Occur. (Analog to digital converter, ΣΔADC)
Bit4	WDTIF	Watch Dog interrupt event flag <0> Not occur. <1> Occur. (Watch dog, WDT)
Bit3	TMBIF	Timer-B interrupt event flag <0> Not occur. <1> Occur. (Clocking/timer B, TMB)
Bit2	TMAIF	Timer-A interrupt event flag <0> Not occur. <1> Occur. (Clocking/timer A, TMA)
Bit0	E0IF	Input pin 0 interrupt event flag <0> Not occur. <1> Occur. (External input pin, PT1.0)

INTF1: Interrupt event flag register 1

Bit	Name	Description
Bit7	TMA1IF	Timer-A1 interrupt event flag <0> Not occur. <1> Occur. (Clocking/timer A1, TMA1)

Bit	Name	Description
Bit5	TXIF	TX interrupt event flag <0> Not occur. <1> Occur. (Communication interface, EUART)
Bit4	RCIF	RC interrupt event flag <0> Not occur. <1> Occur. (Communication interface, EUART)
Bit3	I2CERIF	Peripheral I^2C error interrupt event flag controller <0> An I^2C interrupt event does not occur. <1> An I^2C interrupt event occurs.
Bit2	I2CIF	Peripheral I^2C error interrupt event flag controller <0> An I^2C interrupt event does not occur. <1> An I^2C interrupt event occurs.

PT4INT:PT4 interrupt event flag register

Bit	Name	Description
Bit7	INTF4.7	Interrupt signal generation condition <0> Falling edge (1 → 0). <1> Rising edge (0 → 1).
Bit6	INTF4.6	Interrupt signal generation condition <0> Falling edge (1 → 0). <1> Rising edge (0 → 1).

PT4INTE: PT4 interrupt enable control register

Bit	Name	Description
Bit7	INTE4.7	External input pin PT4.7 interrupt event enable controller <0> Disable. <1> enable.
Bit6	INTE4.6	External input pin PT4.6 interrupt event enable controller <0> Disable. <1> enable.

PT4INTF: PT4 interrupt event flag register

Bit	Name	Description
Bit7	INTF4.7	External input pin PT4.7 interrupt event flag <0> Not occur. <1> Occur.

Bit	Name	Description
Bit6	INTF4.6	External input pin PT4.6 interrupt event flag <0> Not occur. <1> Occur.

6. Hardware multiplier

The H08C instruction set has the processing instructions "MULF and MULL" of the 8x8 hardware multiplier. The operation result of the 8x8 hardware multiplier will be put in the multiplier registers PRODH[7:0] and PRODL[7:0], and will not change any sign of the status register PSTAT[7:0]. PRODH[7:0] and PRODL[7:0] are read-only registers; please pay attention when using them.

The hardware multiplier can execute the signed number operation and the unsigned number operation, as shown in Example 6-1 and Example 6-2.

```
Ex. 1: V1 x V2 = V
MVL     V1          ; Put the value of V1 into the W register.
MVF     BUF0,1,0    ; Put the value of V1 into the BUF0 register of the memory block 0.
MVL     V2          ; Put the value of V2 into the W register.
MULF    BUF0,0      ; Execute V1 x V2 and put the operation result into PRODH/L.
```

Example 6-1 Unsigned number operation

```
Ex. 2: N1 x N2 = N ,s=7,B
MVL     N1          ; Put the value of N1 into the W register.
MVF     BUF0,1,0    ; Put the value of N1 into the BUF0 register of the memory block 0.
MVL     N2          ; Put the value of N2 into the W register.
MVF     BUF1,1,0    ; Put the value of N2 into the BUF1 register.
MULF    BUF0,0      ; Execute V1 x V2 and put the operation result into PRODH/L.
MVFF    PRODH,SWP   ; Put the value of the PRODH register into the SWP register.
BTSZ    BUF0,s      ; Determine whether N1 is a negative number
                    ; If N1 is a negative number…
SUBF    SWP,1,0     ; Put SWP – N2 into the SWP register.
MVF     BUF0,0,0    ; Put the value of N1 into the W register.
BTSZ    BUF1,s      ; Determine whether N2 is a negative number
                    ; If N2 is a negative number…
SUBF    SWP,1,0     ; Put SWP – N1 into SWP and process it by calculation operation
                    ; Then, N = SWP/PRODL.
; ------------------------------------------------------------------------------------
; N1=07Fh, N2=0FFh; PRODH/L = 7E81h is obtained after the operation of the multiplier.
; Determine whether N1 is a negative number; if it is, perform PRODH – N2.
; Determine whether N2 is a negative number; if it is, perform PRODH – N1
; After calculation operation, the value of the signed number, N, is obtained.
; 7Fh x FFh     = 7Fh x ( 0FFh – 100h )
;               = 7Fh x 0FFh – 7Fh x 100h
;               = 7E81h – 7F00h
;               = FF81h
```

Example 6-2 Signed number operation

7. Input/output port (I/O)

Each pin of the I/O port is a port, which can serve as the digital input and output channel. Each pot is controlled by one register.

Summary of registers related to I/O：

PT PT1[0], PT4[7:3]

TRISC4 TRISC4[7:3]

PT4DA **PT4DA**[7:3]

PT4PU PT4PU[7:3]

PT1M1 INTEG0[1:0]

PT4PD1 PDR4.3[1:0]

PT4PD2 PDR4.7[1:0],PDR4.6[1:0],PDR4.5[1:0],PDR4.4[1:0],

PT4M2 PM4.7[0],PM4.6[0]

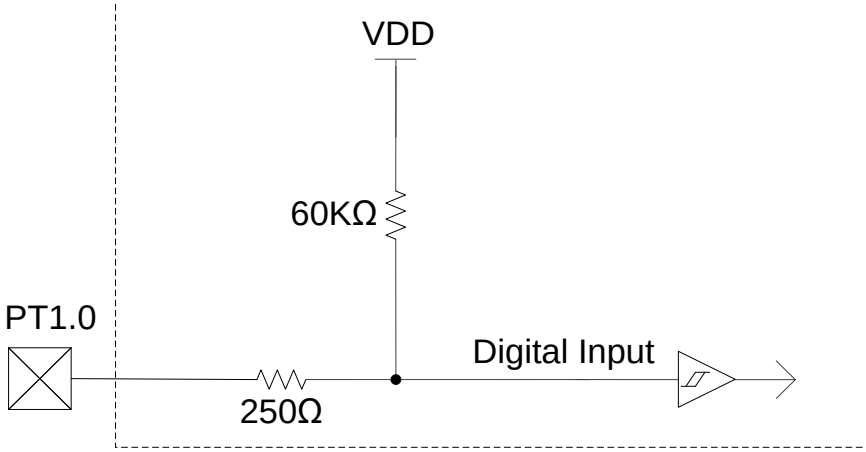


Figure 7-1 PT1.0 I/O structure block diagram

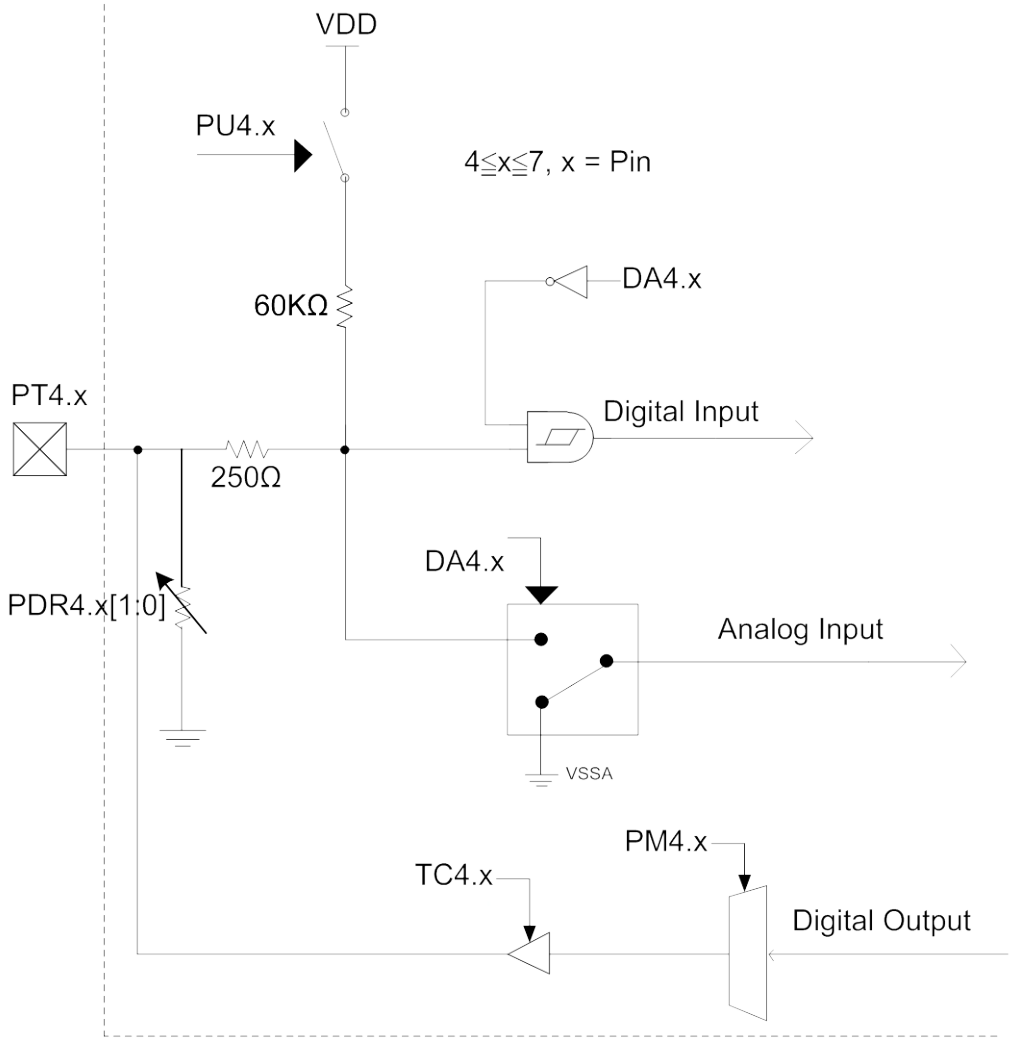


Figure 7-2 PT4.x I/O structure block diagram

7.1. Introduction of registers related to PORT

PORT mainly provides the input and output pins of digital signals.

7.1.1. PTEG interrupt signal generation condition

The condition determines that what change of the I/O input level will generate the interrupt signal; the level change condition can be classified into the ascending edge (0→1) change, the descending edge (1→0) change and the level transition (0→1 or 1→0) change.

7.1.2. PTPU pull-up resistor control register

The register can enable or disable the pull-up resistor function of the I/O; if the register is set as <1>, the function of the I/O is enabled; if the register is set as <0>, the function of the I/O is disabled. If the I/O is set as the digital input state and its external circuit connection method results in the floating phenomenon of the I/O before the chip enters the sleep mode, the pull-up resistor can be enabled so as to avoid that the leakage current occurs because the I/O is floating after the chip enters the sleep mode.

7.1.3. TC I/O control register

The register can select the I/O is under the input state or the output state; if the register is set as <1>, the I/O is under the output state; if the register is set as <0>, the I/O is under the input state. When the I/O is set as the input state, it is necessary to provide a definite input level after the chip enters the sleep mode to avoid that the I/O is under the floating state, or the chip may have leakage current.

7.1.4. PTIO state control register

When the I/O is set as the input state, the current state of the I/O can be read from the register at the corresponding position; if the read value is 1, the current input of the I/O is at high level; if the read value is 0, the current input of the I/O is at low level.

If the I/O is set as the output state, the output state can be controlled by the register at the corresponding position; if the register is set as <1>, the output of the I/O is at high level; if the register is set as <0>, the output of the I/O is at low level.

7.2. I/O Port1

"i": input, "o": output, "a": analog, "c": cmos i/o, "x": not defined, "p": power source

Pin name	Design		Register configuration			Description
	Type	Buffer				
PT1.0	i	S				Digital input pin
INT0	i	s				External interrupt source
VPP	p	p				OTP burning voltage pin

Table 7-1 Functions of PORT1

7.3. I/O Port4

"i": input, "o": output, "a": analog, "c": cmos i/o, "x": not defined, "p": power source

Pin name	Design		Register configuration					Description
	Type	Buffer	TC[0]	DA[0]	PM4.5[1:0]	PM4.6[0]	PM4.7[0]	
PT4.3	i/o	c	x	0	00	0	0	Digital I/O pin
AI3	i	a	0	1	00	0	0	Digital input channel AI3
PSCK	i	s	0	0	00	0	0	OTP read/write interface SCK pin
PT4.4	i/o	c	x	0	00	0	0	Digital I/O pin
AI4	i	a	0	1	00	0	0	Analog input channel AI4
RC	i	s	0	0	01	0	0	EUART communication interface RC pin
PSDI	i	s	0	0	00	0	0	OTP read/write interface SDI pin
SCL	i/o	s	1	0	10	0	0	I^2C communication interface pin
PT4.5	i/o	c	x	0	00	0	0	Digital I/O pin
AI5	i	a	0	1	00	0	0	Analog input channel AI5
TX	o	c	1	0	01	0	0	EUART communication interface TX pin
PSDO	o	c	1	0	00	0	0	OTP read/write interface SDO pin
SDA	i/o	s	1	0	10	0	0	I^2C communication interface pin
PT4.6	i/o	c	x	0	00	0	0	Digital I/O pin
AI6	i	a	0	1	00	0	0	Analog input channel AI6
PWM0	o	c	1	0	00	1	0	PWM0 output port.
INT4.6	i	s	0	0	00	0	0	External interrupt source
PT4.7	i/o	c	x	0	00	0	0	Digital I/O pin
AI7	i	a	0	1	00	0	0	Analog input channel AI7
PWM1	o	c	1	0	00	0	1	PWM1 output port
INT4.7	i	s	0	0	00	0	0	External interrupt source

Table 7-2 Functions of PORT4

7.4. Register description-PORT

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	-	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	-	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
070h	PT1								PT1.0	xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
074h	PT1M1	-	-	-	-			INTEG0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
075h	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
076h	TRISC4	TC4.7	TC4.6	TC4.5	TC4.4	TC4.3	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
077h	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
078h	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
079h	PT4PD1	PDR4.3[1:0]		-	-	-	-	-		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Ah	PT4PD2	PDR4.7[1:0]		PDR4.6[1:0]		PDR4.5[1:0]		PDR4.4[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Bh	PT4INT	INTG4.7	INTG4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Ch	PT4INTE	INTE4.7	INTE4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Dh	PT4INTF	INTF4.7	INTF4.6							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07Eh	PT4M2	-	PM4.7[0]	-	PM4.6[0]					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 7-3 PORT control register

INTE0/INTF0/PT4INTE/PT4INTF: please refer to the chapter "Interrupt" for more information.

PT1: PT1 pin state flag and control register

Bit	Name	Description
Bit0	PT1.0	PT1.0 pin state flag and controller

PT1.x	
0	PT1.x input is at low level (L)
1	PT1.x input is at high level (H)

Note: PT1.0 pin is always at the pull-up activation mode.

PT1M1: digital output mode selection register 1

Bit	Name	Description
Bit1~0	INTEG0[1:0]	PT1.0 interrupt signal generation condition

INTEG0[1:0]	Interrupt signal generation condition
00	Descending line (1→0)
01	Ascending line (0→1)
10	Level transition (0→1 or 1→0)
11	Level transition (0→1 or 1→0)

PT4: PT4 pin state flag and control register

Bit	Name	Description
Bit7~3	PT4.x	PT4.x pin state flag and controller, 3≦x≦7

PT4.x	When TC4.x is set as <0>	When TC4.x is set as <1>
0	PT4.x input is at low level (L)	PT4.x output is at low level (L)
1	PT4.x input is at high level (H)	PT4.x output is at high level (H)

TRISC4: PT4 pin characteristic control register

Bit	Name	Description
Bit7~3	TC4.x	PT4.x input output characteristic controller, 3≦x≦7 <0> Disable the output function; the pin only has the input characteristic. <1> Enable the output function; the pin has the output/input characteristics.

PT4DA: PT4 analog input control register

Bit	Name	Description
Bit7~3	DA4.x	PT4.x analog input controller, 3≦x≦7 <0> OFF <1> ON

PT4PU: PT4 pin pull-up resistor control register

Bit	Name	Description
Bit7~3	PU4.x	PT4.x pin pull-up resistor controller, 3≦x≦7 <0> OFF <1> ON

PT4M2: PT4 digital output mode selection register

Bit	Name	Description
Bit6	PM4.7	PT4.7 I/O digital output mode selector. <0> OFF <1> Enable PWM1 output function. (the TRISC4 register should be independently set.)
Bit4	PM4.6	PT4.6 I/O digital output mode selector. <0> OFF <1> Enable PWM0 output function. (The TRISC4 register should be independently set.)

PT4INT: PT4 I/O interrupt signal generation condition

Bit	Name	Description
Bit7~6	INTG4.x	Interrupt signal generation condition (6≦x≦7) <0> Descending line (1→0)) <1> Ascending line (0→1)

PT4PD1: PT4 pull-down resistor selection register

Bit	Name	Description

Bit7~6	PDR4.3[1:0]	PT4.3 pull-down resistor selection controller

PDR4.3[1:0]	Function
00	OFF(default)
01	10K Ω
10	50K Ω
11	100K Ω

PT4PD2: PT4 pull-down resistor selection register

Bit	Name	Description
Bit7~6	PDR4.7[1:0]	PT4.7 pull-down resistor selection controller

PDR4.7[1:0]	Function
00	OFF(default)
01	10K Ω
10	50K Ω
11	100K Ω

Bit5~4	PDR4.6[1:0]	PT4.6 pull-down resistor selection controller

PDR4.6[1:0]	Function
00	OFF(default)
01	10K Ω
10	50K Ω
11	100K Ω

Bit3~2	PDR4.5[1:0]	PT4.5 pull-down resistor selection controller

PDR4.5[1:0]	Function
00	OFF(default)
01	10K Ω
10	50K Ω
11	100K Ω

Bit1~0	PDR4.4[1:0]	PT4.4 pull-down resistor selection controller

PDR4.7[1:0]	Function
00	OFF(default)
01	10K Ω
10	50K Ω
11	100K Ω

8. Watch Dog

The watch dog, WDT, is the watcher of the chip, just like its name; its major function is to generate a wake-up event.

- Operation mode

  A reset signal is generated after the watch dog timer overflows to reset the chip.

  The timer can be reset by software.

- Sleep mode

  The watch dog, WDT, is disabled and cannot be used.

- Standby mode

  The watch dog timer overflows to generate an interrupt event and then wake up chip.

Summary of registers related to WDT:

INTE0	GIE[0], WDTIE[0]
INTF0	WDTIF[0]
PSTAT	TO[0]
WDTCN	ENWDT[0], DWDT[2:0]

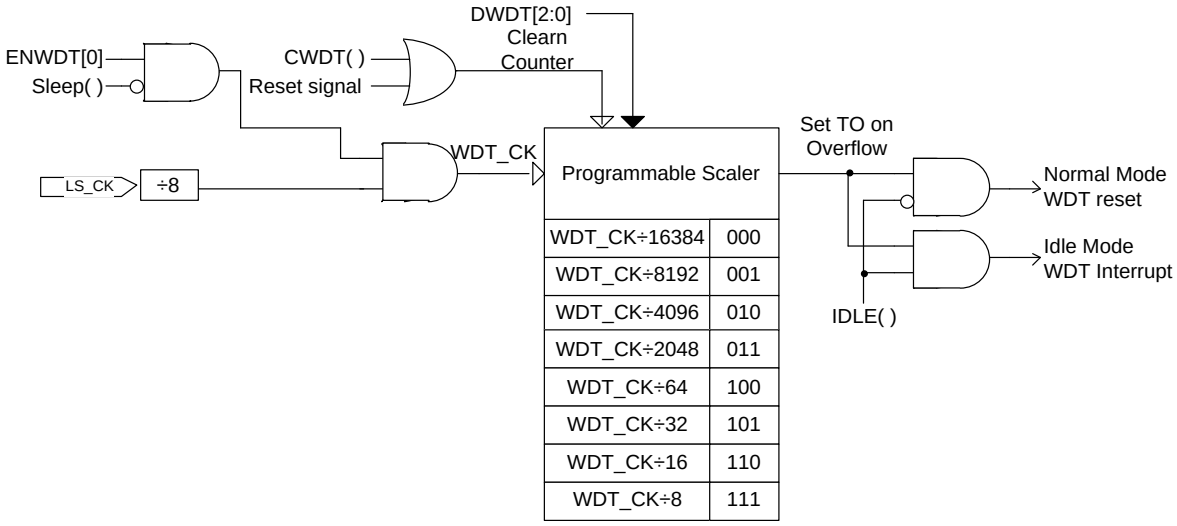


Figure 8-1 Block diagram of watch dog

8.1. Usage description of WDT

8.1.1. Initialization configuration of WDT

WDT's timer controller DWDT[2:0] can determine the working frequency WDT_CK and the overflowing of the WDT timer; the WDT reset signal TO or the interrupt event WDTIF[2] can be generated after the timer overflows.

8.1.2. Interrupt event service of WDT

The WDT interrupt event can only be operated when the chip is under the standby mode; when

2 WDT uses the internal clock source LPO, so can control the chip to operate under Normal Mode and Idle Mode. Under the operation mode, the timer can be reset via software to avoid that the counting of the timer ends and then the chip is reset; however, under the standby mode, the WDT timer cannot be reset by any methods.

WDTIE[0] and GIE[0] are set as <1>, an interrupt event will be generated to set WDTIF[0] as <1> after the WDT timer overflows, and the program counter PC will jump to the interrupt vector <0>x0004h. On the contrary, there is no interrupt occurring when WDTIE[0] and GIE[0] are set as <0>.

8.1.3. Enablement of WDT

WDT should be enabled when the chip is under the operation mode; in other words, the WDT enablement controller ENWDT[0] is set as <1> to enable WDT. After the WDT is enabled, it is impossible to set ENWDT[0] as <1> by software; ENWDT[0] cannot be set as <0> by software no matter under the operation mode or the standby mode. After DWDT[2:0] is set, DWDT will be cleared to be 000b when WDT is reset or interrupted, which should be set again by software.

8.2. WDT control register list and description:

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	-	ADIF	WDTIF	TB1IF	TMAIF	E1IF	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
02Ch	PSTAT			TO						$000 $00.	uu$u u$u.	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0,-
037h	WDTCN					ENWDT	DWDT[2:0]			0000 0000	uuuu $000	-,*,*,* rw1,*,*,*

Table 8-1 Registers related to WDT

INTE0/INTF0: please refer to the chapter "Interrupt" for more information.

PSTAT[7:0] peripheral state flag register

Bit	Name	Description
Bit5	TO	Watch dog (WDT) operation mode counting overflow flag <0> WDT counting overflow event does not occur. <1> WDT reset event has occurred; it should be cleared by BOR, RST or instruction.

WDTCN[7:0] watch dog control register

Bit	Name	Description
Bit3	ENWDT[0]	WDT timer enablement and disablement controller <0> OFF <1> ON
Bit2~0	DWDT[2:0]	Watch dog WDT_CK working frequency selector

DWDT[2:0]	Pre-scale
000	WDT_CK ÷ 16384
001	WDT_CK ÷ 8192
010	WDT_CK ÷ 4096
011	WDT_CK ÷ 2048
100	WDT_CK ÷ 64
101	WDT_CK ÷ 32
110	WDT_CK ÷ 16
111	WDT_CK ÷ 8

9. Timer-A1

The timer-A1 is of 8-bit design structure, and TMA1 can work under the operation mode and the standby mode.

- Up timer
- 8-order overflow value selection
- 8Bit overflow time comparator
- An interrupt event is generated after the overflow occurs.
- The value of the timer can be read.

Summary of TMA1 register:

TMA1CN	ENTMA1[0], TMACL1[0],TMA1S[0], DTMA1[2:0]
TMA1R	TMAR[7:0]
TMA1C	TMAC[7:0]
INTE0	GIE,TA1CIE
INTE1	TA1IE
INTF0	TA1CIF
INTF1	TA1IF

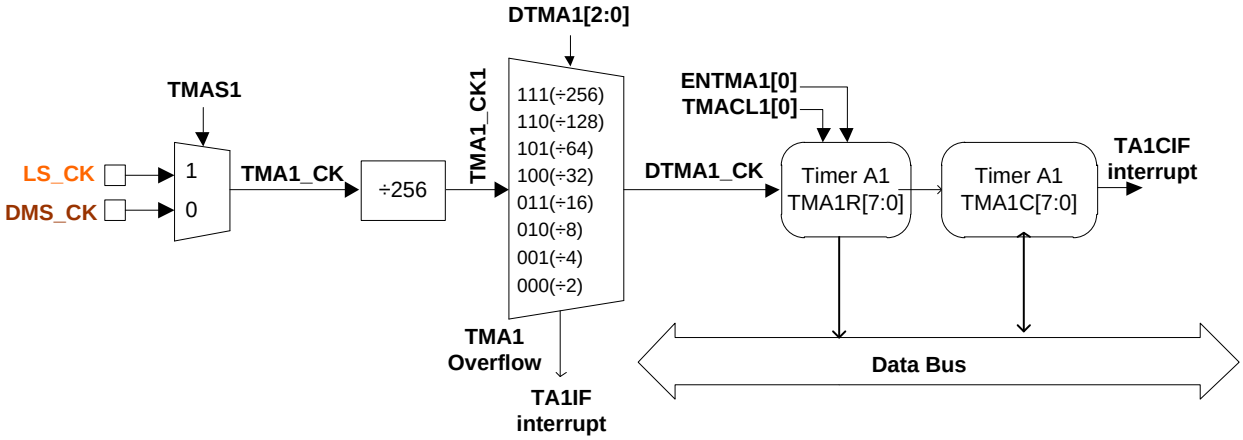


Figure 9-1 Block diagram of Timer-A1

- Operation description:

  Set TMAS1[0] to select the frequency of TMA1_CK, and reduce the frequency by the pre-frequency eliminator 256, and then input which into the DTMA1 pre-frequency eliminator.

  Set ENTMA1[0] as <1> to enable TMA1; on the contrary, set it as <0> to clear TMA1R[7:0].

  DTMA1[2:0] counting condition holds to generate an interrupt event, and make the cumulation of TMA1R[7:0] be added by 1.

  The interrupt event TMA1IF[0] of TMA1 can provide the interrupt service only after TMA1IE[0] is set as <1> and GIE[0] is set as <1>.

  Reading TMA1R[7:0] will not reset the TMA1 timer.

  After the user set TMACL1[0] as <1> to clear all timers of TMA1, TMACL1[0] is automatically set as <0>.

  TMAR1[7:0] can read the value of the TMA accumulating counter, and can clear the counting value of TMAR1[7:0] by written-in operation.

◆ Operation description of comparator of Timer A1:

Set TMAS1 to select the frequency of TMA1_CK; after processed by the pre-frequency eliminator 256m to serve as the frequency source of TMA1_CK1, and then input which into the DTMA1 frequency eliminator.

Set ENTMA1 as <1> to enable TMA1, and clear the timers, TMA1_CK, DTMA1_CK and TMA1R, and then perform the counting from 0; on the contrary, set it as <0> to disable TMA1.

The DTMA1[2:0] counting condition to generate an interrupt event (TA1IF), and make the cumulation of TMA1R[7:0] be added by 1,

The interrupt event TA1IF of TMA1 can provide the interrupt service only after TA1IE is set as <1> and GIE is set as <1>.

Reading TMA1R[7:0] will not reset the TMA1 timer.

The user set TMACL1 as <1> to clear the timers, TMA1_CK and DTMA1_CK，TMACL1[0] is automatically set as <0> by hardware.

TMA1R[7:0] can read the value of the TMA1 accumulating counter, and can clear the counting value of TMA1R[7:0] by written-in operation, and calculate again from TMA1R[7:0]=0.

TMA1C[7:0] is the comparison point register of Timer A1, and can be read an written. When BOR/POR occurs, TMA1C[7:0] will be set as 0. When the value of TMA1R[7:0] is cumulated to be equal to TMA1C[7:0], the TA1CIF flag is set as 1. Likewise, TA1CIF can provide the interrupt service only after TA1CIE is set as <1> and GIE is set as <1>.

If the design is to make the interrupt occurs when TA1IF=1sec but the demand is 60sec interrupt wake-up, it is possible to set TA1CIF=60sec interrupt and enable TA1CIE interrupt request to awake the chip.

9.1. Register description-TMA1

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	-	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024h	INTE1	TA1IE	-							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	-	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
027h	INTF1	TA1IF	-							0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
036h	OSCCN2						HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*,*,*,*,*,*,*,r
044h	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]			-	-	0000 00..	u0uu uu..	*,rw1,*,*,*,*,-,-
045h	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
046h	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0

Table 9-1 TMA1 control register

INTE0/INTE1/INTF0/INTF1: please refer to the chapter "Interrupt" for more information.

OSCCN0/ OSCCN1/OSCCN2: please refer to the chapter "Oscillator, clock source and power consumption management" for more information.

TMA1CN: timer A control register

Bit	Name	Description
Bit7	ENTMA1	Timer-A enablement controller <0> OFF. <1> Enable and clear the counters, TMA1_CK, DTMA1_CK and TMA1R, etc.
Bit6	TMACL1	Reset TMA1 timer. <0>Reset the TMA1 timer. <1>Set TMA1 TMACL1 as <1> to clear timers, TMA1_CK and DTMA1_CK; then, TMACL1 is automatically set as <0> by hardware.
Bit5	TMAS1	TMA1 working frequency selector <0>DMS_CK <1>OSC_LPO

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit4~2</td><td>DTMA1[2:0]</td><td>Enable and disable the controller.
<table>
<tr><th>DTMA1[2:0]</th><th>Frequency allocation selector</th><th>DTMA1[2:0]</th><th>Frequency allocation selector</th></tr>
<tr><td>000</td><td>TMA1_CK ÷ 2</td><td>100</td><td>TMA1_CK ÷ 32</td></tr>
<tr><td>001</td><td>TMA1_CK ÷ 4</td><td>101</td><td>TMA1_CK ÷ 64</td></tr>
<tr><td>010</td><td>TMA1_CK ÷ 8</td><td>110</td><td>TMA1_CK ÷ 128</td></tr>
<tr><td>011</td><td>TMA1_CK ÷ 16</td><td>111</td><td>TMA1_CK ÷ 256</td></tr>
</table>
After calculation, the interrupt time is:
0.512msec~ 65.536msec (at clock source=DMS_CK, and HAO=2MHz)
36.5msec~ 4681.1msec (at clock source=LPC_CK, and LPO=14kHz)
<table>
<tr><th>TMA1_CK(kHz)</th><th>TMA1_CK1(kHz)</th><th>DTMA1[2:0]</th><th>DTMA1_CK(kHz)</th><th>TA1IF(msec)</th></tr>
<tr><td rowspan="3">1000</td><td rowspan="3">3.90625</td><td>000b</td><td>1.953125</td><td>0.512</td></tr>
<tr><td>100b</td><td>0.122070313</td><td>8.192</td></tr>
<tr><td>111b</td><td>0.015258789</td><td>65.536</td></tr>
<tr><td rowspan="3">14</td><td rowspan="3">0.0546875</td><td>000b</td><td>0.02734375</td><td>36.57142857</td></tr>
<tr><td>100b</td><td>0.001708984</td><td>585.1428571</td></tr>
<tr><td>111b</td><td>0.000213623</td><td>4681.142857</td></tr>
</table></td></tr>
</table>

TMA1R: the incremental timer of TMA1 can be read but cannot be written in.

The written-in operation will be considered clearing the counting value of TMA1R[7:0] and then calculating again from TMA1R[7:0]=0.

TMA1C: the comparison point of Timer A1 can be read and written in.

10. 16-bit timerB (TMB)

TimerB (hereinafter TMB); TMB has two PWM outputs, which are PWMA0/1. Each TMB has four operation modes, and the timer of each mode has a special function design to satisfy different applications.

Summary of TMB registers:

INTE0	GIE, TMBIE
INTF0	TMBIF
OSCCN1	DTMB[1:0], TMBS
TB1Flag	PWM6A, PWM5A, PWM4A, PWM3A, PWM2A, PWM1A
TB1CN0	ENTB1, TB1M[1:0], TB1RT[1:0], TB1CL
TB1CN1	PA1IV, PWMA1[2:0], PA0IV, PWMA0[2:0]
TB1R	TB1RH[15:8], TB1RL[7:0]
TB1C0	TB1C0[15:8], TB1C0L[7:0]
TB1C1	TB1C1H[15:8], TB1C1L[7:0]
TB1C2	TB1C2H[15:8], TB1C2L[7:0]

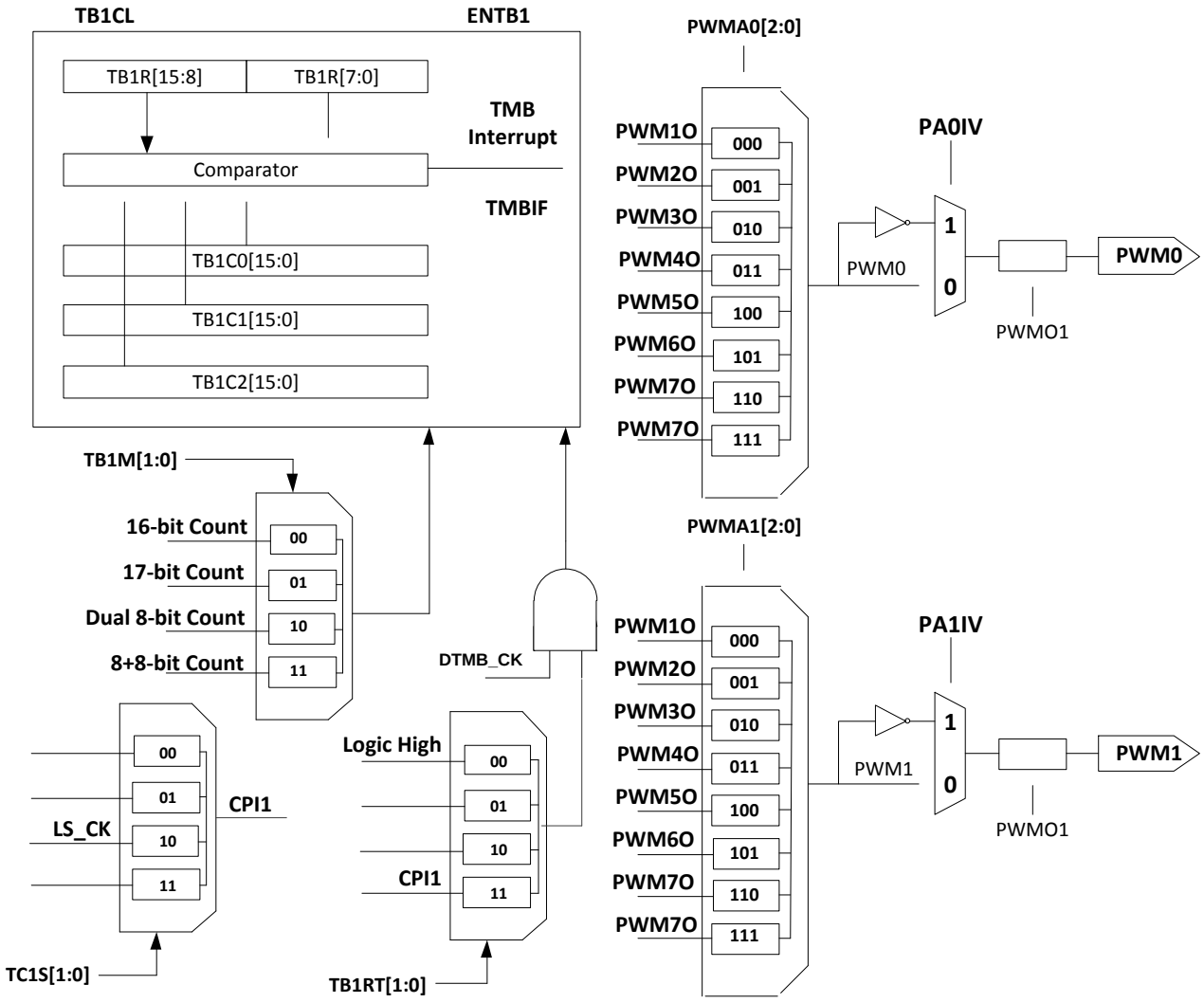


Figure 10-1 Structure diagram of TMB timer

- The counting registers of TMB include:

  Incremental / decremental timer TB1R[15:0]

  Overflow event condition controller TB1C0[15:0]

  PWMA condition controller TB1C1[15:0]

  PWMA condition controller TB1C2[15:0]

  Enablement controller ENTB1[0]

  Mode controller TB1M[1:0]

  Trigger controller TB1RT[1:0]

  Reset controller TB1CL[0]

  PWM0 output waveform selector PWMA0[2:0]

  PWM0 output phase reversal controller PA0IV[0]

  PWM1 output waveform selector PWMA1[2:0]

  PWM1 output phase reversal controller PA1IV[0]

  Working frequency source selector TMBS[0]

  Working frequency pre-frequency eliminator DTMB[1:0]

- Four counting modes of TMB

  16-bit counting

  17-bit counting

  Two 8-bit countings

  8+8-bit counting

- System power consumption operation of TMB

  Operation mode

  Standby mode

  Sleep mode

- TB1R[15:0] reset and re-counting condition

  Reading the registers related to TMB will not make TB1R[15:0] be reset and re-counting.

  Writing data in TB1R[15:0] (read-only), TB1C0[15:0], TB1C1[15:0] and TB1C2 [15:0] will not make TB1R[15:0] be reset and re-count.

  Writing data in TB1CN0 and TB1CN1 control registers will not make TB1R[15:0] be reset and re-count.

  TB1R[15:0] adopts cumulation counting; when the value is higher than TB1C0[15:0], TB1R[15:0] is reset to re-count.

  After the user set TB1CL[0] as <1> to clear the TB1R[15:0] timer, TB1CL[0] is automatically set as <0>.

10.1. Four counting modes of TMB

Timer B has four counting methods, which can be selected via the counting mode selector TB1M[1:0]. Each counting mode has different ways of overflow and interrupt event; the chapter describes the operation methods of the four counting modes.

In addition, different counting modes used together with the PWM condition selector can generate 7 kinds of different PWM waveforms. The following chapters will describe these waveforms one by one.

10.1.1. 16-bit timer

Setting the counting mode selector TB1M[1:0] as <00> makes TMB operate under the 16-bit counting mode; the mode has the following characteristics:

- When the counting of the TB1R[15:0] timer starts, different events can be triggered by setting TB1RT[1:0].
- When the cumulation counting of TB1R[15:0] is equal to TB1C0[15:0], the overflow event TB1IF[0] occurs and TB1R[15:0] is reset to re-count.

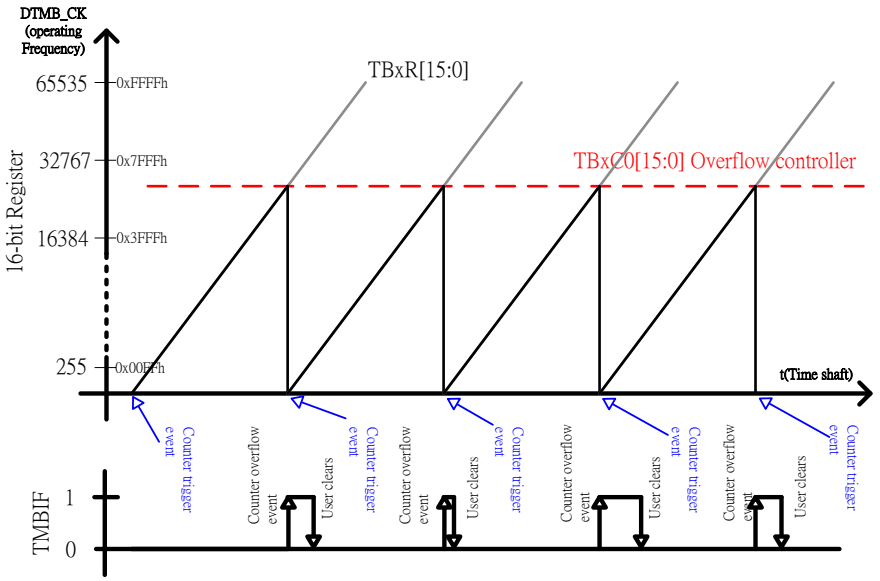


Figure 10-2 Waveform and usage schematic view of 16-bit timer

- Operation description of 16-bit counting mode
  - Initialization
    - Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
    - Set TB1M[1:0] as <00> to plan TMB1 as a 16-bit timer.
    - Write data in TB1C0[15:0].
  - Set TB1RT[1:0] as <00> to select the triggering counting signal as "Always Enable" state (i.e. cycle counting).
  - Set ENTB1[0] as <1> to enable the timer.
    - When the counting value of TB1R[15:0] is equal to TB1C0[15:0], an overflow event occurs to set TB1IF[0] as <1> to reset and perform up-counting again; currently, TB1IE[0]] is set as <1> to generate the interrupt event service.

- During the counting process, the user can set the counting reset controller TB1CL[0] as <1> to re-count, and TB1CL[0] is automatically set as <0>.

■ Set ENTB1[0] as <0> to disable the timer.

10.1.2. 17-bit timer

Set the mode selector TB1M[1:0] as <01> to make TMB operate under the 17-bit counting mode; the mode has the following characteristics:

- When the counting of the TB1R[15:0] timer starts, different events can be triggered by setting TB1RT[1:0].
- When the counting value of TB1R[15:0] is equal to TB1C0[15:0], it will be changed as up-counting after a half instruction period; when the up-counting to TB1R[15:0] is 0000h, the overflow event TB1IF[0] occurs and the up-counting is performed again.

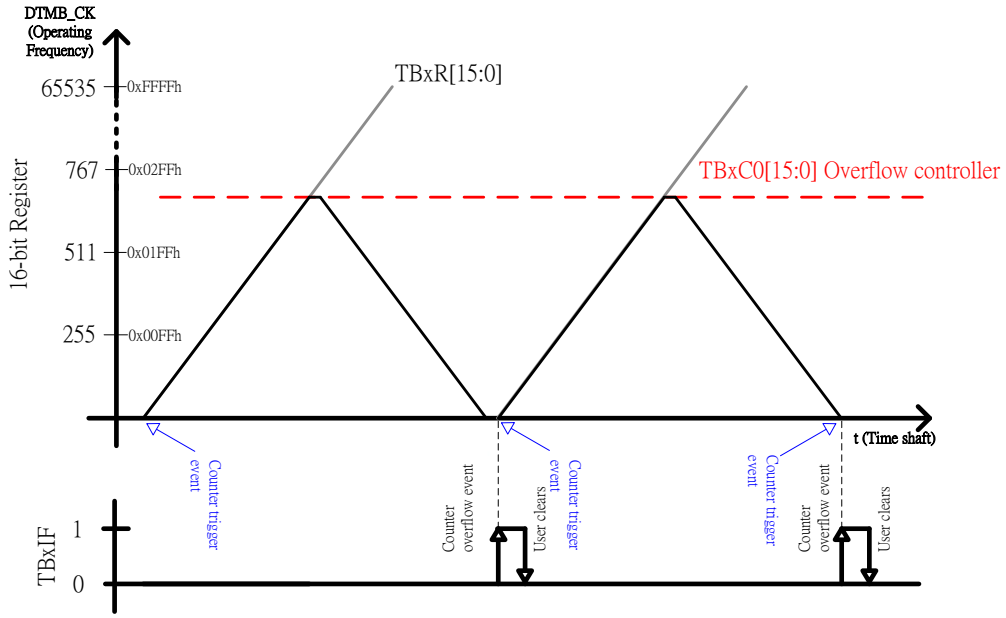


Figure 10-3 Waveform and usage schematic view of 17-bit timer

- Operation description of 17-bit counting mode
  - Initialization
    - Set TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] to determine the working frequency of TMB.
    - Set TB1M[1:0] as <01> to plan TMB1 as a 17-bit timer.
    - Write data in TB1C0[15:0].
  - Set TB1RT[1:0] as <00> to select the triggering counting signal is "Always Enable" state (i.e. cycle counting).
  - Set ENTB1[0] as <1> to enable the timer.
    - When the counting value of TB1R[15:0] is equal to TB1C0[15:0], it will be changed as down-counting after a half instruction period; when the down-counting to TB1R[15:0] is 0000h, an overflow event occurs to set TB1IF[0] as <1> to reset, and the up-counting is performed again; currently, setting TB1IE[0] as <1> will generate an interrupt event service.

- During the counting process, the user can set the counting reset controller TB1CL[0] as <1> to re-count, and TB1CL[0] is automatically set as <0>.

■ Set ENTB1[0] as <0> to disable the timer.

10.1.3. Two 8-bit timers

Set the mode selector TB1M[1:0] as <10> to make TMB operate under the 8-bit counting mode; the mode has the following characteristics:

- When the counting of the two 8-bit timers, TB1R[7:0] and TB1R[15:8], starts, different events can be triggered at the same time by setting TB1RT[1:0].
- When the cumulation counting of TB1R[7:0] is equal to TB1C0[7:0], the overflow event TB1IF[0] occurs and TB1R[7:0] is reset to re-count.
- When the cumulation counting of TB1R[15:8] is equal to TB1C0[15:8], the overflow event occurs and TB1R[15:8] is reset to re-count.

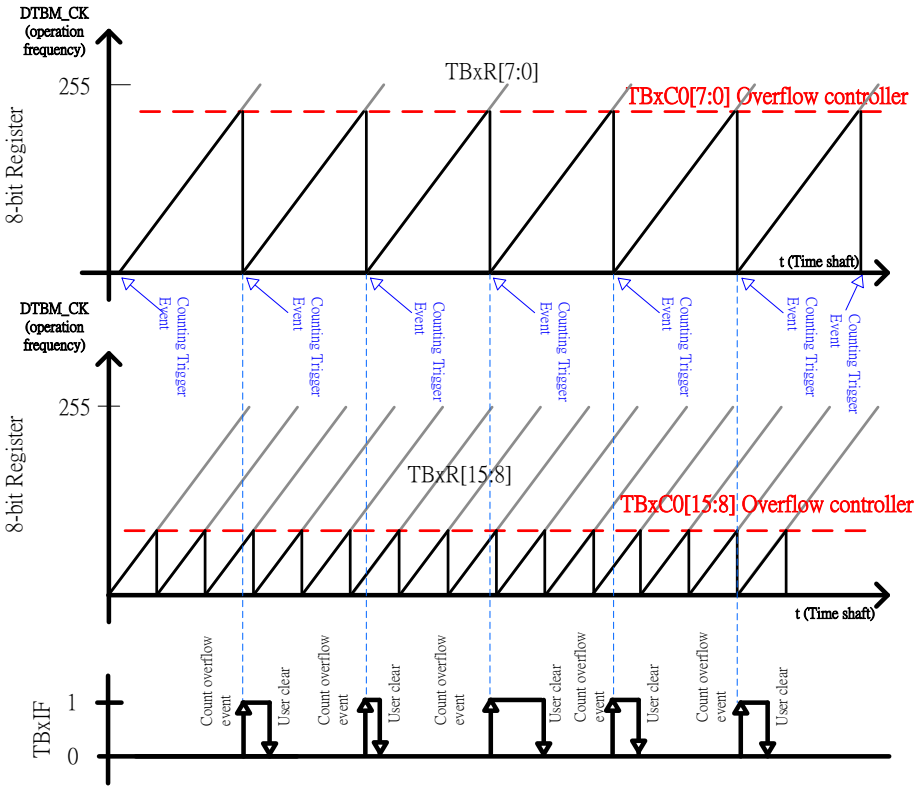


Figure 10-4 Waveform and usage schematic view of two 8-bit timers

- Operation description of two 8-bit counting modes
  - Initialization
    - Set TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] to determine the working frequency of TMB.
    - Set TB1M[1:0] as <10> to plan TMB1 as two 8-bit timers.
    - Write data in TB1C0[7:0] and TB1C0[15:8] respectively.
  - Set TB1RT[1:0] as <00> to select the triggering counting signal is "Always Enable" state (i.e. cycle counting).
  - Set ENTB1[0] as <1> to enable the timer.
    - When the counting value of TB1R[7:0] is equal to TB1C0[7:0], an overflow event occurs to set TB1IF[0] as <1> to reset and perform up-counting again; currently, TB1IE[0]] is set as <1> to generate the interrupt event service.

- When the counting value of TB1R[15:8] is equal to TB1C0[15:8], an overflow event occurs, and TB1R[15:8] is reset to perform up-counting again.
- During the counting process, the user can set the counting reset controller TB1CL[0] as <1> to make TB1R[7:0] and TB1R[15:8] to re-count, and TB1CL[0] is automatically set as <0>.

■ Set ENTB1[0] as <0> to disable the timer.

10.1.4. 8+8-bit timer

Set the mode selector TB1M[1:0] as <11> to make TMB operate under the 8+8-bit counting mode; the mode has the following characteristics:

- When the counting of the 8+8-bit timers, TB1R[15:8] and TB1R[7:0], starts, different events can be triggered by setting TB1RT[1:0].
- When the counting value of TB1R[7:0] is equal to TB1C0[7:0], the overflow event TB1IF[0] occurs and the TB1R[15:8] is cumulated by 1 to reset TB1R[7:0] to re-count.

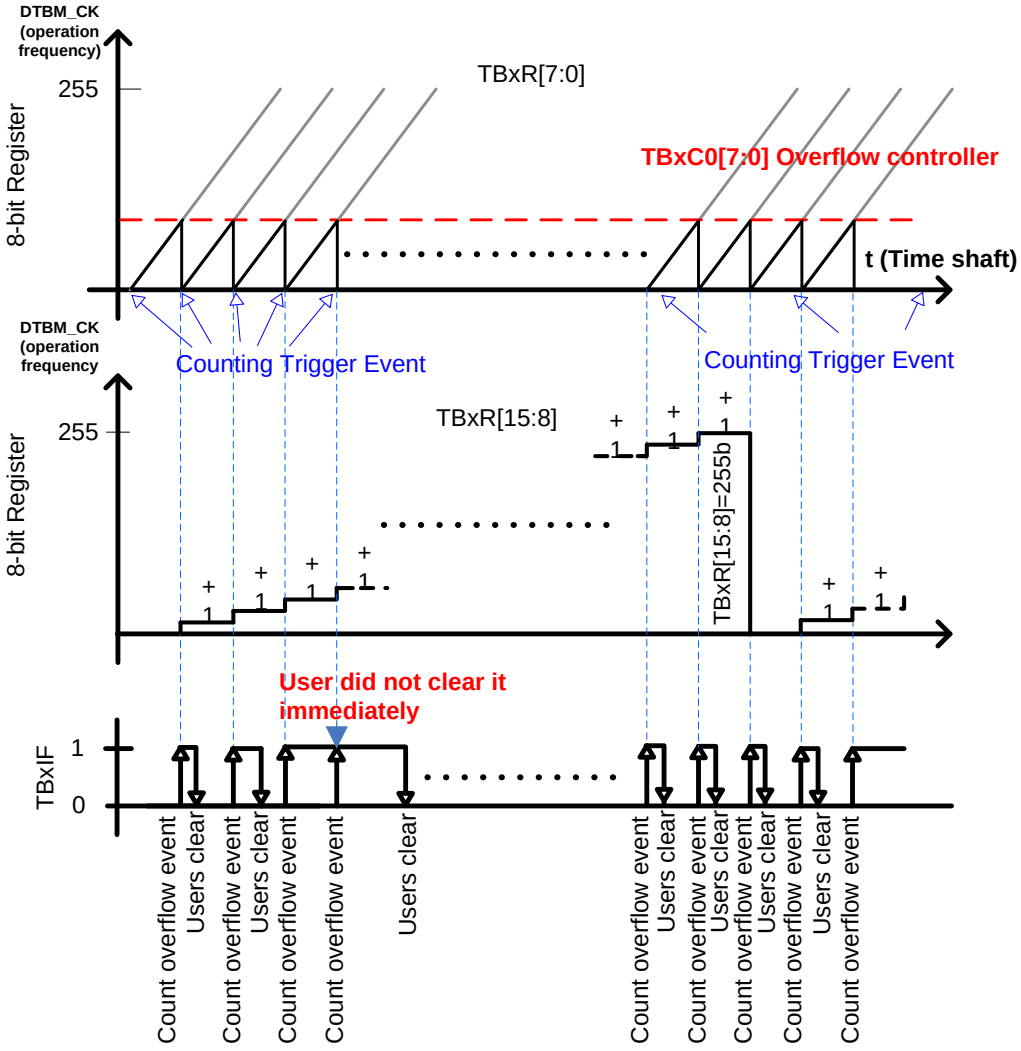


Figure 10-5 Waveform and usage schematic view of two 8+8-bit timer

- Operation description of 8+8-bit counting mode
  - Initialization
    - Set TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] to determine the working frequency of TMB.
    - Set TB1M[1:0] as <11> to plan TMB1 as a 8+8-bit timer.
    - Write data in TB1C0[7:0].

- Set TB1RT[1:0] as <00> to select the triggering counting signal is "Always Enable" state (i.e. cycle counting).
- Set ENTB1[0] as <1> to enable the timer.
  - When the counting value of TB1R[7:0] is equal to TB1C0[7:0], an overflow event occurs to set TB1IF[0] as <1> and the cumulation of the TB1R[15:8] timer is added by 1; currently, setting TB1IE[0] as <1> will generate an interrupt event service to reset and perform up-counting again..
  - When the counting value of TB1R[15:8] is equal to TB1R[15:8]=255b, adding 1 will reset TB1R[15:8] to perform up-counting again.
  - During the counting process, the user can set the counting reset controller TB1CL[0] as <1> to make TB1R[7:0] and TB1R[15:8] re-count at the same time, and TB1CL[0] is automatically set as <0>.
- Set ENTB1[0] as <0> to disable the timer.

10.2. Pulse width modulation (PWM)

When different counting modes of TMB are combined with the pulse width modulation (hereinafter PWM) mode selector, a lot of types of PWM waveforms can be generated, where PWMA0/1 is actually an output pin. The chapter introduces 7 kinds of different usage methods for users' reference.

- ◆ Basic operation description of TMB and PWM output, and their relation
- ■ TMB1 control the outputs of PWMA0 and PWMA1.
  - The PWM mode selectors, PWMA0[2:0] and PWMA1[2:0] can set the output waveform of PWMA0 and PWMA1 is one of PWM1O~PWM7O.
  - The waveform state flags, PWMA1[0]～PWMA6[0], can read "H" or "L" state of PWM1O～PWM6O.
  - By means of the output inverters, PA0IV[0] and PA1IV[0] of PWM, the actual output waveforms of PWMA0 and PWMA1 can be set to be of phase reversal or not.
  - The outputs of PWMA0 and PWMA1 can be performed from the pins, PT7.6(SEG12) and PT7.7(SEG13).
- ◆ The PWM mode selector, PWMA0/1[2:0], can output the waveforms, PWM1O~PWM7O. Please note that when PWM1O~PWM7O are used together with different TMB counting modes, PWM1O~PWM7O can output completely different waveforms; the following chapter will describe basic type and frequently-used application.

10.2.1. PWM1O waveform (16-bit PWM)

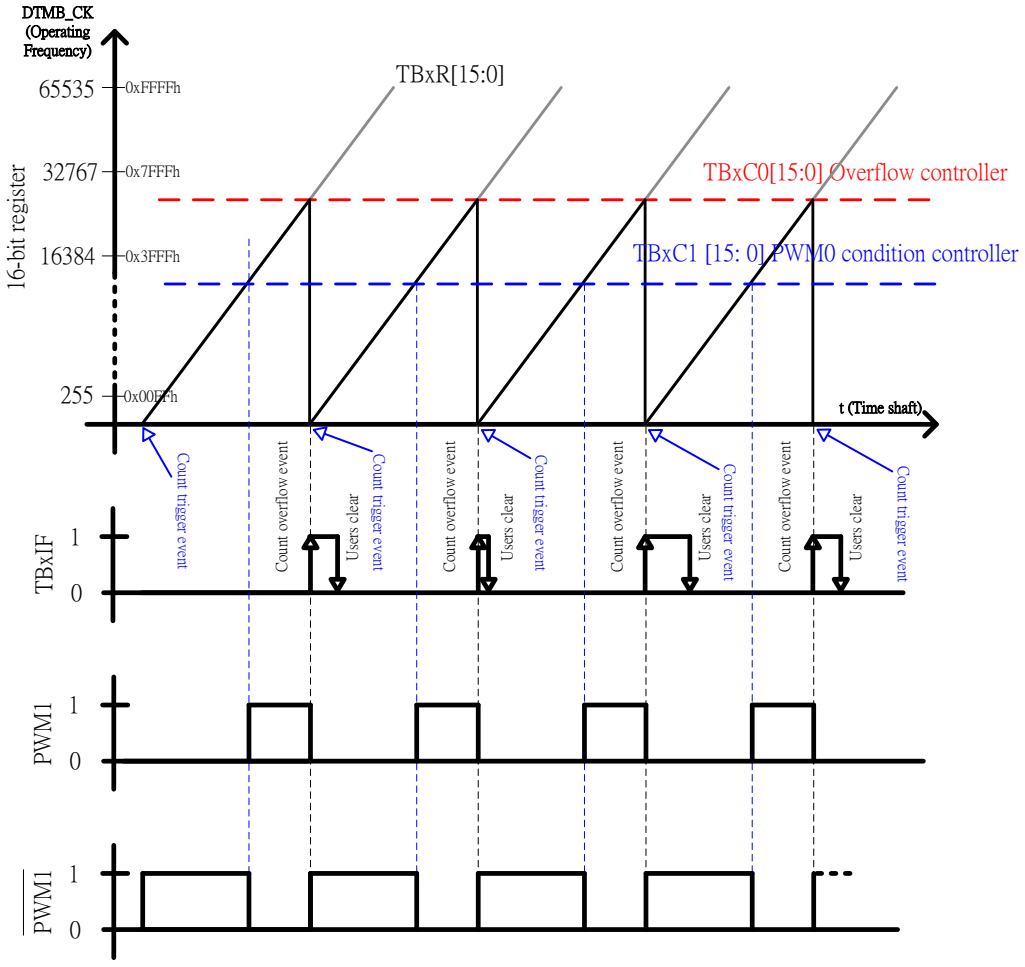


Figure 10-6 Waveform and usage schematic view of PWM1O

- ◆ Operation description of PWM1O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - ● Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - ● Set TB1M[1:0] as <00> to plan TMB1 as a 16-bit timer.
  - ● Set PWMA0/1[2:0] as <000> to output PWM1O waveform.
  - ● Set TB1RT[1:0] as <00> to set the triggering counting signal as Logic High.
  - ● Write data in TB1C0[15:0] to determine the frequency of PWM.
  - ● Write data in TB1C1[15:0] to determine the duty cycle of PWM.
  - ● Set ENTB1[0] as <1> to enable the timer.
- ■ Generate PWM1O waveform
  - ● When the counting value of TB1R[15:0] is equal to TB1C1[15:0], the status of PWM1O is changed from 0➔1.
  - ● When the counting value of TB1R[15:0] is equal to TB1C0[15:0], the status of PWM1O is changed from 1➔0; then, an overflow event is generated to set

TB1IF[0] as <1> and reset it to perform up-counting again; currently, setting TB1IE[0]] as <1> will generate an interrupt event service.

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM1O:

$$\text{PWM1O Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0}[15:0]+1}$$

$$\text{PWM1O Duty Cycle} = \frac{(\text{TB1C0}[15:0]+1) - \text{TB1C1}[15:0]}{\text{TB1C0}[15:0]+1}$$

10.2.2. PWM2O waveform (16-bit PWM)

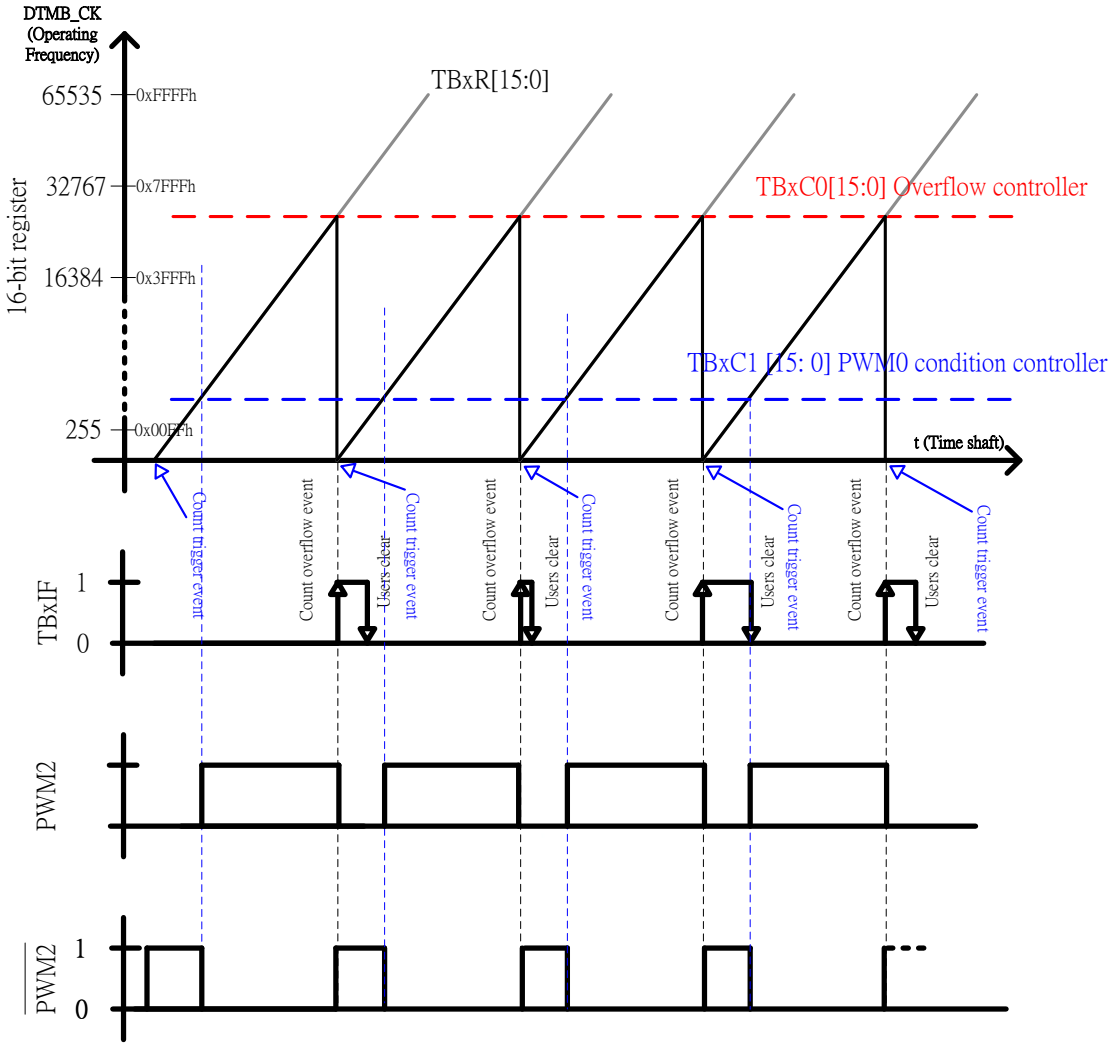


Figure 10-7 Waveform and usage schematic view of PWM2O

- ◆ Operation description of PWM2O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - ● Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - ● Set TB1M[1:0] as <00> to plan TMB1 as a 16-bit timer.
  - ● Set PWMA0/1[2:0] as <001> to output PWM2O waveform.
  - ● Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
  - ● Write data in TB1C0[15:0] to determine the frequency of PWM.
  - ● Write data in TB1C2[15:0] to determine the duty cycle of PWM.
  - ● Set ENTB1[0] as <1> to enable the timer.
- ■ Generate PWM2O waveform
  - ● When the counting value of TB1R[15:0] is equal to TB1C2[15:0], the status of PWM2O is changed from 0➔1.

- When the counting value of TB1R[15:0] is equal to TB1C0[15:0], the status of PWM2O is changed from 1➔0; then, an overflow event is generated to set TB1IF[0] as <1> and reset it to perform up-counting again; currently, setting TB1IE[0]] as <1> will generate an interrupt event service.

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM2O:

$$\text{PWM2O Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0}[15:0]+1}$$

$$\text{PWM2O Duty Cycle} = \frac{(\text{TB1C0}[15:0]+1) - \text{TB1C2}[15:0]}{\text{TB1C0}[15:0]+1}$$

10.2.3. PWM3O waveform (8-bit PWM)

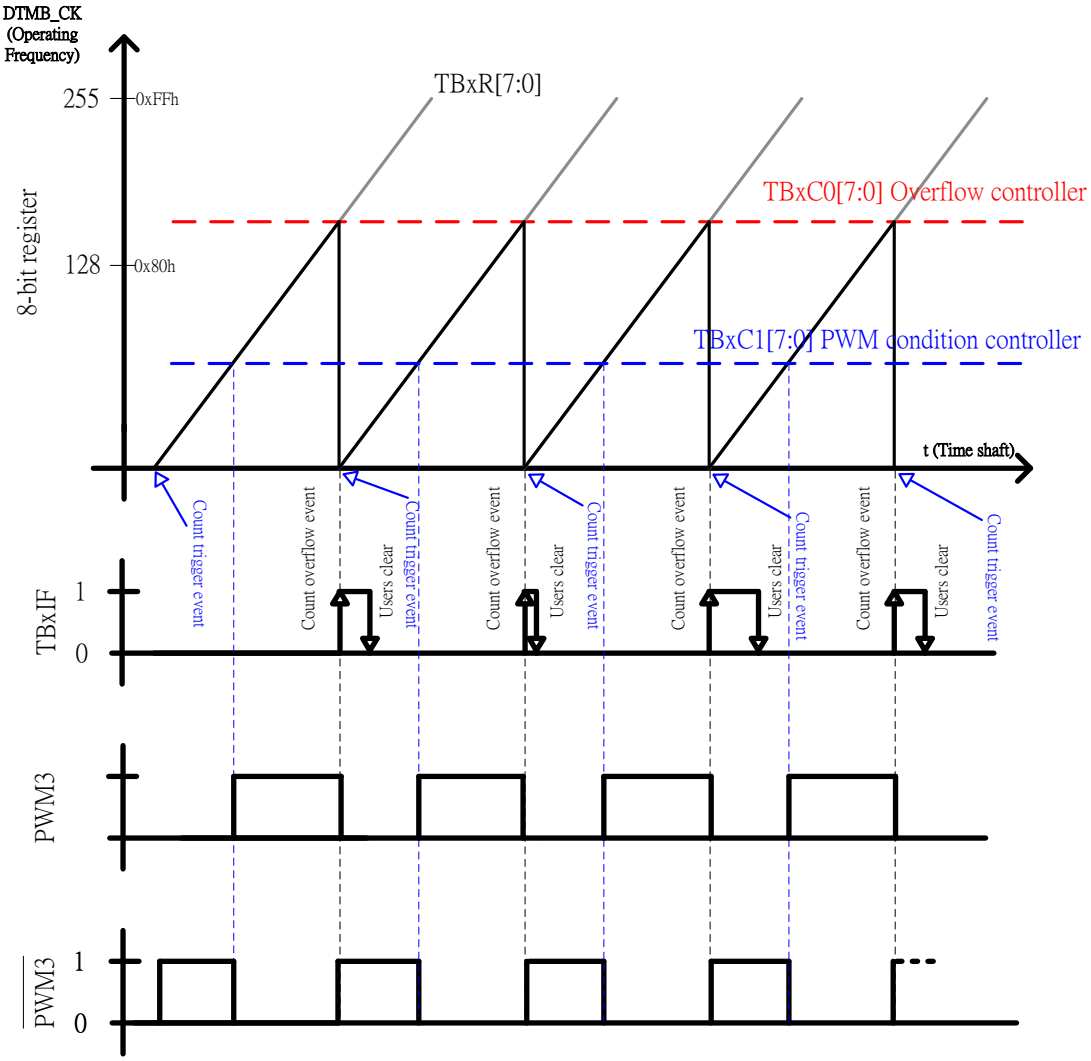


Figure 10-8 Waveform and usage schematic view of PWM3O

- ◆ Operation description of PWM3O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - Set TB1M[1:0] as <10> to plan TMB1 as two 8-bit timers.
  - Set PWMA0/1[2:0] as <010> to output PWM3O waveform.
  - Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
  - Write data in TB1C0L[7:0] to determine the frequency of PWM.
  - Write data in TB1C1L[7:0] to determine the duty cycle of PWM.
  - Set ENTB1[0] as <1> to enable the timer.
- ■ Generate PWM3O waveform
  - When the counting value of TB1RL[7:0] is equal to TB1C1L[7:0], the status of PWM3O is changed from 0➔1.

- When the counting value of TB1RL[7:0] is equal to TB1C0L[7:0], the status of PWM3O is changed from 1➔0; then, an overflow event is generated to set TB1IF[0] as <1> and reset it to perform up-counting again; currently, setting TB1IE[0]] as <1> will generate an interrupt event service.

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM3O:

$$\text{PWM3O Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0L}[7:0]+1}$$

$$\text{PWM3O Duty Cycle} = \frac{(\text{TB1C0L}[7:0]+1) - \text{TB1C1L}[7:0]}{\text{TB1C0L}[7:0]+1}$$

10.2.4. PWM4O waveform (8-bit PWM)

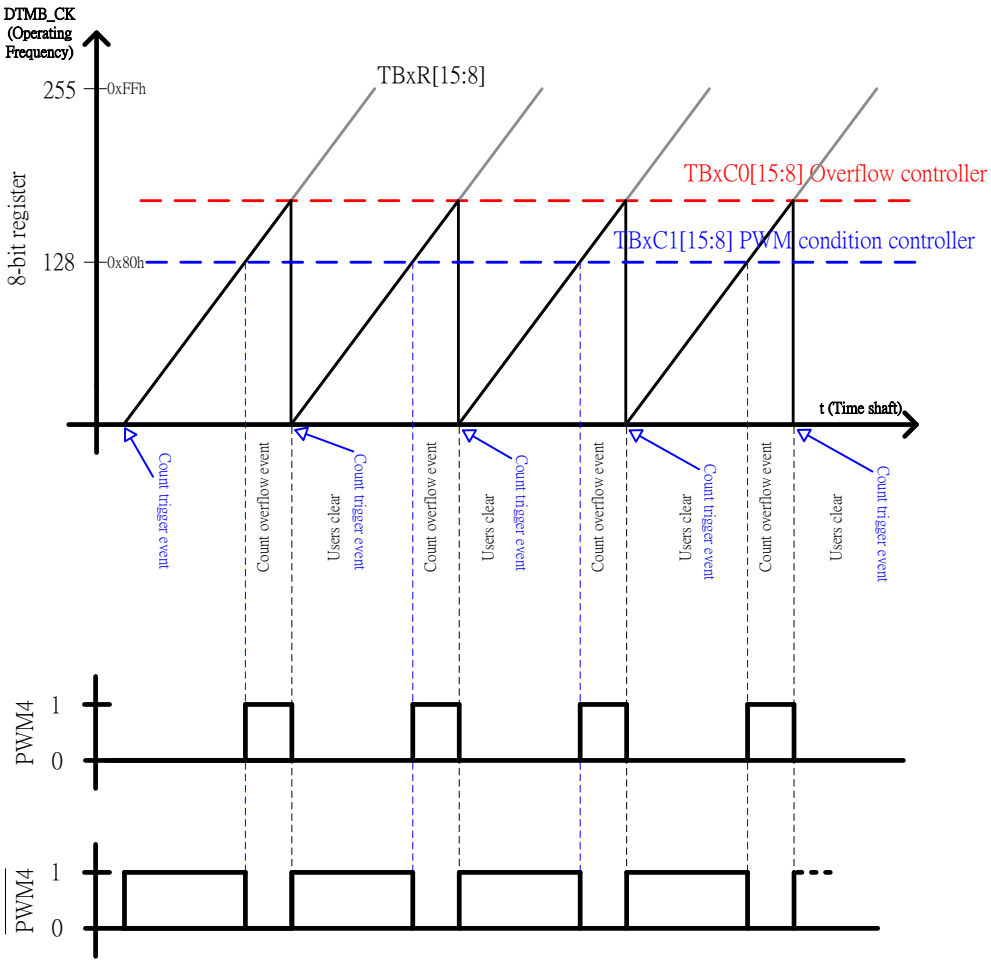


Figure 10-9 Waveform and usage schematic view of PWM4O

- ◆ Output operation description of PWM4O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - Set TB1M[1:0] as <10> to plan TMB1 as two 8-bit timers.
  - Set PWMA0/1[2:0] as <011> to output PWM4O waveform.
  - Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
  - Write data in TB1C0H[15:8] to determine the frequency of PWM.
  - Write data in TB1C2L[7:0] to determine the duty cycle of PWM.
  - Set ENTB1[0] as <1> to enable the timer.
- ■ Generate PWM4O waveform.
  - When the counting value of TB1R[7:0] is equal to TB1C2L[7:0], the status of PWM4O is changed from 0➔1.
  - When the counting value of TB1RL[7:0] is equal to TB1C0H[15:8], the status of PWM4O is changed from 1➔0 and reset it to perform up-counting again.

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM4O:

$$\text{PWM4O Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0H}[15:8]+1}$$

$$\text{PWM4O Duty Cycle} = \frac{(\text{TB1C0H}[15:8]+1) - \text{TB1C2L}[7:0]}{\text{TB1C0H}[15:8]+1}$$

10.2.5. PWM5O waveform (8+8-bit PWM)

Setting the TMB timer as the 8+8-bit mode and setting the output waveform of PWM as PWM5O can obtain 8+8bit PWM output.

8+8-bit PWM is composed of the control registers, TB1R[7:0], TB1C0[7:0], TB1C1[7:0] and TB1C2[7:0], etc., and internal digital circuit, where TB1R[7:0] is the accumulating counter, TB1C0[7:0] is the PWM frequency controller, TB1C1[7:0] is the PWM duty cycle controller, TB1C2[7:0] is the 8+8-bit PWM duty cycle fine-tuner.

The configuration and description of the duty cycle trimmer, TB1C2[7:0], of the 8+8-bit PWM are as shown in the following table:

Configuration / Weighting amount	TB1C2[7:0]							
	80h	40h	20h	10h	08h	04h	02h	01h
Fine-tuning of duty cycle of PWM	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
Description	2 times of TMB overflows; one half is N+1 and the other is N.	4 times of TMB overflows; one half is N+1 and the other is N.	8 times of TMB overflows; one half is N+1 and the other is N.	16 times of TMB overflows; one half is N+1 and the other is N.	32 times of TMB overflows; one half is N+1 and the other is N.	64 times of TMB overflows; one half is N+1 and the other is N.	128 times of TMB overflows; one half is N+1 and the other is N.	256 times of TMB overflows; one half is N+1 and the other is N.

Table 10-1 Configuration table of duty cycle trimmer

- ◆ The description of the duty cycle trimmer TB1C0[7:0], where N is the width of the duty cycle (*PS: N = TB1C0[7:0] – TB1C1[7:0]*).
- ■ Basic type
  - ● Set TB1C2[7:0] as 80h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 2 output cycles; one output is N+1 and the other one output is N.
  - ● Set TB1C2[7:0] as 40h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes

one set of 4 output cycles; two output are N+1 and the other two outputs are N.

- Set TB1C2[7:0] as 20h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 8 output cycles; 4 consecutive outputs are N+1 and the other 4 outputs are N.
- Set TB1C2[7:0] as 10h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 16 output cycles; 8 consecutive outputs are N+1 and the other 8 outputs are N.
- Set TB1C2[7:0] as 08h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 32 output cycles; 16 consecutive outputs are N+1 and the other 16 outputs are N.
- Set TB1C2[7:0] as 04h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 64 output cycles; 32 consecutive outputs are N+1 and the other 32 outputs are N.
- Set TB1C2[7:0] as 02h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 128 output cycles; 64 consecutive outputs are N+1 and the other 64 outputs are N.
- Set TB1C2[7:0] as 01h to make the waveform of the duty cycle of PWM generate the outputs of N+1 and N. In other words, the waveform includes one set of 256 output cycles; 128 consecutive outputs are N+1 and the other 128 outputs are N.

■ Logic operation OR superposition type

(the chapter only illustrates 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)

- Set TB1C2[7:0] as C0h(1/2+1/4) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 4 output cycles; one output is N+1 and the other 3 outputs are N.
- Set TB1C2[7:0] as A0h(1/2+1/8) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 8 output cycles; 2 outputs are N+1 and the other 6 outputs are N.
- Set TB1C2[7:0] as 90h(1/2+1/16) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform

includes one set of 16 output cycles; 4 outputs are N+1 and the other 12 outputs are N.

- Set TB1C2[7:0] as 88h(1/2+1/32) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 32 output cycles; 8 outputs are N+1 and the other 24 outputs are N.
- Set TB1C2[7:0] as 84h(1/2+1/64) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 64 output cycles; 16 outputs are N+1 and the other 40 outputs are N.
- Set TB1C2[7:0] as 90h(1/2+1/128) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 128 output cycles; 32 outputs are N+1 and the other 96 outputs are N.
- Set TB1C2[7:0] as 90h(1/2+1/256) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 256 output cycles; 64 outputs are N+1 and the other 192 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 8 output cycles; 1 output is N+1 and the other 7 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 8 output cycles; 1 output is N+1 and the other 7 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8+1/16) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 16 output cycles; 1 output is N+1 and the other 15 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8+1/16+1/32) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 32 output cycles; 1 output is N+1 and the other 31 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8+1/16+1/32+1/64) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 64 output cycles; 1 output is N+1 and the other 63 outputs are N.

- Set TB1C2[7:0] as C0h(1/2+1/4+1/8+1/16+1/32+1/64+1/128) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 128 output cycles; 1 output is N+1 and the other 127 outputs are N.
- Set TB1C2[7:0] as C0h(1/2+1/4+1/8+1/16+1/32+1/64+1/128+1/256) to make the waveform of the duty cycle of PWM to generate the outputs of N+1 and N. In other words, the waveform includes one set of 256 output cycles; 1 output is N+1 and the other 255 outputs are N.

◆ The following Table 10-2 and Figure 10-10 partially show the waveform change of 8+8-bit PWM when TB1C2[7:0] is under different configurations for users' reference.

8+8bit PWM			Overflow times of TBN																		
Type	TBC2 [7:0]	Logic operation	0	1	2	3	4	5	6	7	8	9	10	~	127	128	~	252	253	254	255
Basic waveform	80h	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
	40h	1/4	N	N	N+1	N	N	N	N+1	N	N	N	N+1	~	N	N	~	N	N	N+1	N
	20h	1/8	N	N	N	N	N+1	N	N	N	N	N	N	~	N	N	~	N+1	N	N	N
	10h	1/16	N	N	N	N	N	N	N	N	N+1	N	N	~	N	N	~	N	N	N	N
	08h	1/32	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	04h	1/64	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	02h	1/128	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	01h	1/256	N	N	N	N	N	N	N	N	N	N	N	~	N	N+1	~	N	N	N	N
Logic opeation superposition	C0h	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
	A0h	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
	E0h	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
	F0h	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
	A1h	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
	F1h	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
	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 PWM5O output waveform table

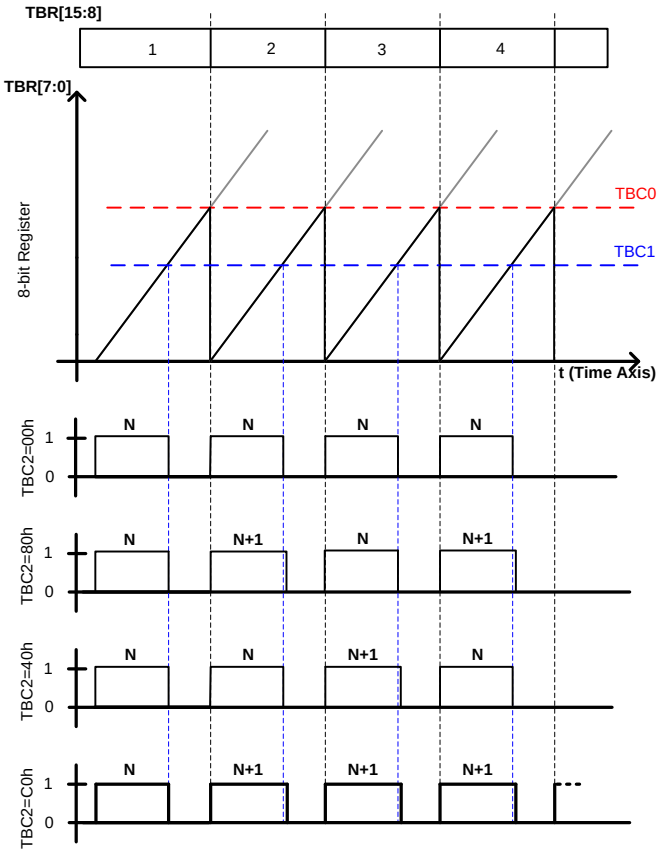


Figure 10-10 Waveform and usage schematic view of PWM5O

◆ Operation description of PWM5O

■ Initialization (the frequency and duty cycle configuration of PWM)

- Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
- Set TB1M[1:0] as <11> to plan TMB1 as a 8+8-bit timer.
- Set PWMA0/1[2:0] as <100> to output PWM5O waveform.
- Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
- Write data in TB1C0L[7:0] to determine the frequency of PWM.
- Write data in TB1C1L[7:0] to determine the duty cycle of PWM.
- Write data in TB1C2L[7:0] to determine the fine-tuning method of the duty cycle of PWM.
- Set ENTB1[0] as <1> to enable the timer.

■ Generate PWM5O waveform.

- When the counting value of TB1RL[7:0] is equal to TB1C1L[7:0], the status of PWM5O is changed from 0➔1.
- When the counting value of TB1RL[7:0] is equal to TB1C0L[7:0], the status of PWM5O is changed from 1➔0; then, an overflow event is generated to set

TB1IF[0] as <1> and reset it to perform up-counting again; currently, setting TB1IE[0]] as <1> will generate an interrupt event service..

- Currently, the data set by TB1C2L[7:0] will adjust the output of PWM5O as N+1 and N, as shown in Table 10-1, where N＝TB1C1[7:0]－TB1C0[7:0].

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM5O:

$$\text{PWM5O Frequency} = \frac{\text{DTMB\_CK}}{\text{TBxC0}[7:0]+1}$$

$$\text{PWM5O Duty Cycle} = \frac{(\text{TB1C0L}[7:0]+1) - \text{TB1C1L}[7:0] + \textit{TB1C2L}[7:0]/256}{\text{TB1C0L}[7:0]+1}$$

10.2.6. PWM6O waveform (two 16-bit PWM waveforms)

Setting the TMB timer to be under the 17-bit mode and set the output waveform of PWM as PWM6O can generate two 16-bit PWM waveforms.

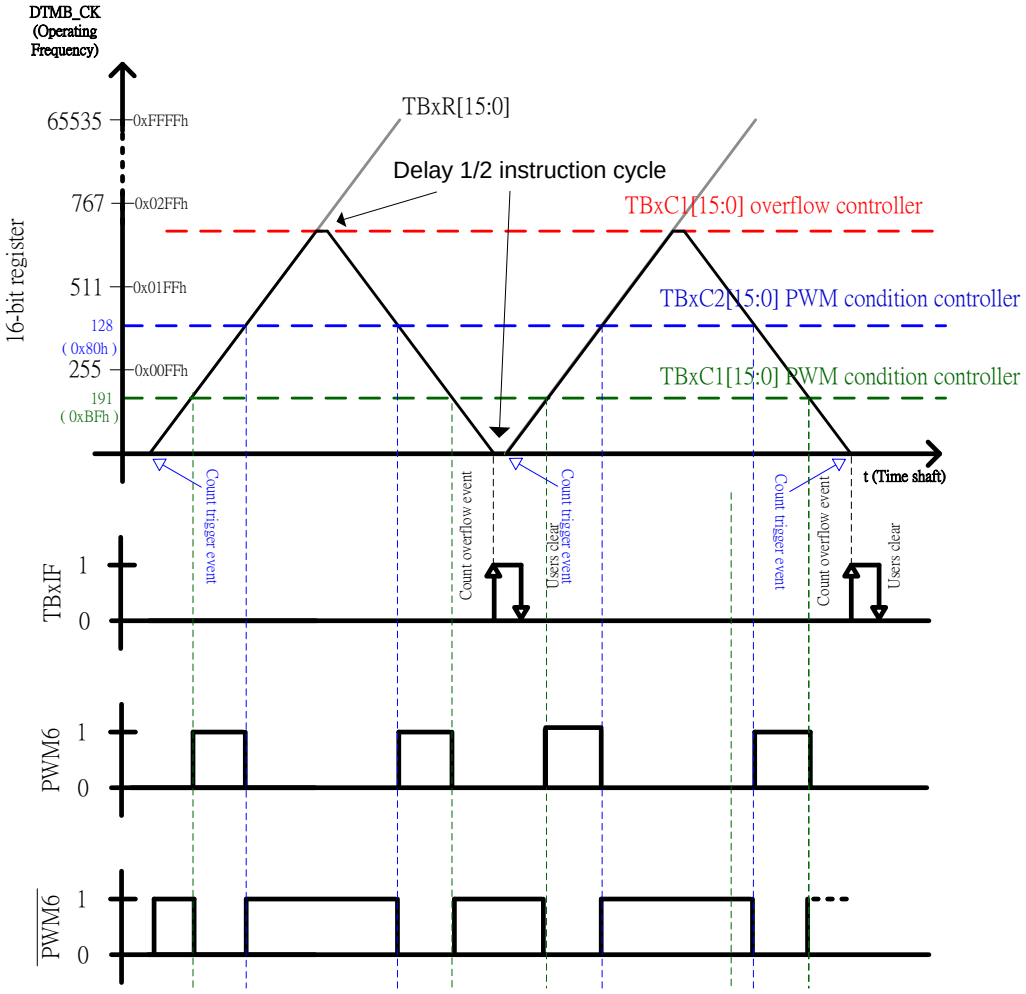


Figure 10-11 Waveform and usage schematic view of PWM6O

- ◆ Output operation description of 17-bit PWM
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - ● Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - ● Set TB1M[1:0] as <01> to plan TMB1 as a 17-bit timer.
  - ● Set PWMA0/1[2:0] as <101> to output PWM6O waveform.
  - ● Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
  - ● Write data in TB1C0H[15:8] to determine the frequency of PWM.
  - ● Write data in TB1C1L[15:0] and TB1C2[15:0] to determine the duty cycle of PWM.
  - ● Set ENTB1[0] as <1> to enable the timer.
- ■ Generate dual PWM6O waveforms.
  - ● Condition of first waveform

- ✓ When the up-counting value of TB1R[15:0] is equal to TB1C1[15:0], the status of PWM6O is changed from 0➔1.
- ✓ When the up-counting value of TB1R[15:0] is further equal to TB1C2[15:0], the status of PWM6O is changed from 1➔0.
- ✓ Next, when the counting value of TB1R[15:0] is equal to TB1C0[15:0], TB1R[15:0] is changed to down-counting.
- ● Condition of second waveform
- ✓ When the up-counting value of TB1R[15:0] is equal to TB1C2[15:0], the status of PWM6O is changed from 0➔1.
- ✓ When the up-counting value of TB1R[15:0] is further equal to TB1C1[15:0], the status of PWM6O is changed from 1➔0.
- ✓ Next, when the counting value of TB1R[15:0] is equal to 0x0000h, an overflow event generates to set TB1IF[0] as <1> to reset it and perform up-counting again; meanwhile, setting TB1IE[0]] as <1> to generate an interrupt event service.

■ Output control of PWM

- ● Set PM4.7/PM4.6] as <1> to enable PWM Mode.
- ● Set TC4.7/TC4.6[0] as <1> to enable the output function.
- ● Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.

■ Set ENTB1[0] as <0> to disable the timer and the output of PWM.

■ The frequency and duty cycle calculation of PWM6O fail to be described in the chapter because it will generate special waveforms.

10.2.7. PWM7O waveform (16-bit PWM waveform)

Setting the TMB timer be under the 16-bit mode and setting the output waveform of PWM as PWM7O can generate a periodical PWM waveform.

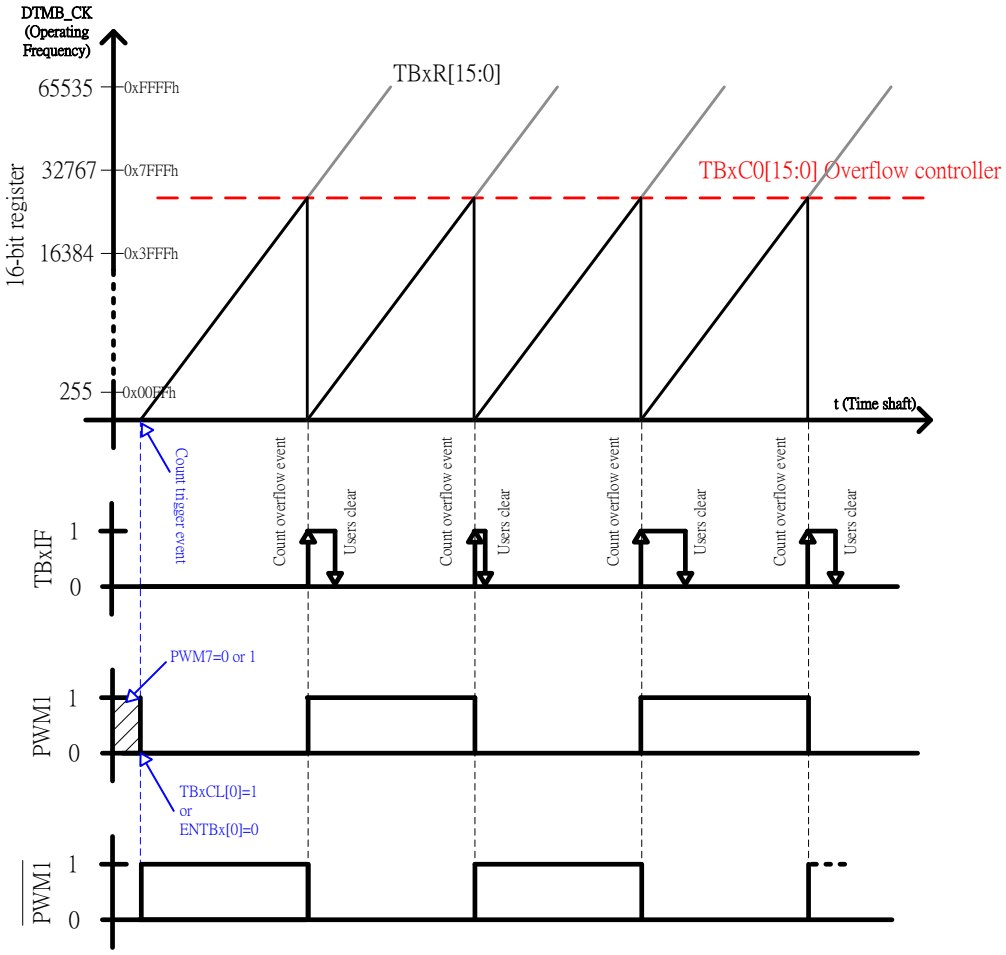


Figure 10-12 Waveform and usage schematic view of PWM7O

- ◆ Operation description of PWM7O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the working frequency source of TMB; setting DTMB[1:0] can determine the working frequency of TMB.
  - Set TB1M[1:0] as <00> to plan TMB1 as a 16-bit timer.
  - Set PWMA0/1[2:0] as <11> to output PWM7O waveform.
  - Set TB1RT[1:0] as <00> to set the triggering counting signal as "Always Enable" (i.e. cycle counting).
  - Write data in TB1C0[15:0] to determine the frequency of PWM.
  - Set ENTB1[0] as <1> to enable the timer.
- ■ Generate PWM7O waveform.
  - When TMB1 is not enabled, the state of PWM7O is not determined; however, when ENTB1[0] is set as <1> or TB1CL[0] is set as <1>, PWM7O outputs 0 until an overflow event occurs; then, PWM7O is transited to output 1; currently,

when the overflow event occurs again, PWM7O is transited to output 0 to generate a periodical waveform,

- When counting value of TB1R[15:0] is equal to TB1C0[15:0] again, PWM7O is transited and an overflow event generates to set TB1IF[0] as <1> to reset it and perform the up-counting again; meanwhile, setting TB1IE[0]] as <1> will generate an interrupt event service.

- Output control of PWM
  - Set PM4.7/PM4.6] as <1> to enable PWM Mode.
  - Set TC4.7/TC4.6[0] as <1> to enable the output function.
  - Set PA0/1IV[0] to determine whether the output waveform of the pin is of phase reversal or not.
- Set ENTB1[0] as <0> to disable the timer and the output of PWM.
- Frequency and duty cycle calculation formulas of PWM7O:

$$\text{PWM7O Frequency} = \frac{\text{DTMB\_CK}}{\text{TBxC0}[15:0]+1} \div 2$$

$$\text{PWM7O Duty Cycle} = 50\%$$

10.3. List and description of TMB1 control register:

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE			E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-				TB1IF				.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
035h	OSCCN1		-			DTMB[1:0]		TMBS		0000 0000	uuuu uuu.	*,*,*,*,*,*,*,-
036h	OSCCN2	-	-	-	-	-	HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*,*,*,*,*,*,*,r
04Eh	TB1Flag	-	-	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu	-,-,r,r r,r,r,r
04Fh	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	-	-	0000 0000	uuuu u0uu	*,*,*,*,*,rw1,*,*
050h	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
051h	TB1RH	TimerB1 counter Register [15:8]				-	-			xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
052h	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
053h	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
054h	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
055h	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
056h	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
057h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
058h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
059h	TCCN0	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 10-3 Registers related to TMB1/2/3

INTE0/INTF0: please refer to the chapter "Interrupt" for more information.

OSCCN0/OSCCN1/OSCCN2: please refer to the chapter "Oscillator, clock source and power consumption management" for more information.

TB1Flag: PWM waveform state flag generated by TMB1 timer

Bit	Name	Description
Bit6~0	PWMxA	PWMx waveform state, 1≦x≦6 <0> Low level L <1> High level H

TB1CN0: TMB1 timer control register

Bit	Name	Description
Bit7	ENTB1	Enable and disable TMB1 <0> OFF. <1> ON
Bit6~5	TB1M[1:0]	Timer TMB1 operation modes <00> 16-bit timer <01> 17-bit timer <10> Two 8-bit timers <11> 8+8-bit timer
Bit4~3	TB1RT[1:0]	Timer TMB1 counting triggering selector <00> Logic High <11> CPI1

Bit	Name	Description
Bit2	TB1CL	TB1R counting reset controller <0> Not reset counting. <1> Reset counting. (Setting <1> can reset the counting; the counting will be set as <0> after the timer is reset.)

TB1CN1: TMB1 timer control register

Bit	Name	Description
Bit7	PA1IV	Pin PWMAx waveform output phase (0≦x≦1)
Bit3	PA0IV	<0> Reversed phase. <1> Co-phase.
Bit6~4	PWMA1[2:0]	Pin PWMAx waveform output selector (0≦x≦1)
Bit2~0	PWMA0[2:0]	

PWMAx[2:0]	Output selector	PWMAx[2:0]	Output selector
000	PWM1O	100	PWM5O
001	PWM2O	101	PWM6O
010	PWM3O	110	PWM7O
011	PWM4O	111	PWM7O

TB1R: TMB1 timer

Bit	Name	Description
Bit15~8	TB1RH[7:0]	TMB1 timer
Bit7~0	TB1RL[7:0]	

TB1C0: TMB1 overflow control

Bit	Name	Description
Bit15~8	TB1C0RH[7:0]	TMB1 timer overflow control
Bit7~0	TB1C0RL[7:0]	

TB1C1: PWMA condition control 1

Bit	Name	Description
Bit15~8	TB1C1RH[7:0]	PWMA condition control 1
Bit7~0	TB1C1RL[7:0]	

TB1C2: PWMA condition control 2

Bit	Name	Description
Bit15~8	TB1C2RH[7:0]	PWMA condition control 2
Bit7~0	TB1C2RL[7:0]	

TCCN0: TMC timer control register

Bit	Name	Description
Bit6~5	TC1S[1:0]	Timer TC1 event input selector <10> LS_CK

11. Power System

The power system PWR has a linear stabilized power source VDDA and an analog circuit common-ground power source ACM, which provides the analog peripheral circuits for the chip and can be used to properly drive the external circuits.

Summary of PWR register:

PWRCN ENBGR, LDOC[2:0], LDOM[1:0], ENLDO

AD1CN5 LDOPL

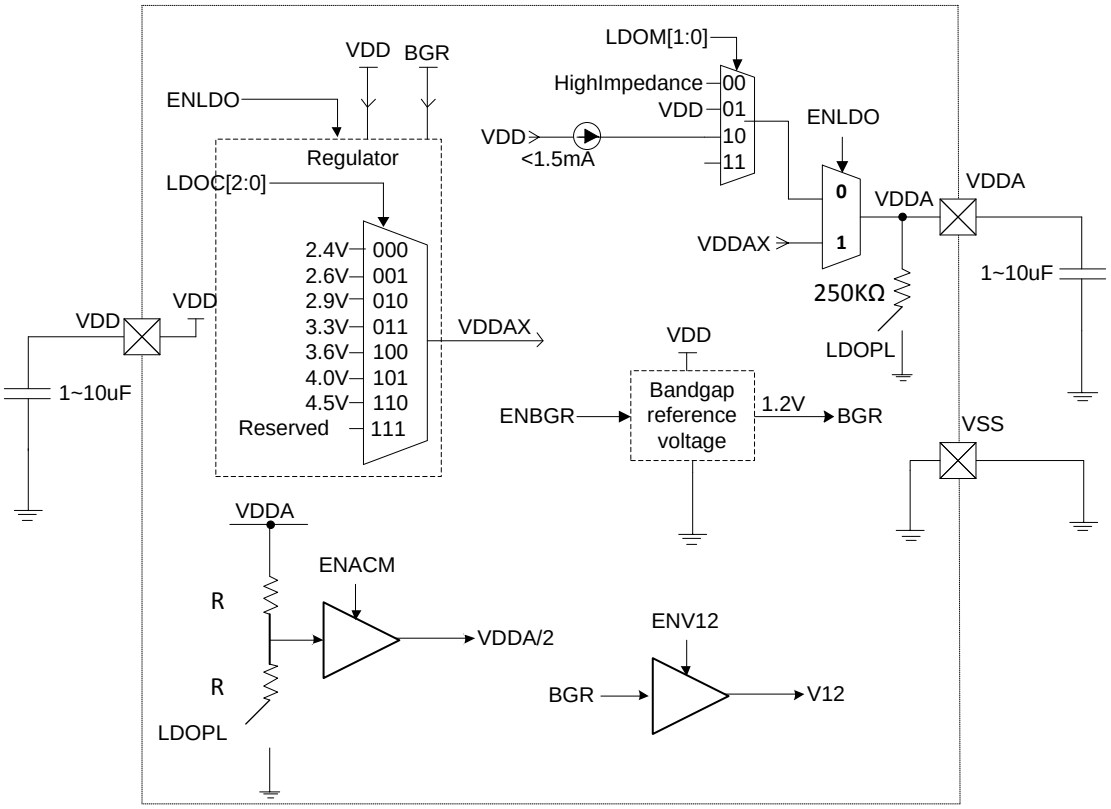


Figure 11-1 Block diagram of power system

11.1. Usage description of VDDA

11.1.1. VDDA initialization configuration:

The voltage stabilization selector LDOC[2:0] can set the voltage outputted by the VDDA pin to be 2.4V~4.5V; there are 7 voltage orders in total. As VDDA is a linear stabilized power source; when using it, it is necessary to notice whether the voltage value of the VDD working voltage is lower than the setting value of the VDDA output voltage so as to avoid that an unpredictable circuit error operation occurs.

11.1.2. External bias voltage used by VDDA:

VDDA can adopt the external voltage input design; when the user wants to provide a voltage source by himself, it is necessary externally input the voltage into the VDDA pin. When using the above method, it is necessary to turn off VDDA; in other words, LDOM [1:0] should be set as 00. Please note that this method may influence the performance of the analog circuit, so the user needs to be careful when using this method.

11.1.3. VDDA enablement

Setting ENLDO[0] as <1> can enable the VDDA voltage stabilizer. It is necessary to avoid that ΣΔADC is under the enablement state when enabling the VDDA voltage stabilizer; besides, ΣΔADC can be enabled only after the VDDA voltage is stable. When it is connected to an external 1uF (10uF) stabilized capacitor, the necessary stabilization time is 500uS(5mS).

11.2. Register description-PWR

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
033h	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]		ENLDO		0000 0000	uuuu u00u	*,*,*,* *,wr0,wr0,*

Table 11-1 PWR register

PWRCN: power system control register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENBGR</td><td>Internal reference voltage controller
<0> OFF
<1> ON; it should be set as '1' before enabling ADC and TPS; then, ADC and TPS can be enabled.
This bit and HAO are linked, so long as HAO is open. Even if this bit writes 0, actually BGR is still open.</td></tr>
<tr><td>Bit6~4</td><td>LDOC[2:0]</td><td>VDDAX output voltage selector
When ENLDO is '1', the set voltage can be outputted to the VDDA pin.
<table>
<tr><th>LDOC[2:0]</th><th>VDDAX output voltage</th><th>LDOC[2:0]</th><th>VDDAX output voltage</th></tr>
<tr><td>000</td><td>2.4V</td><td>100</td><td>3.6V</td></tr>
<tr><td>001</td><td>2.6V</td><td>101</td><td>4.0V</td></tr>
<tr><td>010</td><td>2.9V</td><td>110</td><td>4.5V</td></tr>
<tr><td>011</td><td>3.3V</td><td>111</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Bit3~2</td><td>LDOM[1:0]</td><td>VDDA output selector
When ENLDO is '0, the setting can be outputted to the VDDA pin.
<00> Disable the high input impedance mode.
(when VDDA selects external inputs, but VDD needs to be higher than VDDA at least 0.2V)
<01> Output the VDD voltage.
(When VDD short to VDDA or SOP8 package, the need to use this mode)
<10> Output VDD voltage, 1.5mA current limit
(When the external capacitance of VDDA is greater than that of 1uF or the power supply capability of the system is weak. When the system is running, this mode is selected to charge the external capacitor of VDDA. When the VDDA capacitor reaches a certain potential, it will switch to normal mode)
<11> RSV</td></tr>
<tr><td>Bit1</td><td>ENLDO</td><td>Internal linear voltage stabilizer controller
<0> OFF
<1> ON
Note: when using the SOP8 package for VDD/VDDA to share the Pin configuration, ENLDO please select 0</td></tr>
</table>

AD1CN5: ΣΔADC control register 5

Bit	Name	Description
Bit4	LDOPL	Internal resistance 250k Ω pull down switch. <0> OFF。 <1> ON。 The following points must be made LDOPL 1, otherwise the result will not be as expected 1. Use internal LDO output 2.ADC reference voltage selection VDDA / 2-VSS

12. Analog-to-Digital Converter, ΣΔADC

ΣΔADC is a high-resolution over sampling sigma delta analog-to-digital converter, which can provide 24-bit output. It includes three major parts, a multiple-function input multiplexer, an input buffer, a ΣΔ modulator (ΣΔAD, Sigma Delta Modulator) and a comb filter.

- Multi-function input multiplexer

  The multiplexer can be switched to select different input channels, and one chip can perform various measurements.

  The multiplexer's input channel can be short-circuited to eliminate the zero offset of ADC.

  The multiplexer has a built-in temperature sensing circuit able to output voltage.

- ΣΔ modulator

  The modulator can adjust the magnifying power of the input voltage, and the magnifying power can be selected to be 0.25, 0.5, 1, 2, 4, 8 and16 times.

  The magnifying power of the reference voltage can be selected to be 1 or 1/2.

  The modulator can provide a 4-bit DC input bias voltage setting.

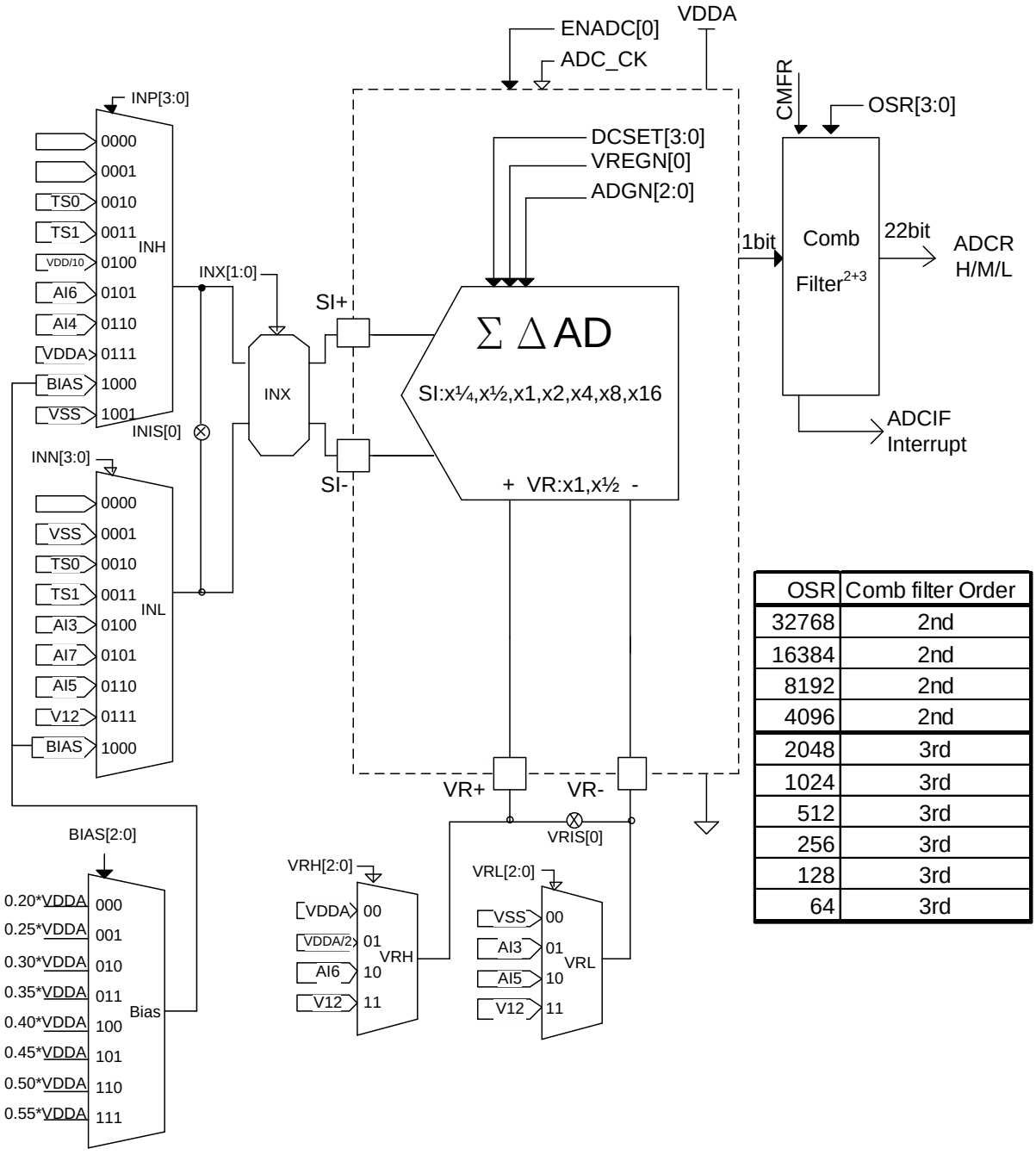
- Comb filter

  The filer can adjust OSR (Over Sampling Ratio)= 64~32768.

  The filter can generate an interrupt event.

Summary of ΣΔADC register:

AD1CN0	ENAD1, OSR[3:0], CMFR
AD1CN1	VREGN, ADGN[2:0]
AD1CN2	BIAS[2:0], DCSET[3:0]
AD1CN3	INP[3:0], INN[3:0]
AD1CN4	VRH[1:0], VRL[1:0], INX[1:0] , VRIS, INIS
AD1CN5	ENACM, ENV12, VCMS,LDOPL, ENBS, ENTPS, TPSCH
AIXM1	APDR1[1:0]
AIXM2	APDR2[1:0]



OSR	Comb filter Order
32768	2nd
16384	2nd
8192	2nd
4096	2nd
2048	3rd
1024	3rd
512	3rd
256	3rd
128	3rd
64	3rd

Figure 12-1 Block diagram of ΣΔADC

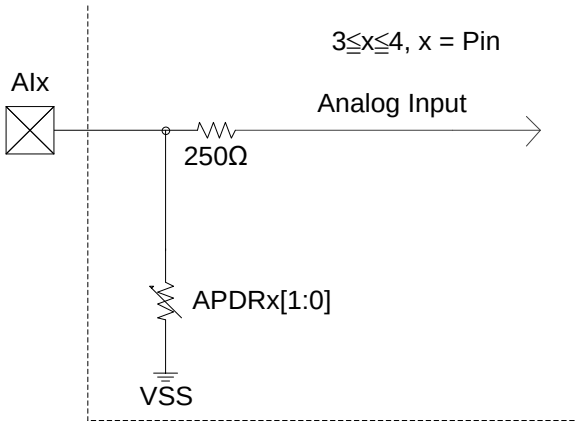


Figure 12-2 Block diagram of analog channel input structure

12.1. Usage description of ΣΔADC

12.1.1. ΣΔADC initialization configuration

12.1.1.1. Configuration method of working frequency

The sampling frequency of ΣΔADC can be provided by DHS_CK after the sampling frequency selector ADCCK[0] configures the working frequency of ΣΔADC, and the highest sampling frequency cannot be higher than 500KHz. The higher sampling frequency can obtain better resolution under the same output speed, but its input impedance will also decrease (please refer to 12.2 Input characteristics of analog channel). When the frequency of DHS_CK is higher than the maximal allowed value, it is necessary to perform the frequency adjustment via the sampling frequency pre-frequency eliminator DADC[1:0].

When using ICE simulation, the frequency source selection of ADC can only be chosen as. CPUS[0]=1b, DHS[1:0]=01b, ADCD[1:0]=00b.

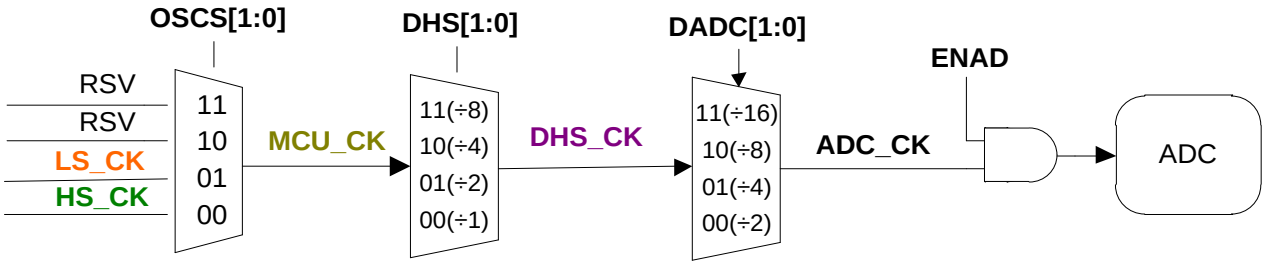


Figure 12-3 Block diagram of working frequency of ΣΔADC

12.1.1.2. Configuration method of multi-function input multiplexer

ΣΔADC adopts 2-order ΣΔ modulator, and the magnifying power and the bias voltage of its signal to be measured and reference voltage can be adjusted by the following configurations.

- When the ΔVR± magnifying power adjuster VREGN[0] is set as <1>, the signal of the reference voltage will be adjusted by 1/2 magnifying power, and will change ΔSI± = (SI+ − SI-) and ΔVR± = (VR+ − VR-) of the input signal; if it is set as <0>, the signal of the reference voltage will be adjusted by 1 magnifying power
- The input signal can be configured by the magnifying adjuster ADGN[2:0] to reach up to 16-time signal magnifying power, as shown in Table 12-2(a).
- If the input signal SI± passes through the DC input bias voltage adjuster DCSET[3:0], the zero position of the input signal can be adjusted to increase the measurement range. The voltage bias method adopts the magnifying power of the weighting reference signal VR±, as shown in Table 12-2(b).
- When measuring signals, please note that the matching problem of the external input signal impedance and the ADC. Please refer to 12.2 "Input characteristics of analog channel" for more information.

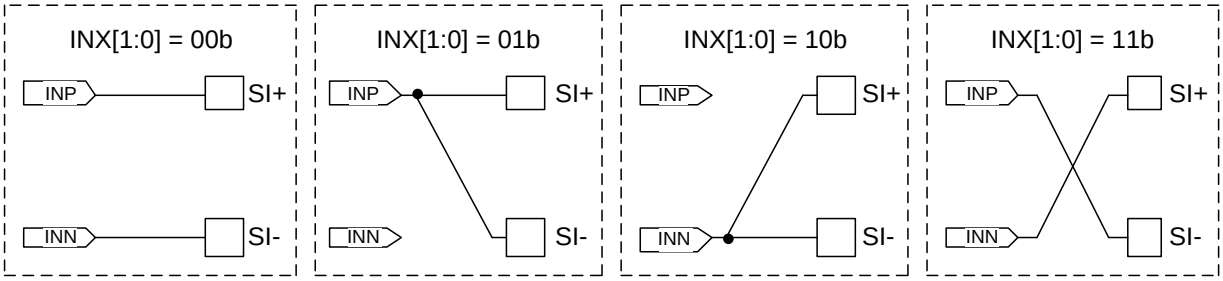


Figure 12-4 Four combination methods of INX input signal transposer

Configuration / Input	ADGN[2:0]							
	000	001	010	011	100	101	110	111
AD Gain	x1/4	x1/2	x1	x2	x4	x8	x16	-

Table 12-2 (a)ADGN[2:0] magnifying configuration table

Configuration / Input	DCSET[3:0]							
	0000	0001	0010	0011	0100	0101	0110	0111
SI±	+0	+1/8 * Vref	+2/8 * Vref	+3/8 * Vref	+4/8 * Vref	+5/8 * Vref	+6/8 * Vref	+7/8 * Vref
Configuration / Input	DCSET[3:0]							
	1000	1001	1010	1011	1100	1101	1110	1111
SI±	-0	-1/8 * Vref	-2/8 * Vref	-3/8 * Vref	-4/8 * Vref	-5/8 * Vref	-6/8 * Vref	-7/8 * Vref

Unit: VR±

Table 12-2(b) Weighting reference voltage magnifying power list of SI± input signal

After the ΣΔ modulator is adjusted by its own magnifying power bias voltage, the calculation equations of its equivalent signal to be measured ΔSI_I and equivalent reference voltage ΔVR_I are respectively as follows:

Equation 12-1

$$\Delta SI\_I = ADGN \times \Delta SI \pm +(DCSET \times \Delta VR \pm)$$

Equation 12 -2

$$\Delta VR\_I = VREGN \times VR \pm$$

It should be note that it is suggested that the equivalent reference voltage ΔVR_I is 0.8V~1.2V and the equivalent reference voltage ΔVR_I is ±0.9 xΔVR_I in order to achieve higher resolution and linearity of the output of the ΣΔ modulator.

12.1.1.3. Configuration method of comb filter

The ΣΔ modulator output 1-bit data to the 2-order comb filter, and then converted into 24-bit value by the comb filer to be saved in the AD1O[23:0] register. The updating speed of the data of AD1O[23:0] is just the output speed of ΣΔADC; the calculation method is the ratio of the sampling frequency of ΣΔADC to the output speed frequency of ΣΔADC; the output speed frequency of ΣΔADC is also called OSR (Over Sampling Ratio).

Thus, the output speed of ΣΔADC is ADC_CK÷OSR and the OSR value can be configured by OSR[3:0] to generate different output turnover frequencies of ΣΔADC, as shown in Table 12-2(c).

Configuration / ADC_CK	OSR[3:0]										
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
	32768	32768	16384	8192	4096	2048	1024	512	256	128	64
500k	15	15	30	61	122	244	488	976	1953	3906	7812
250K	7	7	15	30	61	122	244	488	976	1953	3906
125K	3	3	7	15	30	61	122	244	488	976	1953

Table 12-2(c) Simple table of over sampling frequency configuration of ΣΔADC

AD1O[23:0] is composed of AD1H[7:0], AD1M[7:0] and AD1L[7:0] (AD1O[2]= AD1O[1]=AD1O[0]), which is used to store the 24-bit data output by the Comb Filter, but bit0 and bit1 are always fixed to zero.The data format composition of the comb filter is as shown in Table 12-3.

+FSR/-FSR: maximal measurement range of positive phase and negative phase

	Equivalent signal to be measured	AD1O[23:0]	
		Hexadecimal system	Binary system
Two-polarity output 2's complement format	ΔVR_I	7FFFFC	0111 1111-1111 1111-1111 1100
	$\Delta VR\_I \times \frac{1}{2^{21}}$	000004	0000 0000-0000-0000-000 0100
	0	000000	0000 0000-0000 0000-0000 0000
	$-\Delta VR\_I \times \frac{1}{2^{21}}$	FFFFFC	1111 1111-1111 1111-1111 1100
	$-\Delta VR\_I$	800000	1000 0000-0000 0000-0000-0000

Table 12-3 Relation table of AD1O[23:0] and input signals

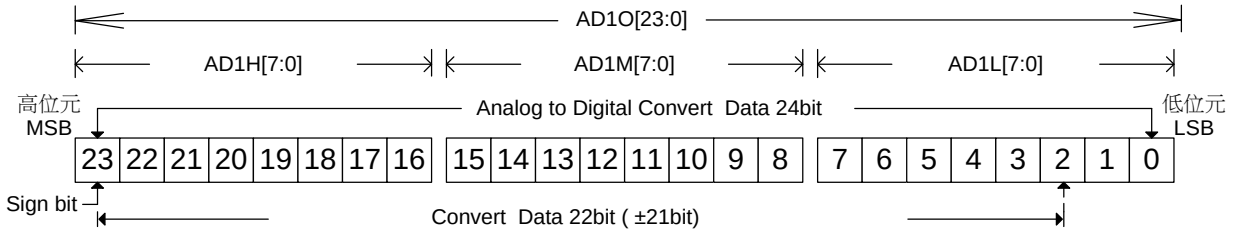


Figure 12-5 Schematic view of resolution of AD1O[23:0]

12.2. Input characteristics of analog channel

ΣΔADC uses the switched capacitance circuit to process analog signals; so as to make sure that the voltage of the sampling capacitor can obtain the correct value when the input buffer is not in use, the maximal output impedance of the input signal should be limited, and have a mutual-limited relation with the sampling frequency and the signal magnifying power selection of ΣΔADC.

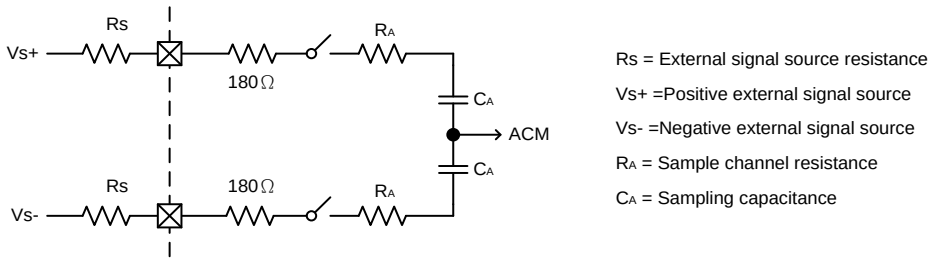


Figure 12-6 AIx input capacitors and impedance modules

As shown in figure 12-6, when the input signal is directly inputted without passing through the buffer, it is necessary to consider the effects of the input signal internal resistance Rs, the sampling frequency ADC_CK of ΣΔADC, parasitic resistances R_A and capacitances C_A. The relevant calculation methods are as follows:

Equation 12-3

$$t_s > (R_s + R_A + 180\Omega) \times C_A \times [\ln(2^{ENOB} \times Gain) + 2]$$

t_s: minimal sampling time of ΣΔADC

ENOB: desired effective numbers of ΣΔADC

Gain：(ΣΔAD Gain)

Equation 12 -4

$$F_s = \frac{1}{2 \times t_s}$$

Fs: shortest sampling frequency of ΣΔADC

As ΣΔADC includes PGA and ΣΔAD, the two parts are designed to have their own R_A and C_A value respectively; besides, the calculation of the minimal sampling time t_s is based on the parts directly matching the input signal.

ΣΔAD Gain	C_A	R_A
x1/4		
x1/2		
x1	0.5pF	10k Ω
x2	1pF	10k Ω
x4	2pF	10k Ω
x8	4pF	5k Ω
X16		

Table 12-4(a) Relation table of ΣΔADC Gain, and R_A and C_A

VR Gain	C_A	R_A
x1/2	0.25pF	10k Ω
X1	0.5pF	10 kΩ

Table 12-4(b) Relation table of VR Gain, and R_A and C_A

The major application of ΣΔADC is to measure the low-frequency signals; however, in the real world, the signals to be measured may include a lot of high-frequency noise; according to the signal sampling principle, after being sampled, the high-frequency noise exceeding the sampling frequency will generate zero shift and low-frequency noise, and then result in measurement errors. Accordingly, we suggest that differential signal to be measured and the reference voltage of the chip are added by a 10nF~100nF filtering capacitor to better the measurement accuracy.

In the ADC input channel configuration, in order to get better efficiency. The choice of differential signal (it is recommended to use the IC's internal signal source or input by the external AIX) configuration.

12.3. Absolute temperature sensor, TPS

The absolute temperature sensor is composed of bipolar junction transistors (BJT); the change between its voltage signal and its temperature is a curve passing through 0K, and it has the following characteristics:

- When the environmental temperature is 0K, the voltage value outputted by the temperature sensor is $V_{TPS@0K}$ =0V.
- Via the measurement method, the offset voltage ($V_{ADC\text{-}OFFSET}$) of the analog-to-digital converter ADC can be cancelled by the asymmetry ($I_{S1} \neq I_{S2}$) of BJT.
- The temperature calibration can be performed just by single-point calibration.

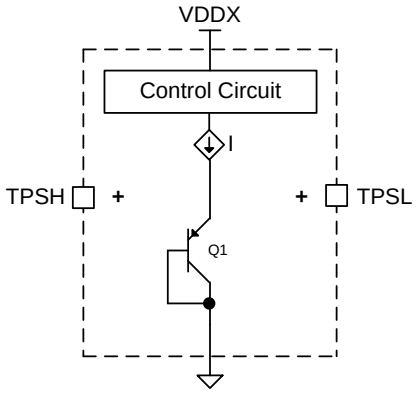


Figure 12-7 Block diagram of application of absolute temperature sensor

12.3.1. Initialization configuration and calculation method of TPS

- TPS is enabled, with the exception of ENAD1 set to 1b, ENTPS must also be set to 1b.
- ADC input signal selector settings INP = 0010b, INN = 0010b, TPSCH = 0b measurement signal ADC_{TPS0}

- ADC input signal selector settings INP = 0011b, INN = 0011b, TPSCH = 1b measurement signal ADC_{TPS1}
- At the same temperature T_A (° C), ΣΔ ADC measures the ADC_{TPS0} and ADC_{TPS1} values, adds the two numbers and averages them to get the $ADC_{TPS\ @\ TA}$ value corresponding to the measured TPS at temperature T_A.
- The output value TPS of V_{TPS} is a linear curve to the temperature change, the gain G_{TPS} can be derived (or called slope).

Equation 12 -5 TPS gain equation

$$G_{TPS} = \frac{ADC_{TPS@T_A}}{(273.15 + T_{offset} + T_A)K}$$

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

$ADC_{TPS@Ta}$: ADC values measured at calibration temperature

K : °C + 273.15

T_{offset} : Temperature offset

- TPS in the temperature conversion is not ideal, so in fact not at ˚C = K-273.15
  Instead ˚C = K + KT = K + (- 273.15-Toffset)
  For the KT values, refer to the TPS specification in the IC Data sheet ADC section.
  HY15P41 KT value of -284, ˚C = K-284 -> K=˚C+284

12.3.2. Example description of TPS

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 INP=0010b=TS0, INN=0010b=TS0, AD1CN5[TPSCH]=0b and AD1CN5[ENTPS]=1b, which allows ADC to measure a digital code ADC_{TPS0}=5897634.

(2) Set INP=0011b=TS1, INN=0011b=TS1, AD1CN5[TPSCH]=1b and AD1CN5[ENTPS]=1b, which allows ADC to measure a digital code ADC_{TPS1}=5827679.

(3) Calculate ADCTPS@25=(ADCTPS0 +ADCTPS1)/2=5862656. This action eliminates the Offset of the Temperature Sensor.

(4) Calculate G_{TPS}

$$G_{TPS} = \frac{ADC_{TPS@T_A}}{(273.15 + T_{offset} + T_A)K} = \frac{5862656}{(284+25)K} = 18973$$

(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

$$T_X = \frac{ADC_{TPS@65}}{G_{TPS}} - \left[273.15 + T_{offset}\right] = \frac{6630103}{18973} - 284 = 65.45^{o}C$$

12.4. Register description-ΣΔADC

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	-	ADIE	WDTIE	TB1IE	-	-	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
026h	INTF0	-	-	ADIF	WDTIF	TB1IF	-	-	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
033h	PWRCN	ENBGR	LDOC[2:0]			LDOM[0]	LDOM	ENLDO	CSFON	0000 0000	uuuu u00u	*,*,*,* *,wr0,wr0,*
03Ah	AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu	-,-,*,*,*,*,*,*
03Bh	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
03Ch	AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
03Dh	AD1CN0	ENAD1	-	-	OSR[3:0]				CMFR	000. 0000	uuu. uuuu	*,*,*,*,*,*,*,*
03Eh	AD1CN1	-	-	VREGN			ADGN[2:0]			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03Fh	AD1CN2	-	BIAS[2:0]			DCSET[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
040h	AD1CN3	INP[3:0]				INN[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
041h	AD1CN4	VRH[1:0]		VRL[1:0]		INX[1:0]		VRIS	INIS	0010 0000	uuuu uuuu	*,*,*,*,*,*,*,*
042h	AD1CN5	ENACM	ENV12	VCMS	LDOPL	ENBS	-	ENTPS	TPSCH	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
047h	AIXM1	APDR3[1:0]		-	-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
048h	AIXM2	-	-	-	-	-	-	APDR5[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 12-5 ΣΔADC register

INTE0/INTF0: please refer to the chapter "Interrupt" for more information.

PWRCN: please refer to the chapter "Power System" for more information.

AD1O[23:0] Analog-to-digital converting register: please refer to the chapter "Configuration Method of Comb Filter" for more information.

AD1H[7:0] AD1 Analog-to-digital converting high-byte data register

AD1M[7:0] AD1 Analog-to-digital converting data register

AD1L[7:0] AD1 Analog-to-digital converting low-byte data register

AD1CN0: ΣΔADC register 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENAD1</td><td>ΣΔADC enablement controller
<0>OFF
<1>ON</td></tr>
<tr><td>Bit4~1</td><td>OSR<3:0></td><td>ΣΔADC over-sampling rate frequency eliminator
<table>
<tr><th>OSR<3:0></th><th>OSR</th><th>OSR<3:0></th><th>OSR</th></tr>
<tr><td>0000</td><td>32768</td><td>1000</td><td>256</td></tr>
<tr><td>0001</td><td>32768</td><td>1001</td><td>128</td></tr>
<tr><td>0010</td><td>16384</td><td>1010</td><td>64</td></tr>
<tr><td>0011</td><td>8192</td><td>1011</td><td>32768</td></tr>
<tr><td>0100</td><td>4096</td><td>1100</td><td>32768</td></tr>
<tr><td>0101</td><td>2048</td><td>1101</td><td>32768</td></tr>
<tr><td>0110</td><td>1024</td><td>1110</td><td>32768</td></tr>
<tr><td>0111</td><td>512</td><td>1111</td><td>32768</td></tr>
</table></td></tr>
<tr><td>Bit0</td><td>CMFR</td><td>Σ Δ ADC and comb filter reset controller
<0> Not reset</td></tr>
</table>

Bit	Name	Description
		<1> Reset; A reset occurs when the write action occurs

AD1CN1: ΣΔADC control register 1

Bit	Name	Description
Bit5	VREGN	VR± magnifying power adjuster <0> x1 <1> x1/2
Bit2~0	ADGN[2:0]	AD magnifying adjuster

ADGN[2:0]	Gain	ADGN[2:0]	Gain
000	x1/4	100	x4
001	x1/2	101	x8
010	x1	110	x16
011	x2	111	RSVD

AD1CN2: ΣΔADC control register 2

Bit	Name	Description
Bit6~4	BIAS[2:0]	Bias voltage value setting

BIAS [2:0]	Voltage value (V)	BIAS [2:0]	Voltage value (V)
000	VDDA*0.20	100	VDDA*0.40
001	VDDA*0.25	101	VDDA*0.45
010	VDDA*0.30	110	VDDA*0.50
011	VDDA*0.35	111	VDDA*0.55

Bit3~0 DCSET[3:0] SI± bias voltage adjuster

DCSET<3:0>	Offset	DCSET<3:0>	Offset
0000	+0*(REFP – REFN)	1000	-0*(REFP – REFN)
0001	+0*(REFP – REFN)	1001	-0*(REFP – REFN)
0010	+2/8*(REFP – REFN)	1010	-2/8*(REFP – REFN)
0011	+2/8*(REFP – REFN)	1011	-2/8*(REFP – REFN)
0100	+4/8*(REFP – REFN)	1100	-4/8*(REFP – REFN)
0101	+4/8*(REFP – REFN)	1101	-4/8*(REFP – REFN)
0110	+6/8*(REFP – REFN)	1110	-6/8*(REFP – REFN)
0111	+6/8*(REFP – REFN)	1111	-6/8*(REFP – REFN)

AD1CN3: ΣΔADC control register 3

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~4</td><td>INP[3:0]</td><td>SI±“+” input signal selector
<table>
<tr><th>INP[3:0]</th><th>ADC input channel</th><th>INP[3:0]</th><th>ADC input channel</th></tr>
<tr><td>0000</td><td>-</td><td>1000</td><td>BIAS</td></tr>
<tr><td>0001</td><td>-</td><td>1001</td><td>VSS</td></tr>
<tr><td>0010</td><td>TS0</td><td>1010</td><td>-</td></tr>
<tr><td>0011</td><td>TS1</td><td>1011</td><td>-</td></tr>
<tr><td>0100</td><td>VDD/10</td><td>1100</td><td>-</td></tr>
<tr><td>0101</td><td>AI6</td><td>1101</td><td>-</td></tr>
<tr><td>0110</td><td>AI4</td><td>1110</td><td>-</td></tr>
<tr><td>0111</td><td>VDDA</td><td>1111</td><td>-</td></tr>
</table></td></tr>
<tr><td>Bit3~0</td><td>INN[3:0]</td><td>SI±“-” input signal selector
<table>
<tr><th>INN[3:0]</th><th>ADC input channel</th><th>INN[3:0]</th><th>ADC input channel</th></tr>
<tr><td>0000</td><td>-</td><td>1000</td><td>BIAS</td></tr>
<tr><td>0001</td><td>VSS</td><td>1001</td><td>-</td></tr>
<tr><td>0010</td><td>TS0</td><td>1010</td><td>-</td></tr>
<tr><td>0011</td><td>TS1</td><td>1011</td><td>-</td></tr>
<tr><td>0100</td><td>AI3</td><td>1100</td><td>-</td></tr>
<tr><td>0101</td><td>AI7</td><td>1101</td><td>-</td></tr>
<tr><td>0110</td><td>AI5</td><td>1110</td><td>-</td></tr>
<tr><td>0111</td><td>V12</td><td>1111</td><td>-</td></tr>
</table></td></tr>
</table>

AD1CN4: ΣΔADC control register 4

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~6</td><td>VRH[1:0]</td><td>VR±“+” voltage signal selector
<table>
<tr><th>VRH[1:0]</th><th>ADC input channel</th></tr>
<tr><td>00</td><td>VDDA</td></tr>
<tr><td>01</td><td>VDDA/2</td></tr>
<tr><td>10</td><td>AI6</td></tr>
<tr><td>11</td><td>V12</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit5~4</td><td>VRL[1:0]</td><td>VR±"-" voltage signal selector
<table>
<tr><th>VRL[1:0]</th><th>ADC input channel</th></tr>
<tr><td>00</td><td>VSS</td></tr>
<tr><td>01</td><td>AI3</td></tr>
<tr><td>10</td><td>AI5</td></tr>
<tr><td>11</td><td>V12</td></tr>
</table></td></tr>
<tr><td>Bit3~2</td><td>INX</td><td>SI± input signal transposer
<11> INP→ADL,INN→ADH
<10> INN→ADH & ADL,INP floating connection
<01> INP floating connection, INP→ADH & ADH
<00> INP→ADH,INN→ADL</td></tr>
<tr><td>Bit1</td><td>VRIS[0]</td><td>VR+/VR- short-circuited controller
<0> Not short-circuited
<1> Short-circuited</td></tr>
<tr><td>Bit0</td><td>INIS</td><td>SI± input signal short-circuit controller
<0> Not short-circuited
<1> Short-circuited</td></tr>
</table>

AD1CN5: ΣΔADC control register 5

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENACM</td><td>ADC Common Mode Voltage
<0> OFF.
<1> ON.</td></tr>
<tr><td>Bit6</td><td>ENV12</td><td>Voltage Source 1.2V (Buffer Enable)
<0> OFF.
<1> ON.</td></tr>
<tr><td>Bit5</td><td>VCMS</td><td>ADC Common Voltage.
<0> VDDA/2。
<1> 1.2V。
The choice of VCMS has a great effect on the performance of the ADC measurement, and the ideal ACM voltage is equal to the ADC reference voltage.

Example : The ADC reference voltage input is selected as AI6-AI5
When VDDA = 5V AI6-AI5 and other schools to 1.2V VCMS please choose 1
When VDDA = 5V AI6-AI5 and other schools for the 3V VCMS please use 0</td></tr>
<tr><td>Bit4</td><td>LDOPL</td><td>Internal resistance 250k Ω pull down switch.</td></tr>
</table>

Bit	Name	Description
		<0> OFF。 <1> ON。 The following points must be made LDOPL 1, otherwise the result will not be as expected 1. Use internal LDO output 2.ADC reference voltage selection VDDA / 2-VSS
Bit3	ENBS	ADC Input Channel, Bias Voltage Enable Switch. <0> OFF。 <1> ON。
Bit1	ENTPS	Internal TPS enablement control <0> OFF <1> ON; it needs to set a corresponding ADC network.
Bit0	TPSCH	TPS output voltage reverse control <0> Normal <1> Reverse

AIXM1: AIx Port pull-down resistor selection register

Bit	Name	Description
Bit7~6	APDR3[1:0]	AI3 pin pull-down resistor selector

APDR3[1:0]	Function
00	OFF(Default)
01	10K Ω
10	50K Ω
11	100K Ω

AIXM2: AIx Port pull-down resistor selection register

Bit	Name	Description
Bit1~0	APDR4[1:0]	AI4 pin pull-down resistor selector

APDR4[1:0]	Function
00	OFF(Default)
01	10K Ω
10	50K Ω
11	100K Ω

13. Inter-Integrated Circuit Serial interface

The I^2C communication interface includes two operation modes, master mode and slave mode; the master mode can use the transmission controller (Tx Controller) to transmit the signals with the I^2C package format to the I^2C Bus, and use the clock generator to determine the desired transmission speed. On the other hand, the slave controller can receive the signals from the I^2C Bus, and then receive the communication requests of the host machine on the bus by the slave mode, and then use the transmission controller to return the data needed by the host machine; in addition, the data receiving circuit included by the slave controller is also a channel for the master controller to receive the returned data.

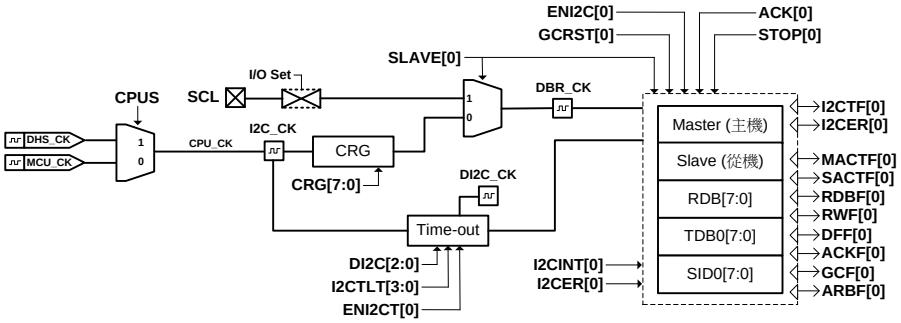


Figure 13-1 I^2C system structure diagram

- Function characteristics of I^2C serial interface:
  - The standard I^2C serial interface includes 2 pins, which are the serial data (SDA) and the serial clock (SCL).
  - The pin is the output structure of the open drain, which needs external resistors to achieve high-level output.
  - The standard I^2C serial interface can be configured to be the master, the slave or the master/slave mode.
  - Programmable clock, which allows adjusting the transmission speed of I^2C.
  - The data transmission between the master and the slave is dual-direction.
  - I^2C allows quite high working voltage range.
  - The reference design of I^2C uses a 7-bit address space, but keeps 16 addresses, so a set of buses can have at most 112 nodes for communication.

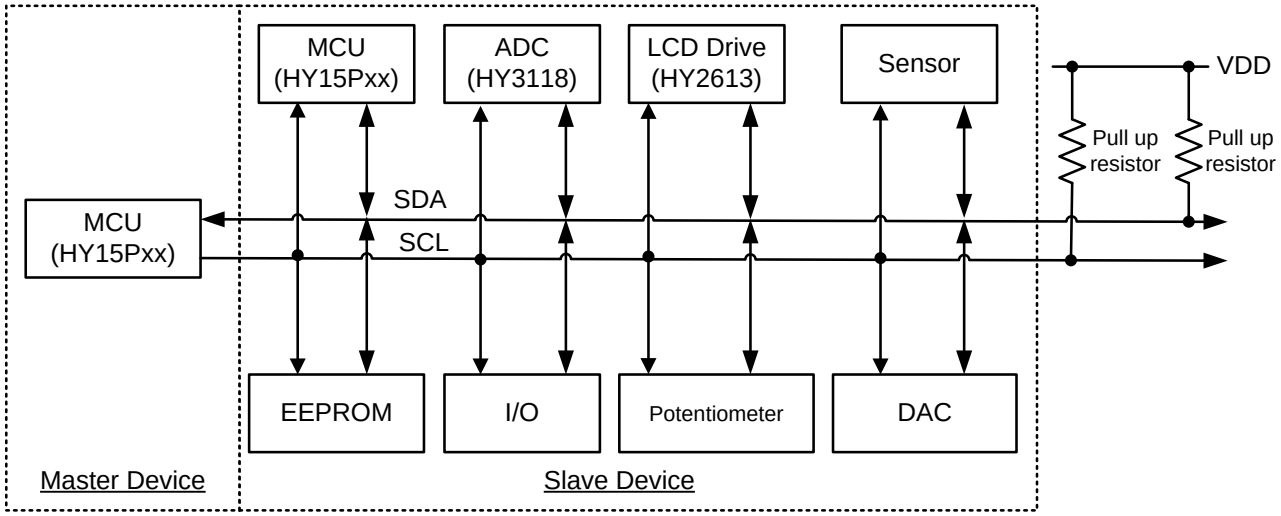


Figure 13-2 Schematic view of I²C communication connection

- I²C serial interface signal:
  - Start signal (START): when SCL of the host machine is at high level, SDA is transmitted and changed from high level to low level, and the data transmission ends.
  - Data signal (DATA) or address (ADDRESS) signal: the I²C serial interface protocol requires that the data in the SDA can be changed only after SCL is at low level.
  - Acknowledge signal (Acknowledge): after the device (slave) receiving data receives the 8th bit, the device transmits a low-level signal to the device (master) transmitting data to show that the data have been received.
  - Stop signal (STOP): when SCL of the host machine is at high level, SDA is transmitted and changed from low level to high level, and the data transmission ends.

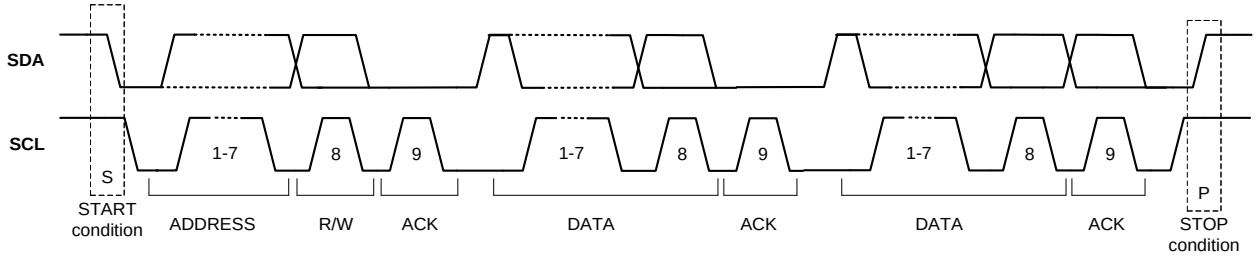


Figure 13-3 Timing diagram of I²C bus

13.1. Data transmission speed calculation

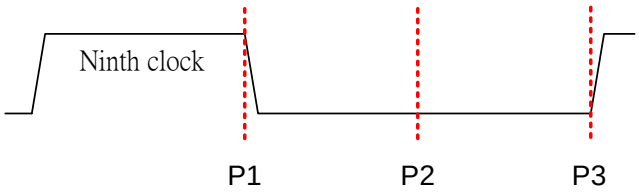
- Master Mode

  The internal register CRG[7:0] of I^{2}C can control the data transmission speed of the master mode; the value of CRG[7:0] generates the pin signal of SCL of the host machine via the internal timer, so the data transmission speed can be calculated by the following equation according to the frequency of the clock source I2C_CK of I^{2}C:

$$\text{Data Baud Rate(Hz)}=\frac{\text{I2C\_CK}}{[4\times(CRG[7:0]+1)]}$$

- Slave Mode

  When the Master end uses a standard hardware I2C, or a device that judges the SCL state to action, the CRG[7:0] recommends a value of 01H. If the master is a device that uses I / O emulation, the ninth clock length may be adjusted according to the actual situation. Formula calculation b is as follows:



- P1~P2 is determined by the ISR time, the ISR time determined by the I2CINT=0
- the time of P2~P3 is determined by CRG[7:0]
  Min=(CRG+1)*(CPU_CK Cycle)
  Max=2*(CRG+1)*(CPU_CK Cycle)

13.2. Time-out function

- The time-out control is to avoid that the I^{2}C controller locks the I^{2}C communication bus; so as to provide enough time for MCU to deal with the requirements of the I^{2}C controller during the operation process of I^{2}C, the I^{2}C controller will pull SCL to be low after each response bit to make the master not transmit the next clock signal; in other words, the clock stretching occurs. However, the MCU is too busy or fails to reply to the request of the I^{2}C controller for any reason, SCL of the I^{2}C communication bus may be locked to be at low.
- For the purpose of preventing from the above situations, the time-out controller allows the user to use the working frequency eliminator DI2C[2:0] and the time condition controller I2CTLT[3:0] to determine the time-out condition of the SCL being at low level. The condition handling includes the following status:
- When detecting the time of SCL being pulled to low level by the host machine, the I^{2}C controller will force SCL to be released after the condition is satisfied and then transmit an interrupt event to CPU.
- When SCL is released to be high because failing to reach time-out time, the internal timer of the time-out controller will be reset, and then re-count after SCL is pulled to be low again.

13.3. Communication flow diagram of I2C serial interface

- Terms of I^2C serial interface
  - (SPIA): stands for the instruction transmitting to the ACT control register, where S is the "Start" instruction, P is the "Stop" instruction, I is the interrupt flag, and A is the "Acknowledge" instruction.
  - SPIA: stands for the read value of the Action control register, which can be used to determine whether the interrupt flag or other instructions are finished.
  - STA: read the value of the Status register (STA), which can show the current operation state of the I^2C circuit.

  The following flow chart will respectively use the "gray-background circular frame", the "white-background circular frame" and the "rectangular frame" shown in figure 16-4 to show the status of the I^2C interface:

  Gray-background circular frame: stands for the I^2C status with the interrupt flag.

  White-background circular frame: stands for the I^2C status without the interrupt flag and needed by to actively read by MCU.

  Rectangular frame: means that MCU should set the instruction to I^2C.

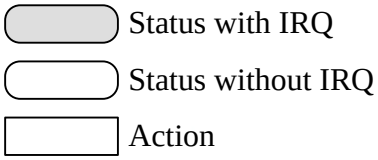


Figure 13-4 Symbols of flow chart

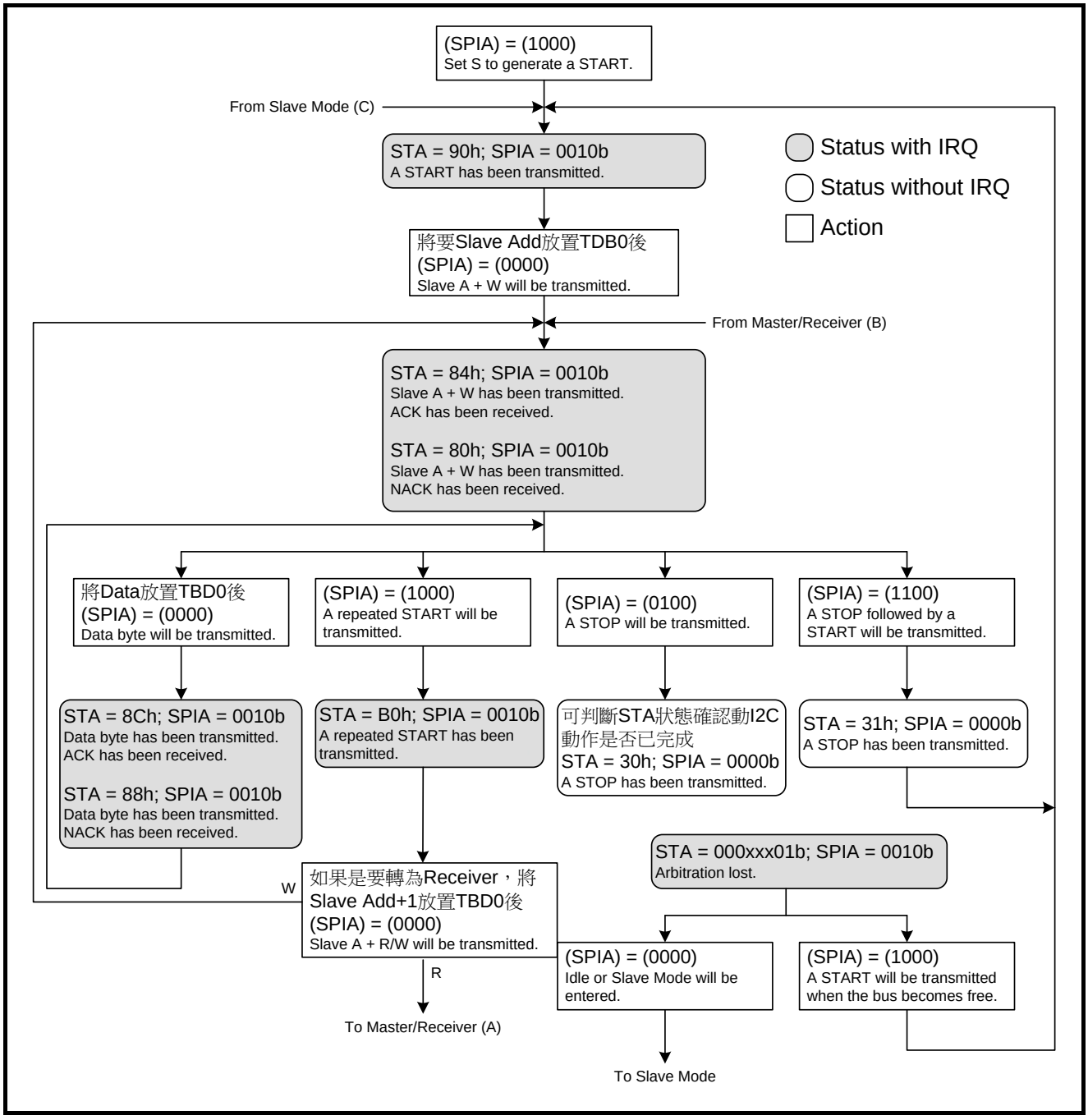


Figure 13-5 Master Transmitter Mode

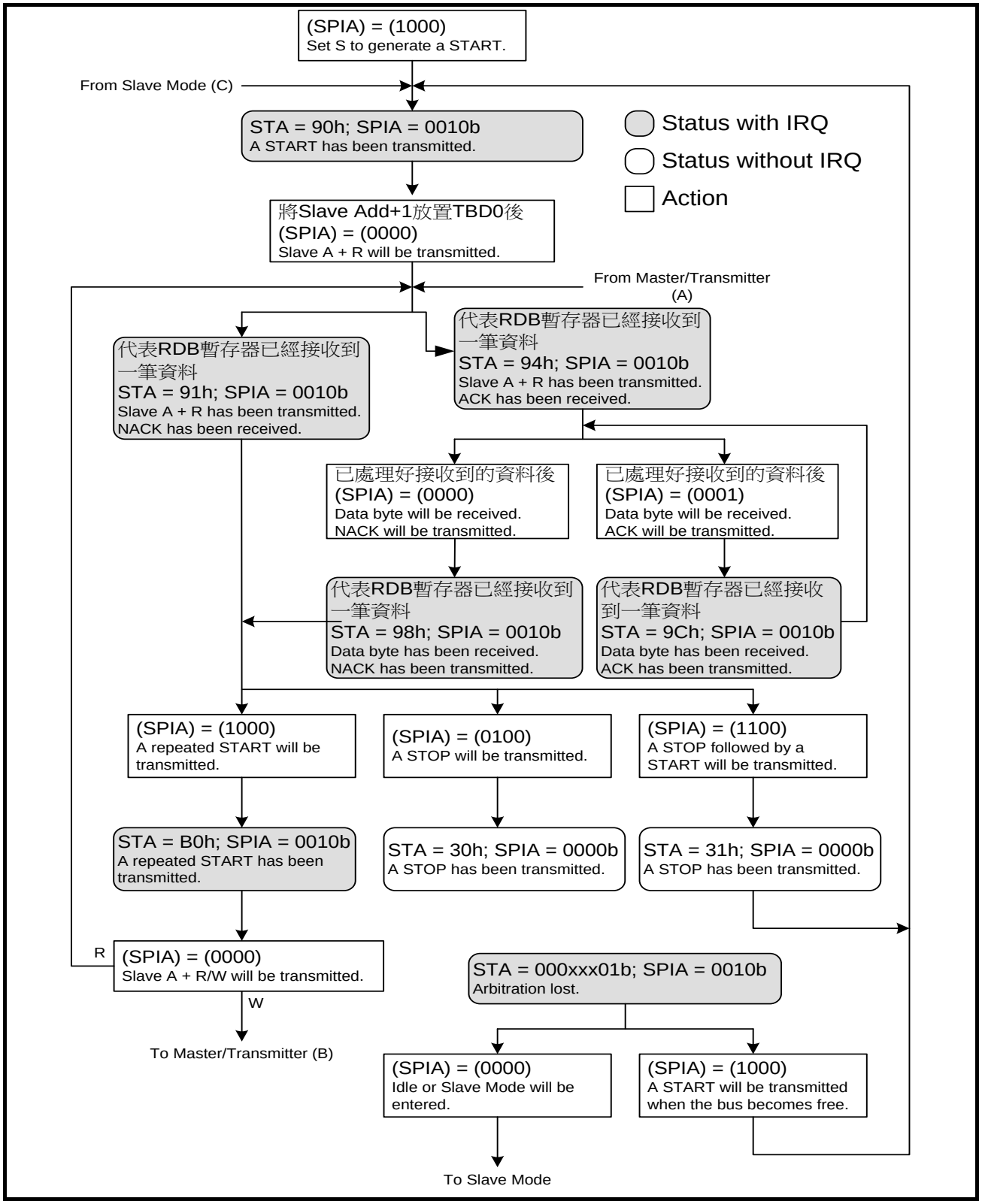


Figure 13-6 Master Receiver Mode

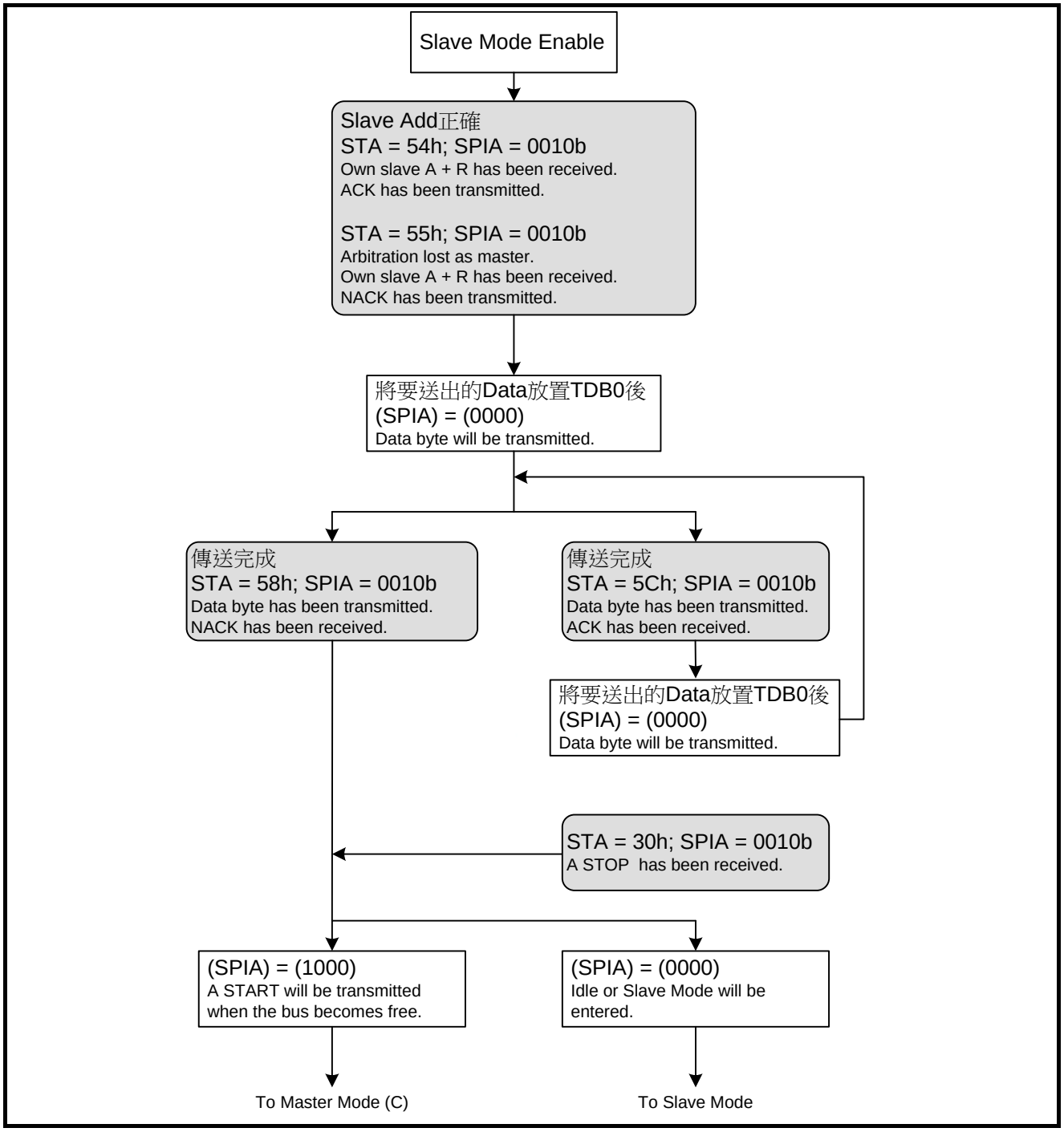


Figure 13 -7 Slave Transmitter Mode

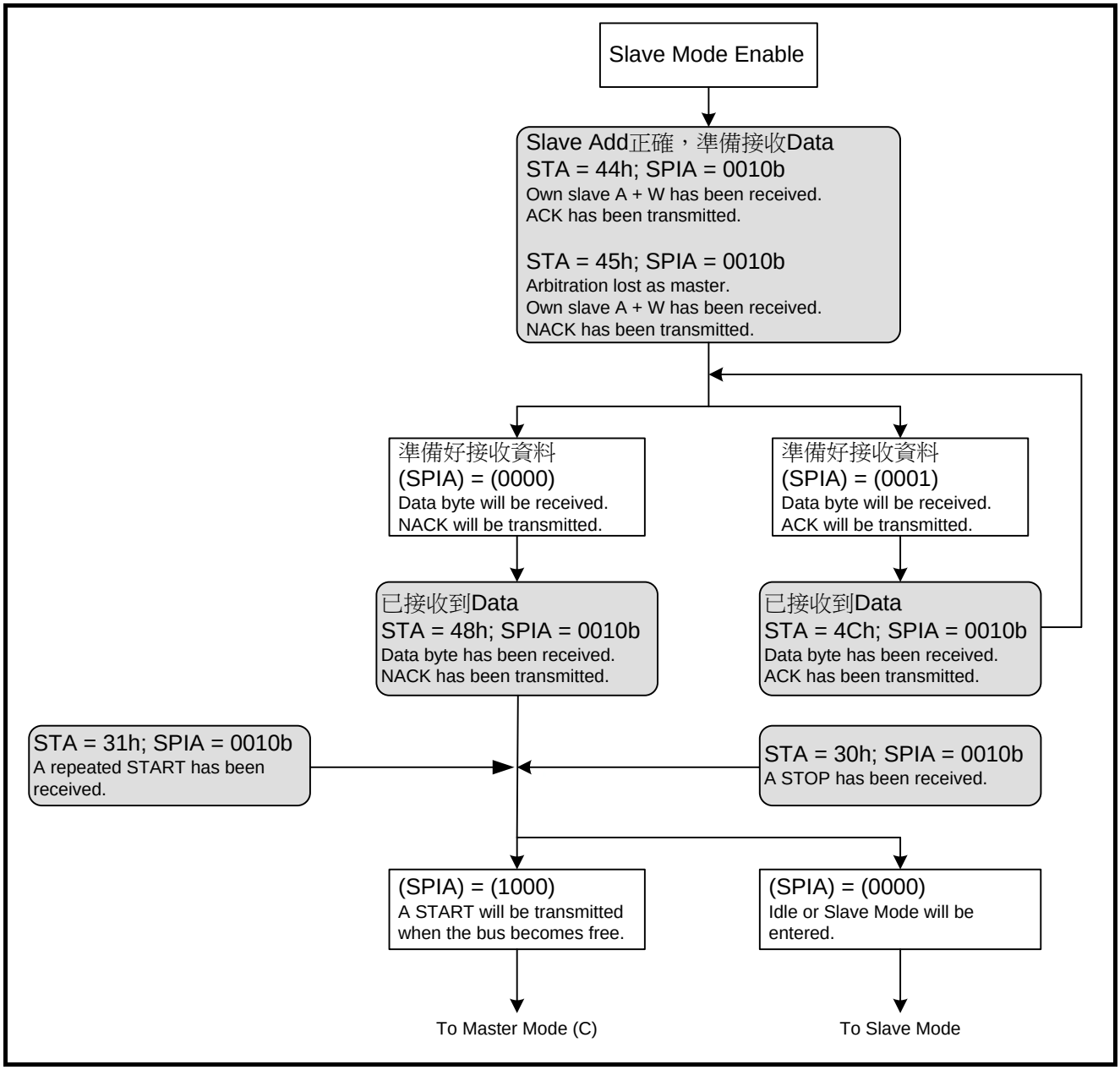


Figure 13-8 Slave Receiver Mode

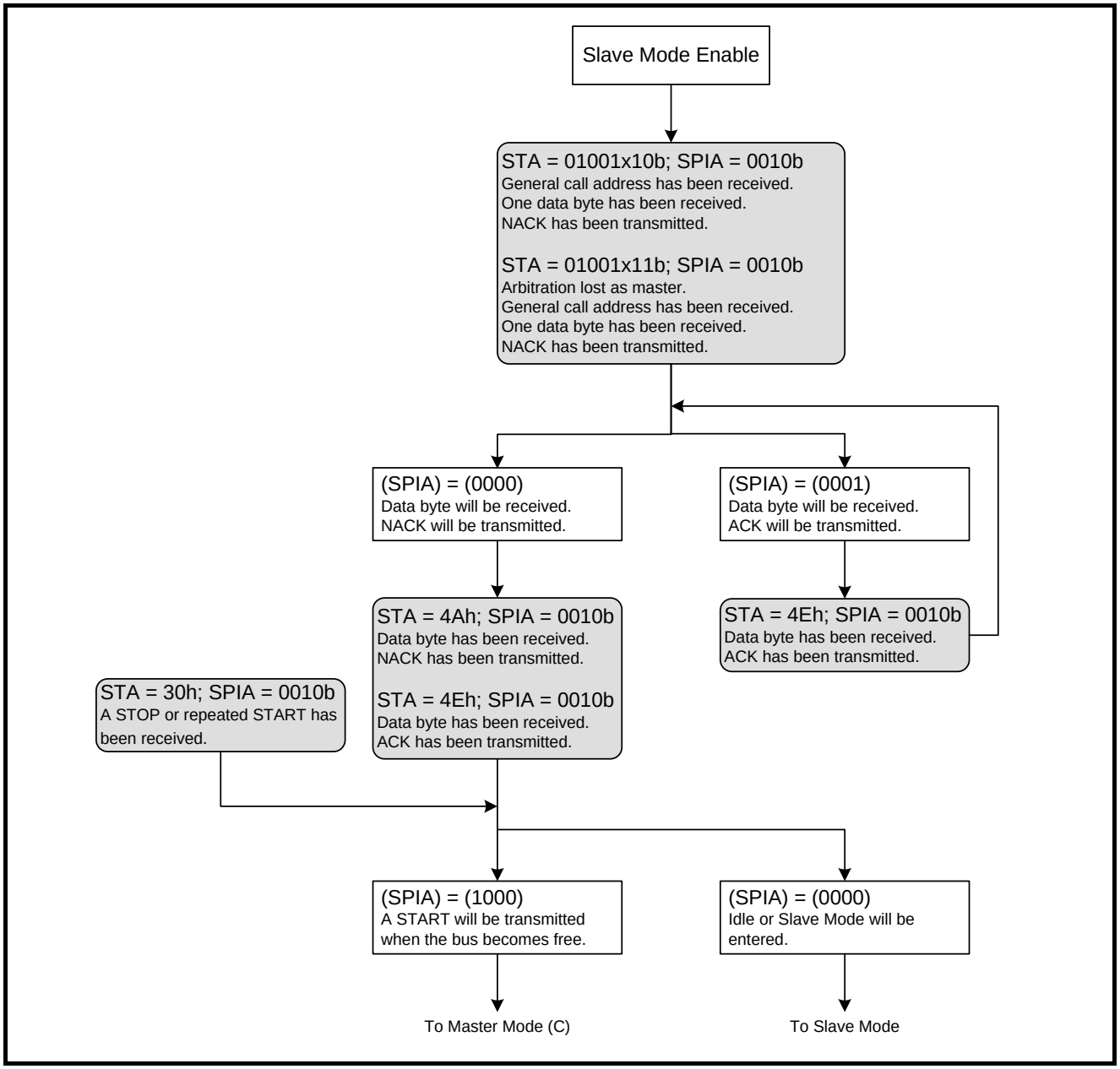


Figure 13-9 General Call Mode

13.4. Description of I2C register

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE			E0IE	0000 0000	0uuu uuuu	*,*,*,* *,*,*,*
024h	INTE1	TA1IE		TXIE	RCIE	I2CERIE	I2CIE			0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
027h	INTF1	-	-	TXIF	RCIF	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
061h	CFG	Rsv.					GCRst	ENI2CT	ENI2C	.... .000	.... .uuu	-,-,-,- -,*,*,*
062h	ACT	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
063h	STA	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*,*,*,* *,*,*,*
064h	CRG	CRG[7:0]								0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
065h	TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
066h	RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
067h	TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
068h	SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu	*,*,*,* *,*,*,*

Table 13-1 I^{2}C register

INTE0/INTE1/INTF1: please refer to the chapter "Interrupt" for more information.

TRISC4: PT4 pin characteristic control register: please refer to the chapter "Input/output port, I/O" for more information.

PT4M2: PT4 digital output mode selection register: please refer to the chapter "Input/output port, I/O" for more information.

CFG：I^{2}C configuration register

Bit	Name	Description
Bit2	GCRST	I^{2}C general call reset enablement control <0>Disable <1>Enable ※ After the I^{2}C slave mode and the GCRST function are enabled at the same time, if the I^{2}C controller receives the general call ID 00h and the first piece of data is "06h", the "General Call Reset" condition holds; currently, the interrupt signal going to be originally transmitted to the processor of the host machine will be replaced by the reset signal, which provides the function that an external host machine can reset the function of the chip of the host machine via the I^{2}C bus.
Bit1	ENI2CT	Enable I^{2}C time-out monitoring function bit <0>Disable <1>Enable the I^{2}C Time-out monitoring function
Bit0	ENI2C	Enable I^{2}C function control bit <0>Disable <1>Enable the I^{2}C communication interface ※ Notice: When ENI2C is disabled, the internal clock of I^{2}C will be disabled; only the configuration register can perform the write operation, but other registers cannot be written by data.

ACT: Action Register

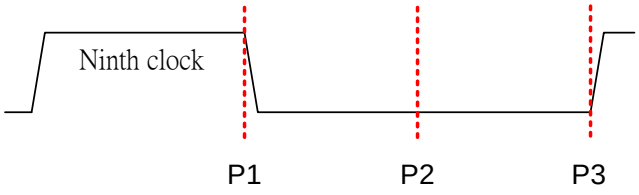
Bit	Name	Description
Bit7	SLAVE	Slave enablement control <0> Disable <1> Enable
Bit4	I2CER	Error interrupt flag <0> Normal; writing 0 will clear the error interrupt flag to make I^2C proceed to the next state. <1> Error interrupt occurs.
Bit3	START	Start order bit <0> Normal <1> Generate the "Start" signal on the I^2C bus.
Bit2	STOP	Stop order bit <0> Normal <1> Generate the "Stop" signal on the I^2C bus.
Bit1	I2CINT	Interrupt flag <0> Normal; writing 0 will clear the interrupt flag to make I^2C proceed to the next state. <1> The I^2C interrupt occurs.
Bit0	ACK	ACK(Acknowledge) acknowledge bit <0> Reply ACK or reply NACK. <1> ACK has been replied.

STA: I^2C status register

Bit	Name	Description
Bit7	MACTF	Master Mode Active Flag <0> Disable <1> Enable
Bit6	SACTF	Slave Mode Active Flag <0> Disable <1> Enable
Bit5	RDBF	Received Stop/Repeat-Start Flag <0> Normal <1> Received stop flag or repeat-start flag has been transmitted or received.
Bit4	RWF	Read/Write State Flag <0> The write command has been transmitted or received. <1> The read command has been transmitted or received.

Bit	Name	Description
Bit3	DFF	Data Field Flag <0> Normal <1> The I²C data are transmitted or received.
Bit2	ACKF	ACK flag (Acknowledge Flag) <0> ACK has yet to be transmitted or received. <1> ACK has been transmitted or received.
Bit1	GCF	General Call Flag <0> Normal <1> Currently General Call Operation
Bit0	ARBF	Arbitration Lost Flag <0> Normal <1> Arbitration is lost

CRG: I²C clock control register

Bit	Name	Description
Bit7~0	CRG[7:0]	I²C Bus Data Baud Rate Control Master Mode: The data transmission of the I²C bus is determined by the clock signal of the SCL pin, and the clock reset of the SCL pin can be calculated by the following equation according to the frequencies CPU_CK and CRG of the clock source of the I²C circuit: $\text{Data Baud Rate(Hz)} = \frac{\text{I2C\_CK}}{[4 \times (CRG[7:0] + 1)]}$ Slave Mode:  • P1~P2 is determined by the ISR time, the ISR time determined by the I2CINT=0 • the time of P2~P3 is determined by CRG[7:0] Min=(CRG+1)*(CPU_CK Cycle) Max=2*(CRG+1)*(CPU_CK Cycle)

TOC: I²C time-out control register

Bit	Name	Description
Bit7	I2CTF	Time-out flag <1> I²C Bus Clock Stretching Time-out <0> Normal

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit6~4</td><td>DI2C[2:0]</td><td>Time-out Clock Pre-scale
<table>
<tr><th>DI2C[2:0]</th><th>Pre-scale</th><th>DI2C[2:0]</th><th>Pre-scale</th></tr>
<tr><td>000</td><td>CLKPS = CPU_CK / 1</td><td>100</td><td>CLKPS = CPU_CK / 16</td></tr>
<tr><td>001</td><td>CLKPS = CPU_CK / 2</td><td>101</td><td>CLKPS = CPU_CK / 32</td></tr>
<tr><td>010</td><td>CLKPS = CPU_CK / 4</td><td>110</td><td>CLKPS = CPU_CK / 64</td></tr>
<tr><td>011</td><td>CLKPS = CPU_CK / 8</td><td>111</td><td>CLKPS = CPU_CK / 128</td></tr>
</table></td></tr>
<tr><td>Bit3~0</td><td>I2CTLT[3:0]</td><td>Time-out Limit; the occurrence of time-out is triggered after CLKPS counts for I2CTLT + 1 times.
<table>
<tr><th>I2CTLT[3:0]</th><th>Limit</th><th>I2CTLT[3:0]</th><th>Limit</th></tr>
<tr><td>0000</td><td>1 x CLKPS Cycle</td><td>1000</td><td>9 x CLKPS Cycle</td></tr>
<tr><td>0001</td><td>2 x CLKPS Cycle</td><td>1001</td><td>10 x CLKPS Cycle</td></tr>
<tr><td>0010</td><td>3 x CLKPS Cycle</td><td>1010</td><td>11 x CLKPS Cycle</td></tr>
<tr><td>0011</td><td>4 x CLKPS Cycle</td><td>1011</td><td>12 x CLKPS Cycle</td></tr>
<tr><td>0100</td><td>5 x CLKPS Cycle</td><td>1100</td><td>13 x CLKPS Cycle</td></tr>
<tr><td>0101</td><td>6 x CLKPS Cycle</td><td>1101</td><td>14 x CLKPS Cycle</td></tr>
<tr><td>0110</td><td>7 x CLKPS Cycle</td><td>1110</td><td>15 x CLKPS Cycle</td></tr>
<tr><td>0111</td><td>8 x CLKPS Cycle</td><td>1111</td><td>16 x CLKPS Cycle</td></tr>
</table></td></tr>
</table>

RDB: Data receiving register

Bit	Name	Description
Bit7~1	RDB[7:1]	The content is the receiving addresses (A7~A1) or the data (D7~D1).
Bit0	RDB[0]	The content is to receive the read/write command or the data (D0).

TDB0: Data transmission register

Bit	Name	Description
Bit7~1	TDB0[7:1]	The content is the receiving addresses (A7~A1) or the data (D7~D1).
Bit0	TDB[0]	The content is to receive the read/write command or the data (D0).
※Notice: When the host machine is not at the state of transmitting address or data during the communication process, the register should be set as FFh becasue SDA Bus may be locked at low when the bit 7 of TDB0 is 0.		

SID0: Slave mode ID code configuration register

Bit	Name	Description
Bit7~1	SID[7:1]	Slave ID codes (A7~A1)
Bit0	SIDV[0]	Effective control of slave ID code <0> the slave ID code is invalid. <1> the slave ID code is valid.

14. Enhanced Universal Asynchronous Receiver Transmitter

The peripherals of the enhanced universal asynchronous receiver transmitter (EUART) can also be called by the serial communication interface or SCI; EUART can not only be configured to be the full-duplex asynchronous system able to communicate with CRT terminals and personal computers, but also can be configured to the half-duplex synchronous system able to communicate with A/D or D/A integrated circuits, serial EEPROM and other external communication devices.

EUART has two additional functions, the frame error test and the automatic address identification when compared with the standard URTA; the frame error test can determine whether a frame is effective by testing the stop bit of the information of the frame. The automatic address identification function can automatically compare the content of the frame of the received address with the address of the computer on a chip, and generate a serial interrupt only after they are matched with each other. Regarding the two functions of the version, the former is finished by the built-in hardware circuit and the latter is finished by the software of the user.

EUART can be configured to be at the following work and debug modes:

- The full-duplex asynchronous mode has the following functions:
  Baud transmission rate generator
- Debug mode
  Frame error detection[3]
  Overrun error detection[4]
  Hardware parity check code
- Data transmission and reception
  Asynchronous transmission (8 bits or 9 bits)
  Asynchronous transmission (8 bits or 9 bits)
- Byte reception automatic wake-up function

EUART registers include:

UR0CON	ENSP[0],ENTX[0],TX9[0],TX9D[0],PARITY[0]
UR0STA	RC9D [0],PERR[0],FERR[0],OERR[0],RCIDL[0],TRMT[0],ABDOVF[0]
BA0CON	ENCR[0],RC9[0],ENADD[0],ENABD[0]
BR0GR[15:0]	BG0RH[7:0], BG0RL[7:0]
TX0G	TX0G[7:0]
RCREG	RCREG[7:0]

[3] Frame error detection (FERR): UART fails to receive the initial bit; in other words, it means that we cannot understand when the received signal starts and ends; which usually results from the noise in the signal line, so the UART cannot obtain the correct data from the transmit shift register.

[4] Overrun error detection (OERR): the latest piece of data has covered the previous data which not be taken.

14.1. Usage description of EUART

14.1.1. Transmission configuration steps of asynchronous data

- After ENSP is set as 1, PT4.4 and PT4.5 are automatically set as the UART pin.
- Configure the TXIE bit of the INTE1 register and the GIE bit of the INTE0 register to determine whether the transmission of the interrupt enablement is permitted. (The default configuration of the TXIF bit of the INTF1 register is high, and the relevant interrupt enablement should be configured after confirmation)
- Configure the BG0RH and BG0RL registers to determine the proper value of the BAUD rate.
- Configure the TX9 bit of the UR0CN register to determine whether to enable the data transmission function of the data of the 9th bit. (If the transmission function of the data of the 9th bit is enabled, the data should be filled in to the TX9D bit. The 9th bit may be an address or data).
- Configure the ENTX bit of the UR0CN register to enable the data transmission function.
- Write data in the TX0R register to determine the transmitted data. (Enable the transmission after the data are written in)

14.1.2. Reception configuration steps of asynchronous data

- After ENSP is set as 1, PT4.4 and PT4.5 are automatically set as the UART pins.
- Configure the INTE2, the RCIE bit of the register and the GIE bit of the INTE1 register to determine whether to allow receiving the interrupt enablement.
- Configure the BG0RH and BG0RL register to determine the proper value of the BAUD rate.
- Configure the ENSP bit of the UR0CN register to enable the EUART serial I/O module.
- Configure the RC9 bit of the BA0CN register to determine whether to enable the data receiving function of the data of the 9th bit.
- Configure the ENCR bit of the BA0CN register to enable the data receiving function.
- Read the RC9D bit of the UR0STA register to capture the data of the 9th bit of the received data (when RC9 is configured), and then determine whether an error occurs during the receiving process.
- Read the RCREG register to capture the data of the 8 bits in total of the received data.
- Read the FERR bit of the UR0STA register is configured, and confirm whether the read data is wrong; it is possible to clear the ENCR bit to release the FERR bit.

14.1.3. Reception configuration steps of asynchronous data (9 bits, RS-485 mode)

- After ENSP is set as 1, PT4.4 and PT4.5 are automatically set as the UART pins.
- Configure the BG0RH and BG0RL registers to determine the proper value of the BAUD rate.
- Configure the ENSP bit of the UR0CN register to enable the EUART serial I/O module.
- Configure the RC9 bit of the BA0CN register to determine whether to enable the data receiving function of the data of the 9th bit.
- Configure the ENADD bit of the BA0CN register to enable the address test function.
- Configure the ENCR bit of the BA0CN register to enable the data receiving function.
- Configure the INTE2, the RCIE bit of the register and the GIE bit of the INTE1 register to determine whether to allow receiving the interrupt enablement.
- Read the RC9D bit of the UR0STA register to capture the data of the 9th bit of the received data (when RC9 is configured), and then determine whether an error occurs during the receiving process.
- Read the RCREG register to capture the data of the 8 bits in total of the received data.
- Read the FERR bit of the UR0STA register is configured, and confirm whether the read data is wrong; it is possible to clear the ENCR bit to release the FERR bit.
- Configure the ENADD bit of the BA0CN register to disable the address test, and make the next piece of data be received.

14.2. Serial BAUD transmission rate generator (BRG)

BRG is a dedicated 13-bit generator and support the asynchronous mode of EUART. The BG0R[15:0] register is the cycle controller of an independent operation timer. Table 14-1 is the serial transmission BAUD rate calculation equation, but only can be applied to the master mode.

Under the condition that the targeted serial transmission BAUD rate is given and the working clock is CPU_CK, the equation of Table 14-1 can be used to calculate the approximate integer of the BG0R [15:0] register in order to confirm the error of the serial transmission BAUD rate. Example 14-1 describes the calculation methods of the serial transmission BAUD rate and the error rate.

BRG/EUART MODE	Calculation method of serial transmission BAUD rate
13 bits/asynchronous	CPU_CK÷[4 (n + 1)]
CPU_CK=working frequency; n = value of BG0RH:BG0RL register pair	

Table 14-1 Serial transmission BAUD rate equation

When working under the asynchronous mode, its working frequency is CPU_CK (2MHz), but the targeted serial transmission BAUD rate is 9600bps. Calculate BG0R[15:0]=< ? > and then BG0RH[7:0]:BG0RL[7:0]=< ? >

Known equation: targeted serial transmission BAUD rate＝CPU_CK÷(4 (<BG0R[15:0]>+1)):

Thus, BG0R[15:0]＝((CPU_CK÷targeted serial transmission BAUD rate)÷4)–1

＝((2000000÷9600)÷4)–1

＝51.08

≒51 then, BG0RH[7:0]=<00>, BG0RL[7:0]=<33>;

PS: 33 is hexadecimal system.

In fact, the Calculation result of BRG is: actual serial transmission BAUD rate＝ 2000000÷4(51+1))＝9615.38

Therefore, there is an error existing, and its calculation method is:

Error rate＝(Actual serial transmission BAUD rate–Targeted serial transmission BAUD rate)/Targeted serial transmission BAUD rate

＝(9615-9600)/9600

＝0.16%

Example 14-1 Calculation of error of serial transmission BAUD rate

14.2.1. Operations under power consumption management mode

The clock of the ship is used to generate the needed serial transmission BAUD rate. When entering a kind of power consumption management mode, the new clock source may work under a different frequency. It may need to adjust the value of the BG0R[15:0] register.

14.2.2. Sampling method of RC

The sampling circuit can perform sampling at the central point of the period of the transmission

BAUD rate in order to determine whether the RC pin is high level or low level.

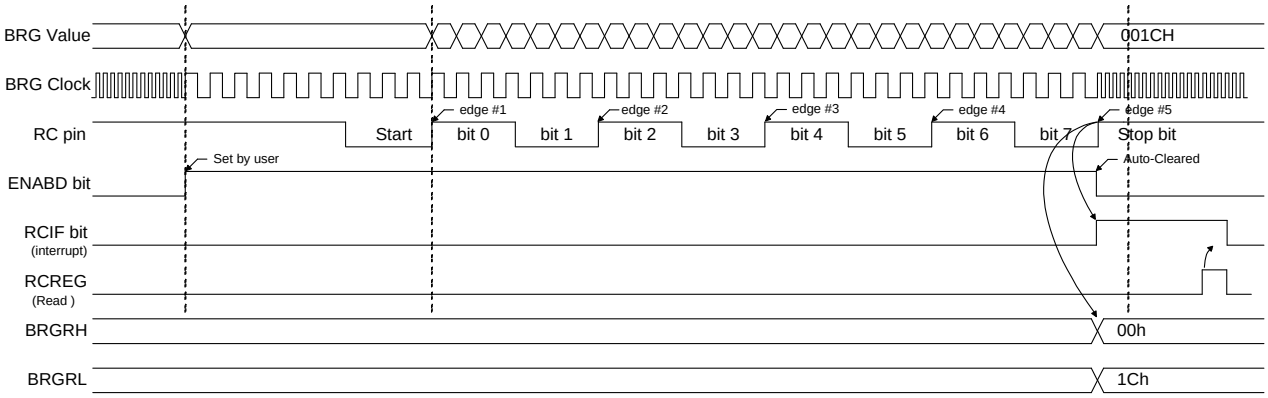
14.2.3. Automatic BAUD rate

The EUART module supports the automatic test and calibration functions, which is also called automatic BAUD rate. The automatic BAUD rate can be effective only after the wake-up enablement controller WUE[0] is set as 0, and then the automatic BAUD rate enablement controller ENABD[0] is set as 1.

After the start state is received, the automatic BAUD rate test function begins (the received value should be 055H, and cannot be the even parity check). After the automatic test and calibration are finished, the calculated result will be written in BG0RH[7:0] and BG0RL[7:0]; the relevant timing is as shown in figure 14-1.

When the calculation of BG0R[15:0] overflows, it content from 01FFFH to 00000H overflows; thus, the automatic BAUD overflow flag ABDOVF[0] is set as 1; the user can use an instruction to set ABDOVF[0] as 0 or set ENABD[0] as 0. After ABDOVF[0] is set as 1, the state of ENABD[0] still remains at 1; the relevant timing is as shown in figure 14-2.

Automatic BAUD rate calculation



Note：The ABD sequence requires the EUART module to be configured in WUE = 0

Figure 14-1 Waveforms of automatic BAUD rate calculation

BRG Overflow Sequence

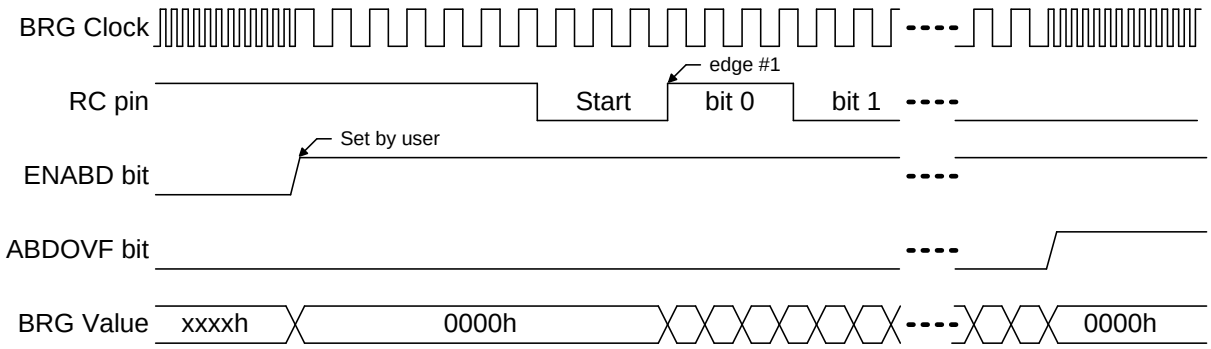


Figure 14-2 Waveforms of automatic BAUD rate overflow (ABDOVF)

14.3. Hardware parity check

- EUART supports the hardware odd/even test function; the test bit is saved in the 9th data bit. When using it, the parity check (ENADD[0]=0) is performed according to the register configuration of the user; the relevant settings are as shown in Table 14-2.

Transmit/receive data of 8/9 bits		PARITY	State
TX9	RC9		
0	0	0	Transmit/receive data without parity check information.
0	0	1	Transmit/receive data without parity check information.
0	1	0	Receive data with parity check function, even parity.
0	1	1	Receive data with parity check function, odd parity.
1	0	0	Transmit data with parity check code, even parity.
1	0	1	Transmit data with parity check code, odd parity.
1	1	0	
1	1	1	
PS: when RC9[0] is set as 1, the parity check function is enabled and PERR[0] is set as 1 when the odd/even parity is incorrect. If RC9[0] and ENADD[0] are set as 1 at the same time, the value of incorrect PERR[0] bit is neglected.			

Table 14-2 Parity check state table

14.4. EUART asynchronous mode

- The mode uses the standard "Non-Return-to-Zero (NRZ) format"; it is just one initial bit added by the data of 8 or 9 bits and the last bit is the stop bit; the most frequently-used data format is 8-bit. Besides, the 13-bit serial transmission BAUD rate generator dedicated for the chip can generate the standard serial transmission BAUD rate via the working clock oscillator.
- Further, the first data which EUART transmits or receives is the least significant bit; the function of the transmitter and that of the receiver are independent from each other, but they adopt the same data format and serial transmission BAUD rate. It further supports the hardware odd/even test function, and the test bit is stored in the 9th data bit.

14.4.1. EUART asynchronous transmitter (UART TXIF/RCIF flag from 0->1 and interrupt occurs)

Figure 14-3 is the timing diagram of the EUART transmitter, and the core of the transmitter is to transmit the data inside the transmit shift register (TSR), but the user cannot read/write TSR.

TSR can obtain the data from the read/write buffer register TX0R[7:0]. The data in the TX0R[7:0] register can be written in via software, and will not load the data from the TSR register before the transmission of the stop bit previously loaded is finished; once the transmission of the stop bit is finished, the new data of the TX0R register (if the data exist) will be loaded into TSR. Once the TX0R register transmits the data to the TSR register, TX0R register is empty (no written-in operation occurs again); meanwhile, the mark bit TXIF is set from 1 to 0 (when the ENTX bit of the UR0CN register is set, the TXIF bit will be set as 1). However, TXIF will not be immediately cleared as 0 after TX0R is inputted by new data, but will be cleared to be 0 at the second instruction period after the new data are inputted. When TXIF is set as 0, it will be set as 1 again after one instruction period. It is possible to set the interrupt allowance bit TXIE as 1 or clear it to be 0 to permit/prohibit the interrupt. No matter how the status of TXIE is, TXIF will be set from 1 to 0 after the interrupt occurs, and cannot be cleared to be 0 by software; besides, it will be set as 1 again after one instruction period. If the data inside the TSR register has yet to be completely transmitted and the data are written into the TX0R register at this time, TXIF will be cleared to be 0 at the second instruction period after the new data are inputted, and TXIF will be set as 1 until STOP BIT occurs.

Thus, after the new data are inputted in TX0R, it is necessary to search TXIF right wary, and its returned value cannot serve as the reference. TXIF stands for the state of the TX0R register and the other bit TRMT stands for the status of the TSR register. TRMT is the read-only bit, and it is set as 1 when the TSR register is empty (the loading operation has yet to be executed again). The TRMT bit is irrelevant to any interrupt logic; thus, if it needs to confirm whether the TSR register is empty, the user can only check the bit. Please refer to following figure 14-4 and figure 14-5 for more information about the timing diagram of the data asynchronous transmission.

- The UART operation is irrelevant to the CPU instruction period in addition to read/write.
- TXIF and RCIF stand for the interrupt purpose and are irrelevant to any other events.
- It is necessary to pay attention to the relative operation speed when using CPU to check the peripheral components.

EUART TRANSMIT BLOCK DIAGRAM

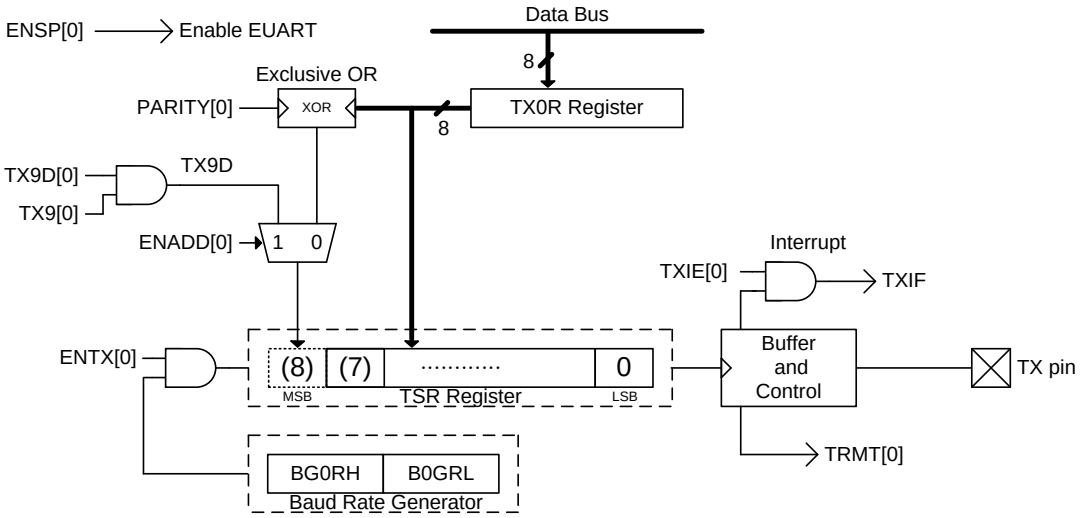


Figure 14-3 EUART transmission block diagram

ASYNCHRONOUS TRANSMISSION

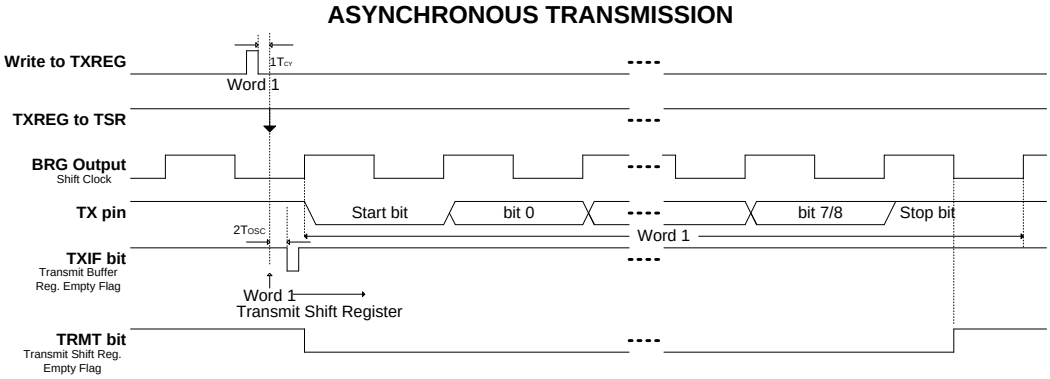


Figure 14-4 Timing diagram of asynchronous transmission

ASYNCHRONOUS TRANSMISSION (Back to Back)

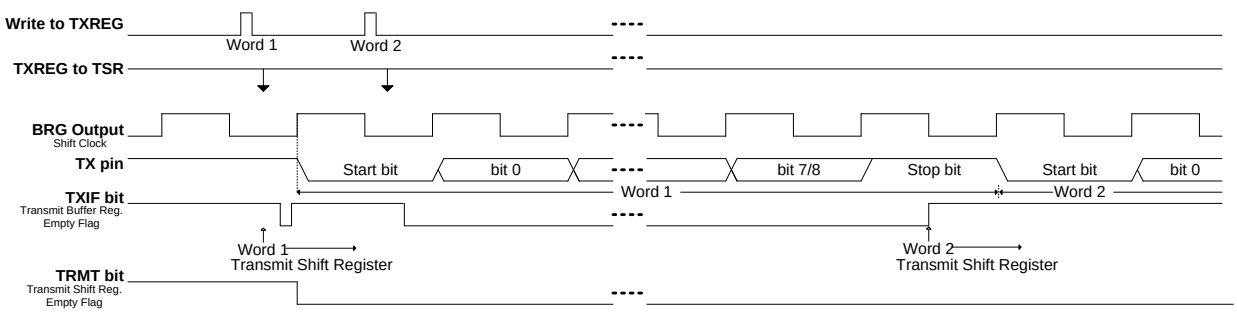


Figure 14-5 Asynchronous transmission timing (back to back)

◆ EUART asynchronous receiver

Figure 14-6 and figure 14-7 shows the block diagram of the receiver. figure 14-8 shows the asynchronous receiving timing. The RC pin receives the data to drive the data recovery circuit. The data recovery circuit is actually a high-speed shifter with the working frequency of 13-bit serial transmission BAUD rate; besides, the working frequency of the main receiving serial shifter is equal to the BAUD rate or OSC_RC2M. The mode is usually applied to the RS-232 system.

If the RC pin fails to receive a complete byte (start, 8(9) bit data, end) when receiving data, the FERR bit will be set as 1; it is possible to the FERR bit by clearing the ENCR bit.

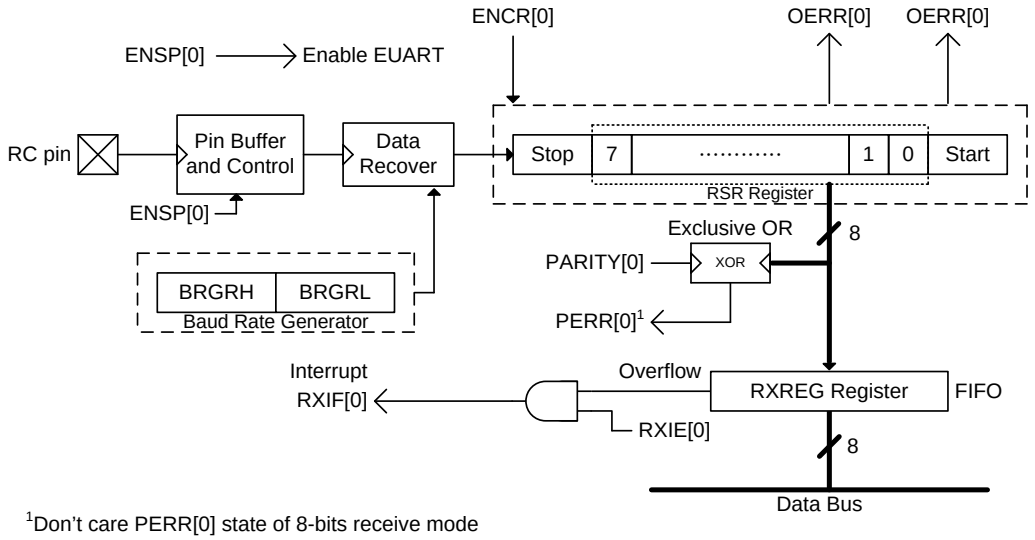
When the RC pin has received 2 pieces of complete byte data (they have yet to read the data from the RCREG register), the OERR will be set as 1, it is possible to clear the OERR bit by clearing the ENCR bit.

After the complete data are completely received, the RCIF bit of the INTF2 register will be configured, and the RCIF bit cannot be cleared by instruction when being configured; executing the operation of reading the RCREG register can clear the RCIF state.

The RCIDL bit of the UR0STA register can show whether it is under the receiving state. The user can indirectly determine whether the data reception is finished accordingly.

When receiving the data, the hardware will perform the operation of "exclusive or" for the 8-bit data; if RC9 is set as 1, it will perform the operation of "exclusive or" for the received RC9D data (9 bits in total). After the above operation, it will perform the operation of "exclusive or" with the PARITY bit set by the user, and then show the operation result in the PERR bit. If the received data are correct, PERR is set as 0; if the received data are wrong, PERR is set as 1. The FERR bit cannot be cleared by instruction; after the next piece of data is correctly received, FERR will be set as 0.

EUART 8-BITs RECEIVE BLOCK DIAGRAM



[1]Don't care PERR[0] state of 8-bits receive mode

Figure 14-6 EUART 8-bits receiving block diagram

EUART 9-BITs RECEIVE BLOCK DIAGRAM

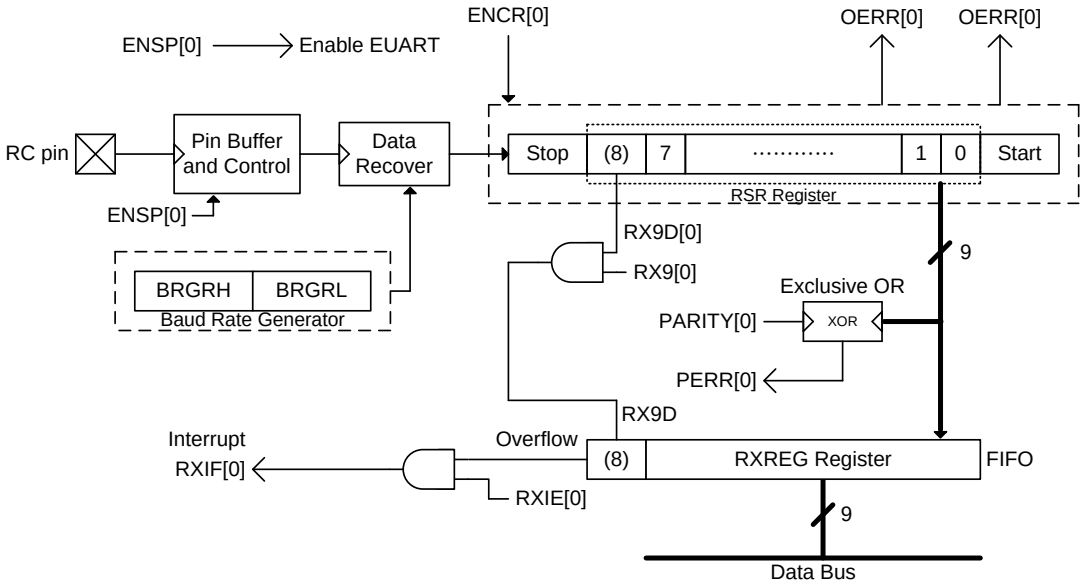


Figure 14-7 Block diagram of EUART 9-bits reception

ASYNCHRONOUS RECEPTION

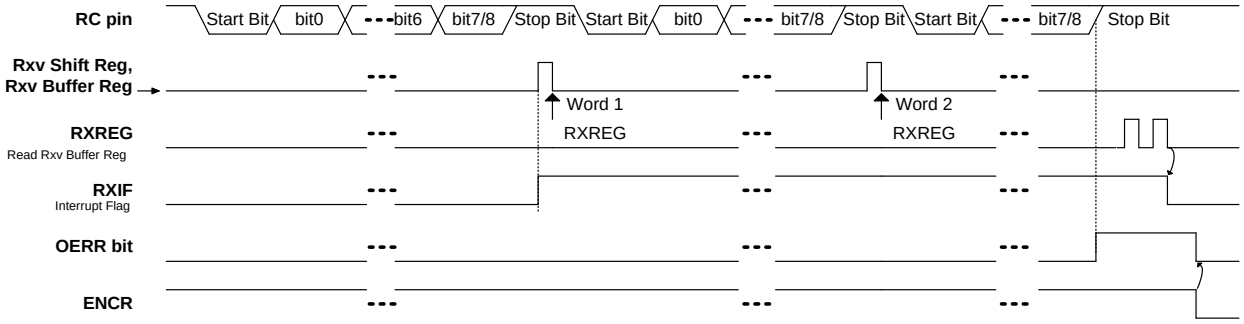


Figure 14-8 Asynchronous receiving timing

- 9-bit mode of address test function

  The mode is usually applied to the RS-485 system. The user can configure the asynchronous reception operation with the address test function according to the chapter of the usage description of EUART. The user can determine whether it is the address test or the data test by the configuration of the ENADD bit of the BA0CN register.

- Byte reception automatic wake-up

  Under the sleep or the idle mode, all clock sources of EUART will temporarily stop. Thus, the serial transmission BAUD rate generator is under the non-enable status (ILDE UART will operate), and cannot receive the correct byte. The automatic wake-up function can allow the controller to be waken when an event occurs in the RC line; the function needs to set the WUE bit of the UR0CN register as 1 when EUART operates under the asynchronous mode (Sleep needs to be set as 1, but Idle does not need) so as to enable the automatic wake-up function. After the function is enabled, the typical reception operation on RC will be

prohibited, and EUART stays under the idle state, and monitors the wake-up event (which is irrelevant to the operation mode of CPU).

The wake-up event means the transfer from high level to the low level on the RC line. After the wake-up event, the module will generate a RCIF interrupt; under the normal operation mode, the interrupt and the Q clock will synchronously generate; please refer to the following figure 14-9; if the chip is under the sleep mode or the idle mode, both of which are not synchronous; please refer to the description of the following figure 14-10. The interrupt condition can be cleared by reading the RCREG register.

After the wake-up event, when the level transfer from low to high occurs on the RC line, the WUE bit is automatically cleared. Meanwhile, the EUART module will return to the normal working mode from the idle state. Accordingly, the user can know that the event ends.

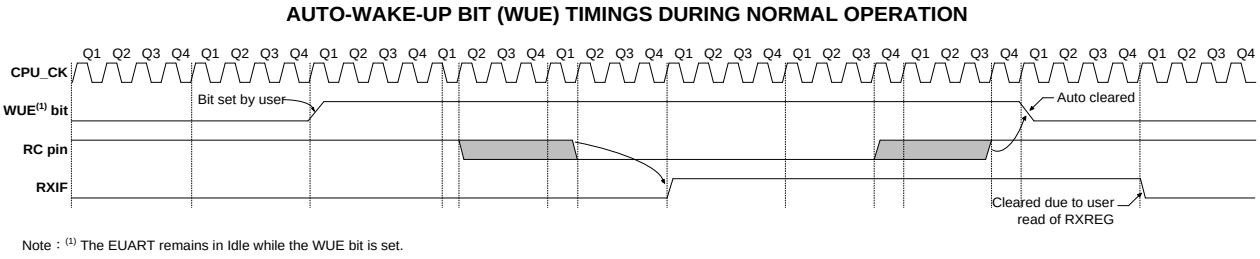


Figure 14-9 Automatic wake-up timing under normal mode

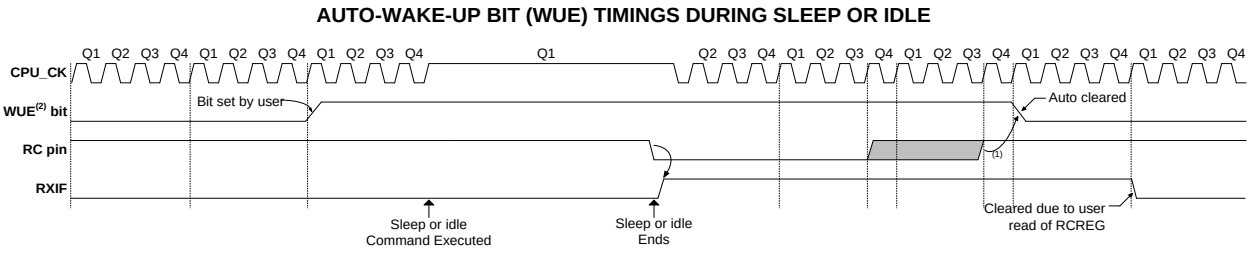


Figure 14-10 Automatic wake-up timing under sleep mode or standby mode

- Notices of using automatic wake-up function

  As the automatic wake-up function is realized by testing the jump of the ascending edge of RC, any state change on the pin before the stop bit may generate a wrong ending signal, and then result in data or frame error. Thus, the full 00H should be transmitted first so as to make sure that the correct transmission is made. (If it is 9bit, all of the 9bit should be 0)

  In addition, it is necessary to take the oscillation starting time of the oscillator into consideration; in particular, it is necessary to pay more attention when the oscillators with longer oscillation starting delay are adopted. Alternatively, the wake-up signal byte should be long enough, and has long enough time interval so as to make the selected oscillator have sufficient time to oscillate and make sure that EUART is correctly initialized.

- Notices of using WUE bit

  Using the timing of the WUE and the RCIF events to determine the effectiveness of the received data may result in confusion. As described above, setting the WUE bit as 1 will

make EUART enter the idle mode. The wake-up event will generate a reception interrupt, and set the RCIF bit as 1. Afterward, when the ascending edge of RC occurs, the WUE bit is cleared. Then, the interrupt condition is cleared by reading the RCREG register. Generally speaking, the data in the awaken RCREG are not effective, and should be abandoned. Clearing the WUE bit to be 0 (or still be 1) and setting the RCIF bit as 1 cannot make sure that the data received in RCREG are complete. The user should consider using the firmware to verify whether the data are completely received. So as to make sure that the effective data are not lost, it is necessary to check the RCIDL bit to verify whether the data are still received. If the data are not being receiving, the WUE bit can be set as 1 to make the chip enter the sleep mode.

14.5. Register description- UART

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
023h	INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024h	INTE1	-		TXIE	RCIE					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
027h	INTF1	-		TXIF	RCIF					0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
069h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	*,*,*,*,*,-,-,*
06Ah	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r r,r,r,rw0
06Bh	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD	 0000	.... uuuu	-,-,-,- *,*,*,*'
06Ch	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-,-,-,* *,*,*,*
06Dh	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06Eh	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06Fh	RCREG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
070h	PT1				PT1.4	PT1.3				xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
071h	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
073h	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
075h	PT1M2	-	PM1.7[0]	-	PM1.6[0]	-	PM1.5[0]	-	PM1.4[0]	xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*

Table 14-3 UART register

INTE0/INTE1/INTF1: please refer to the chapter "Interrupt" for more information.

TRISC4: PT4 pin characteristic control register: please refer to the chapter "Input/output port, I/O" for more information.

PT4M2: PT4 digital output mode selection register: please refer to the chapter "Input/output port, I/O" for more information.

UR0CN: UART control register

Bit	Name	Description
Bit7	ENSP	UART port function enablement bit <0> Disable the UART port, and set the configuration of TX and RC pins for I/O to use. <1> Enable the UART port, and set the configuration of the TX and RC pin for the UART port to use. PS: when the UART serial port is enabled, it is necessary to properly configure the usage of the input pin or output pin.
Bit6	ENTX	UART transmission function enablement bit <0> Disable <1> Enable
Bit5	TX9	Transmit 9th bit function enablement <0> Disable <1> Enable
Bit4	TX9D	Transmit 9th bit data <0> The data are"0". <1> The data are"1".
Bit3	PARITY	Odd/even parity check settings <0> Even parity check

		<1> Odd parity check
Bit0	WUE	Byte reception automatic wake-up enablement bit <0> Disable <1> Enable

UR0STA: UART status register

Bit	Name	Description
Bit6	RC9D	Receive 9th bit data <0> The data are "0". <1> The data are "1".
Bit5	PERR	Data parity check result flag <0> The received parity check is correct. <1> The received parity check is wrong.
Bit4	FERR	UART data reception incompleteness (start, 8(9) bit data, end) flag <0> means the data reception is complete. <1> means the data reception is not complete.
Bit3	OERR	State flag of receiving 2 pieces of data having yet to be handled <0> Not occur. <1> Occur.
Bit2	RCIDL	Flag of being under the reception state or not <0> under the reception state <1> not under the reception state
Bit1	TRMT	Flag of showing state of transmit shift register (TSR) <0> show the TSR register have data. <1> show the TSR register is empty.
Bit0	ABDOVF	Automatic BAUD rate overflow flag <0> Not occur. <1> Occur.

BA0CN: UART received data control register

Bit	Name	Description
Bit3	ENCR	Data reception function enablement bit <0> Disable <1> Enable
Bit2	RC9	Receiving 9th bit function enablement bit <0> Disable <1> Enable
Bit1	ENADD	Address test bit <0> Disable <1> Enable
Bit0	ENABD	Automatic BAUD rate enablement bit <0> Disable <1> Enable

BG0RH/BG0RL: Baudrate control register

TX0R: UART data transmission register

RCREG: UART data receiving register

15. Build-In EPROM

The Build-In EPROM (BIE) function can store the product serial number, the security code and the data generated after the operation of the programs, etc.; the external hardware only needs to connect to VBIE of 8.5V at the VPP/RST pin, and use the burning instruction to uses the function. The storage address range of the HY15P41 series is 00H~3FH, 64 words in total, which is equal to 128 bytes.

The external VBIE power source (8.5V) is used to burn the BIE block, the user can use one instruction to burn the data of one word into the BIE block at a time; when the burning sub-program is called to perform the burning operation for each time, only the data of one word can be burned, and the spent time is about 500msec.

Summary of BIE registers:

BIECN BIEWR[0], BIERD[0]

BIEARL BIE_ADDR[5:0]

BIEDH BIE_DATA[15:8]

BIEDL BIE_DATA[7:0]

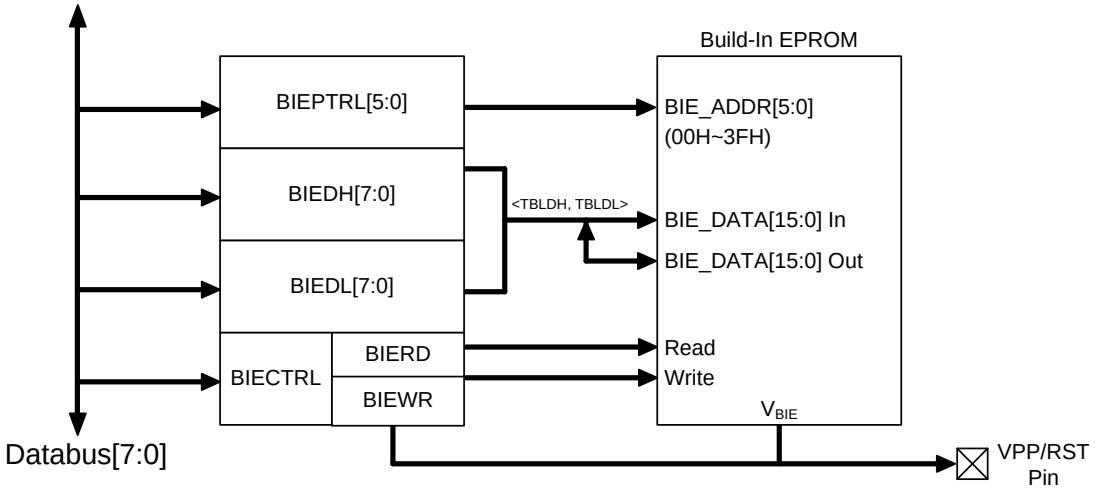


Figure 15-1 Block diagram of BIE

15.1. Register description- BIE

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1												
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**I-RESET**	**R/W**
02Eh	BIECN	1	-	-	-	-	-	BIEWR	BIERD	1... .000	1... .uuu	r1,-,-,- -,*,*,*
030h	BIEARL			BIE Address Register as BIEAL[5:0]						xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 15-1 BIE register

BIECN: BIE control register

Bit	Name	Description
Bit1	BIEWR	Write in the EPROM control bit <0> Can be written in <1> Cannot be written in
Bit0	BIERD	Read EPROM control bit <0> Cannot be read <1> Can be read

BIEARL: EPROM Low Byte address definition

BIEAL[5:0]: OTP address

BIEDH: EPROM High Byte data definition

BIEDL: EPROM Low Byte data definition

16. MODIFICATION RECORD

The following is the important modification of the document, but it does not include the changes of the punctuation and character form.

Document version	Page	Remark
V01	All	First version
V02	20~22	Correct the description of the CLK network.
	56~76	Modify the description of PWM.
	93~118	Optimize the description of UART and I2C.
	120	Modify the description of the BIE register.
V03	79~88	Modify the description of the ADC network configuration.
V04	97~102	Modify the description of the ADC register
	75~76	Modify the description of the PWMxA register
	105,114	Added the description of the I^2C to Slave Mode CRG [7: 0]
	95~97	Optimize the description of absolute temperature sensor TPS