HY17S68

User's Guide

8-Bit RISC-like Mixed Signal Microcontroller
Embedded High Resolution 19-Bit ΣΔADC
Low Noise OPAMP
Digital Multimeter Function

Table of Contents

1. Reading Guidance

1.1. About This User Guide

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.

1.2. Terms and Definition

1.2.1. Glossary

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

1.2.2. Register Related Glossary

[ ]	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

CPU Core (H08D) adopts Harvard architecture concept in order to enhance execution efficiency. Separate program memory and data memory incorporated in program memory address increases user convenience of program writing.

CPU features include：

- Isolated design frame of program memory and data memory upgrades instruction execution speed and CPU efficiency.
- At most 71 instructions including 16-bit look-up-table,8x8 hardware multiplier and program menory block switch and stack control.
- One instruction accomplished data movement from register A to register B without changing work register data.
- One instruction accomplished utmost 16-bit FSR register data movement and address 1MW program memory look-up-table instruction.
- Data memory operation includes program counter(PC),status register (Status) and stack register(Stack) data movement.
- Processor core is H08D core.

2.2. Memory

Memory is composed by program memory (OTP) and data memory (SRAM). Memory size differs from diverse part number; hence product data sheets should be read with extra caution.

Program Memory：

Main Program Memory (MPM)

Program Counter (PC)

Stack (STK)

Data Memory：

Special Register (SR)

General Purpose Register (GPR)

Memory Related Registers：(x：Means it constitutes several registers)

PC[13:0]	PCHSR[4:0],PCLATH[5:0],PCLATL[7:0]
TOS[13:0]	TOSH[4:0],TOSL[7:0]
FSRx[10:0]	FSRxH[2:0],FSRxL[7:0]
INDFx	INDF0[7:0],INDF1[7:0],INDF2[7:0]
POINCx	POINC0[7:0], POINC1[7:0], POINC2[7:0]
PODECx	PODEC0[7:0], PODEC1[7:0], PODEC2[7:0]
PRINCx	PRINC0[7:0], PRINC1[7:0], PRINC2[7:0]
PLUSWx	PLUSW0[7:0], PLUSW1[7:0], PLUSW2[7:0]
SKCN	SKFL[0],SKOV[0],SKUN[0],SKPRT[4:0]
PSTAT	BOR[0],PD[0],TO[0],IDL[0],RST[0], SKERR[0],BOR2LV[0],GCRSTIF[0]
BSRCN	BSR[2:0]

2.2.1. Program Memory

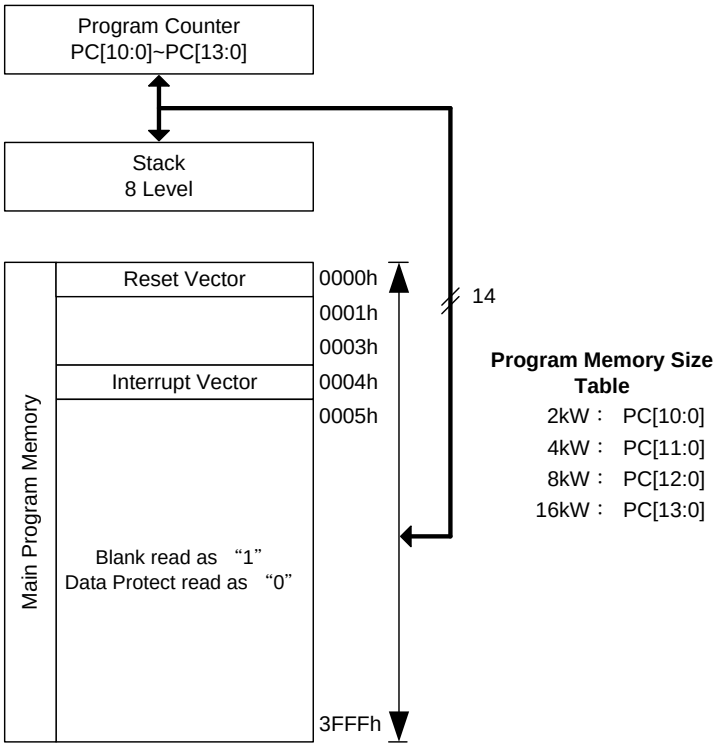


Figure 2-1 Program Memory Architecture

2.2.2. Main Program Memory (MPM)

MPM Architecture is as the following：

- Interrupt Vector
- Reset Vector

Maximum Address ability starting from 0x0000h to 0x3FFFh, the entire capacity is 16384 characters and it will vary with different part numbers.

Before the IC being written, data type of all bits is 1. After programming, the bit will show 1 or 0 according to the written data type. Please be noticed that if the emulation software (HYIDE) compiling option has been configured the programming protection function, all data type will only be read as 0.

2.2.3. Program Counter,PC

Program Counter (PC) includes shift register PCSR and buffer register PCLAT, as Figure 2-2 implicated.

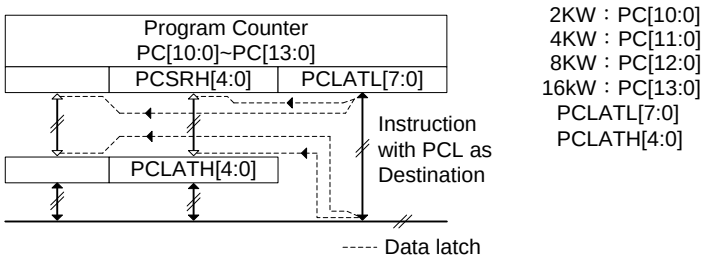


Figure 2-2 Program Counter Architecture

PC[13:0] of the ICE equips with 14 bit data length and is composed by two registers: PCSRH [4:0] and PCLATL [7:0]. PCLATL [7:0] and PCLATH [4:0] can be directly read/written but PCSRH [4:0] cannot. Buffer register, PCLATH [4:0] must be applied to carry out indirect read and write.

◆ To read PC[13:0], PCLATL[7:0] must be read first then to read PCLATH[4:0] in order to obtain

correct data. Any reverse order may result in incorrect data.

◆ To write PC[13:0], PCLATH[4:0] must be written first then to write PCLATL[7:0]. Any reverse

order may result in incorrect data.

2.2.4. Stack,STK

Stack, STK is mainly composed by Stack Index Control Register (SKCN),Stack Error Flag Bit (SKERR) and Stack Error Reset Controller (SKRST).

Stack overflow and stack underflow may result in unexpected result of program execution. If there is a necessary, it is suggested to restart the IC. In the processes of program development, stack reset control bit, SKRST[1] can be configured as <1> through software. When stack overflow or stack underflow take place, reset signal will be generated and SKERR will be set as <1> to restart the IC.

- Stack Full: Configure SKFL[0] as <1>, PC[13:0] is not influenced.
- Stack Underflow: Configure SKUN as <1>, PC[13:0] moves to 0x00000h, SKPRT points to 0 Level. If SKRST is set as <1>, reset signal will be aroused after stack underflow and SKERR may be configured as <1>, SKUN will be <0> after reset.
- Stack Overflow: Configure SKOV as <1>, PC[13:0] is not influenced but SKPRT remains at the last layer and new values may be written in. That is to say, the latest written-in data may be saved after stack full. If SKRST is configured as <1>, reset signal may be generated after stack overflow and SKERR may be set as <1>. SKOV will be set as <0> after reset.
- Error: Configure SKERR as <1>, stack error occurred. If SKRST is configured <1>, reset signal will be generated after stack overflow and SKERR will be set to <1>. SKUN and SKOV will be configured as <0> after reset.

2.2.5. 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[13]	PC[12]	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

SKCN: Stack Controller

Bit	Name	Description
Bit7	SKFL	Stack full flag <0> Not happened <1> Happened
Bit6	SKUN	Stack underflow flag <0> Not happened <1> Happened
Bit5	SKOV	Stack overflow flag <0> Not happened <1> Happened
Bit3~0	SKPRT[2:0]	Stack pointer register <000> The 0 layer, TOS[13:0]=0x0000h ⋮ <110> The 6th layer <111> The 7th layer

PCLATH: Program counter high byte，PC[13:8]

PCLATL: Program counter low byte，PC[7:0]

PSTAT: Status Register

Bit	Name	Description
Bit2	SKERR	Stack error generated reset flag <0> Not happened <1> Happened

2.2.6. Data Memory,DM

Data Memory comprises Special Register (SR) and General Purpose Register (GPR), every 256byte is a Bank. Bank0 and Bank1 are special; include 128byte SR and 128byte GPR respectively. Other Banks only contain GPR, as illustrated in Figure 2-3

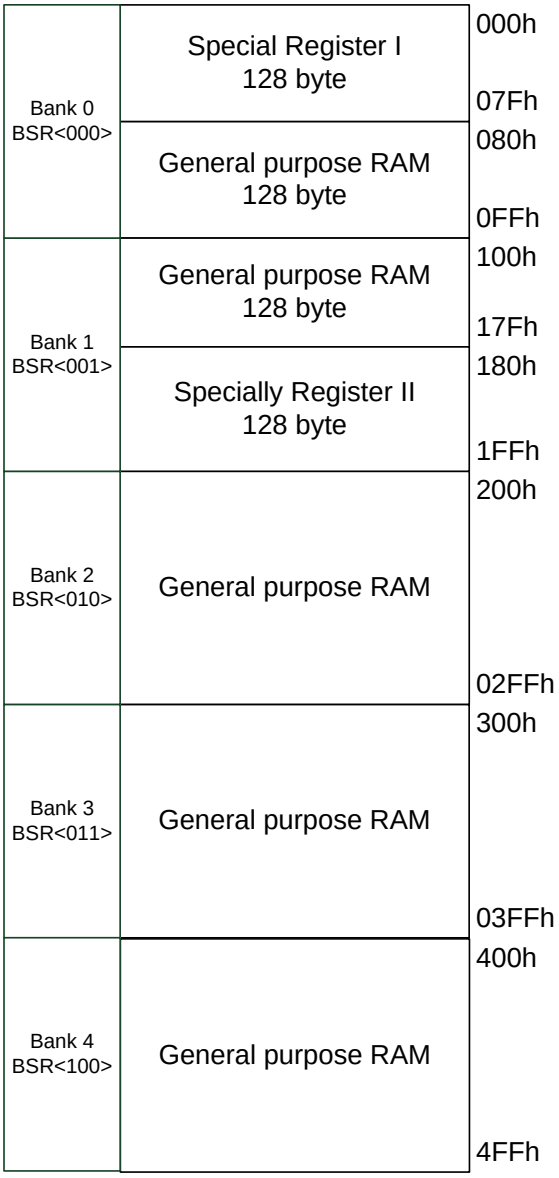


Figure 2-3 DM architecture diagram

2.2.7. Memory and Instruction

The H08 instruction set can be divided into three versions: A,B, and D, which have great differences in memory usage,such as address ability, hardware multiplier, look-up-table instruction, assistant function and arguments definition. Only definition of instruction memory arguments is illustrated in this chapter. Detail description of instruction arguments is depicted in Instruction chapter.

Instructions that contain address operation function of the instruction set have three arguments, namely "f", "d" and "a".

"f" is Data or Data Memory Address

"d" is data storage place after operation

d=0 is saved in WREG register

d=1 is saved in Data Memory Register

"a" is the designated memory operation bank

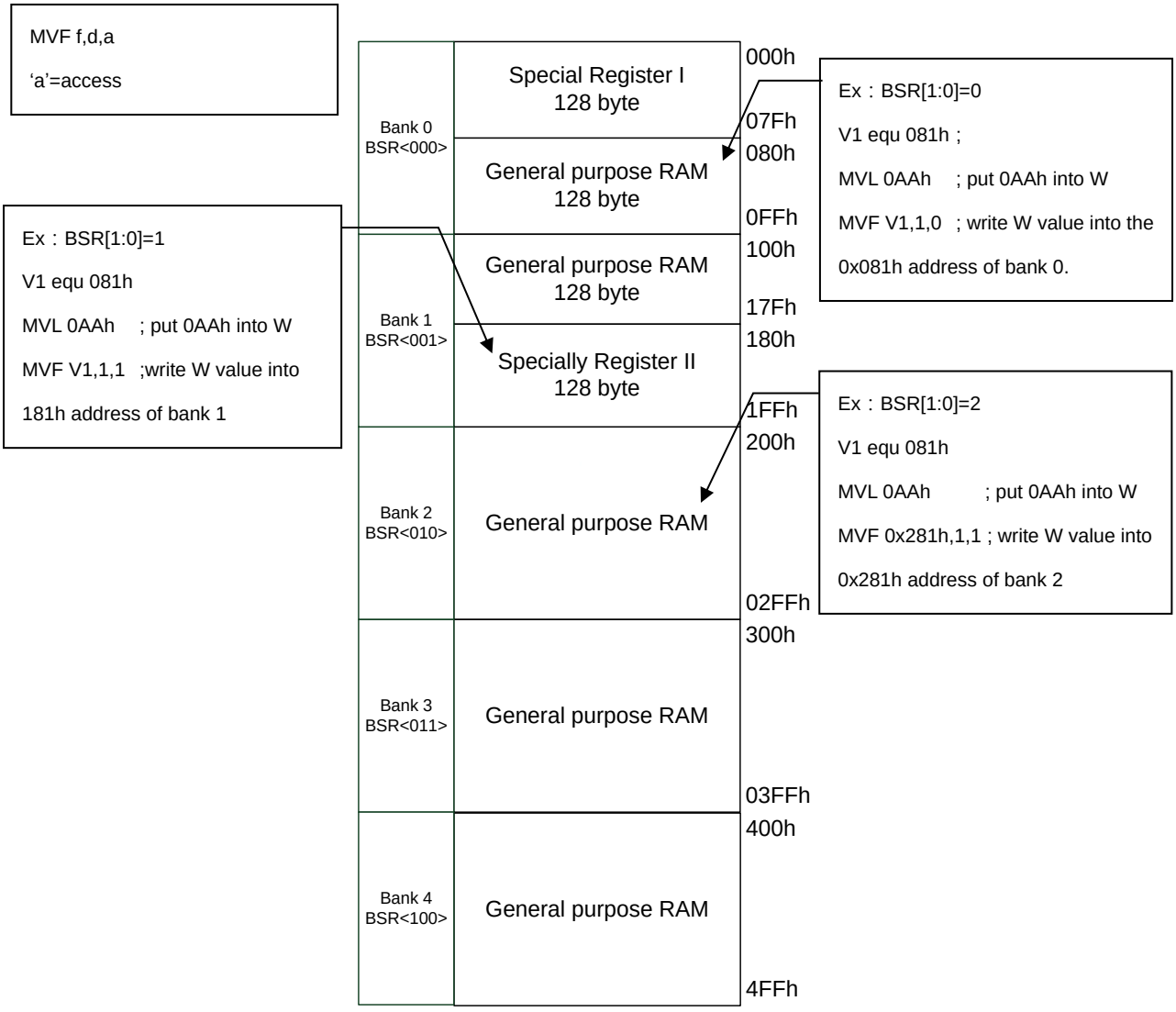
a=0 is operated in bank 0

a=1 is operated in designated bank of BSR[2:0]

2.2.8. Bank Select Control Register

Every 256 byte of data memory is set to be one bank(000h~0FFh). To read or write address 0FFh register data, it is necessary to set correct bank control register BSR [2:0] and instruction argument "a". Description is as follows:

- ◆When a = 0, no matter register BSR [2:0] appoints to which bank, data memory read/write instruction will only show in bank 0.
- ◆When a = 1, read/write instruction of H08D CPU Core to data memory will be in compliance with the assigned bank of BSR[2:0].



Example 2-4 Relation between bank selector example program and data memory

2.2.9. Special Register

Special register comprises CPU Core and peripheral function related registers, mainly are control function registers and data returned registers. Undefined address or address bit of data register will show 0 while reading and writing.

There are several instruction collocation registers contained in special register, only two common types, working register (WREG) and indirect address register (FSR) are introduced herein. Other special registers will be illustrated in depth in other chapter.

2.2.10. Working Register,WREG

Working register is abbreviated as W, which acts as the most frequently used register for data movement, operation and diagnosis.

2.2.11. Indirect Address Register,FSR and INDF

File select register, FSR includes instruction register FSR0 [10:0], FSR1 [10:0] , FSR2 [10:0] and index register, INDF0 [7:0] , INDF1 [7:0] and INDF2 [7:0]. Because of function similarity, only FSR0 is explained in this chapter.

FSR0[10:0] can be separated into two registers, FSR0H[2:0] and FSR0L[7:0]. There is no need to set up BSR [2:0] to address different banks. Through special instruction, only applying one instruction can write 16-bit data.

INDF0 [7:0] is index register that can read FSR0[10:0] appointed address data of data memory.

H08D instruction set supports enhanced index register,the functions are characterized as follows:

- POINC0[7:0]：Events that ensued by read/write POINC0 [7:0] register by instruction.
  - The address value that FSR0 [10:0] pointed to will be sent back first.
  - Then pointer register, FSR0[10:0] value will add 1 and points to the next address.
- PODEC0[7:0]：Events that followed by read/write PODEC0 [7:0] register by instruction.
  - The address value that FSR0 [10:0] pointed to will be sent back first.
  - Then pointer register, FSR0[10:0] value will subtract 1 and points to the last address.
- PRINC0[7:0]：Events that followed by read/write PRINC0[7:0] register by instruction.
  - Pointer register FSR0[10:0] value will add 1 and points to the next address.
  - Current value of FSR0[10:0] appointed to address will be sent back.
- PLUSW0 [7:0]：Events that ensued by read/write PLUSW0 [7:0] register by instruction.
  - Add pointer register, FSR0[10:0] value together with working register, W value.
  - Send back current FSR0[10:0] appointed address value. The register W value will be±128d.

2.2.12. General Purpose Register,GPR

General Purpose Register,GPR is the free area for users to conduct data storage,operation,flag bit setup….etc.

2.2.13. 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	*,*,*,*,*,*,*,*
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								xxxx xxxx	*,*,*,*,*,*,*,*
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								xxxx xxxx	*,*,*,*,*,*,*,*
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	*,*,*,*,*,*,*,*
00Fh	FSR0H	-	-	-	-	-	FSR0[10]	FSR0[9]	FSR0[8]	 xxxx	-,-,-,- *,*,*,*
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	*,*,*,*,*,*,*,*
011h	FSR1H	-	-	-	-	-	FSR1[10]	FSR1[9]	FSR1[8]	xxx	-,-,-,- *,*,*,*
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	*,*,*,*,*,*,*,*
013h	FSR2H	-	-	-	-	-	FSR2[10]	FSR2[9]	FSR2[8]	xxx	-,-,-,- *,*,*,*
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	*,*,*,*,*,*,*,*
016h	TOSH	-	-	TOS[13]	TOS[12]	TOS[11]	TOS[10]	TOS[9]	TOS[8]	..xx xxxx	-,*,*,*,*,*,*,*
017h	TOSL	Top-of-Stack Low Byte (TOS[7:0])								xxxx xxxx	*,*,*,*,*,*,*,*
029h	WREG	Working Register								xxxx xxxx	*,*,*,*,*,*,*,*
02Ah	BSRCN	-	-	-	-	-	BSR[2]	BSR[1]	BSR[0]	 xxxx	-,-,-,- *,*,*,*

Table 2-2 Data Memory Control Register

INDFx/POINCx/PODECx/PRINCx/PLUSWx: Different Functional Index Register (x=0、1、2)

INDFx[7:0]: Please refer to 2.2.11 Indirect Address Register,FSR and INDF description

POINCx[7:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

PODECx[7:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

PRINCx[7:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

PLUSWx[7:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

FSRx: Force Select Register (x=0、1、2)

FSRxH[2:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

FSRxL[7:0]: Please refer to 2.2.11 Indirect Address Register, FSR and INDF description

WREG: Working Register

WREG[7:0]: Please refer to 2.2.10 Working Register,**WREG** description

BSRCN: Memory Bank Read/Write Control Register

Bit	Name	Description
Bit2~0	BSR[2:0]	Memory read/write Bank pointer register <000> Bank 0，address 0x000h~0x0FFh <001> Bank 1，address 0x100h~0x1FFh <010> Bank 2，address 0x200h~0x2FFh。 <011> Bank 3，address 0x300h~0x3FFh。 <100> Bank 4，address 0x400h~0x4FFh。

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	*,*,*,*,*,*,*,*
00FH	FSR0H	-	-	-	-	-	FSR0[10:8]			 xxxx	.... .uuu	-,-,-,-,*,*,*,*
010H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
011H	FSR1H	-	-	-	-	-	FSR1[10:8]			xxx	.... .uuu	-,-,-,-,*,*,*,*
012H	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
013H	FSR2H	-	-	-	-	-	FSR2[10:8]			xxx	.... .uuu	-,-,-,-,*,*,*,*
014H	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
015H	TOSU	-	-	-	-	-	-	-	-	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
016H	TOSH			Top-of-Stack High Byte (TOS[13:8])						.xxx xxxx	..uu uuuu	-,*,*,*,*,*,*,*
017H	TOSL	Top-of-Stack Low Byte (TOS[7:0])								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
018H	SKCN	SKFL	SKUN	SKOV	SKPRT[4:0]					0000 0000	u$$$ $$$$	rw0,rw0,rw0,* *,*,*,*
019H	PCLATU	-	-	-	-	-	-	-	-	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
01AH	PCLATH			PC High Byte for PC[13:8]						..00 0000	..00 0000	*,*,*,*,*,*,*,*
01BH	PCLATL	PC Low Byte for PC[7:0]								0000 0000	0000 0000	*,*,*,*,*,*,*,*
01CH	TBLPTRU	-	-	-	-	-	-	-	-	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
01DH	TBLPTRH	-	-	Program Memory Table Pointer High Byte (TBLPTR[13:8])						..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	CTIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024H	INTE1	TA1IE	SPIIE	TXIE	RCIE	I2CERIE	I2CIE	E3IE	E2IE	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
025H	INTE2	MFCIE	CMPOIE	CMPHOIE	CMPLOIE	CTBOVE	RMSIE	LPFIE	BOR2IE	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
026H	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	CTF	E1IF	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
027H	INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
028H	INTF2	MFCIF	CMPIF	CMPHIF	CMPLIF	CTBOV	RMSF	LPFF	BOR2IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
029H	WREG	Working Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
02AH	BSRCN	-	-	-	-	-	BSR[2:0]			 xxxx	.... uuuu	-,-,-,-,*,*,*,*
02BH	MSTAT	-	-	-	C	DC	N	OV	Z	...x xxxx	...u uuuu	-,-,-,-,*,*,*,*
02CH	PSTAT	BOR	PD	TO	IDL	RST	SKERR	BOR2LV	GCRstIF	$000 $000	uu$u u$uu	rw0,rw0,rw0,rw0 rw0,rw0,r,rw0
02DH	BIECN	1	BLKSEL	-	ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1.00 $000	1.00 $uuu	r1,-,*,* r,*,*,*
02EH	BIEARH	-	-	BIE High Byte Address Register as BIEA[13:8]						0.xx xxxx	u.uu uuuu	*,-,*,*,*,*,*,*
02FH	BIEARL	BIE Low Byte Address Register as BIEA[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
030H	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
031H	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
032H	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]		ENLDO	CSFON	1000 0000	1uuu u00u	*,*,*,*,*,wr0,wr0,*
033H	PWRCN1	ENREFO	ENCMP	ENCNTI	ENCTR	ENVS	SAGND[2:0]			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
034H	PWRCN2	ENPUMP	VGGS	CHP_CKS[1]	CHP_CKS[0]	ENFIR	LDOPL	ENTPS	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
035H	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
036H	OSCCN1	CCOPT	LCPS	DADC[1:0]		DTMB[1:0]		TMBS	LCDS	0000 0000	uuuu uuu.	*,*,*,*,*,*,*,-
037H	OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0001	uuuu uuu1	*,*,*,*,*,*,*,*

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
038H	CSFCN0	SKRST	HAOTR[6:0]							.1..	....	*,*,*,*,*,*,*,*
039H	CSFCN1	BUSCKS	-	-	BOR_TH[2:0]			BORS	ENBOR2	0000 0011	uuuu uuuu	*,*,*,*,*,*,*,*
03AH	WDTCN	ENBZ	BZS	DBZ[1:0]		ENWDT	DWDT[2:0]			0000 0000	uuuu $000	-,*,*,* rw1,*,*,*
03BH	AD1CN0	ENAD1	ENCH	ENINXCH	VREGN	OSR[2:0]			CMFR	000. 0000	uuu. uuuu	*,*,*,*,*,*,*,*
03CH	AD1CN1	ENACM	VCMS	VCINS	TPSCH	TPSCP	ADGN[2:0]			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03DH	AD1CN2	FilterN[1:0]		-	DAFM	DCSET[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03EH	AD1CN3	SAD1FP[3:0]				-	SAD1FN[2:0]			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03FH	AD1CN4	AD1RHBUF	AD1RLBUF	AD1IPBUF	AD1INBUF	INX[1:0]		VRIS	INIS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
040H	AD1CN5	SAD1RH[2:0]			SAD1RL[2:0]			SAD1I[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
041H	RMSCN	ENRMS	ENLPF	ENSQRE	LPFBW[1:0]		ENPKH	RSLPF	RSRMS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
042H	NET0	SDIO	SREFO	SFT1[1:0]		SFUVR[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
043H	NET1	SMODE[7:4]				SMODE[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
044H	NET2	SCMPRH[3:0]				SCMPRL[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
045H	NET3	SCMPI[2:0]			SCMPO	CMPO	CMPHO	CMPLO	CNTI_IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
046H	PA1110	PS11	DS11	FS11	SS11	PS10	DS10	FS10	SS10	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
047H	PA98	PS9	DS9	FS9	SS9	PS8	DS8	FS8	SS8	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
048H	PA76	PS7	DS7	FS7	SS7	PS6	DS6	FS6	SS6	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
049H	PA54	PS5	DS5	FS5	SS5	PS4	DS4	FS4	SS4	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04AH	PA32	PS3	DS3	FS3	SS3	PS2	DS2	FS2	SS2	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04BH	PA10	PS1	DS1	FS1	SS1	PS0	DS0	FS0	SS0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04CH	CTAU	CTA[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04DH	CTAH	CTA[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04EH	CTAL	CTA[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04FH	CTBU	CTB[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
050H	CTBH	CTB[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
051H	CTBL	CTB[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
052H	CTCU	CTC[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
053H	CTCH	CTC[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
054H	CTCL	CTC[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
055H	PKHMAXU	PKHMAX[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
056H	PKHMAXH	PKHMAX[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
057H	PKHMAXL	PKHMAX[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
058H	PKHMINU	PKHMIN[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
059H	PKHMINH	PKHMIN[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05AH	PKHMINL	PKHMIN[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05BH	RMSDATA4	RMS[37:30]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05CH	RMSDATA3	RMS[29:22]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05DH	RMSDATA2	RMS[21:14]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05EH	RMSDATA1	RMS[13:6]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05FH	RMSDATA0	RMS[5:0]						-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
060H	LPFDATAU	LPF[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
061H	LPFDATAH	LPF[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
062H	LPFDATAL	LPF[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
063H	AD1DATAU	AD1[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
064H	AD1DATAH	AD1[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
065H	AD1DATAL	AD1[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
066H	OP1CN0	ENOP1	SOP1P[2:0]			OP1CHOP[1:0]		HS	OPS1	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
067H	ACC	Capacitor array								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
068H	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]			-	-	0000 00.0	u0uu uu.u	*,rw1,*,*,*,-,*
069H	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
06AH	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
06BH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06CH	PT1IN	IN1.7	IN1.6	IN1.5	IN1.4	IN1.3	IN1.2	IN1.1	IN1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06DH	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06EH	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06FH	PT1M1					INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
070H	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
071H	PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
072H	PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 2-4 Data Memory List (continued)

"-"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
073H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
074H	PT2IN	IN2.7	IN2.6	IN2.5	IN2.4	IN2.3	IN2.2	IN2.1	IN2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
075H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
076H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
077H	PT2INT	INTG2.7	INTG2.6	INTG2.5	INTG2.4	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
078H	PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	
079H	PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07AH	MFCCN0	CPRH[1:0]		MFCO	CPIS	CPOR	CPDF	CMPHS	ENMFC	0000 0000	uuuu uuuu	*,*,r,*,*,*,*,*
07BH	MFCCN1	CPRL[2:0]			-	CPPS[1:0]		CPNS[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07CH	MFCCN2	-	-	-	CPDA[4:0]					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07DH	MFCCN3	-	-	-	CPDM[4:0]					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
180H	LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELPCLK	LCDBL	ENLCD	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
181H	LCDCN2									0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
182H	LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
183H	LCDCN4	SSG21	SSG20	SSG19	SSG18	SSG17	SSG16	SSG15	SSG14	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
184H	LCDCN5					SSG37	SSG36	SSG35	SSG34	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
185H	LCDCN6	SSG5[1:0]		SSG4[1:0]		SSG3[1:0]		SSG02[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
186H	LCDCN7	SSG9[1:0]		SSG8[1:0]		SSG7[1:0]		SSG6[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
187H	LCDCN8	SSG13[1:0]		SSG12[1:0]		SSG11[1:0]		SSG10[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
188H	LCDCN9	SSG25[1:0]		SSG24[1:0]		SSG23[1:0]		SSG22[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
189H	LCDCN10	SSG29[1:1]		SSG28[1:1]		SSG27[1:1]		SSG26[1:1]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
18AH	LCDCN11	SSG33[1:1]		SSG32[1:1]		SSG31[1:1]		SSG30[1:1]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
18BH	LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18CH	LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18DH	LCD2	LCD SEG7[4:7] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18EH	LCD3	LCD SEG9[4:7] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18FH	LCD4	LCD SEG11[4:7] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
190H	LCD5	LCD SEG13[4:7] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
191H	LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
192H	LCD7	LCD SEG17[4:7] data				LCD SEG16[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
193H	LCD8	LCD SEG19[4:7] data				LCD SEG18[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
194H	LCD9	LCD SEG21[4:7] data				LCD SEG20[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
195H	LCD10	LCD SEG23[4:7] data				LCD SEG22[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
196H	LCD11	LCD SEG25[4:7] data				LCD SEG24[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
197H	LCD12	LCD SEG27[4:7] data				LCD SEG26[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
198H	LCD13	LCD SEG29[4:7] data				LCD SEG28[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
199H	LCD14	LCD SEG31[4:7] data				LCD SEG30[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19AH	LCD15	LCD SEG33[4:7] data				LCD SEG32[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19BH	LCD16	LCD SEG35[4:7] data				LCD SEG34[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19CH	LCD17	LCD SEG37[4:7] data				LCD SEG36[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19DH	LCD18	LCD SEG39[4:7] data				LCD SEG38[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19EH	LCD19	LCD SEG41[4:7] data				LCD SEG40[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19FH	PT3	PT3.7	PT3.6	PT3.5	PT3.4	PT3.3	PT3.2	PT3.1	PT3.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A0H	PT3IN	IN3.7	IN3.6	IN3.5	IN3.4	IN3.3	IN3.2	IN3.1	IN3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A1H	TRISC3	TC3.7	TC3.6	TC3.5	TC3.4	TC3.3	TC3.2	TC3.1	TC3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A2H	PT3PU	PU3.7	PU3.6	PU3.5	PU3.4	PU3.3	PU3.2	PU3.1	PU3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A3H	PT3M1	PM3.7 ENPCMPO				-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A4H	PT4	-	-	-			PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A5H	PT4IN	-	-	-			IN3.2	IN3.1	IN3.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A6H	TRISC4	-	-	-			TC4.2	TC4.1	TC4.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A7H	PT4PU	-	-	-			PU4.2	PU4.1	PU4.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A8H	PT6	PT6.7	PT6.6	PT6.5	PT6.4	PT6.3	PT6.2	PT6.1	PT6.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A9H	PT6IN	IN6.7	IN6.6	IN6.5	IN6.4	IN6.3	IN6.2	IN6.1	IN6.0	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1AAH	TRISC6	TC6.7	TC6.6	TC6.5	TC6.4	TC6.3	TC6.2	TC6.1	TC6.0	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ABH	PT6PU	PU6.7	PU6.6	PU6.5	PU6.4	PU6.3	PU6.2	PU6.1	PU6.0	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ACH	PT7	PT7.7	PT7.6	PT7.5	PT7.4	PT7.3	PT7.2	PT7.1	PT7.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ADH	PT7IN	IN7.7	IN7.6	IN7.5	IN7.4	IN7.3	IN7.2	IN7.1	IN7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1AEH	TRISC7	TC7.7	TC7.6	TC7.5	TC7.4	TC7.3	TC7.2	TC7.1	TC7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1AFH	PT7PU	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 2-5 Data Memory List (continued)

"-"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
1B0H	PT8	PT8.7	PT8.6	PT8.5	PT8.4	PT8.3	PT8.2	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B1H	PT8IN	IN8.7	IN8.6	IN8.5	IN8.4	IN8.3	IN8.2	IN8.1	IN8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B2H	TRISC8	TC8.7	TC8.6	TC8.5	TC8.4	TC8.3	TC8.2	TC8.1	TC8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B3H	PT8PU	PU8.7	PU8.6	PU8.5	PU8.4	PU8.3	PU8.2	PU8.1	PU8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B4H	PT9	PT9.7	PT9.6	PT9.5	PT9.4	PT9.3	PT9.2	PT9.1	PT9.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B5H	PT9IN	IN9.7	IN9.6	IN9.5	IN9.4	IN9.3	IN9.2	IN9.1	IN9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B6H	TRISC9	TC9.7	TC9.6	TC9.5	TC9.4	TC9.3	TC9.2	TC9.1	TC9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B7H	PT9PU	PU9.7	PU9.6	PU9.5	PU9.4	PU9.3	PU9.2	PU9.1	PU9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B8H	PT10	PT10.7	PT10.6	PT10.5	PT10.4	PT10.3	PT10.2	PT10.1	PT10.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B9H	PT10IN	IN10.7	IN10.6	IN10.5	IN10.4	IN10.3	IN10.2	IN10.1	IN10.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BAH	TRISC10	TC10.7	TC10.6	TC10.5	TC10.4	TC10.3	TC10.2	TC10.1	TC10.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BBH	PT10PU	PU10.7	PU10.6	PU10.5	PU10.4	PU10.3	PU10.2	PU10.1	PU10.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C0H	SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	*,*,*,*-,-,*,*
1C1H	SSPSTA0	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ...0	uu.. ...u	*,*,-,- -,-,-,*
1C2H	SSPBUF0	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1C3H	CFG0	-	-	-	-	-	GCRst	ENI2CT	ENI2C	0000 0000	.... .uuu	-,-,-,- -,*,*,*
1C4H	ACT0	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C5H	STA0	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C6H	CRG0	CRG[7:0]								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C7H	TOC0	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C8H	RDB0	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1C9H	TDB0	TDB0[7:1]							TDB0[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1CAH	SID0	SID0[7:1],The corresponding address of the 7-bit mode							SID0V[0]	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1CBH	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	*,*,*,*,-,-,*
1CCH	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r r,r,r,rw0
1CDH	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD	 0000	.... uuuu	-,-,-,- *,*,*,*
1CEH	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-,-,-,*,*,*,*,*
1CFH	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D0H	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D1H	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
1D2H	TB1Flag		PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	.000 0000	.uuu uuuu	-,r,r,r r,r,r,r
1D3H	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu	*,*,*,* *,rw1,*,*
1D4H	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1D5H	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
1D6H	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
1D7H	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D8H	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D9H	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1DAH	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1DBH	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1DCH	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1DDH	TB1CN2	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	uuuu uuuu
1DEH	DGCON1	-	-	-	DGRST	DGDiv[2:0]			DGEN	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1DFH	DGCON2	-	-	DGRP[5:0]						0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1E0H	DACCN0	DANS[1:0]		DAPS[1:0]		-	-	DALH	ENDA	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1E1H	DACCN1	DABIT[7:0]								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1E2H	FILTER	FreSpect		Frebit	ENSpect	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
080h ~ 0FFh		SRAM as 128Byte								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*
100h ~ 17Fh		SRAM as 128Byte								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*
200h ~ 2FFh		SRAM as 256Byte								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*
300h ~ 3FFh		SRAM as 256Byte, 300h~33Fh set for hardware sinewave first.								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*
400h ~ 4FFh		SRAM as 256Byte								uuuu uuuu	uuuu uuuu	*,*,*,*,*,*,*,*

Table 2-6 Data Memory List(continued)

3. Oscillator,Clock Sources and Power Managed Modes

HY17P series has two clock sources, HAO and LPO as shown in TableTable 3-1. Through clock sources controller register set-up, it helps to flexibly manage CPU and peripheral operating frequency. Moreover, it also appropriately adjusts IC's consumed power as to reach the purpose of energy economy.

Clock source control register：

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

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

OSCCN2 DLCD[1:0], ENXT, XTS[1:0], HAOM[1:0], ENHAO

Symbol	Frequency	Frequency source controller CLKCN[7:0] configuration		Instruction execution status	
		ENHAO	HAOM[1:0]	SLP	IDLE
HAO	-	1	00	STOP	oscillate
	4.9152MHz	1	01	STOP	oscillate
	9.8304MHz	1	10	STOP	oscillate
	9.8304MHz	1	11	STOP	oscillate
LPO	14KHz	After the chip is powered on, it starts to oscillate		STOP	oscillate

Table 3-1 Internal RC oscillator parameters,frequency controller configuration and Instruction execution status

3.1. Oscillator

3.1.1. HAO Oscillator

HAO is internal high speed RC oscillator ; the typical output frequency is 4.9152MHz/9.8304MHz.

When CPU of HY17P Series utilizes other oscillator as operating clock source,HAO oscillator can be turned off by setting ENHAO as <0>.

3.1.2. LPO Oscillator

LPO is internal low speed RC oscillator, the typical output frequency is 14.5KHz. it is mostly implemented in low speed and power efficient CPU operation and Watch Dog Timer clock source.

After executing Sleep instruction, LPO oscillator of HY17P Series products will be shut off and will oscillate automatically as the IC being awakened.

3.2. CPU and Peripheral Circuit Clock Sources

3.2.1. Clock Sources Configuration

Two sets of oscillator output (HS_CK、LS_CK) will firstly pass through pre-operating clock divider to initiate start/stop, switch and prescaling,then to CPU and other peripheral circuits. As shown in FigureFigure 3-1

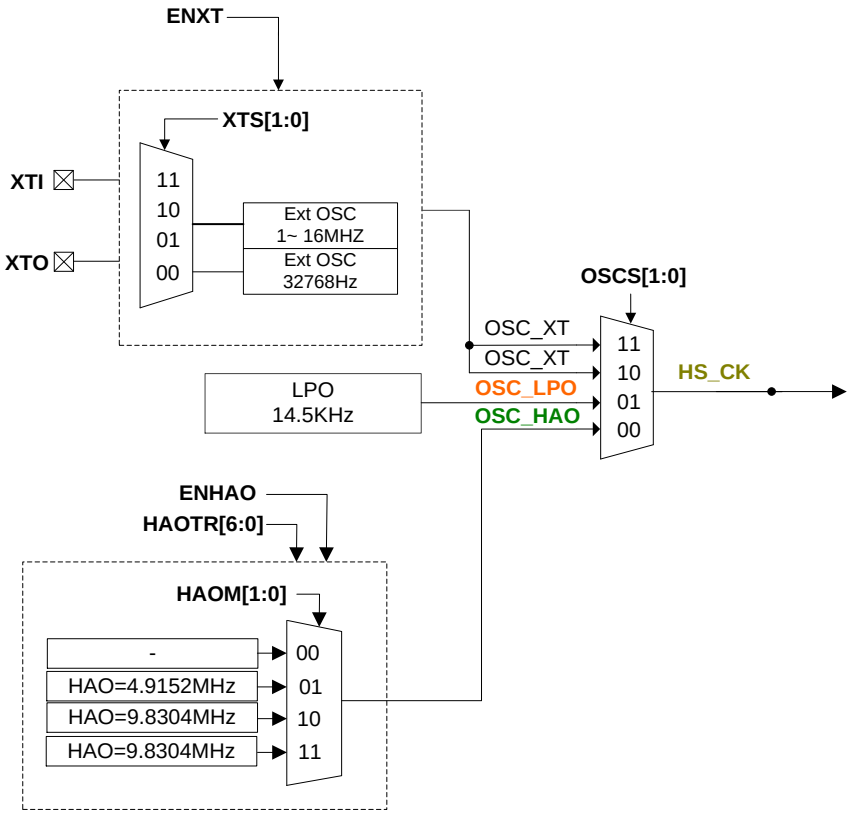


Figure 3-1 Pre-Operating Clock Divider

3.2.2. CPU Clock Source

The CPU has a variety of operating frequencies to choose from. The operating frequency can be selected from HS_CK or DHS_CK through CPUS.

The instruction operating frequency adopts 1/4 CPU_CK design, and divide the frequency into the frequency source of INTR_CK.

- When operating ΣΔADC, It is recommended that the CPU use the source after HS_CK frequency division as the operating frequency to get better performance.
- The relationship between CPU and peripheral operating clock,as shown in figureFigure 3-2
- The relationship between CPU_CK frequency and instruction execution cycle,as shown in tableTable 3-2

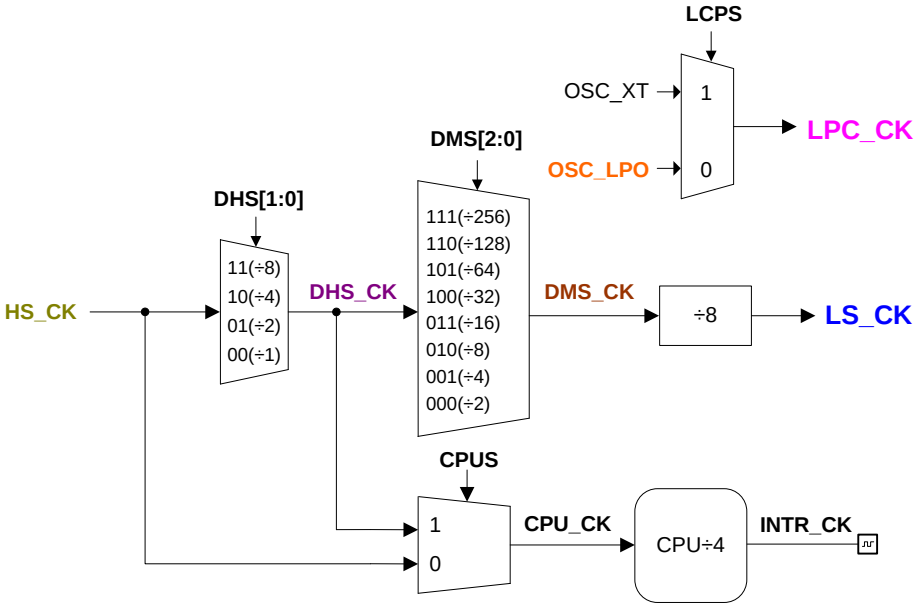


Figure 3-2 CPU and peripheral Operating clock

Operating clock / CPU_CK	CPU	Instruction	
	Frequency	Frequency	Cycle
9.8304MHz	9.8304MHz	2.4576MHz	0.4069us
4.9152MHz	4.9152MHz	1.2288MHz	0.8138us
32.768KHZ	32.768KHZ	16.384KHz	122.07us
14.5KHz	14.5KHz	3.625KHz	275.86us

Table 3-2 CPU_CK frequency and instruction execution cycle

3.2.3. CPU Peripheral Operating Clock Source

Clock source of HY17P6x series peripheral circuit are configured by different distribution controller and prescaler. The configuration will be fully illustrated in each peripheral unit; hence FigureFigure 3-3 only presents the peripheral clock sources configuration.

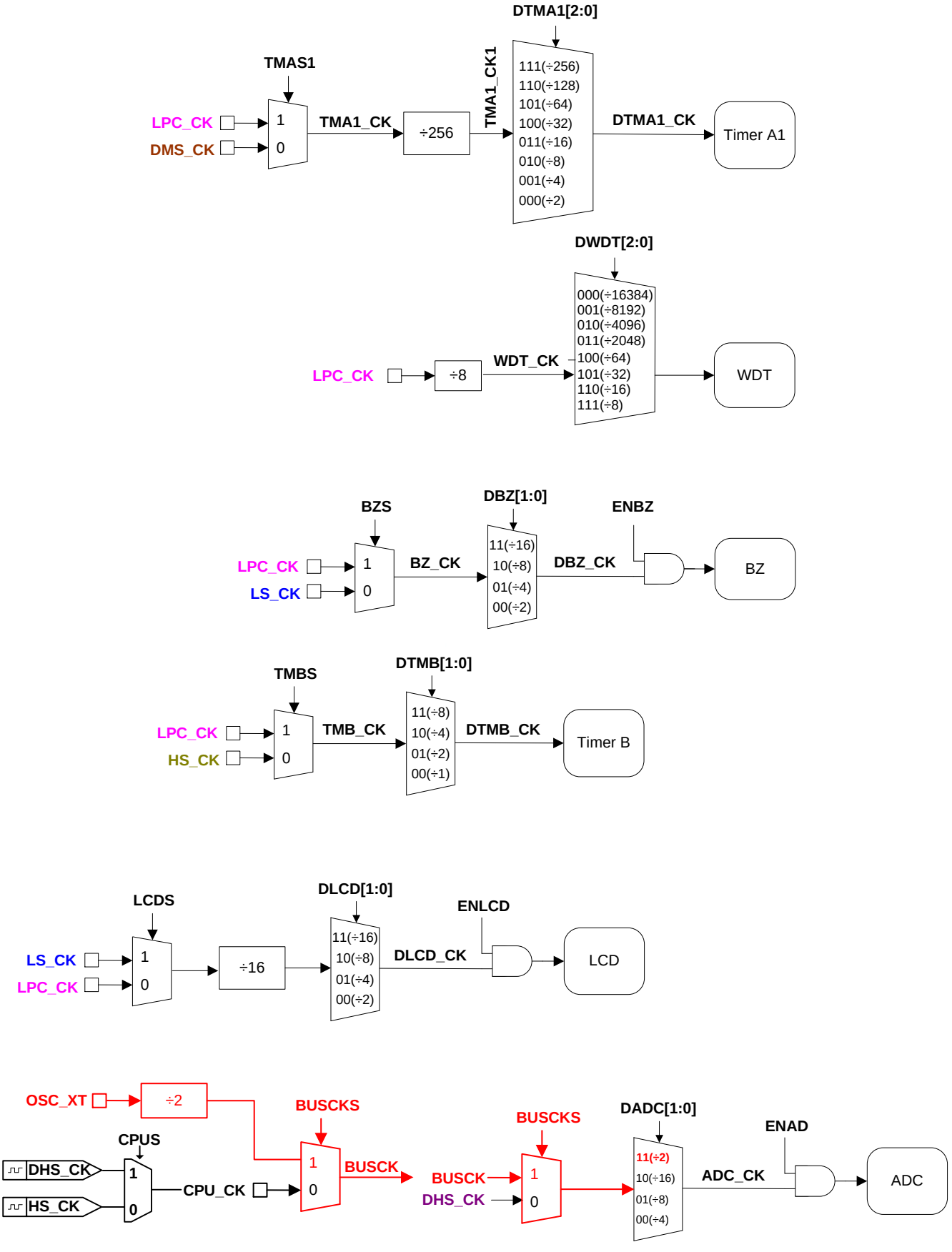


Figure 3-3 Configuration of Peripheral Operating Clock Source

3.1. Register Description-Operating 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**
032H	PWRCN								CSFON	0000 0000	uuuu u00u	*,*,*,* *,wr0,wr0,*
035H	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
036H	OSCCN1		LCPS	DADC[1:0]		DTMB[1:0]		TMBS	LCDS	0000 0000	uuuu uuu.	*,*,*,* *,*,*,-
037H	OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*,*,*,* *,*,*,r
038H	CSFCN0			HAOTR[5:0]						0	....u	-,-,-,- -,-,-,*

Table 3-3 Operating Clock Source Control Register

OSCCN0[7:0] : Chip operating frequency control register 0

Bit	Name	Description
Bit7~6	OSCS[1:0]	Frequency selector of HS_CK **OSCS[1:0]** / **Pre-scale** 00 / OSC_HAO 01 / OSC_LPO 10 / OSC_XT 11 / OSC_XT
Bit5~4	DHS[1:0]	Prescal selector of HS_CK **DHS[1:0]** / **Pre-scale** 00 / HS_CK ÷ 1 01 / HS_CK ÷ 2 10 / HS_CK ÷ 4 11 / HS_CK ÷ 8
Bit3~1	DMS[2:0]	Prescal selector of DMS_CK **DMS[2:0]** / **Pre-scale** / **DMS[2:0]** / **Pre-scale** 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
Bit0	CUPS	Frequency selector of CPU_CK <0>HS_CK <1>DHS_CK

OSCCN1[7:0] : Chip operating frequency control register 1

Bit	Name	Description
Bit6	LCPS	Frequency selector of LPC_CK <0>OSC_LPO <1>OSC_XT

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit5~4</td><td>DADC[1:0]</td><td>Prescal selector of ADC_CK
<table>
<tr><th>DADC[1:0]</th><th>Pre-scale</th></tr>
<tr><td>00</td><td>DHS_CK ÷ 4</td></tr>
<tr><td>01</td><td>DHS_CK ÷ 8</td></tr>
<tr><td>10</td><td>DHS_CK ÷ 16</td></tr>
<tr><td>11</td><td>DHS_CK ÷ 2(only for HY17P68/HY17P60B)</td></tr>
</table></td></tr>
<tr><td>Bit3~2</td><td>DTMB[1:0]</td><td>Prescal 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>LPC_CK</td></tr>
<tr><td>Bit0</td><td>LCDS</td><td>Frequency selector of LCD_CK
<0> LPC_CK
<1> LS_CK</td></tr>
</table>

OSCCN2[7:0] : Chip operating frequency control register 2

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~6</td><td>DLCD[1:0]</td><td>Prescal selector of DLCD_CK
Note: In LCD mode,DLCD is the frequency prescal Controller,but in LED mode is the LED duty selection controller
<table>
<tr><th>DLCD[1:0]</th><th>Pre-scale</th></tr>
<tr><td>00</td><td>LCD_CK ÷ 2</td></tr>
<tr><td>01</td><td>LCD_CK ÷ 4</td></tr>
<tr><td>10</td><td>LCD_CK ÷ 8</td></tr>
<tr><td>11</td><td>LCD_CK ÷ 16</td></tr>
</table></td></tr>
<tr><td>Bit5</td><td>ENXT</td><td>Control bit of starting external crystal/resonator
<0>Disable
<1>Enable</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit4~3</td><td>XTS[1:0]</td><td>Control bit of external oscillator frequency selection
<table>
<tr><th>XTS[1:0]</th><th>Frequency range of external oscillator</th></tr>
<tr><td>00</td><td>~ 32768Hz</td></tr>
<tr><td>01</td><td>~ 32768Hz</td></tr>
<tr><td>10</td><td>2M (low power)</td></tr>
<tr><td>11</td><td>2~ 16MHz</td></tr>
</table></td></tr>
<tr><td>Bit2~1</td><td>HAOM[1:0]</td><td>Control bit of internal oscillator HAO frequency selection
<table>
<tr><th>HAOM[1:0]</th><th>HAO oscillation frequency</th></tr>
<tr><td>00</td><td>reserve</td></tr>
<tr><td>01</td><td>4.9152MHz</td></tr>
<tr><td>10</td><td>9.8304MHz</td></tr>
<tr><td>11</td><td>9.8304MHz</td></tr>
</table></td></tr>
<tr><td>Bit0</td><td>ENHAO</td><td>Control bit of starting internal HAO
<0> Disable
<1> Enable</td></tr>
</table>

CSFCN0[7:0] Special control bit register 0

Bit	Name	Description
Bit6~0	HAOTR	HAO frequency center adjustment controller <0000000>Adjustment 34.00%(maximum) . <1000000> Center point 0.00% . <1111111>Adjustment -34%(minimum)

※ CSFCN0 can be operated by users in normal mode, this bit is protected,It is necessary to set CSFON[0] to 1 to modify this bit setting.

CSFCN1: Special control bit register 1

Bit	Name	Description
Bit7	BUSCKS	BUS frequency source selector(only for HY17P68/HY17P60B) <0> Frequency source selection CPU_CK <1> Frequency source selection OSC_XT

※ CSFCN1 can be operated by users in normal mode, this bit is protected,It is necessary to set CSFON[0] to 1 to modify this bit setting.

Reset event only BOR / RST will clear the register.

4. RESET

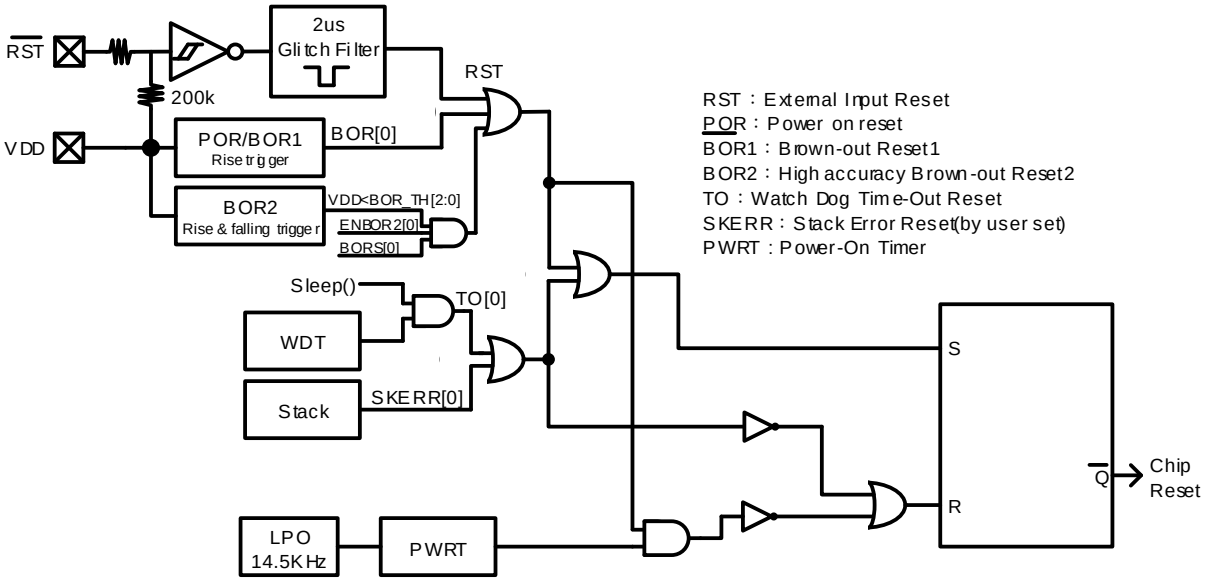


Figure 4-1 Reset Block Diagram

These reset events can be divided into hardware reset and software reset as depicted in Table Table 4-1. After CPU reset, program will start from 0x0000h.

Reset Type	Event	Symbol	Description
Hardware Reset	BOR RST	A-RESET	CPU restarts. It is until the end of internal oscillator counting, will CPU return to normal operating status.
Software Reset	WDT SKERR	I-RESET	Only partial registers will be cleared, CPU gets back to normal operating status soon.

Table 4-1 Reset Level

4.1. Reset Events Description

4.1.1. BOR Power Interference Reset

When CPU is interfered during power on process or by external power, it will enter into normal operating voltage from abnormal over-low operating voltage. If CPU cannot enter into reset status in over-low operating voltage, it may result in CPU crash and abnormal peripheral circuit operations. Thus, when BOR circuit detects that operating voltage is interfered and the voltage lowers than default value, reset signal will be aroused and the IC will enter into restart status. Until operating voltage returned to default value, will the reset signal be released and the IC enters into normal run mode.

4.1.2. BOR1 is power saving BOR

When BOR reset occurred, BOR flag of register PSTAT[7:0] will be configured as <1> to record the occurred event. BOR1 circuit of HY17P Series will produce about 0.2uA current consumption; there is no other program or configuration methods can shut it off.

4.1.3. BOR2 is precision BOR

BOR2 will be activated by POR/BOR1 by default, and the BOR2 function can be turned off by software. The off bit ENBOR2[0] can be operated by users in normal mode. This bit is protected and can be set effectively only when the switch is activated.

The trigger BOR2 behavior can be set to chip Reset or interrupt the BOR2IF flag.

The control bit BORS[0] can be operated by the user in normal mode. This bit is protected, and CSFON[0] needs to be activated to set it effectively.

4.1.4. SKERR Stack Error Reset

When reset signal is generated from program stack overflow/underflow, the IC will access into quick restart status. For recording the occurred events, SKERR flag of register, PSTAT[7:0] is set as <1> while SKERR stack error reset happened. Detailed instruction description will be elaborated on Memory chapter.

4.2. Status Registers

IC operating status is displayed in reset register,PSTAT[7:0]; the interrelation is indicated in TableTable 4-2

"0" : not happened,"1" : happened,"u" : unchanged,"-" : unused

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

Table 4-2 Interrelation of Reset Status Flags

4.2.1. Timing chart of reset state

The timing chart from the occurrence of the hardware reset signal until the chip enters the operating state,as shown in figureFigure 4-2

After different reset signals occur, the time when the chip enters the operating state

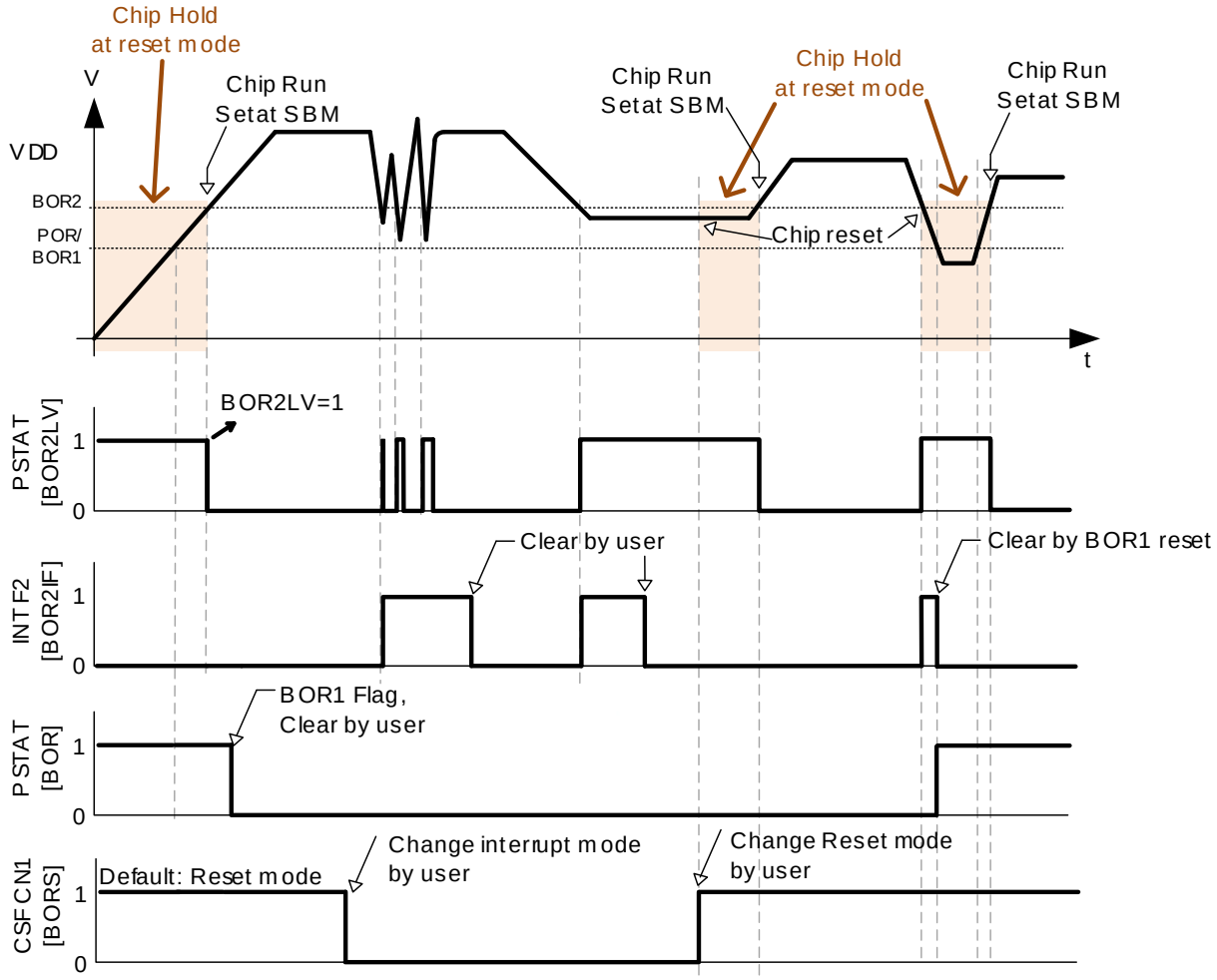


Figure 4-2 Reset and operation mode and status flag timing chart

*. After BOR2 is triggered, Power up count:

"-" : no definition

Reset Signal	Delay Time			Operating Status		
	Symbol	T1	T2	Run	Idle	Sleep
BOR	t_{RST}	T1 + T2		Valid	Valid	Valid
SKERR	-	-		Valid	Invalid	Invalid

Table 4 3 Interrelation of reset status delay time and operating status

1024 HAO+ 1024 LPO= 1024*(1/4.9152Mhz)+1024*(1/14.5Khz)= 70.829msec.

T1 => LPO (14.5KHz)

T2 => HAO (4.9152MHz)

4.3. Register Description-Reset Status

"-"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**
025H	INTE2								BOR2IE	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
028H	INTF2								BOR2IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
02CH	PSTAT	BOR	PD	TO	IDL	RST	SKERR	BOR2LV	GCRstIF	$000 $000	uu$u u$uu	rw0,rw0,rw0,rw0 rw0,rw0,r,rw0
032H	PWRCN								CSFON	1000 0000	1uuu u00u	*,*,*,*,*,wr0,wr0,*
038H	CSFCN0	SKRST								.1..	....	-,-,-,- -,-,-,-
039H	CSFCN1	-	-	-	BOR_TH[2:0]			BORS	ENBOR2	0.00 0011	0.uu uuuu	*,-,*,*,*,*,*,*

Table 4-3 Reset Register

INTE2/INTF2: Please refer to the Interrupt, Interrupt chapter

PSTAT: Status Register

Bit	Name	Description
Bit7	BOR	Power interference reset flag <0> Cleared through instruction <1> Set up <1> to operate BOR
Bit6	PD	Sleep mode flag <0> Cleared through BOR, RST or instruction <1> Set up <1> to execute SLEEP instruction
Bit5	TO	WDT operation mode count overflow flag <0>Not happened <1>Happened；Cleared through BOR, RST or instruction
Bit4	IDL	Idle Mode flag <0> Cleared through BOR, RST or instruction <1> Set up <1> to execute IDLE instruction
Bit3	RST	External RST pin low level reset event flag <0> Not happened <1> Happened；Cleared through BOR or instruction
Bit2	SKERR	Stack error reset flag <0> Cleared through BOR, RST or instruction <1> Stack error set up <1>
Bit1	BOR2LV	BOR2 Status flag <0> VDD voltage >BOR_TH[2:0] <1> VDD voltage <=BOR_TH[2:0]
Bit0	GCRstIF	I^2C Reset command flag <0> Not happened <1> Happened

CSFCN0: Special control bit register 0

Bit	Name	Description
Bit7	SKRST	Stack error reset controller <0> Turn off error reset chip <1> Enable error reset chip

※ CSFCN0 can be operated by users in normal mode, this bit is protected,It is necessary to set CSFON[0] to 1 to modify this bit setting.

CSFCN1: Special control bit register 1

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit4~2</td><td>BOR_TH[2:0]</td><td>BOR2 detection voltage setting
<table>
<tr><th>BOR_TH[2:0]</th><th>BOR2 voltage</th><th>Description</th></tr>
<tr><td>000</td><td>1.7V</td><td>Chip default value at power up</td></tr>
<tr><td>001</td><td>2.0V</td><td>When the battery uses 1.5V*2, the battery is 2V (=1V*2) means low voltage</td></tr>
<tr><td>010</td><td>2.2V</td><td></td></tr>
<tr><td>011</td><td>2.5V</td><td>for VDDA=2.4V mode, VDD>=2.45V</td></tr>
<tr><td>100</td><td>2.75V</td><td>When the battery uses 1.5V*3, the battery is 2.7V (=0.9V*3) means low voltage</td></tr>
<tr><td>101</td><td>3.0V</td><td>When the battery uses 1.5V*3, the battery is 3V (=1V*3) means low voltage</td></tr>
<tr><td>110</td><td>3.65V</td><td>When the battery uses 1.5V*4, the battery is 3.6V (=0.9V*4) means low voltage</td></tr>
<tr><td>111</td><td>4.0V</td><td>When the battery uses 1.5V*4, the battery is 4V (=1V*4) means low voltage</td></tr>
</table></td></tr>
<tr><td>Bit1</td><td>BORS</td><td>BOR2 behavior setting
<0> BOR2 is an interrupt wake up function. When BOR2IE=1 and BOR2IF=1,an interrupt event is generated.
<1> BOR2 is the chip reset function. When BOR2IF=0, the chip will be reset. The chip power on default value.</td></tr>
<tr><td>Bit0</td><td>ENBOR2</td><td>BOR2 enable and disable the controller
<0> Disable
<0> Enable</td></tr>
</table>

※ CSFCN1 can be operated by users in normal mode, this bit is protected,It is necessary to set CSFON[0] to 1 to modify this bit setting.

Reset event only BOR / RST will clear the register.

5. Interrupt

Interrupt involves interrupt starting controller, INTE and interrupt event flag, INTF. If interrupt event occurs while interrupt service established, PC will move to interrupt vector address, 0x0004h of program memory to execute interrupt service program.

Interrupt Control Register：

INTE0	GIE, TA1CIE, ADIE, WDTIE, TB1IE, CTIE, E1IE, E0IE
INTE1	TA1IE, SPIIE, TXIE, RCIE, I2CERIE, I2CIE, E3IE, E2IE
INTE2	MFCIE, CMPOIE, CMPHOIE, CMPLOIE, CTBOVE, RMSIE, LPFIE, BOR2IE
INTF0	TA1CIF, ADIF, WDTIF, TB1IF, CTF, E1IF, E0IF
INTF1	TA1IF, SPIIF, TXIF, RCIF, I2CERIF, I2CIF, E3IF, E2IF
INTF2	MFCIF, CMPOIF, CMPHOIF, CMPLOIF, CTBOV, RMSF, LPFF, BOR2IF
PT1M1	INTEG1[1:0], INTEG0[1:0]
PT1INT	INTEG7, INTEG6, INTEG5, INTEG4, INTEG3, INTEG2
PT1INTE	INTE1.7, INTE1.6, INTE1.5, INTE1.4
PT1INTF	INTF1.7, INTF1.6, INTF1.5, INTF1.4

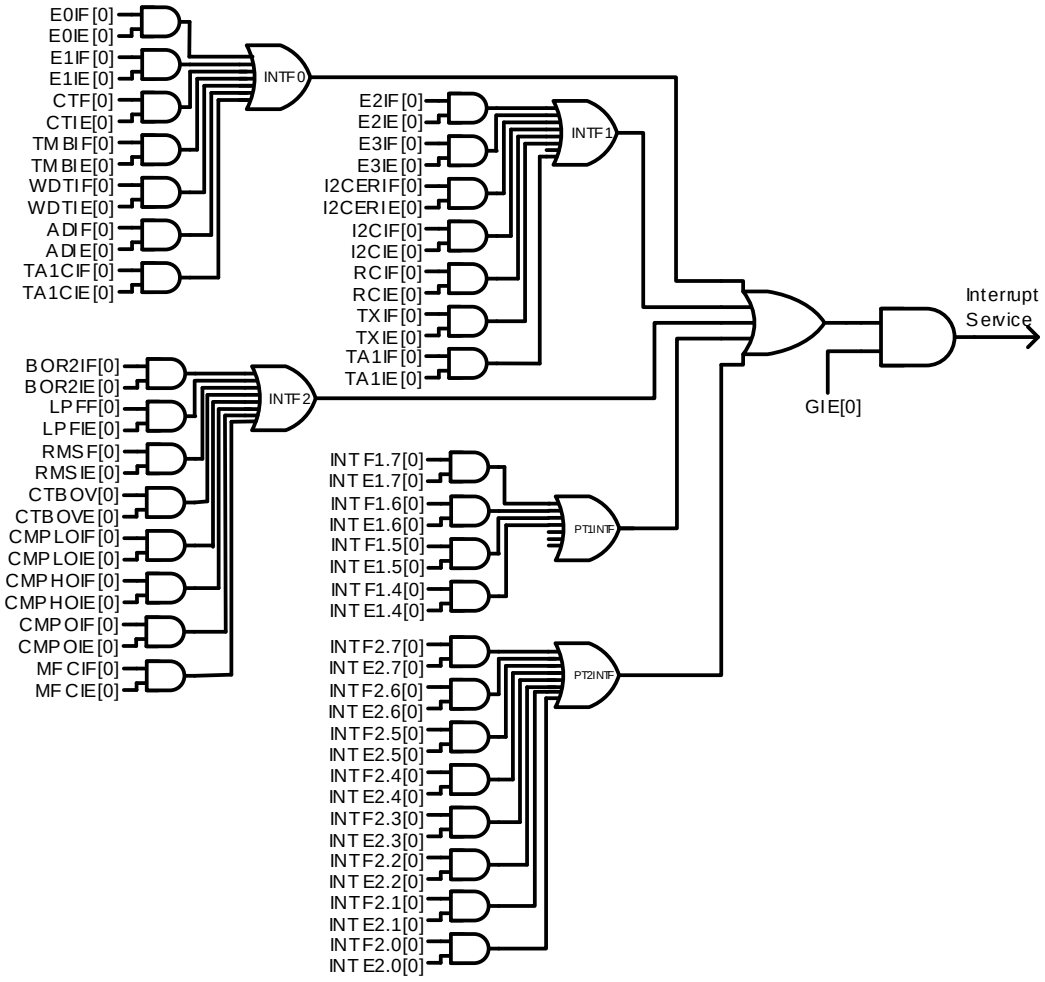


Figure 5-1 Interrupt vector block diagram

There are two layers of interrupt service event controller. The top layer is global interrupt enable, GIE and the bottom layer is the starting control bit for interrupt event.

- To initiate interrupt event, set up the control bit that relatives to interrupt control register, INTEx [7:0] to be <1>. Conversely, set up< 0> to close the interrupt event.
- To initiate interrupt service, set up the control bit, GIE that relatives to interrupt control register INTE0 [7:0] to be <1>. Conversely, set up< 0> to close the interrupt service.

GIE will automatically be set up as <0> when it enters into interrupt service vector. After interrupt service program has been completely executed and would like to return to interrupt occurred address, interrupt return instruction, RETI can be applied directly. At this time, GIE will automatically be set up as <1>. Moreover, if return instruction RET is conducted, GIE status will remain as 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	CTIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024H	INTE1	TA1IE	SPIIE	TXIE	RCIE	I2CERIE	I2CIE	E3IE	E2IE	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
025H	INTE2	MFCIE	CMPOIE	CMPHOIE	CMPLOIE	CTBOVE	RMSIE	LPFIE	BOR2IE	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
026H	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	CTF	E1IF	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
027H	INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
028H	INTF2	MFCIF	CMPOIF	CMPHOIF	CMPLOIF	CTBOV	RMSF	LPFF	BOR2IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06FH	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
070H	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
071H	PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
072H	PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
077H	PT2INT	INTG2.7	INTG2.6	INTG2.5	INTG2.4	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
078H	PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	
079H	PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 5-1 Interrupt Registers

INTE0: Interrupt Starting Control Register 0

Bit	Name	Description
Bit7	GIE	Global interrupt enable <0> Shut off <1> Start
Bit6	TA1CIE	Timer-A1 Compare event starting controller <0> Shut off <1> Start
Bit5	ADIE	ADC interrupt event starting controller <0> Shut off <1> Start
Bit4	WDTIE	Watch Dog interrupt event starting controller <0> Shut off <1> Start
Bit3	TB1IE	Timer-B interrupt event starting controller <0> Shut off <1> Start
Bit2	CTIE	Frequency Counter interrupt event starting controller <0> Shut off <1> Start
Bit1	E1IE	Interrupt event starting controller of input pin 1 <0> Shut off <1> Start (external input pin, PT1.1)
Bit0	E0IE	Interrupt event starting controller of input pin 0 <0> Shut off <1> Start (external input pin, PT1.0)

INTE1: Interrupt Starting Control Register 1

Bit	Name	Description
Bit7	TA1IE	Timer-A1 interrupt event starting controller <0> Shut off <1> Start
Bit6	SPIIE	SPI interrupt event starting controller(Only For HY17P68) <0> Shut off <1> Start
Bit5	TXIE	TX interrupt event starting controller <0> Shut off <1> Start
Bit4	RCIE	RC interrupt event starting controller <0> Shut off <1> Start
Bit3	I2CERIE	Peripheral I^2C error interrupt vector service controller <0> Disable I^2C error interrupt vector service <1> Enable I^2C error interrupt vector service
Bit2	I2CIE	Peripheral I^2C interrupt vector service controller <0> Disable I^2C interrupt vector service <1> Enable I^2C interrupt vector service
Bit1	E3IE	Interrupt event starting controller of input pin 3 <0> Shut off <1> Start (external input pin, PT1.3)
Bit0	E2IE	Interrupt event starting controller of input pin 2 <0> Shut off <1> Start (external input pin, PT1.2)

INTE2: Interrupt Starting Control Register 2

Bit	Name	Description
Bit7	MFCIE	MFC interrupt event starting controller <0> Shut off <1> Start
Bit6	CMPOIE	CMPO interrupt event starting controller (Only For HY17P68) <0> Shut off <1> Start
Bit5	CMPHOIE	CMPHO interrupt event starting controller (Only For HY17P68) <0> Shut off <1> Start

Bit	Name	Description
Bit4	CMPLOIE	CMPLO interrupt event starting controller (Only For HY17P68) <0> Shut off <1> Start
Bit3	CTBOVE	CTB Over Flow interrupt event starting controller <0> Shut off <1> Start
Bit2	RMSIE	RMS Converter interrupt event starting controller <0> Shut off <1> Start
Bit1	LPFIE	Low Pass Filter interrupt event starting controller <0> Shut off <1> Start
Bit0	BOR2IE	BOR2 interrupt event starting controller <0> Shut off <1> Start

INTF0: Interrupt Event Flag Register 0

Bit	Name	Description
Bit6	TA1CIF	Timer-A1 interrupt Compare event flag <0> Not happened <1> Happened
Bit5	ADIF	ADC interrupt event flag <0> Not happened <1> Happened
Bit4	WDTIF	Watch Dog interrupt event flag <0> Not happened <1> Happened
Bit3	TB1IF	Timer-B interrupt event flag <0> Not happened <1> Happened
Bit2	CTF	Frequency Counter interrupt event flag <0> Not happened <1> Happened
Bit1	E1IF	Interrupt event flag of input pin 1 <0> Not happened <1> Happened (External input pin, PT1.1)

Bit	Name	Description
Bit0	E0IF	Interrupt event flag of input pin 0 <0> Not happened <1> Happened (External input pin, PT1.0)

INTF1: Interrupt Event Flag Register 1

Bit	Name	Description
Bit7	TA1IF	Timer-A1 interrupt event flag <0> Not happened <1> Happened
Bit6	SPIIF	SPI interrupt event flag (Only For HY17P68) <0> Not happened <1> Happened
Bit5	TXIF	TX interrupt event flag <0> Not happened <1> Happened
Bit4	RCIF	RC interrupt event flag <0> Not happened <1> Happened
Bit3	I2CERIF	Peripheral I^2C error interrupt event flag <0> Not happened <1> Happened
Bit2	I2CIF	Peripheral I^2C interrupt event flag <0> Not happened <1> Happened
Bit1	E3IF	Interrupt event flag of input pin 3 <0> Not happened <1> Happened (External input pin, PT1.3)
Bit0	E2IF	Interrupt event flag of input pin 2 <0> Not happened <1> Happened (External input pin, PT1.2)

INTF2: Interrupt Event Flag Register 2

Bit	Name	Description
Bit7	MFCIF	MFC State event flag <0> Not happened <1> Happened (rising edge 0→1 change)

Bit	Name	Description
Bit6	CMPOIF	CMPO State event flag (Only For HY17P68) <0> Not happened <1> Happened (rising edge 0→1 change)
Bit5	CMPHOIF	CMPHO State event flag (Only For HY17P68) <0> Not happened <1> Happened (rising edge 0→1 change)
Bit4	CMPLOIF	CMPLO State event flag (Only For HY17P68) <0> Not happened <1> Happened (rising edge 0→1 change)
Bit3	CTBOV	CTB State event flag <0> Not happened <1> When CTB[23:0] Over Flow,it will be set to 1
Bit2	RMSF	RMS Converter interrupt event flag <0> Not happened <1> Happened
Bit1	LPFF	Low Pass Filter interrupt event flag <0> Not happened <1> Happened
Bit0	BOR2IF	BOR2 interrupt event flag <0> Not happened <1> Happened

PT1M1: Digital Output Mode Selection Register 1

Bit	Name	Description
Bit3~2	INTEG1[1:0]	PT1.x Interrupt signal generated conditions (0≦x≦1)
Bit1~0	INTEG0[1:0]	

INTEGx[1:0]	Interrupt signal generated conditions
00	Falling edge (1→0)
01	Rising edge (0→1)
10	Potential conversion (0→1 or 1→0)
11	Potential conversion (0→1 or 1→0)

PT1INT: PT1 Interrupt Signal Generated Conditions

Bit	Name	Description
Bit7~2	INTEG.x	Interrupt event starting controller (2≦x≦7) <0> Falling edge (1→0)。(Default) <1> Rising edge (0→1)。

PT1INTE: PT1 I/O Interrupt Event Starting Controller

Bit	Name	Description
Bit7~4	INTE1.x	Interrupt Event Starting Controller (4≦x≦7) <0>Disable (Default) <1>Enable

PT1INTF: PT1 I/O Interrupt Event Flag

Bit	Name	Description
Bit7~4	INTF1.x	Interrupt Event Flag (4≦x≦7) <0>Disable (Default) <1>Enable

PT2INT: PT2 I/O Interrupt Signal Generated Conditions

Bit	Name	Description
Bit7~0	INTG2.x	Interrupt signal generated conditions (0≦x≦7) <0> Falling edge (1→0)。(Default) <1> Rising edge (0→1)。

PT2INTE: PT2 I/O Interrupt Event Starting Controller

Bit	Name	Description
Bit7~0	INTE2.x	Interrupt Event Starting Controller (0≦x≦7) <0>Disable (Default) <1>Enable

PT2INTF: PT2 I/O Interrupt Event Flag

Bit	Name	Description
Bit7~0	INTF2.x	Interrupt Event Flag (0≦x≦7) <0>Disable (Default) <1>Enable

6. Hardware Multiplier

H08D instruction set has 8x8 hardware multiplier processing instruction "MULF and MULL". The operation outcome of 8x8 hardware multiplier will store at multiplier register PRODH[7:0] and PRODL[7:0], and will not change any sign of PSTAT[7:0] status register. Users must be cautioned that PRODH[7:0] and PRODL[7:0] are read only registers.

Hardware multiplier can conduct signed value and unsigned value operation, as ExampleExample 6-1 and ExampleExample 6-2 illustrates.

```
Example 1：V1 x V2 = V
MVL     V1
MVF     BUF0,1,0    ; Put V1 value in register BUF0 of memory Bank 0
MVL     V2          ; Put V2 value to register W
MULF    BUF0,0      ; Execute V1 x V2 and place the result to PRODH/PRODL
```

Example 6-1 Unsigned Value Operation

```
Example 2：N1 x N2 = N ,s=7,B
MVL     N1          ; Put N1 value to register W
MVF     BUF0,1,0    ; Put N1 value in register BUF0 of memory Bank 0
MVL     N2          ; Put N2 value to register W
MVF     BUF1,1,0    ; Put N2 value to register BUF1
MULF    BUF0,0      ; Execute V1 x V2 and place the result to PRODH/L
MVFF    PRODH,SWP   ; Put register PRODH value to register SWP
BTSZ    BUF0,s      ; Judge N1, if it is negative then
SUBF    SWP,1,0     ; Place SWP – N2 to register SWP
MVF     BUF0,0,0    ; Put N1 value to register W
BTSZ    BUF1,s      ; Judge N2, if it is negative then
SUBF    SWP,1,0     ; Place SWP – N1 to SWP. After operation, N = SWP/PRODL
; ------------------------------------------------------------------------------------
; N1=07Fh,N2=0FFh, after multiplier operation, obtained PRODH/L = 7E81h
; Determine whether N1 is minus, if so, then set PRODH – N2
; Determine whether N2 is minus, if so, then set PRODH – N1
; After operation, signed N value will be acquired.
; 7Fh x FFh     = 7Fh x ( 0FFh – 100h )
;               = 7Fh x 0FFh – 7Fh x 100h
;               = 7E81h – 7F00h
;               = FF81h
```

Example 6-2 Signed Value Operation

7. Input/Output Port,I/O

Each pin of the input/output port is a port, which can be used as a digital input and output channel.Every port is controlled by a set of registers.

I/O Related Registers：

PT	PT1[7:0], PT2[7:0], PT3[7:0], PT4[4:0], PT6[7:4], PT7[7:0], PT8[7:0], PT9[7:0], PT10[7:0], PT11[3:0]
PT1IN	PT1IN[7:0], PT2IN[7:0], PT3IN[7:0], PT4IN[4:0], PT6IN[7:4], PT7IN[7:0], PT8IN[7:0], PT9IN[7:0], PT10IN[7:0], PT11IN[3:0]
TRISC	TC1[7:0], TC2[7:0], TC3[7:0], TC4[4:0], TC6[7:4], TC7[7:0], TC8[7:0], TC9[7:0], TC10[7:0], TC11[3:0],
PTPU	PU1[7:0], PU2[7:0], PU3[7:0], PU4[4:0], PU6[7:4], PU7[7:0], PU8[7:0], PU9[7:0], PU10[7:0], PU11[3:0]

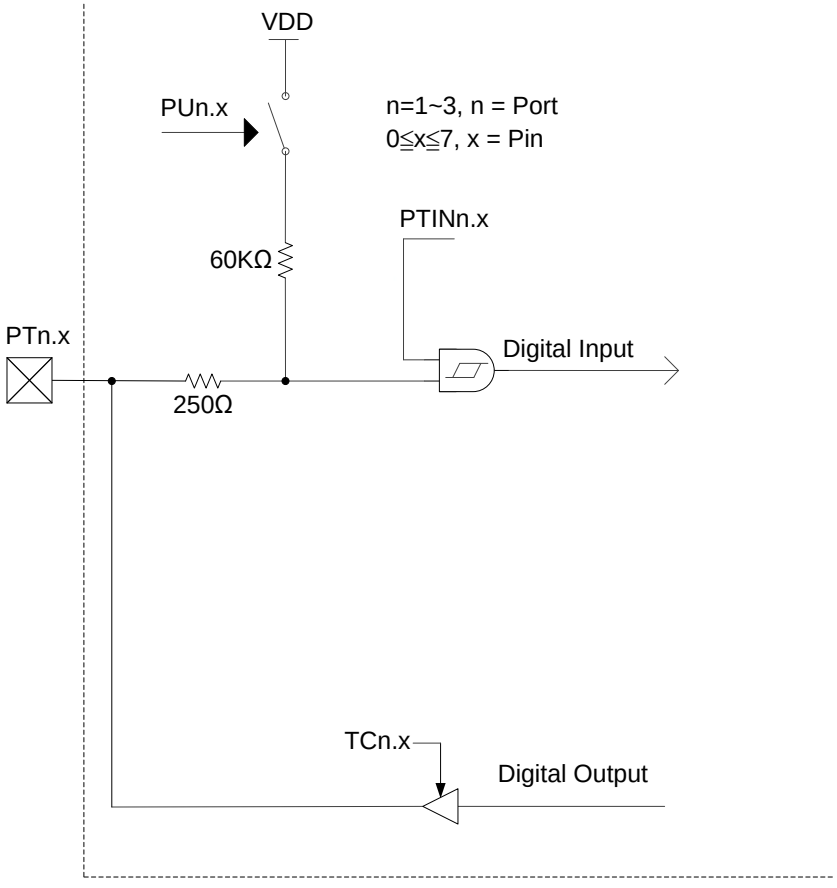


Figure 7-1 I/O PORT1~4 architecture block diagram

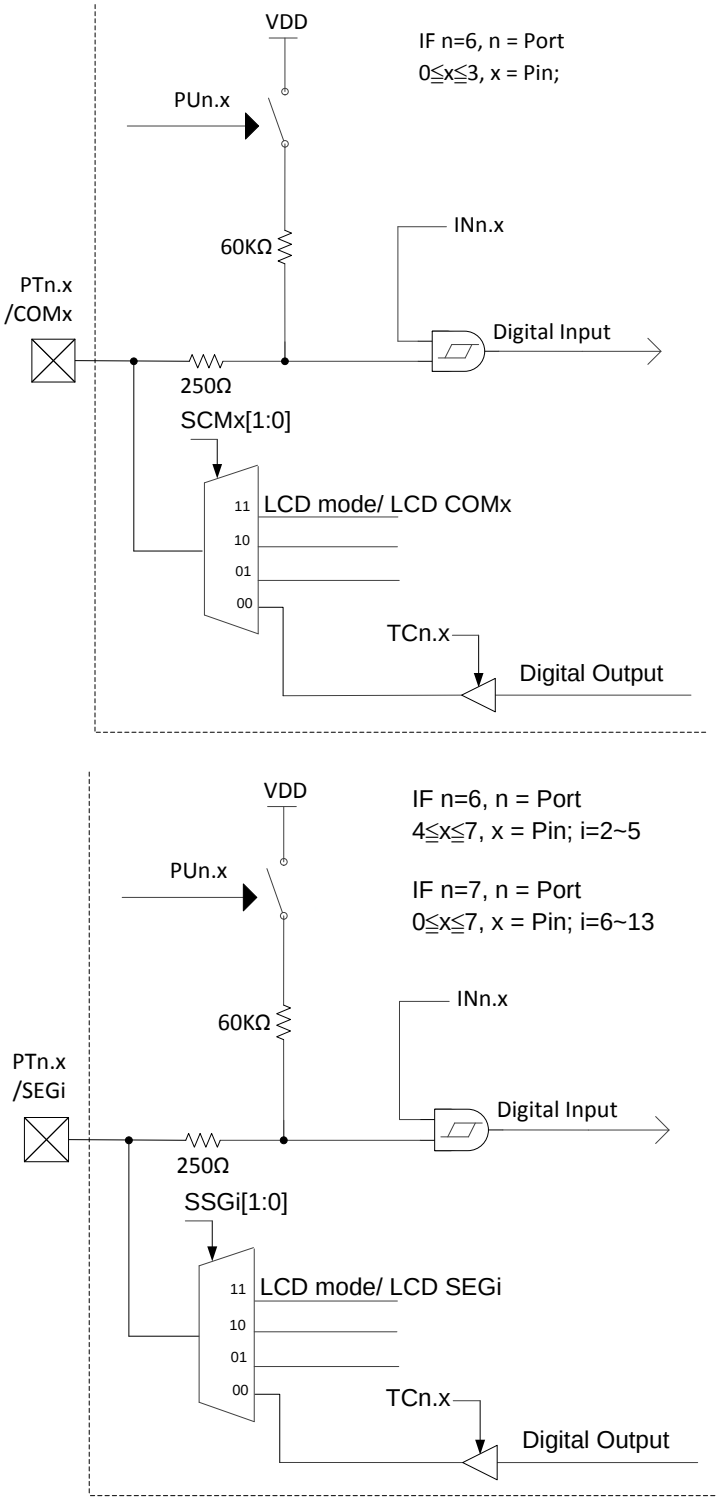


Figure 7-2 I/O PORT6~7 architecture block diagram

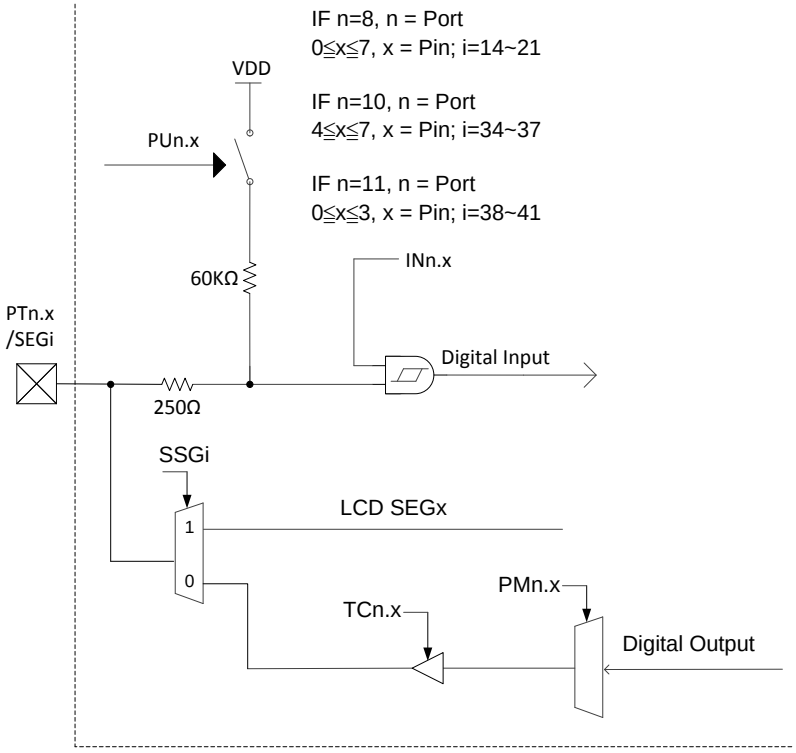


Figure 7-3 I/O PORT8,10,11 architecture block diagram

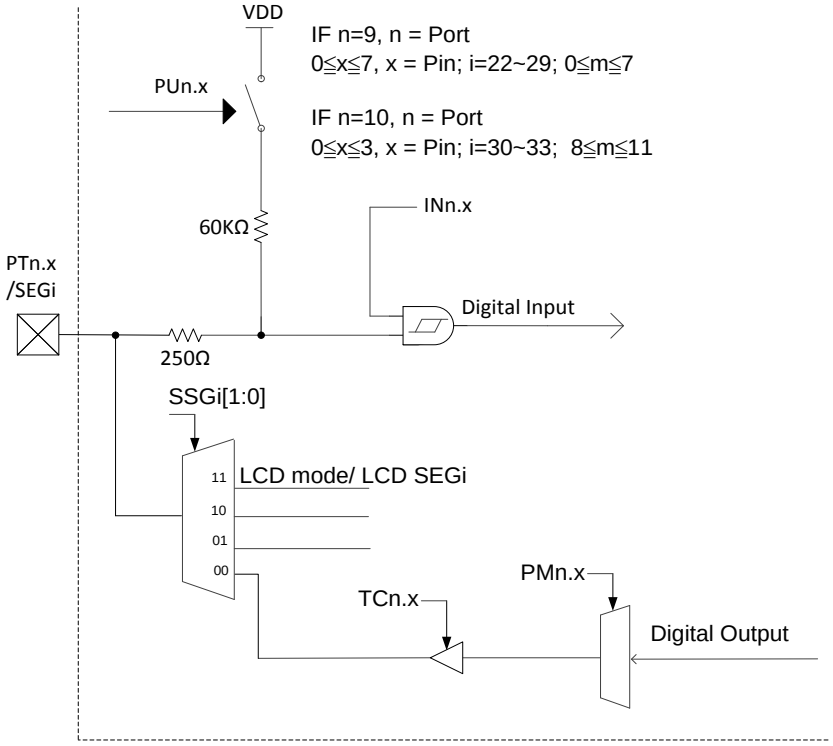


Figure 7-4 I/O PORT9~10 architecture block diagram

7.1. PORT Related Register Introduction

PORT mainly offers digital or analog signal I/O pin.

7.1.1. PTEG Interrupt Signal Generated Conditions

Diverse I/O external input potential produces different interrupt signals. Potential change conditions can be separated into rising edge (0→1) change, falling edge (1→0) change and potential conversion (0→1 or 1→0) change.

7.1.2. PTPU Pull-Up Resistor Control Register

Configure the register as <1>; the I/O is enabled with the pull-up resistor function. Configure the register set <0>, the function is disabled. Before the IC gets into sleep mode, if I/O is configured as digital input status, the external circuit connected way will cause I/O floating phenomenon. At this time, the pull-up resistor may be enabled as to avoid current leakage.

7.1.3. TC Input/Output Control Register

Selecting I/O to be input or output, set <1> and the I/O is in output status, set <0> and the I/O is in input status. If I/O is configured as input status, an explicit input potential must be made once the IC enters into sleep mode so as to avoid IC leakage status.

7.1.4. PTIO Status Control Register

When I/O is set as input, in corresponding register address, current I/O status can be read. If the value is 1, the I/O inputs high potential, if the value is 0 and the I/O inputs low potential.

When I/O is set as output, the correlative register site can control output status. If <1> is set, the I/O outputs high potential. If the value is set as <0>, the I/O outputs low potential.

7.2. Buzzer

BZ can produce several different frequencies to drive external buzzer. The operating frequency source is selected by BZS, BZ operating frequency prescaler, DBZ[2:0] can set diverse output frequency.

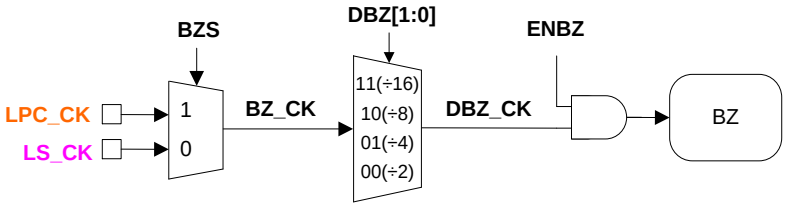


Figure 7-5 BZ Block Diagram

7.3. 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
06BH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
06CH	PT1IN	IN1.7	IN1.6	IN1.5	IN1.4	IN1.3	IN1.2	IN1.1	IN1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06DH	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06EH	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
06FH	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
070H	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
071H	PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
072H	PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
073H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
074H	PT2IN	IN2.7	IN2.6	IN2.5	IN2.4	IN2.3	IN2.2	IN2.1	IN2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
075H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
076H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
077H	PT2INT	INTG2.7	INTG2.6	INTG2.5	INTG2.4	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
078H	PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	
079H	PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
19FH	PT3	PT3.7	PT3.6	PT3.5	PT3.4	PT3.3	PT3.2	PT3.1	PT3.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A0H	PT3IN	IN3.7	IN3.6	IN3.5	IN3.4	IN3.3	IN3.2	IN3.1	IN3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A1H	TRISC3	TC3.7	TC3.6	TC3.5	TC3.4	TC3.3	TC3.2	TC3.1	TC3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A2H	PT3PU	PU3.7	PU3.6	PU3.5	PU3.4	PU3.3	PU3.2	PU3.1	PU3.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A3H	PT3M1	PM3.7[0] ENPCMPO	-	-	-	-	-	-	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A4H	PT4	-	-	-	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A5H	PT4IN	-	-	-	IN4.4	IN4.3	IN4.2	IN4.1	IN4.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A6H	TRISC4	-	-	-	TC4.4	TC4.3	TC4.2	TC4.1	TC4.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A7H	PT4PU	-	-	-	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	xxx0 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1A8H	PT6	PT6.7	PT6.6	PT6.5	PT6.4	-	-	-	-	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1A9H	PT6IN	IN6.7	IN6.6	IN6.5	IN6.4	-	-	-	-	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1AAH	TRISC6	TC6.7	TC6.6	TC6.5	TC6.4	-	-	-	-	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ABH	PT6PU	PU6.7	PU6.6	PU6.5	PU6.4	-	-	-	-	0000 xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ACH	PT7	PT7.7	PT7.6	PT7.5	PT7.4	PT7.3	PT7.2	PT7.1	PT7.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1ADH	PT7IN	IN7.7	IN7.6	IN7.5	IN7.4	IN7.3	IN7.2	IN7.1	IN7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1AEH	TRISC7	TC7.7	TC7.6	TC7.5	TC7.4	TC7.3	TC7.2	TC7.1	TC7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1AFH	PT7PU	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B0H	PT8	PT8.7	PT8.6	PT8.5	PT8.4	PT8.3	PT8.2	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B1H	PT8IN	IN8.7	IN8.6	IN8.5	IN8.4	IN8.3	IN8.2	IN8.1	IN8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B2H	TRISC8	TC8.7	TC8.6	TC8.5	TC8.4	TC8.3	TC8.2	TC8.1	TC8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B3H	PT8PU	PU8.7	PU8.6	PU8.5	PU8.4	PU8.3	PU8.2	PU8.1	PU8.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B4H	PT9	PT9.7	PT9.6	PT9.5	PT9.4	PT9.3	PT9.2	PT9.1	PT9.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B5H	PT9IN	IN9.7	IN9.6	IN9.5	IN9.4	IN9.3	IN9.2	IN9.1	IN9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B6H	TRISC9	TC9.7	TC9.6	TC9.5	TC9.4	TC9.3	TC9.2	TC9.1	TC9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B7H	PT9PU	PU9.7	PU9.6	PU9.5	PU9.4	PU9.3	PU9.2	PU9.1	PU9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1B8H	PT10	PT10.7	PT10.6	PT10.5	PT10.4	PT10.3	PT10.2	PT10.1	PT10.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1B9H	PT9IN	IN9.7	IN9.6	IN9.5	IN9.4	IN9.3	IN9.2	IN9.1	IN9.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BAH	TRISC10	TC10.7	TC10.6	TC10.5	TC10.4	TC10.3	TC10.2	TC10.1	TC10.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BBH	PT10PU	PU10.7	PU10.6	PU10.5	PU10.4	PU10.3	PU10.2	PU10.1	PU10.0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BCH	PT11	-	-	-	-	PT11.3	PT11.2	PT11.1	PT11.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1BDH	PT11IN	-	-	-	-	IN11.3	IN11.2	IN11.1	IN1.0	xxxx 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BEH	TRISC11	-	-	-	-	TC11.3	TC11.2	TC11.1	TC11.0	xxxx 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1BFH	PT11PU	-	-	-	-	PU11.3	PU11.2	PU11.1	PU11.0	xxxx 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 7-1 PORT Control Register

INTE0/INTF0: Please refer to Interrupt chapter

INTE0/INTE1/PT1INTE/PT2INTE: Please refer to Interrupt chapter

/INTF0/INTF1/PT1INTF/PT2INTF: Please refer to Interrupt chapter

WDTCN[7:0] WDT Control Register

Bit	Name	Description
Bit7	ENBZ	Buzzer enable and disable controller <0>Disable <1>Enable
Bit6	BZS	Buzzer operating frequency selector <0>LS_CK <1>LPC_CK
Bit5~4	DBZ[1:0]	Buzzer output frequency controller

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

PT1: PT1 Status Control Register

Bit	Name	Description
Bit7~0	PT1.x	PT1.x status flag and controller 0≦x≦7

PT1.x	When TC1.x is set to <0>	When TC1.x is set to <1>
0	PT1.x Input is low(L)	PT1.x Output is low (L)
1	PT1.x Input is high(H)	PT1.x Output is high(H)

PT1IN: PT1 Input Control Register

Bit	Name	Description
Bit7~0	IN1.x	PT1.x digital input controller，0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN1.x to change the PT1.x state normally.

TRISC1: PT1 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC1.x	PT1.x input/output control bit，0≦x≦7 <0> Disable the output function,the pin is only for the input function. <1> Enable the output function,the pin is Output/input function

PT1PU: PT1 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU1.x	PT1.x pull-up resistor pin control bit，0≦x≦7 <0>Disable <1>Enable

PT1M1: Digital Output Mode Selection Register 1

Bit	Name	Description
Bit3~2	INTEG1[1:0]	PT1.x Interrupt signal generated conditions (0≦x≦1)
Bit1~0	INTEG0[1:0]	

INTEGx[1:0]	Interrupt signal generated conditions
00	Falling edge (1→0)
01	Rising edge (0→1)
10	Potential conversion (0→1 or 1→0)
11	Potential conversion (0→1 or 1→0)

PT1INT: PT1 Interrupt Signal Generated Conditions

Bit	Name	Description
Bit7~2	INTEG.x	Interrupt event starting controller (2≦x≦7) <0> Falling edge (1→0) (Default) <1> Rising edge (0→1)

PT2: PT2 Status Control Register

Bit	Name	Description
Bit7~0	PT2.x	PT2.x status flag and controller，0≦x≦7

PT2.x	When TC2.x is set to <0>	When TC2.x is set to <1>
0	PT2.x Input is low(L)	PT2.x Output is low (L)
1	PT2.x Input is high(H)	PT2.x Output is high(H)

PT2IN: PT2 Input Control Register

Bit	Name	Description
Bit7~0	IN2.x	PT2.x pin digital input controller，0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN2.x to change the PT2.x state normally.

TRISC2: PT2 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC2.x	PT2.x input/output control bit · 0≦x≦7 <0> Disable the output function,the pin is only for the input function. <1> Enable the output function,the pin is Output/input function

PT2PU: PT2 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU2.x	PT2.x pull-up resistor pin control bit · 0≦x≦7 <0>Disable <1>Enable

PT2INT: I/O Interrupt Signal Generated Conditions

Bit	Name	Description
Bit7~0	INTEGx	Interrupt event starting controller (0≦x≦7) <0> Falling edge (1→0) (Default) <1> Rising edge (0→1)

PT3: PT3 Status Control Register

Bit	Name	Description
Bit7~0	PT3.x	PT3.x status flag and controller · 0≦x≦7

PT3.x	When TC3.x is set to <0>	When TC3.x is set to <1>
0 (Default)	PT3.x Input is low(L)	PT3.x Output is low (L)
1	PT3.x Input is high(H)	PT3.x Output is high(H)

TRISC3: PT3 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC3.x	PT3.x input/output control bit · 0≦x≦7 <0> Disable the output function,the pin is only for the input function. (Default) <1> Enable the output function,the pin is Output/input function

PT3IN: PT3 Input Control Register

Bit	Name	Description
Bit7~0	IN3.x	PT3.x digital input controller · 0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN3.x to change the PT3.x state normally.

PT3PU: PT3 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU3.x	PT3.x pull-up resistor pin control bit，0≦x≦7 <0>Disable(Default) <1>Enable

PT3M1: PT3 Digital Output Mode Selection Register 1

Bit	Name	Description
Bit7	PM3.7 ENPCMPO	PT3.7 output selection controller <0> Disable，Use PT3.7 as a general I/O。(Default) <1> Enable CMPO,output through PT3.7 Set CMPO to output from PT3.7 and also need to set PT3.7 to Output mode

PT4: PT4 Status Flag and Control Register

Bit	Name	Description
Bit2~0	PT4.x	PT4.x status flag and controller，0≦x≦2

PT4.x	When TC4.x is set to <0>	When TC4.x is set to <1>
0 (Default)	PT4.x Input is low (L)	PT4.x Output is low (L)
1	PT4.x Input is high (H)	PT4.x Output is high (H)

TRISC4: PT4 Input/Output Control Register

Bit	Name	Description
Bit2~0	TC4.x	PT4.x input/output control bit，0≦x≦2 <0> Disable the output function,the pin is only for the input function (Default) <1>Enable the output function,the pin is Output/input function

PT4IN: PT4 Input Control Register

Bit	Name	Description
Bit2~0	IN4.x	PT4.x digital input controller，0≦x≦2 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN4.x to change the PT4.x state normally.

PT4PU: PT4 Pull Up Resistor Control Register

Bit	Name	Description
Bit2~0	PU4.x	PT4.x pull-up resistor pin control bit，0≦x≦2 <0>Disable (Default) <1>Enable

PT6: PT6 Status Flag and Control Register

Bit	Name	Description
Bit7~0	PT6.x	PT6.x status flag and controller，0≦x≦7

PT6.x	When TC6.x is set to <0>	When TC6.x is set to <1>
0 (Default)	-	PT6.x Output is low (L)
1	-	PT6.x Output is high (H)

PT6IN: PT6 Input Control Register

Bit	Name	Description
Bit7~0	IN6.x	PT6.x digital input controller，0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN6.x to change the PT6.x state normally.

TRISC6: PT6 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC6.x	PT6.x input/output control bit，0≦x≦7 <0> Disable the output function,the pin is only for the input function (Default) <1>Enable the output function,the pin is Output/input function

PT6PU: PT6 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU6.x	PT6.x pull-up resistor pin control bit，0≦x≦7 <0>Disable <1>Enable

PT7: PT7 Status Control Register

Bit	Name	Description
Bit7~0	PT7.x	PT7.x status flag and controller，0≦x≦7

PT7.x	When TC7.x is set to <0>	When TC7.x is set to <1>
0 (Default)	-	PT7.x Output is low (L)
1	-	PT7.x Output is high (H)

PT7IN: PT7 Input Control Register

Bit	Name	Description
Bit7~0	IN7.x	PT7.x pin digital input controller，0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN7.x to change the PT7.x state normally.

TRISC7: PT7 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC7.x	PT7.x input/output control bit，0≦x≦7 <0> Disable the output function,the pin is only for the input function. (Default) <1> Enable the output function,the pin is Output/input function

PT7PU: PT7 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU7.x	PT7.x pull-up resistor pin control bit，0≦x≦7 <0>Disable <1>Enable

PT8: PT8 Status Flag and Control Register

Bit	Name	Description
Bit7~0	PT8.x	PT8.x status flag and controller，0≦x≦7

PT8.x	When TC8.x is set to <0>	When TC8.x is set to<1>
0 (Default)	-	PT8.x Output is low (L)
1	-	PT8.x Output is high (H)

PT8IN: PT8 Input Control Register

Bit	Name	Description
Bit7~0	IN8.x	PT8.x digital input controller，0≦x≦7 <0> Disable digital input function

		<1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN8.x to change the PT8.x state normally

TRISC8: PT8 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC8.x	PT8.x input/output control bit · 0≦x≦7 <0> Disable the output function,the pin is only for the input function (Default) <1>Enable the output function,the pin is Output/input function

PT8PU: PT8 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU8.x	PT8.x pull-up resistor pin control bit · 0≦x≦7 <0>Disable <1>Enable

PT9: PT9 Status Flag and Control Register

Bit	Name	Description
Bit7~0	PT9.x	PT9.x status flag and controller · 0≦x≦7

PT7.x	When TC9.x is set to <0>	When TC9.x is set to <1>
0 (Default)	-	PT7.x Output is low (L)
1	-	PT7.x Output is high (H)

PT9IN: PT9 Input Control Register

Bit	Name	Description
Bit7~0	IN9.x	PT9.x digital input controller · 0≦x≦7 <0> Disable digital input function <1> Enable digital input function Note: When IO is used as an output mode (When you want to read the output status), It is necessary to enable IN9.x to change the PT9.x state normally.

TRISC9: PT9 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC9.x	PT9.x input/output control bit · 0≦x≦7 <0> Disable the output function,the pin is only for the input function (Default) <1>Enable the output function,the pin is Output/input function

PT9PU: PT9 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU9.x	PT9.x pull-up resistor pin control bit，0≦x≦7 <0>Disable <1>Enable

PT10: PT10 Status Control Register

Bit	Name	Description
Bit7~0	PT10.x	PT10.x status flag and controller，0≦x≦7

PT10.x	When TC10.x is set to <0>	When TC10.x is set to <1>
0 (Default)	-	PT10.x Output is low (L)
1	-	PT10.x Output is high (H)

TRISC10: PT10 Input/Output Control Register

Bit	Name	Description
Bit7~0	TC10.x	PT10.x input/output control bit，0≦x≦7 <0> Disable the output function,the pin is only for the input function. (Default) <1> Enable the output function,the pin is Output/input function

PT10PU: PT10 Pull Up Resistor Control Register

Bit	Name	Description
Bit7~0	PU10.x	PT10.x pull-up resistor pin control bit，0≦x≦7 <0>Disable <1>Enable

8. Multi-Function Comparator,MFC

The chip has an embedded low-power, Rail-to-Rail multi-function comparator for the comparing analog signals. It has the interrupt function; when the comparison result generates, the interrupt signal also generates; and it can increase the operability for users. It has different configuration settings for different applications.Used as a comparison of analog signals. With interrupt function, when the comparison result is generated, an interrupt signal can be generated to increase the user's operability. It can have different configurations for different applications.

- Features of CMP include:
  - Rail-to-Rail input range
  - Low operating current
  - 2us peak pulse filter
  - Buit-in 32 nodes 5-bit step resistor

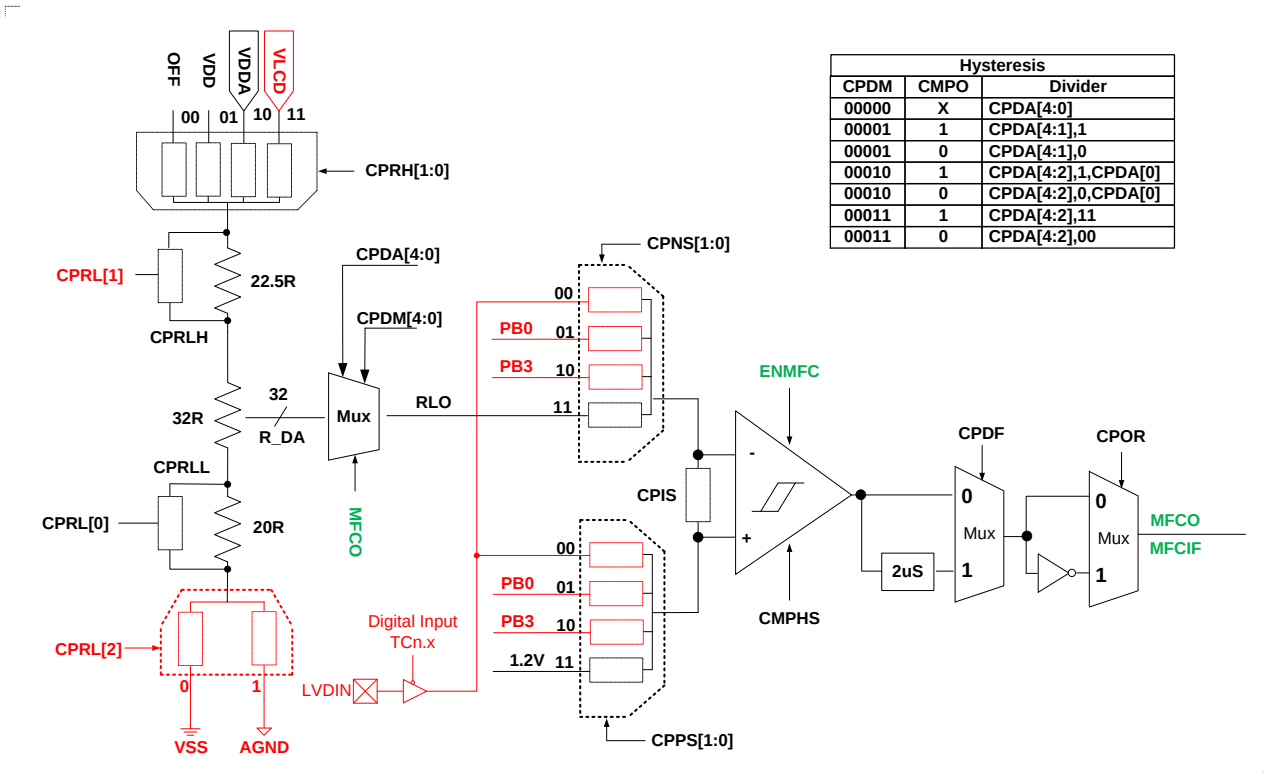
MFC Related Registers:

MFCCN0 CPRH[1:0], MFCO, CPIS, CPOR, CPDF, CMPHS, ENMFC

MFCCN1 CPRL[2:0], CPPS[1:0], CPNS[1:0]

MFCCN2 CPDA[4:0]

MFCCN3 CPDM[4:0]



Hysteresis		
CPDM	CMPO	Divider
00000	X	CPDA[4:0]
00001	1	CPDA[4:1],1
00001	0	CPDA[4:1],0
00010	1	CPDA[4:2],1,CPDA[0]
00010	0	CPDA[4:2],0,CPDA[0]
00011	1	CPDA[4:2],11
00011	0	CPDA[4:2],00

Figure 8-1 Low voltage detection block diagram

8.1. Function Description

8.1.1. Multplexing input channel selector

The input channel of the comparator is composed two parts; one is the input channel of the comparator, which can be set by the controller CPPS[1:0]/CPNS[1:0] to respectively set the positive input channel and the negative input channel of the comparator. When the control bit CPIS as <1> to realize the short-circuit between the positive input end and the negative input end; on the contrary, if the CPIS is set as <0>, the comparator can work normally.

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

The comparator has a built-in multi-node resistor, and the resistor includes three parts: 22.5R, 32R and 20R. The 32R resistor is connected to a 32-stage resistor node selector; the selector divides the 32R resistor into 32 nodes, which can be set by the controllers CPDA [4:0] and CPDM [4:0] to select different resistor nodes to output different voltages to the input channel RLO of the comparator. If the control bits CPRLH and CPRLL are set as <1>, the short circuit between the 22.5R resistor and 20R resistor can be achieved, which can adjust the resistor node voltage. The controller CPRH [1:0] can be used to select different voltage sources to increase the output range of the node voltage.

The hysteresis controller CPDM [4:0] is linked up with the node selector CPDA [4:0]; each bit of the hysteresis controller CPDM [4:0] is corresponding to the control of the enablement and disablement of the each bit of the controller CPDA [4:0] respectively. When the corresponding bit of the hysteresis controller CPDM[4:0] is set as <1>, the hysteresis function of the corresponding bit of the node controller CPDA[4:0] will be enabled and the status of the bit is consistent with the output status of the comparator; that is CPDA[X]=CMPO. In this way, the node selector will be switched between the two nodes.When using a hysteresis controller, CPOR[0] must be set to 1.

CPDM[4:0]	MFCO output status	CPDA[4:0] Hysteresis switch period	CPDM[4:0]	MFCO output status	CPDA[4:0] Hysteresis switch period
00000	0	uuuuu	10000	0	0uuuu
	1	uuuuu		1	1uuuu
00001	0	uuuu0	10001	0	0uuu0
	1	uuuu1		1	1uuu1
00010	0	uuu0u	10010	0	0uu0u
	1	uuu1u		1	1uu1u
00011	0	uuu00	10011	0	0uu00
	1	uuu11		1	1uu11
00100	0	uu0uu	10100	0	0u0uu
	1	uu1uu		1	1u1uu
00101	0	uu0u0	10101	0	0u0u0
	1	uu1u1		1	1u1u1

CPDM[4:0]	MFCO output status	CPDA[4:0] Hysteresis switch period	CPDM[4:0]	MFCO output status	CPDA[4:0] Hysteresis switch period
00110	0	uu00u	10110	0	0u00u
	1	uu11u		1	1u11u
00111	0	uu000	10111	0	0u000
	1	uu111		1	1u111
01000	0	u0uuu	11000	0	00uuu
	1	u1uuu		1	11uuu
01001	0	u0uu0	11001	0	00uu0
	1	u1uu1		1	11uu1
01010	0	u0u0u	11010	0	00u0u
	1	u1u1u		1	11u1u
01011	0	u0u00	1011	0	00u00
	1	u1u11		1	11u11
01100	0	u00uu	11100	0	000uu
	1	u11uu		1	111uu
01101	0	u00u0	11101	0	000u0
	1	u11u1		1	111u1
01110	0	u000u	11110	0	0000u
	1	u111u		1	1111u
01111	0	u0000	11111	0	00000
	1	u1111		1	11111
'u' means no change					

Table 8-1 Hysteresis control CPDM[4:0] configuration and values

8.1.3. Comparator interrupt output

The output of the comparator is a digital output, and MFCO can trigger an interrupt flag.

When the output signal (MFCO) generates a Rising edge (0→1) to trigger an interrupt event, MFCIF[0] will be set to <1>. MFCIF[0] can be cleared. When an interrupt event occurs next time, the MFCO output signal needs to go through 1->0->1.

8.2. Register Description -MFC

"-"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**
07AH	MFCCN0	CPRH[1:0]		MFCO	CPIS	CPOR	CPDF	CMPHS	ENMFC	0000 0000	uuuu uuuu	*,*,r,*,*,*,*,*
07BH	MFCCN1	CPRL[2:0]			-	CPPS[1:0]		CPNS[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07CH	MFCCN2	-	-	-	CPDA[4:0]					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
07DH	MFCCN3	-	-	-	CPDM[4:0]					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 8-1 MFC Control Register

MFCCN0: MFC Control Register 0

Bit	Name	Description
Bit7~6	CPRH[1:0]	The voltage source selection of the built-in step resistor of the comparator **CPRH[1:0]** / **voltage source** 00 / Disable，no voltage supply, under high impedance 01 / VDD 10 / VDDA 11 / VLCD
Bit5	MFCO	The comparison result input transferring out status of the comparator <0> Negative input signal > Positive input signal <1> Positive input signal > Negative input signal
Bit4	CPIS	The short-circuit switch control of the comparator <0> short-circuit switch opens.(Open=OFF) <1> short-circuit switch closes.(Closed=ON) (For testing, not recommended)
Bit3	CPOR	The digital output phase control of the comparator <0> Normal output <1> Inversed output
Bit2	CPDF	The output low-pass filer enable control of the comparator <0> Disable, the output of the comparator does not pass through the 2us low-pass filter. <1> Enable, the output of the comparator passes through the 2us low-pass filter.
Bit1	CMPHS	The high-speed mode enable control of the comparator <0> Low power mode <1> Normal mode
Bit0	ENMFC	Comparator function enable control <0> Disable (the output status is 0) <1> Enable

MFCCN1: MFC Control Register 1

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>CPRL[2]</td><td>Built-in step resistance grounding switch control of the comparator
<0> Connect the low section resistor to VSS
<1> Connect the low section resistor to AGND</td></tr>
<tr><td>Bit6</td><td>CPRL[1]</td><td>Built-in step resistance (22.5R) short-circuit switch control of the comparator
<0> The short-circuit switch opens, the low step resistor is not short-circuited.
<1> The short-circuit switch closes, the low step resistor is short-circuited.</td></tr>
<tr><td>Bit5</td><td>CPRL[0]</td><td>Built-in step resistance (20R) short-circuit switch control of the comparator
<0> The short-circuit switch opens, the low step resistor is not short-circuited.
<1> The short-circuit switch closes, the low step resistor is short-circuited.</td></tr>
<tr><td>Bit3~2</td><td>CPPS[1:0]</td><td>The positive input end selection of the comparator
<table>
<tr><th>CPPS[1:0]</th><th>positive input end</th></tr>
<tr><td>00</td><td>LVDIN pin</td></tr>
<tr><td>01</td><td>PB0</td></tr>
<tr><td>10</td><td>PB3</td></tr>
<tr><td>11</td><td>V12</td></tr>
</table></td></tr>
<tr><td>Bit1~0</td><td>CPNS[1:0]</td><td>The negative input end selection of the comparator
<table>
<tr><th>CPNS[1:0]</th><th>negative input end</th></tr>
<tr><td>00</td><td>LVDIN pin</td></tr>
<tr><td>01</td><td>PB0</td></tr>
<tr><td>10</td><td>PB3</td></tr>
<tr><td>11</td><td>RLO</td></tr>
</table></td></tr>
</table>

MFCCN2: MFC Control Register 2

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit4~0</td><td>CPDA[4:0]</td><td>The voltage divider node setting of the built-in multi-point resistor in the comparator
<table>
<tr><th>CPDA[4:0]</th><th>Voltage division node</th><th>CPDA[4:0]</th><th>Voltage division node</th></tr>
<tr><td>00000</td><td>0</td><td>10000</td><td>16/32(CPRLH–CPRLL)</td></tr>
<tr><td>00001</td><td>1/32(CPRLH–CPRLL)</td><td>10001</td><td>17/32(CPRLH–CPRLL)</td></tr>
<tr><td>00010</td><td>2/32(CPRLH–CPRLL)</td><td>10010</td><td>18/32(CPRLH–CPRLL)</td></tr>
<tr><td>00011</td><td>3/32(CPRLH–CPRLL)</td><td>10011</td><td>19/32(CPRLH–CPRLL)</td></tr>
<tr><td>00100</td><td>4/32(CPRLH–CPRLL)</td><td>10100</td><td>20/32(CPRLH–CPRLL)</td></tr>
<tr><td>00101</td><td>5/32(CPRLH–CPRLL)</td><td>10101</td><td>21/32(CPRLH–CPRLL)</td></tr>
<tr><td>00110</td><td>6/32(CPRLH–CPRLL)</td><td>10110</td><td>22/32(CPRLH–CPRLL)</td></tr>
<tr><td>00111</td><td>7/32(CPRLH–CPRLL)</td><td>10111</td><td>23/32(CPRLH–CPRLL)</td></tr>
<tr><td>01000</td><td>8/32(CPRLH–CPRLL)</td><td>11000</td><td>24/32(CPRLH–CPRLL)</td></tr>
<tr><td>01001</td><td>9/32(CPRLH–CPRLL)</td><td>11001</td><td>25/32(CPRLH–CPRLL)</td></tr>
<tr><td>01010</td><td>10/32(CPRLH–CPRLL)</td><td>11010</td><td>26/32(CPRLH–CPRLL)</td></tr>
<tr><td>01011</td><td>11/32(CPRLH–CPRLL)</td><td>11011</td><td>27/32(CPRLH–CPRLL)</td></tr>
<tr><td>01100</td><td>12/32(CPRLH–CPRLL)</td><td>11100</td><td>28/32(CPRLH–CPRLL)</td></tr>
<tr><td>01101</td><td>13/32(CPRLH–CPRLL)</td><td>11101</td><td>29/32(CPRLH–CPRLL)</td></tr>
<tr><td>01110</td><td>14/32(CPRLH–CPRLL)</td><td>11110</td><td>30/32(CPRLH–CPRLL)</td></tr>
<tr><td>01111</td><td>15/32(CPRLH–CPRLL)</td><td>11111</td><td>31/32(CPRLH–CPRLL)</td></tr>
</table></td></tr>
</table>

MFCCN3: MFC Control Register 3

Bit	Name	Description
Bit4	CPDM[4]	The CPDA[4] outputs hystersis enable control，and the value of the CPDA[4] is controlled by the CMPO, and keep consistent. <0> Disable <1> Enable，CPDA[4]= MFCO
Bit3	CPDM[3]	The CPDA[3] outputs hystersis enable control，and the value of the CPDA[3] is controlled by the CMPO, and keep consistent. <0> Disable <1> Enable，CPDA[3]= MFCO
Bit2	CPDM[2]	The CPDA[2] outputs hystersis enable control，and the value of the CPDA[2] is controlled by the CMPO, and keep consistent. <0> Disable <1> Enable，CPDA[2]= MFCO
Bit1	CPDM[1]	The CPDA[1] outputs hystersis enable control，and the value of the CPDA[1] is controlled by the CMPO, and keep consistent. <0> Disable <1> Enable，CPDA[1]= MFCO
Bit0	CPDM[0]	The CPDA[0] outputs hystersis enable control，and the value of the CPDA[0] is controlled by the CMPO, and keep consistent. <0> Disable <1> Enable，CPDA[0]= MFCO

9. WATCH DOG TIMER,WDT

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

- Operation mode

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

  The WDT can be cleared by using software.

- SLEEP mode

  The WDT is disabled, and cannot work.

- IDLE mode

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

WDT Related Registers：

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

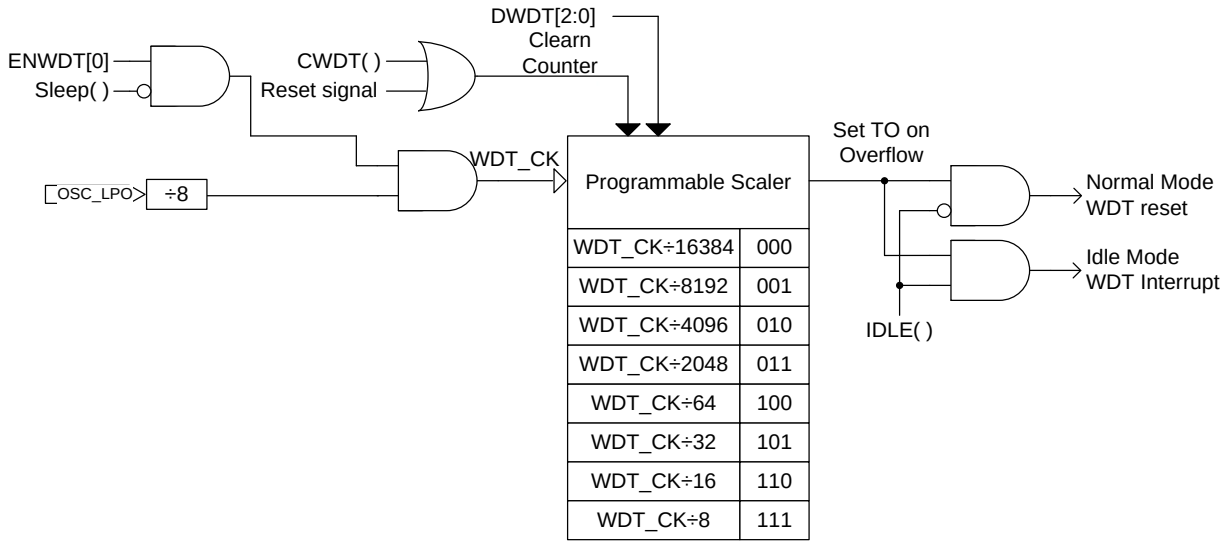


Figure 9-1 WDT block diagram

9.1. WDT Manual

9.1.1. WDT Initial Configuration

WDT counting controller, DWDT[2:0] can decide WDT counter operating frequency WDT_CK and overflow.WDT reset signal TO or interrupt event,WDTIF[2] will be produced after the counter being overflowed.

[2] WDT uses internal clock source, LPO. It can be operated under Run Mode and Idle Mode. Under Run Mode, WDT can be zeroed by software as to avoid IC reset. However, WDT cannot be zeroed by any means under Idle Mode.

9.1.2. WDT Interrupt Event Service

WDT interrupt event can only be operated while the IC is in Idle Mode. When WDTIE[0] and GIE[0] is configured as <1>, after WDT overflowed, interrupt event, WDTIF[0] is set as <1> and PC will jump to interrupt vector address, <0>x0004h. By contrast, WDTIE[0] and GIE[0] is configured as <0>, no interrupt will take place.

9.1.3. WDT Initiation

WDT must be started when the IC is under Run Mode. That is to say, WDT starting controller, ENWDT[0] must be set as <1> to initiate WDT. Once the WDT is started, ENWDT[0] cannot be configured as <0> by software. It is not possible to use software to set ENWDT[0] to <0> no matter in running mode or standby mode. After DWDT[2:0] is set, when a WDT reset or interrupt occurs, DWDT will be cleared to 000b and needs to be reset by software.

9.2. Register Description of the WDT

"-"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,-
03Ah	WDTCN					ENWDT	DWDT[2:0]			0000 0000	uuuu $000	-,*,*,* rw1,*,*,*

Table 9-1 WDT Control Register

INTE0/INTF0: Please refer to Interrupt Chapter

PSTAT[7:0] Status flag register

Bit	Name	Description
Bit5	TO	WDT operation mode count overflow flag <0>Not happened <1>Happened；Cleared through BOR, RST or instruction

WDTCN[7:0] WDT Control Register

Bit	Name	Description
Bit3	ENWDT[0]	WDT Starting Controller <0> Disable <1> Enable
Bit2~0	DWDT[2:0]	Configure WDT overflow time

DWDT[2:0]	Pre-scale	DWDT[2:0]	Pre-scale
000	WDT_CK ÷ 256 ÷ 64	100	WDT_CK ÷ 64
001	WDT_CK ÷ 256 ÷ 32	101	WDT_CK ÷ 32
010	WDT_CK ÷ 256 ÷ 16	110	WDT_CK ÷ 16
011	WDT_CK ÷ 256 ÷ 8	111	WDT_CK ÷ 8

10. Power System,PWR

Power System (Power System, hereafter referred to as PWR).The PWR includes a linear power supply(VDDA) and analog circuit common ground power supply ACM, which provides chip analog peripheral circuits can be properly used to drive an external circuit.

PWR Related Registers：

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

PWRCN1 ENREFO[0], ENVS, SAGND[2:0]

PWRCN2 ENPUMP, VGGS

AD1CN1 ENBRCH[0]

AD1CN5 LDOPL[0]

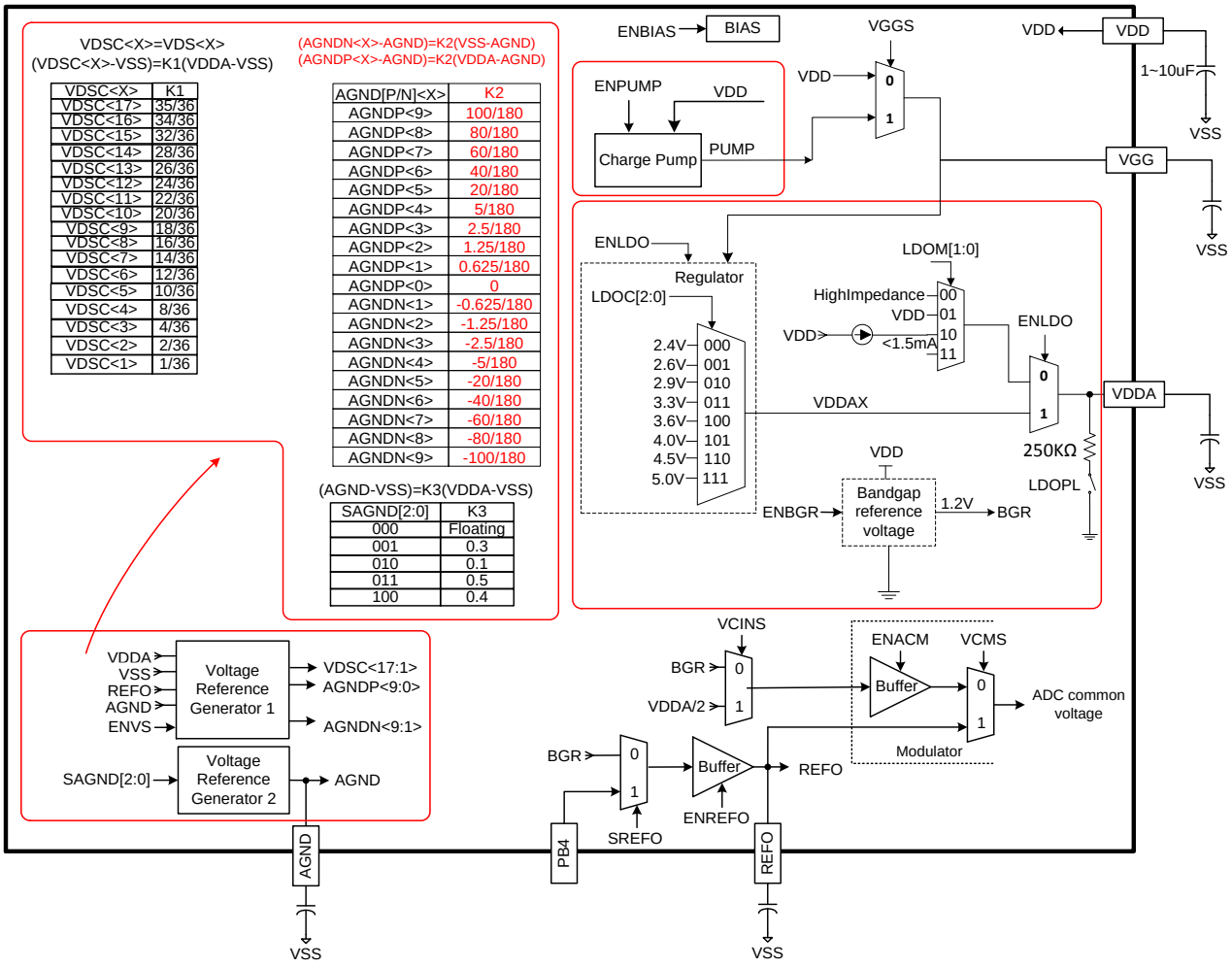


Figure 10-1 Power System Block Diagram

10.1. VDDA Manual

10.1.1. VDDA Initial configuration：

The regulators selector LDOC [2:0] can be set VDDA pin output voltage from 2.4V~5.0V, total of 8-segment of voltage can be selected.VDDA is a linear regulated power supply.

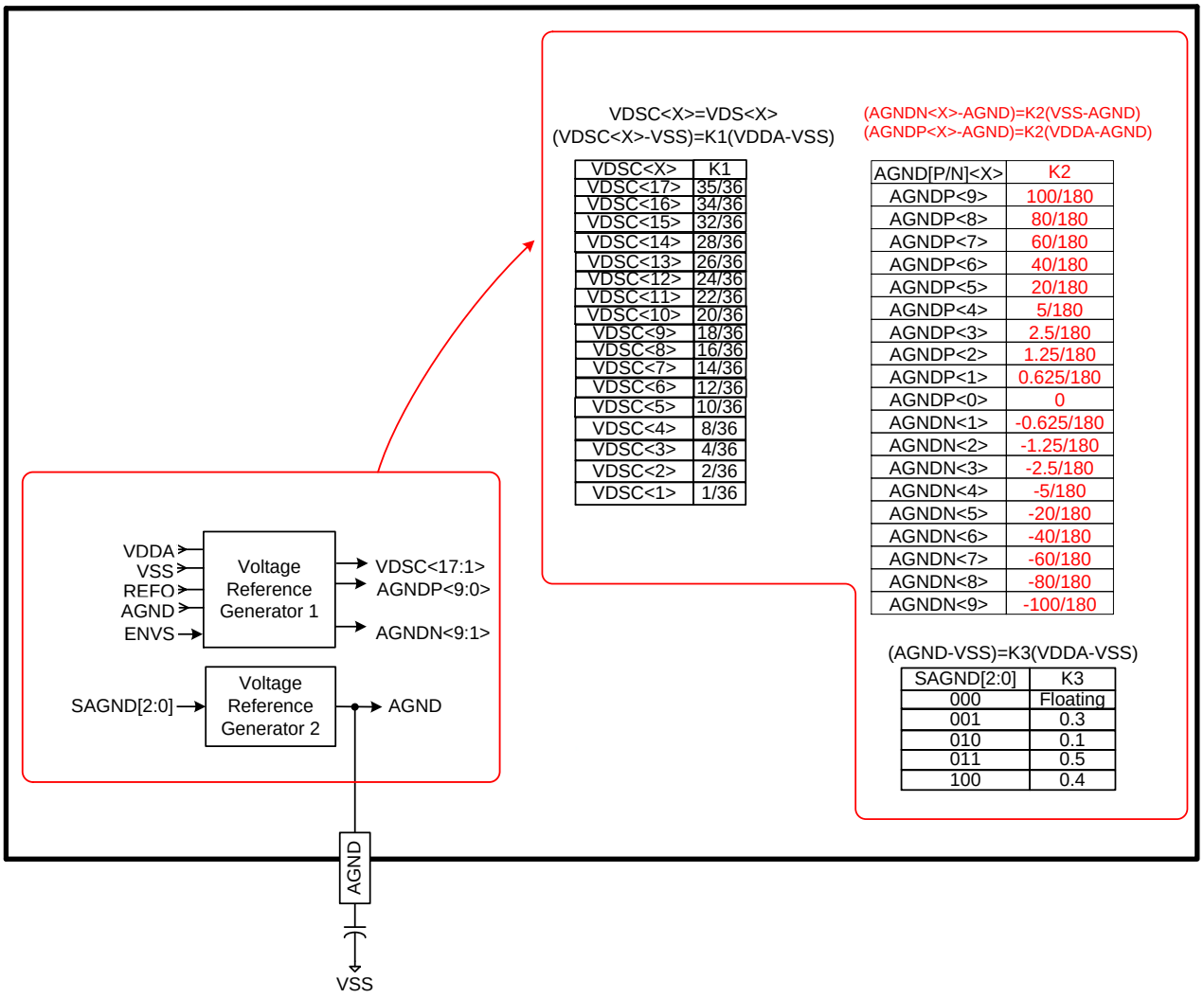
10.1.2. VDDA using external input :

VDDA can be designed by external input voltage, if users would like to provide alternative voltage sources, the voltage must input from VDDA pin. Using this method, VDDA must be closed, which means LDOM [1:0] must be configured as 0. Moreover, this method may impact analog circuit performance, so it should be dealt with extra caution.

10.1.3. VDDA Initiation:

Configure ENLDO[0] as <1> to initiate VDDA regulator. Oppositely, if ENLDO[0] is configured as <0>, VDDA will be shut off. To start VDDA, ΣΔADC cannot in enabled status. It must wait after VDDA voltage is stabilized then to start ΣΔADC. When external 1uF(10uF) regulated capacitor is connected, it requires 500uS(5mS) to stabilize.

10.2. Voltage Reference Generator(VRG)



VDSC<X>	K1
VDSC<17>	35/36
VDSC<16>	34/36
VDSC<15>	32/36
VDSC<14>	28/36
VDSC<13>	26/36
VDSC<12>	24/36
VDSC<11>	22/36
VDSC<10>	20/36
VDSC<9>	18/36
VDSC<8>	16/36
VDSC<7>	14/36
VDSC<6>	12/36
VDSC<5>	10/36
VDSC<4>	8/36
VDSC<3>	4/36
VDSC<2>	2/36
VDSC<1>	1/36

AGND[P/N]<X>	K2
AGNDP<9>	100/180
AGNDP<8>	80/180
AGNDP<7>	60/180
AGNDP<6>	40/180
AGNDP<5>	20/180
AGNDP<4>	5/180
AGNDP<3>	2.5/180
AGNDP<2>	1.25/180
AGNDP<1>	0.625/180
AGNDP<0>	0
AGNDN<1>	-0.625/180
AGNDN<2>	-1.25/180
AGNDN<3>	-2.5/180
AGNDN<4>	-5/180
AGNDN<5>	-20/180
AGNDN<6>	-40/180
AGNDN<7>	-60/180
AGNDN<8>	-80/180
AGNDN<9>	-100/180

SAGND[2:0]	K3
000	Floating
001	0.3
010	0.1
011	0.5
100	0.4

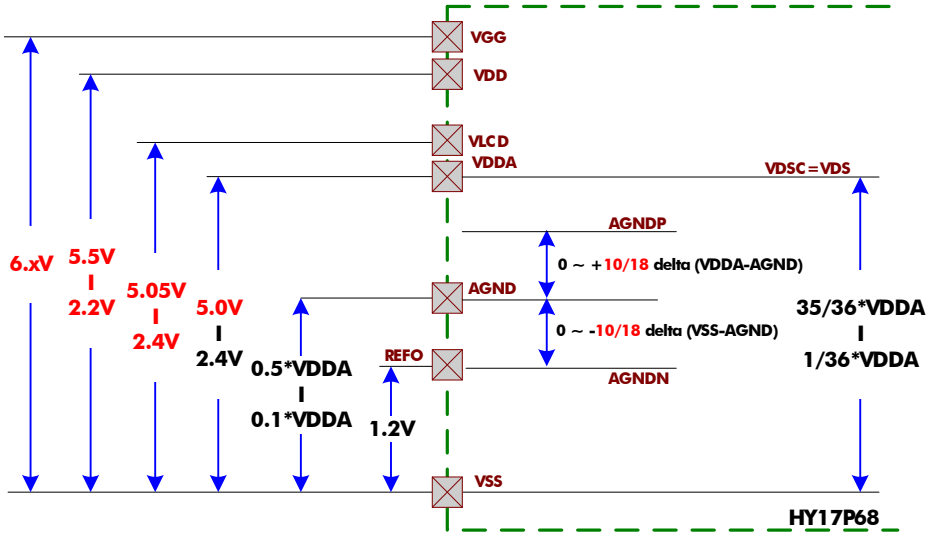
Figure 10-2 Voltage Reference Generator(VRG)

Voltage reference generator(VRG) of different reference voltages, and is supplied to the ADC and Comparator. Wherein VDSC <N> (N = 17 ~ 1), with respect to (VDDA.VSS) voltage.

AGNDP<N>(N=9~0) is the voltage relative to (VDDA. AGND).

AGNDN<N>(N=9~1) is the voltage relative to (VSS.-AGND).

VDSC<N> is the node obtained by voltage division of (VDDA,VSS), AGNDP<N> is the node obtained by voltage division of (VDDA, AGND), and AGNDN<N> is the node obtained by voltage division of (VSS, -AGND).



10.3. 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**
032H	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]		ENLDO		1000 0000	1uuu u00u	*,*,*,*,wr0,wr0,*
033H	PWRCN1	ENREFO				ENVS	SAGND2:0]			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
034H	PWRCN2	ENPUMP	VGGS	-	-	ENFIR	LDOPL	ENTPS	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
042H	NET0		SREFO							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 10-1 PWR Register

PWRCN: PWR Control Register

Bit	Name	Description
Bit7	ENBGR	Internal reference voltage controller <0> Disable <1> Enable · When using ADC and TPS,you need to set <1> first, and then turn on. Note: This bit is linked with HAO, if HAO is on. Even if this bit writes 0, BGR is still enabled
Bit6~4	LDOC[2:0]	Output voltage selector of the VDDA When ENLDO is set to 1, the set voltage will be output to the VDDA pin.
Bit3~2	LDOM[1:0]	Output selector of the VDDA When ENLDO is set to 0, the set will be output to the VDDA pin. <00> Close, with highimpedance mode <01> Output VDD voltage <10> Pull high to VDD by 1.5mA. (It is use to initial VDDA when a small current)

LDOC[2:0]	VDDA	LDOC[2:0]	VDDA
000	2.4V	100	3.6V
001	2.6V	101	4.0V
010	2.9V	110	4.5V
011	3.3V	111	5.0V

Bit	Name	Description
		<11> Reserve
Bit1	ENLDO	Internal linear regulator controller <0>Disable <1>Enable

PWRCN1: PWR Control Register 1

Bit	Name	Description
Bit7	ENREFO	REFO Buffer controller <0> Disable, REFO pin is in floating state <1> Enable
Bit3	ENVS	VRG(Voltage Reference Generator) Enable controller <0> Disable <1> Enable
Bit2~0	SAGND[2:0]	Voltage selector of the AGND <000> Disable, AGND pin is in floating state <001> 0.3xVDDA <010> 0.1xVDDA <011> 0.5xVDDA <100> 0.4xVDDA

PWRCN2: PWR Control Register 2

Bit	Name	Description
Bit7	ENPUMP	Charge pump regulator enable controller(Only For HY17P68) <0> Disable(Default) <1> Enable ※When charge pump is started, an external capacitor must be added to the VGG pin, otherwise the chip will consume a lot of current. ※When charge pump is started, The ADC must be turned on (ENAD1=1)
Bit6	VGGS	VGG Charge pump regulator selector(Only For HY17P68) <0> VGG=VDD (Default) <1> VGG≈2 * VDD, To enable the charge pump function, it is valid only after ENPUMP is enabled.
Bit2	LDOPL	Internal 250kΩ resistance pull-down switch <0> Disable(Default) <1>Enable · To use the internal LDO output, be sure to set LDOPL to 1, otherwise the result will not be as expected.
Bit1	ENTPS	Internal TPS enable controller <0> Disable

Bit	Name	Description
		<1> Enable, Need to set up relative ADC network

NET0: Measurement network setting control register 0

Bit	Name	Description
Bit6	SREFO	REFO Buffer input source selection controller <0> Select internal Band-gap Voltage Reference (Default) <1> Select PB<4> pin

11. Auto Range DMM Multi-Function Network,MFN(Only For HY17P68)

Except for HY17P60 which does not have this function, all others have an Auto Range DMM Multi-Function Network (MFN), which is used to realize the Auto Range measurement function.

- Voltage/resistor/capacitor switch measurement
- Constant voltage/current output
- Positive/negative electrode differential

MFN Related Registers:

NET0	SDIO, SREFO, SFT1[1:0], SFUVR[3:0]
NET1	SMODE[7:0]
PA98	PS8, DS8, FS8,SS8
PA76	PS7, DS7, FS7, SS7, PS6, DS6, FS6, SS6
PA54	PS5, DS5, FS5, SS5, PS4, DS4, FS4, SS4
PA32	PS3, DS3, FS3, SS3, PS2, DS2, FS2, SS2
PA10	PS1, DS1, FS1, SS1, PS0, DS0, FS0, SS0
ACC	ACC[7:0]

11.1. Analog Input Network (Only For HY17P68)

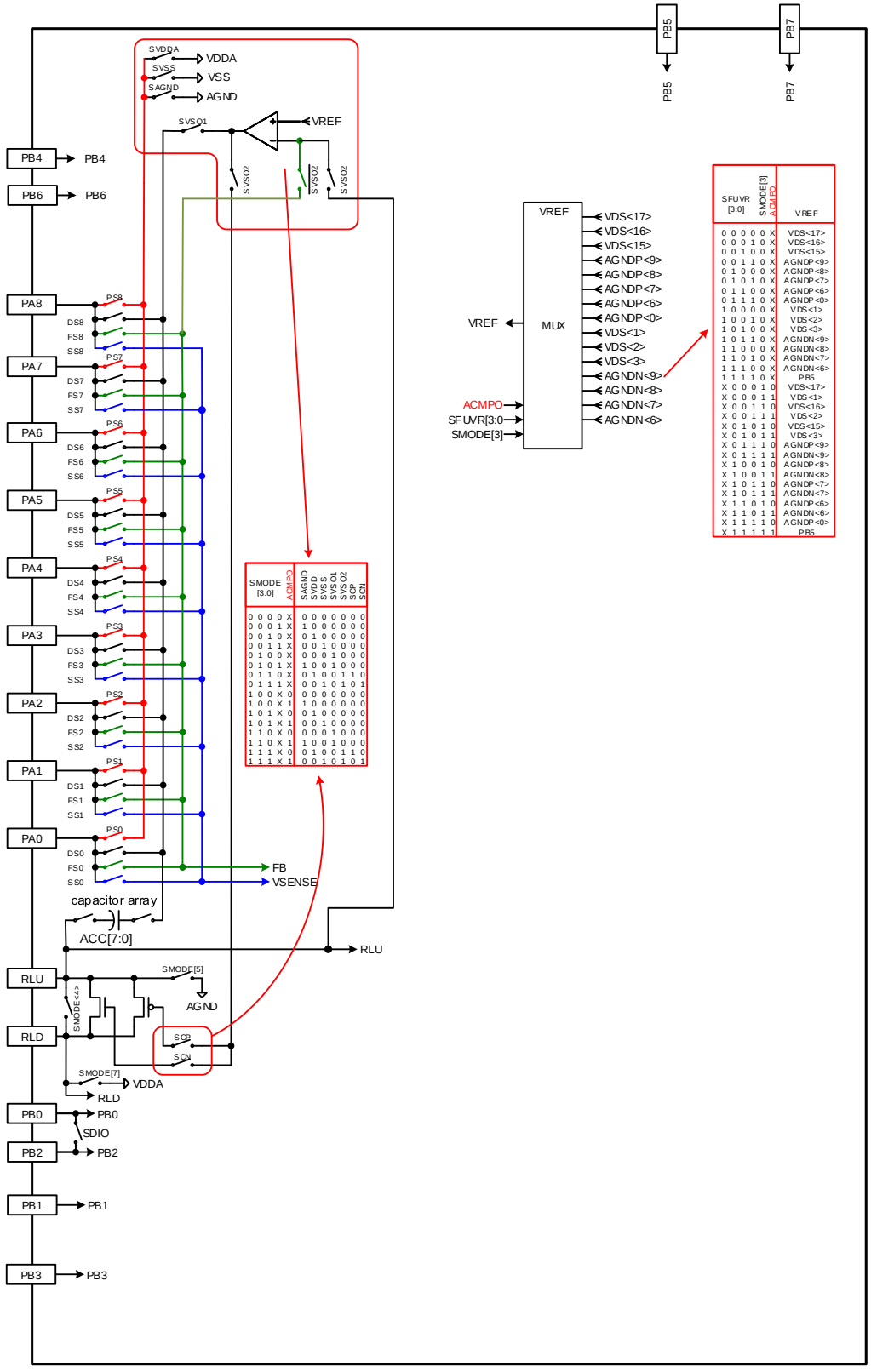


Figure 11-1 Analog Input Network Block Diagram

11.2. Capacitor array description

Capacitor array can be used as bandwidth compensation for ACV measurement. The capacitance value is as follows. The capacitance value is controlled by the register bit ACC[7:0].

$$capacitor\ array\ = ACC[7:0] \times 0.2pF$$

Example 1：

Supposed ACC[7:0]=01010101b=85,

Then total compensation capacitance value: 85*0.2 pF = 17 pF

Example 2：

Supposed ACC[7:0]=01100011b=99,

Then total compensation capacitance value: 99*0.2pF = 19.8 pF

11.3. Register Description - Multi-Function Network

"-"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**
42H	NET0	SDIO	SREFO	SFT1[1:0]		SFUVR[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
043H	NET1	SMODE[7:4]				SMODE[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
047H	PA98	-	-	-	-	PS8	DS8	FS8	SS8	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
048H	PA76	PS7	DS7	FS7	SS7	PS6	DS6	FS6	SS6	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
049H	PA54	PS5	DS5	FS5	SS5	PS4	DS4	FS4	SS4	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04AH	PA32	PS3	DS3	FS3	SS3	PS2	DS2	FS2	SS2	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04BH	PA10	PS1	DS1	FS1	SS1	PS0	DS0	FS0	SS0	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
067H	ACC	Capacitor array								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 11-1 Multi-Function Network Register

PA98/PA76/PA54/PA32/PA10: Control Register (Only For HY17P68)

Bit	PA98	PA76	PA54	PA32	PA10	Description
Bit7	-	PS7	PS5	PS3	PS1	PA<n> Power select control bit (n=7,5,3,1) 1：Connect 0：Disconnect
Bit6	-	DS7	DS5	DS3	DS1	PA<n> OP3 output select control bit (n=7,5,3,1) 1：Connect 0：Disconnect
Bit5	-	FS7	FS5	FS3	FS1	PA<n> Feedback select control bit (n=7,5,3,1) 1：Connect 0：Disconnect
Bit4	-	SS7	SS5	SS3	SS1	PA<n> Sense select control bit (n=7,5,3,1) 1：Connect 0：Disconnect
Bit3	PS8	PS6	PS4	PS2	PS0	PA<n> Power select control bit (n=8,6,4,2,0) 1：Connect 0：Disconnect
Bit2	DS8	DS6	DS4	DS2	DS0	PA<n> OP3 output select control bit (n=8,6,4,2,0) 1：Connect 0：Disconnect

Bit	PA98	PA76	PA54	PA32	PA10	Description
Bit1	FS8	FS6	FS4	FS2	FS0	PA<n> Feedback select control bit (n=8,6,4,2,0) 1：Connect 0：Disconnect
Bit0	SS8	SS6	SS4	SS2	SS0	PA<n> Sense select control bit (n=8,6,4,2,0) 1：Connect 0：Disconnect

NET0: Measurement network setting control register 0

Bit	Name	Description
Bit7	SDIO	Short control bit of PB<0> and PB<2> <0> Open。 <1> Short。
Bit6	SREFO	REFO Buffer input source selection controller <0> Select internal Band-gap Voltage Reference (Default) <1> Select PB<4> pin
Bit5~4	SFT1[1:0]	Pre-filter path
Bit3~0	SFUVR[3:0]	Select voltage reference source(Only For HY17P68)

SFT1[1:0]	Pre-filter
00	100kΩ
01	10kΩ
10	0Ω；Short
11	Close；Without Pre-filter path

SFUVR[3:0]	VREF	SFUVR[3:0]	VREF
0000	VDSC<17>	1000	VDSC<1>
0001	VDSC<16>	1001	VDSC<2>
0010	VDSC<15>	1010	VDSC<3>
0011	AGNDP<9>	1011	AGNDN<9>
0100	AGNDP<8>	1100	AGNDN<8>
0101	AGNDP<7>	1101	AGNDN<7>
0110	AGNDP<6>	1110	AGNDN<6>
0111	AGNDP<0>	1111	PB5

NET1: Measurement network setting control register 1 (Only For HY17P68)

Bit	Name	Description
Bit7	SMODE[7]	RLD and VDDA short-circuit the control bit <0> Open。 <1> Short。
Bit5	SMODE[5]	RLU and AGND short-circuit the control bit <0> Open。 <1> Short。
Bit4	SMODE[4]	RLU and RLD short-circuit the control bit <0> Open。 <1> Short。
Bit3~0	SMODE[3:0]	Analog voltage / current output control bit

SMODE [3:0]	ACMPO	SAGND	SVDD	SVSS	SVSO1	SVSO2	SCP	SCN
0 0 0 0	X	0	0	0	0	0	0	0
0 0 0 1	X	1	0	0	0	0	0	0
0 0 1 0	X	0	1	0	0	0	0	0
0 0 1 1	X	0	0	1	0	0	0	0
0 1 0 0	X	0	0	0	1	0	0	0
0 1 0 1	X	1	0	0	1	0	0	0
0 1 1 0	X	0	1	0	0	1	1	0
0 1 1 1	X	0	0	1	0	1	0	1
1 0 0 X	0	0	0	0	0	0	0	0
1 0 0 X	1	1	0	0	0	0	0	0
1 0 1 X	0	0	1	0	0	0	0	0
1 0 1 X	1	0	0	1	0	0	0	0
1 1 0 X	0	0	0	0	1	0	0	0
1 1 0 X	1	1	0	0	1	0	0	0
1 1 1 X	0	0	1	0	0	1	1	0
1 1 1 X	1	0	0	1	0	1	0	1

12. Sigma Delta Analog to Digital Converter ,ΣΔADC

ΣΔADC is a high resolution over sampling sigma delta analog-to-digital converter that equips with multi-channel 24 bit output. ΣΔADC consists of four main categories, multi-functional input multiplexer, input buffer and pre-low noise programmable gain amplifier (PGA), Sigma Delta Modulator(ΣΔADC) and comb filter.

- Multi-Functional Input Multiplexer
  Can switch to diversified set of input channels, single IC can execute several measurements
  Input channel can short, eliminating ADC zero point drift
  Built-in temperature sensor circuit voltage output
- ΣΔ Modulator
  Adjustable input voltage amplification: 1/2~ 8 amplifications
  Selectable reference voltage amplification: 1 or 1/2
  4 bit direct current input bias configuration
- Comb Filter
  Can adjust OSR(Over Sampling Ratio)= 32~61440
  Support 2^{nd} + 3^{rd} architecture
  Can produce interrupt event

ΣΔADC Related Registers：

AD1CN0	ENAD1, ENCH[0], ENINXCH[0], VREGN, OSR[2:0], CMFR
AD1CN1	ENACM, VCMS, VCINS, TPSCP, TPSCH, ADGN[2:0]
AD1CN2	FilterN[1:0], DAFM[0], DCSET[3:0]
AD1CN3	SAD1FP[3:0], SAD1FN[2:0]
AD1CN4	AD1RHBUF, AD1RLBUF, AD1IPBUF, AD1INBUF, INX, VRIS, INIS
AD1CN5	SAD1RH[2:0], SAD1RL[2:0], SAD1I[1:0]

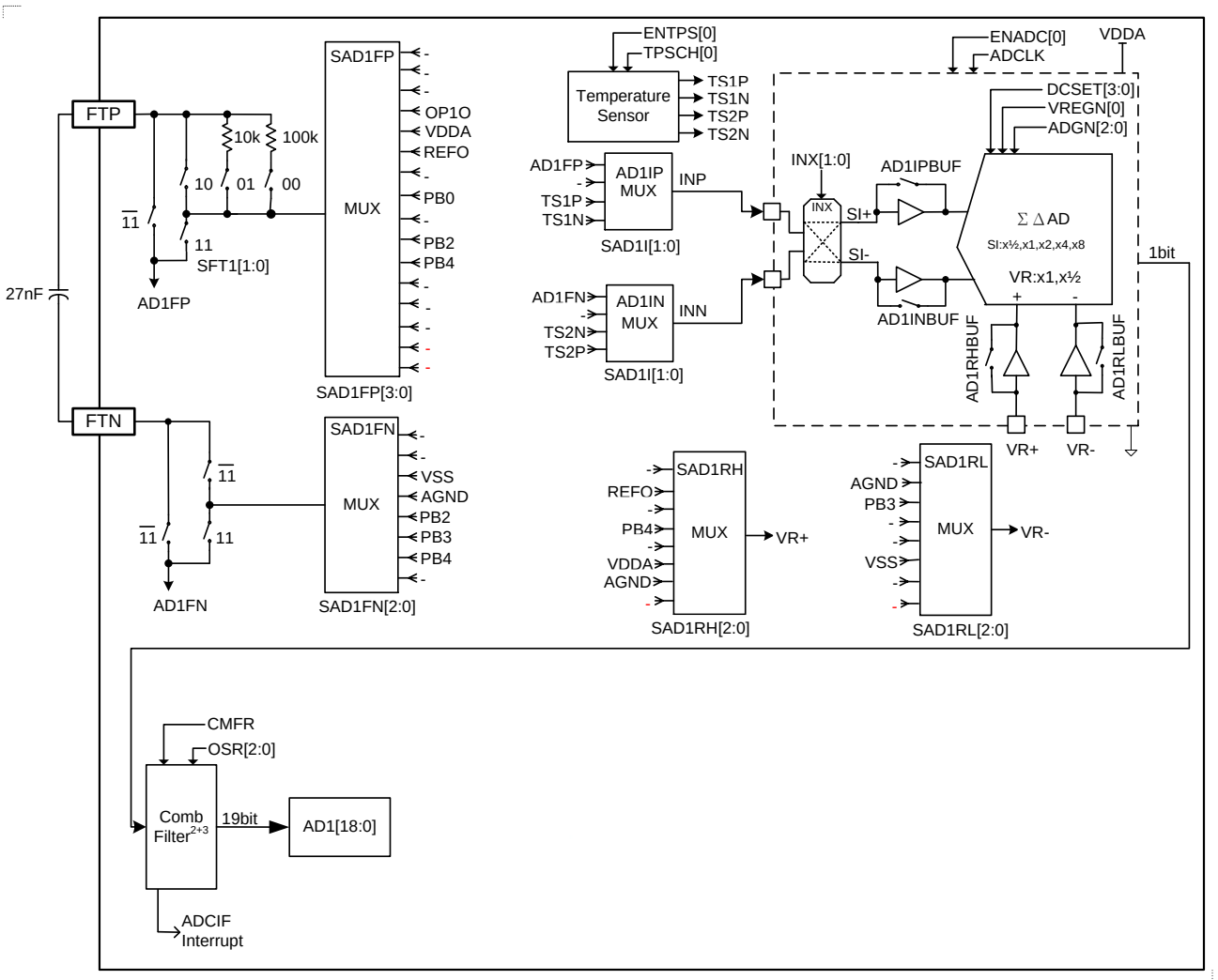


Figure 12-1a HY17P60B ΣΔADC Block Diagram

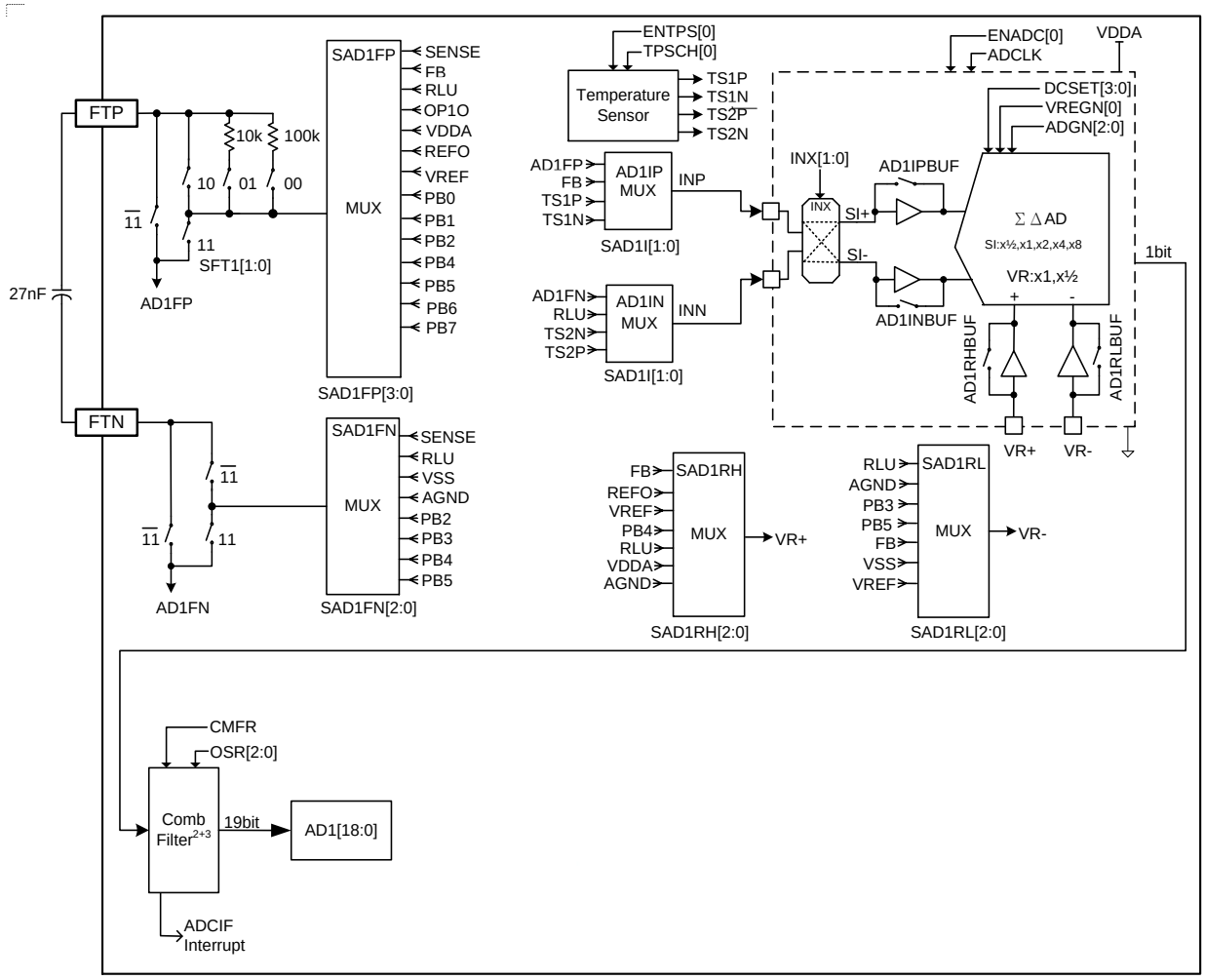


Figure 12-1b HY17P68 ΣΔADC Block Diagram

12.1. ΣΔADC Manual

12.1.1. ΣΔADC Initial Configuration

12.1.1.1. Operating Frequency Configuration

The sampling frequency of ΣΔADC can be set by the sampling frequency selector ADCCK[0]. The operating frequency of ΣΔADC is provided by DHS_CK, and its maximum sampling frequency cannot be greater than 1MHz.Faster sampling Frequency can get better resolution at the same output speed, but it's input impedance will also be reduced (refer to: 12.2 Analog channel input characteristices). When the DHS_CK frequency exceeds the maximum allowable value, the frequency must be adjusted through the sampling frequency prescaler DADC[1:0].

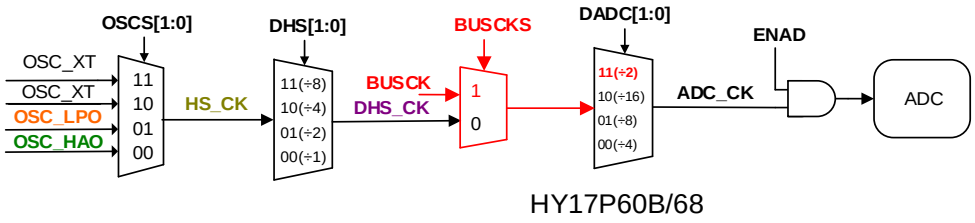


Figure 12-2 ΣΔADC Operating Frequency Block Diagram

12.1.1.2. Configuration of Multi-Functional Input Multiplexer

ΣΔADC is adopted second order ΣΔ modulator, The signal for testing and reference voltage can be taken magnification and bias voltage adjustment via the following settings.

- When ΔVR± magnification adjuster VREGN[0] is set as <1>, it will take 1/2 magnification adjustment for reference voltage signal, and also change the ratio of input signals ΔSI± = (SI+ - SI-) and ΔVR± = (VR+ - VR-). When it is set as <0>, it takes 1 time of adjustment.
- Via the setting of magnification adjuster ADGN[1:0], input signal can be up to max 8 times of signal magnification, as in Table 12-1(a).
- Input signal SI± can adjust the input signal zero position to increase measurement range via DC input bias adjuster DCSET[3:0]. Bias method is adopted magnification value of weighted reference signal VR±, as in Table 12-1(b).
- When taking signal measuring, it shall note the matching problem of external input signal impedance and ADC. See Analog Channel Input Characteristics

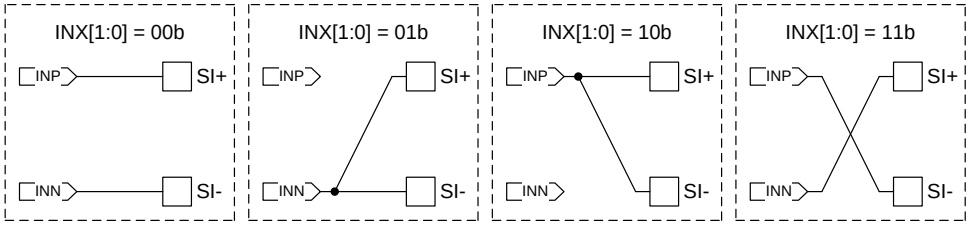


Figure 12-3 4 Combinations of INX Input Signal Convertor

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

Table 12-1 (a)ADGN[2:0] Amplification Configuration

Configuration	DCSET[3:0]							
Input	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	DCSET[3:0]							
Input	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 12-1 (b) SI± Input Signal Weighted Voltage Reference Chart

After ΣΔ modulator takes pre-PGA and modulator magnification bias adjustment, the calculation equation of equivalent signal for testing ΔSI_I and equivalent reference voltage ΔVR_I are as the following respectively:

Equation 12 -1

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

Equation 12 -2

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

Notes: To ensure ΣΔ modulator output get higher resolution and degree of linearity, equivalent reference voltage ΔVR_I is suggested as ΔVR_I=0.8V~1.2V , and equivalent signal for testing ΔSI_I is operated at ΔSI_I=±0.9 x ΔVR_I.

12.1.1.3. Comb Filter Configuration

ΣΔ modulator output1-bit data is sent to second order Comb Filter, then it is covered into 24-bit value via Comb Filter and stored in ADCR[23:0] register. The update speed of ADCR[23:0] data is the output speed of ΣΔADC, and the calculation mode is the ratio of ΣΔADC sampling frequency to ΣΔADC output speed. ΣΔADC output speed frequency is also called OSR (Over Sampling Ratio).

Thus, ΣΔADC output rate is ADC_CK÷OSR. However, OSR value can be set by OSR[2:0] in order to generate different ΣΔADC output conversion frequency, as Table 12-1(c).

Configuration	OSR[2:0]							
ADC_CK	32	64	128	256	7680	15360	30720	61440
1228.8kHz	38400	19200	9600	4800	160	80	40	20
614.4kHz	19200	9600	4800	2400	80	40	20	10

Table 12-1 (c) ΣΔADC Over-Sampling Rate Configuration

AD1[23:0] is constituted by AD1DATAU [7:0] , AD1DATAH [7:0] and AD1DATAL [7:0] where 24-bit Comb Filter outputted data is stored. Comb Filter data format is presented in Figure 12-2.

+FSR/-FSR：Optimum positive and negative measurement range

(For the actual maximum value, please refer to the description of OSR[2:0])

	Equivalent unmeasured signal	AD1[23:0]	
		Hexadecimal	Binary
The output format is two complement	ΔVR_I	7FFFFF	0111-1111 1111-1111 1111-1111
	$\Delta VR\_I \times \frac{1}{2^{19}}$	000001	0000-0000 0000-0000 0000-0001
	0	000000	0000-0000 0000-0000 0000-0000
	$-\Delta VR\_I \times \frac{1}{2^{19}}$	800000	1000-0000 0000-0000 0000-0000
	$-\Delta VR\_I$	FFFFFF	1111-1111 1111-1111 1111-1111

Table 12-2 AD1[23:0] and Input Signal Correlation

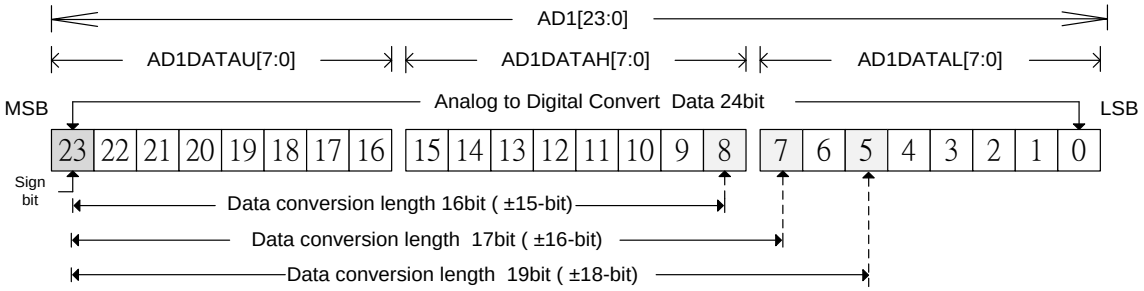


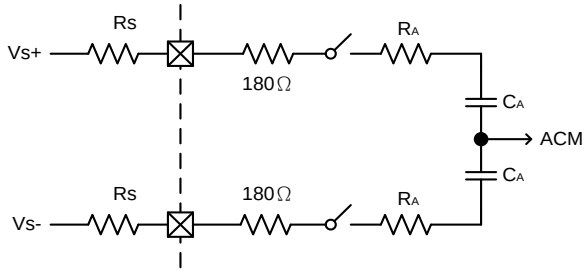
Figure 12-4 AD1[23:0] Resolution Chart

12.1.1.4. ADC Usage Note

- When starting ΣΔADC, must be set ADC Common Voltage(ACM), may be selected VCMS = 0b (ACM = VDDA / 2) or VCMS = 1b (ACM = 1.2V).
- If the internal VDDA voltage regulation is enabled,in addition to ENLDO=1b,LDOPL=1b is also required for normal operation.
- LDOPL bit and VCMS bit are related.
  If LDOPL=1b,VCMS can choose VCMS=0b(ACM=VDDA/2),or VCMS=1b(ACM=1.2V) to use;
  If LDOPL=0b, VCMS can only choose VCMS=1b(ACM=1.2V) to use.
- If VDDA is selected an external input, must set ENLDO=0b to turn off LDO,switch LDOM=00b (high impedance),to allow external input voltage.if LDOPL=0b, the pull down resistance can be turned off to save power.therefore, after setting LDOPL=0b,need to set ADC Common Voltage, VCMS=1b(ACM=1.2V)

12.2. Analog Channel Input Characteristics

ΣΔADC adopts switched capacitor circuit to process analog signals. When input buffer is not used, in order to acquire accurate sampling capacitor voltage value, the highest output impedance of input signal must be confined. Moreover, it will have impeditive interrelation between ΣΔADC sampling frequency and signal amplification.



Rs = External signal source resistance
Vs+ =Positive external signal source
Vs- =Negative external signal source
RA = Sample channel resistance
CA = Sampling capacitance

Figure 12-5 AIx Input Capatitor and Impedance Module

As Figure 12-5 illustrated, if input signal does not pass through buffer, further consideration of input signal impedance, Rs and ΣΔADC sampling frequency, ADC_CK and parasitical resistor RA, capacitor CA effect must also been taken into account.Related calculation is given by :

Equation 12-3

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

ts：ΣΔADC shortest sampling time

ENOB：Expected ΣΔADC effective bit

Gain：(ΣΔAD Gain)

Equation 12 -4

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

Fs：ΣΔADC shortest sampling frequency

ΣΔADC is composed by PGA and ΣΔAD, these two parts have separate RA and CA value in design. The shortest sampling time, ts are calculated by direct and input signal matching consideration.

ΣΔAD Gain	CA	RA
x1/2		
x1	0.5pF	10kΩ
x2	1pF	10kΩ
x4	2pF	10kΩ
x8	4pF	5kΩ

Table 12-3(a) ΣΔADC Gain and RA and CA Relation

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

Table 12-3(b) VR Gain and R_A and C_A Relation

ΣΔADC is mainly applied to low frequency signal measurement. Nevertheless, unmeasured signal includes much more high frequency noise in the real world. Based on signal sampling theory, any high frequency noise that exceeds sampling frequency will produce zero point drift and low frequency noise. Furthermore, it will cause measurement deviation.

Hence, it is suggested to add on 10nF~100nF filter capacitor in IC differential unmeasured signal and voltage reference end to strengthen measurement accuracy.

12.3. Absolute Temperature Sensor,TPS

Absolute temperature sensor is composed by diode (BJT). Its voltage signal to temperature change is a curve that passes through 0°K that equips with the following features.

- Temperature sensor in ambient temperature 0°K, its output voltage: $V_{TPS@0K}$=0V.
- Through measurement method, ADC bias ($V_{ADC-OFFSET}$) and BJT asymmetry (IS1≠IS2) can be offset automatically.
- Single point temperature calibration.

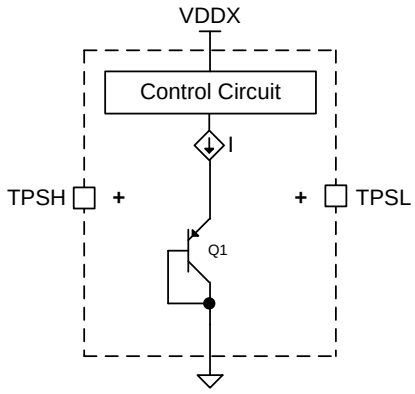


Figure 12-6 Block Diagram of Absolute Temperature Sensor Application

12.3.1. TPS Initial Configuration and Calculation

When ΣΔADC is enabled, the TPS function is automatically enabled immediately.

Configuring input signal selector, INP[1:0] and INN[1:0] as INP=[10]、INN=[10] to measure voltage signal, V_{TPS0}. Configuring INP=[11]、INN=[11] to measure voltage signal, V_{TPS1}. Recommend removal of offset when doing chopper,, TPSLCH control bit must be different, such as when measuring V_{TPS0} TPSCH = 0b, when the measurement V_{TPS1} TPSCH = 1b.

Under the same temperature TA(°C), after ΣΔADC measured the value of V T_{PS0} and V_{TPS1}, add the value together and get the mean, TPS corresponding voltage value, $V_{TPS@TA}$ can be acquired.

The response of TPS output voltage V_{TPS} to temperature change is a linear curve, thus, the gain, GTPS(or called as slope) is derived.

Equation 12 -5 TPS Gain

$$G_{TPS} = \frac{V_{TPS@T_A} - V_{TPS@0K}}{(273.15 + T_{offset} + T_A) - (0)} = \frac{V_{TPS@T_A}}{289.15 + T_A}$$

12.3.2. TPS Example Description

(1) Set INP=0011b=TS1、INN=0011b=TS1、AD1CN5[TPSCH]=1b、AD1CN5[ENTPS]=1b, ADC measured a digital code, V_{TPS0}Code.

(2) Set INP=0010b=TS0、INN=0010b=TS0、AD1CN5[TPSCH]=0b、AD1CN5[ENTPS]=1b, ADC measured a digital code, V_{TPS1}Code.

(3) Calculate V_{TPS}Code=(V_{TPS0}Code + V_{TPS1}Code)/2, this move can erase Temperature Sensor Offset.

(4) Supposed that one point was calculated at 25℃, and then V_{TPS}Code@25℃ can be obtained.Due to the fact that there is a level shift, an offset will be added, Temperature curve slope, G can be gained as follows:

$$G = \frac{V_{TPS}Code@\ 25^oC}{25 + 273.15 + T_{OS}}$$ ， T_{OS} is offset, about 11K。

(5) Supposed that the temperature to be measured is T_X℃，then we can gained：

$$T_X = \frac{V_{TPS}Code@T_X{}^oC}{G} - [273.15 + T_{OS}] \quad {}^oC$$

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						0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
026H	INTF0	-		ADIF						.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
034H	PWRCN2	ENPUMP	VGGS	CHP_CKS[1:0]		ENFIR	LDOPL	ENTPS	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
03BH	AD1CN0	ENAD1	ENCH	ENINXCH	VREGN	OSR[2:0]			CMFR	000. 0000	uuu. uuuu	*,*,*,*,*,*,*,*
03CH	AD1CN1	ENACM	VCMS	VCINS	TPSCH	TPSCP	ADGN[2:0]			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03DH	AD1CN2	FilterN[1:0]		-	DAFM	DCSET[3:0]				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03EH	AD1CN3	SAD1FP[3:0]				-	SAD1FN[2:0]			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
03FH	AD1CN4	AD1RHBUF	AD1RLBUF	AD1IPBUF	AD1INBUF	INX[1:0]		VRIS	INIS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
040H	AD1CN5	SAD1RH[2:0]			SAD1RL[2:0]			SAD1I[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
063H	AD1DATAU	AD1[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
064H	AD1DATAH	AD1[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
065H	AD1DATAL	AD1[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r

Table 12-1 ΣΔADC Register

INTE0/INTF0: Please refer to Interrupt Chapter

PWRCN2: Power System Control Register 2

Bit	Name	Description
Bit5~4	CHP_CKS[1:0]	ADC chop clock frequency select

CHP_CKS[1:0]	Description
00	divdie by 128(Default),In DC measurement recommended settings
01	divide by 16
10	divide by 8
11	Not chop，In AC measurement recommended settings

AD1CN0: ΣΔADC Control Register 0

Bit	Name	Description
Bit7	ENAD1	ΣΔADC Enable Control <0> Disable <1> Enable
Bit6	ENCH[0]	ADC Chopper Controller <0> Disable (Default) <1> Enable Note: Chopper result: 1 bit output XOR chopper clock.must set ENINXCH first, and then enable ENCH last.
Bit5	ENINXCH[0]	Control ADC input end INX[1:0] automatic switch <0> Disable, INX keeps user settings (Default) <1> Enable automatic switching; respectively control INX[1:0]=00b and INX[1:0]=11b to switch alternately.
Bit4	VREGN	ADC Reference Gain Setting

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td><0> x1
<1> x1/2</td></tr>
<tr><td>Bit3~1</td><td>OSR[2:0]</td><td>ΣΔADC Over-sampling rate frequency eliminator
<table>
<tr><th>OSR[2:0]</th><th>OSR1</th><th>Comb filter Order</th><th>+Max</th><th>-Min</th></tr>
<tr><td>000</td><td>32</td><td>3rd</td><td>3FFFFH</td><td>40000H</td></tr>
<tr><td>001</td><td>64</td><td>3rd</td><td>3FFFFH</td><td>40000H</td></tr>
<tr><td>010</td><td>128</td><td>3rd</td><td>3FFFFH</td><td>40000H</td></tr>
<tr><td>011</td><td>256</td><td>3rd</td><td>3FFFFH</td><td>40000H</td></tr>
<tr><td>100</td><td>7680</td><td>2nd</td><td>38401H</td><td>47C00H</td></tr>
<tr><td>101</td><td>15360</td><td>2nd</td><td>38401H</td><td>47C00H</td></tr>
<tr><td>110</td><td>30720</td><td>2nd</td><td>38401H</td><td>47C00H</td></tr>
<tr><td>111</td><td>61440</td><td>2nd</td><td>38401H</td><td>47C00H</td></tr>
</table>
※ When OSR1 is 32, ADC Chopper is prohibited
※ The maximum value(Comb Filter Gain Factor) of ADC output code will be different under different OSR settings, as shown in the table above:</td></tr>
<tr><td>Bit0</td><td>CMFR</td><td>ΣΔADC and Comb Filter Reset Controller
<0> Not reset
<1> Reset; Reset immediately after writing,After reset, the hardware will automatically give up 1~2 AD conversions according to the Comb filter.</td></tr>
</table>

AD1CN1: ΣΔADC Control Register 1

Bit	Name	Description
Bit7	ENACM	ADC Common Mode Voltage <0> Disable <1> Enable
Bit6	VCMS	VCM buffer selection control bit <0> through ADC internal buffer <1> through ADC external buffer (Can be connected to PAD)
Bit5	VCINS	ADC Common Voltage(ACM). <0> 1.2V <1> VDDA/2 ,At the same time, LDOPL must be set to "1".
Bit4	TPSCH	TPS output voltage reverse control <0> normal <1> Reverse
Bit3	TPSCP	TPS Chopper Controller <0> Disable

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td><1> Enable</td></tr>
<tr><td>Bit2~0</td><td>ADGN[2:0]</td><td>AD Gain Setting
<table>
<tr><th>ADGN[2:0]</th><th>Gain</th><th>ADGN[2:0]</th><th>Gain</th></tr>
<tr><td>000</td><td>RSVD</td><td>100</td><td>x4</td></tr>
<tr><td>001</td><td>x1/2</td><td>101</td><td>x8</td></tr>
<tr><td>010</td><td>x1</td><td>110</td><td>RSVD</td></tr>
<tr><td>011</td><td>x2</td><td>111</td><td>RSVD</td></tr>
</table></td></tr>
</table>

AD1CN2: ΣΔADC Control Register 2

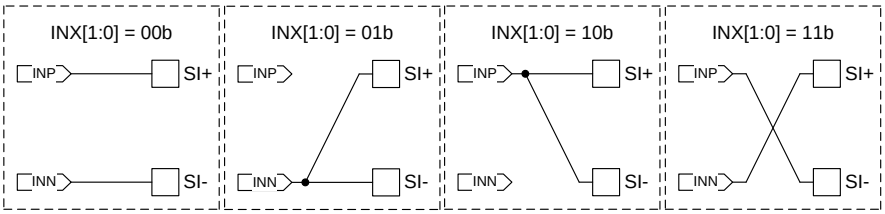
<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~6</td><td>FilterN[1:0]</td><td>Delay setting after ADC channel switching
<table>
<tr><th>FilterN[1:0]</th><th>Delay setting</th></tr>
<tr><td>00</td><td>After channel switching, automatically delay 64* ADC Clock time</td></tr>
<tr><td>01</td><td>After channel switching, automatically delay 128* ADC Clock time</td></tr>
<tr><td>10</td><td>After channel switching, automatically delay 256* ADC Clock time</td></tr>
<tr><td>11</td><td>After channel switching, automatically delay 512* ADC Clock time</td></tr>
</table></td></tr>
<tr><td>Bit4</td><td>DAFM[0]</td><td>Comb filter output data format.
<0> Normal data output (default)
<1> Chopper Result data output. (ADC1 + (ADC2))/2, the next one is: (ADC2 + ADC3)/2...and so on.</td></tr>
<tr><td>Bit3~0</td><td>DCSET[3:0]</td><td>SI± Bias Adjuster
<table>
<tr><th>DCSET[3:0]</th><th>Offset</th><th>DCSET[3:0]</th><th>Offset</th></tr>
<tr><td>0000</td><td>+0*(REFP – REFN)</td><td>1000</td><td>-0*(REFP – REFN)</td></tr>
<tr><td>0001</td><td>+1/8*(REFP – REFN)</td><td>1001</td><td>-1/8*(REFP – REFN)</td></tr>
<tr><td>0010</td><td>+2/8*(REFP – REFN)</td><td>1010</td><td>-2/8*(REFP – REFN)</td></tr>
<tr><td>0011</td><td>+3/8*(REFP – REFN)</td><td>1011</td><td>-3/8*(REFP – REFN)</td></tr>
<tr><td>0100</td><td>+4/8*(REFP – REFN)</td><td>1100</td><td>-4/8*(REFP – REFN)</td></tr>
<tr><td>0101</td><td>+5/8*(REFP – REFN)</td><td>1101</td><td>-5/8*(REFP – REFN)</td></tr>
<tr><td>0110</td><td>+6/8*(REFP – REFN)</td><td>1110</td><td>-6/8*(REFP – REFN)</td></tr>
<tr><td>0111</td><td>+7/8*(REFP – REFN)</td><td>1111</td><td>-7/8*(REFP – REFN)</td></tr>
</table></td></tr>
</table>

AD1CN3: ΣΔADC Control Register 3

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~6</td><td>SAD1FP[3:0]</td><td>
HY17P60B :Pre-Filter Positive Input Signal Selector
<table>
<tr><th>SAD1FP[3:0]</th><th>Filter Positive Input</th><th>SAD1FP[3:0]</th><th>Filter Positive Input</th></tr>
<tr><td>0000</td><td>-</td><td>1000</td><td>-</td></tr>
<tr><td>0001</td><td>-</td><td>1001</td><td>PB2</td></tr>
<tr><td>0010</td><td>-</td><td>1010</td><td>PB4</td></tr>
<tr><td>0011</td><td>OP1O</td><td>1011</td><td>-</td></tr>
<tr><td>0100</td><td>VDDA</td><td>1100</td><td>-</td></tr>
<tr><td>0101</td><td>REFO</td><td>1101</td><td>-</td></tr>
<tr><td>0110</td><td>-</td><td>1110</td><td>-</td></tr>
<tr><td>0111</td><td>PB0</td><td>1111</td><td>-</td></tr>
</table>
HY17P68 :Pre-Filter Positive Input Signal Selector
<table>
<tr><th>SAD1FP[3:0]</th><th>Filter Positive Input</th><th>SAD1FP[3:0]</th><th>Filter Positive Input</th></tr>
<tr><td>0000</td><td>SENSE</td><td>1000</td><td>PB1</td></tr>
<tr><td>0001</td><td>FB</td><td>1001</td><td>PB2</td></tr>
<tr><td>0010</td><td>RLU</td><td>1010</td><td>PB4</td></tr>
<tr><td>0011</td><td>OP1O</td><td>1011</td><td>PB5</td></tr>
<tr><td>0100</td><td>VDDA</td><td>1100</td><td>PB6</td></tr>
<tr><td>0101</td><td>REFO</td><td>1101</td><td>PB7</td></tr>
<tr><td>0110</td><td>VREF</td><td>1110</td><td>RSVD</td></tr>
<tr><td>0111</td><td>PB0</td><td>1111</td><td>RSVD</td></tr>
</table>
</td></tr>
<tr><td>Bit2~0</td><td>SAD1FN[2:0]</td><td>
HY17P60B :Pre-Filter Negative Input Signal Selector
<table>
<tr><th>SAD1FN[2:0]</th><th>Filter Negative Input</th><th>SAD1FN[2:0]</th><th>Filter Negative Input</th></tr>
<tr><td>000</td><td>-</td><td>100</td><td>PB2</td></tr>
<tr><td>001</td><td>-</td><td>101</td><td>PB3</td></tr>
<tr><td>010</td><td>VSS</td><td>110</td><td>PB4</td></tr>
<tr><td>011</td><td>AGND</td><td>111</td><td>-</td></tr>
</table>
HY17P68 :Pre-Filter Negative Input Signal Selector
<table>
<tr><th>SAD1FN[2:0]</th><th>Filter Negative Input</th><th>SAD1FN[2:0]</th><th>Filter Negative Input</th></tr>
<tr><td>000</td><td>SENSE</td><td>100</td><td>PB2</td></tr>
<tr><td>001</td><td>RLU</td><td>101</td><td>PB3</td></tr>
</table>
</td></tr>
</table>

			010	VSS	110	PB4	
			011	AGND	111	PB5	

AD1CN4: ΣΔADC Control Register 4

Bit	Name	Description
Bit7	AD1RHBUF	Can configure whether to let ADC positive reference signal passes through buffer <0> Disable(default) <1> Enable
Bit6	AD1RLBUF	can configure whether to let ADC negative reference signal passes through buffer <0> Disable (default) <1> Enable
Bit5	AD1IPBUF	Can configure whether to let ADC positive input signal passes through buffer. <0> Disable (default) <1> Enable
Bit4	AD1INBUF	Can configure whether to let ADC negative input signal passes through buffer. <0> Disable (default) <1> Enable
Bit3~2	INX	SI± Input Signal Convertor <11> INP→SI-,INN→ SI+ <10> INP→ SI+ & SI, INN floating <01> INP floating, INN→ SI+ & SI- <00> INP→ SI+,INN→ SI- 
Bit1	VRIS	VR± Input Signal Short Selector <0> Not short <1> Short (Reserved for testing by the manufacturer, not recommended)
Bit0	INIS	SI± Input Signal Short Selector <0> Not short <1> Short (Reserved for testing by the manufacturer, not recommended)

AD1CN5: ΣΔADC Control Register 5

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~5</td><td>SAD1RH[2:0]</td><td>
HY17P60 :AD1 Reference Voltage Positive End Input Selection Control Bit
<table>
<tr><th>SAD1RH[2:0]</th><th>SADFRP</th><th>SAD1RH[2:0]</th><th>SADFRP</th></tr>
<tr><td>000</td><td>-</td><td>100</td><td>-</td></tr>
<tr><td>001</td><td>REFO</td><td>101</td><td>VDDA</td></tr>
<tr><td>010</td><td>-</td><td>110</td><td>AGND</td></tr>
<tr><td>011</td><td>PB4</td><td>111</td><td>-</td></tr>
</table>
HY17P68 :AD1 Reference Voltage Positive End Input Selection Control Bit
<table>
<tr><th>SAD1RH[2:0]</th><th>SADFRP</th><th>SAD1RH[2:0]</th><th>SADFRP</th></tr>
<tr><td>000</td><td>FB</td><td>100</td><td>RLU</td></tr>
<tr><td>001</td><td>REFO</td><td>101</td><td>VDDA</td></tr>
<tr><td>010</td><td>VREF</td><td>110</td><td>AGND</td></tr>
<tr><td>011</td><td>PB4</td><td>111</td><td>RSVD</td></tr>
</table>
</td></tr>
<tr><td>Bit4~2</td><td>SAD1RL[2:0]</td><td>
HY17P60 :AD1 Reference Voltage Negative End Input Selection Control Bit
<table>
<tr><th>SAD1RL[2:0]</th><th>SADFRN</th><th>SAD1RL[2:0]</th><th>SADFRN</th></tr>
<tr><td>000</td><td>-</td><td>100</td><td>-</td></tr>
<tr><td>001</td><td>AGND</td><td>101</td><td>VSS</td></tr>
<tr><td>010</td><td>PB3</td><td>110</td><td>-</td></tr>
<tr><td>011</td><td>-</td><td>111</td><td>-</td></tr>
</table>
HY17P68 :AD1 Reference Voltage Negative End Input Selection Control Bit
<table>
<tr><th>SAD1RL[2:0]</th><th>SADFRN</th><th>SAD1RL[2:0]</th><th>SADFRN</th></tr>
<tr><td>000</td><td>RLU</td><td>100</td><td>FB</td></tr>
<tr><td>001</td><td>AGND</td><td>101</td><td>VSS</td></tr>
<tr><td>010</td><td>PB3</td><td>110</td><td>VREF</td></tr>
<tr><td>011</td><td>PB5</td><td>111</td><td>RSVD</td></tr>
</table>
</td></tr>
</table>

Bit1~0	SAD1I[1:0]	**HY17P60 :**AD1 Signal Input Selection Control Bit

SAD1I[1:0]	AD1IP	AD1IN
00	AD1FP	AD1FN
01	-	-
10	TS1P	TS2N
11	TS1N	TS2P

HY17P68 :AD1 Signal Input Selection Control Bit

SAD1I[1:0]	AD1IP	AD1IN
00	AD1FP	AD1FN
01	FB	RLU
10	TS1P	TS2N
11	TS1N	TS2P

AD1[18:0] Analog to Digital Conversion Register

AD1DATAU[7:0] AD1 Analog to Digital Conversion Data Register(AD1[18:11])

AD1DATAH[7:0] AD1 Analog to Digital Conversion Data Register (AD1[10:3])

AD1DATAL[7:5] AD1 Analog to Digital Conversion Data Register (AD1[2:0]), Must be read first when using.

13. Digital Signal Processing,DSP

Digital Signal Processing (Digital Signal Processing · hereafter referred to as DSP) includes digital calculation functions such as Low Pass Filter, RMS Coverter, and Peak Hold.

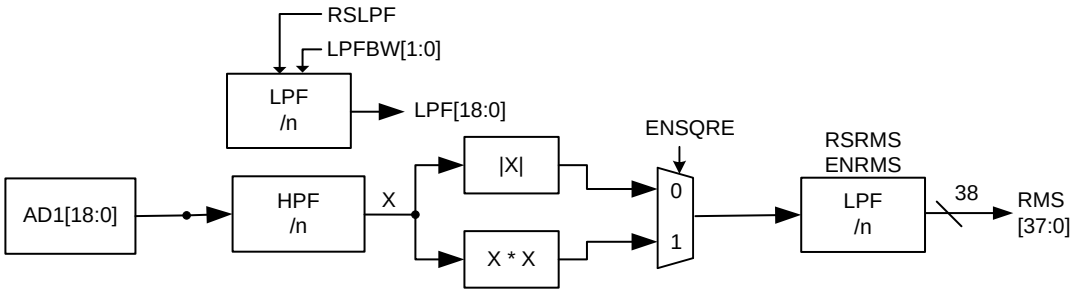


Figure 13-1 HY17P60B DSP Block Diagram

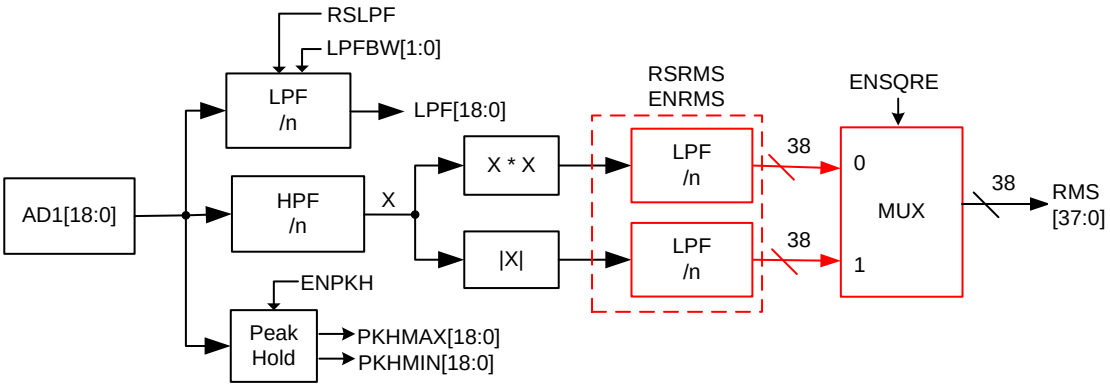
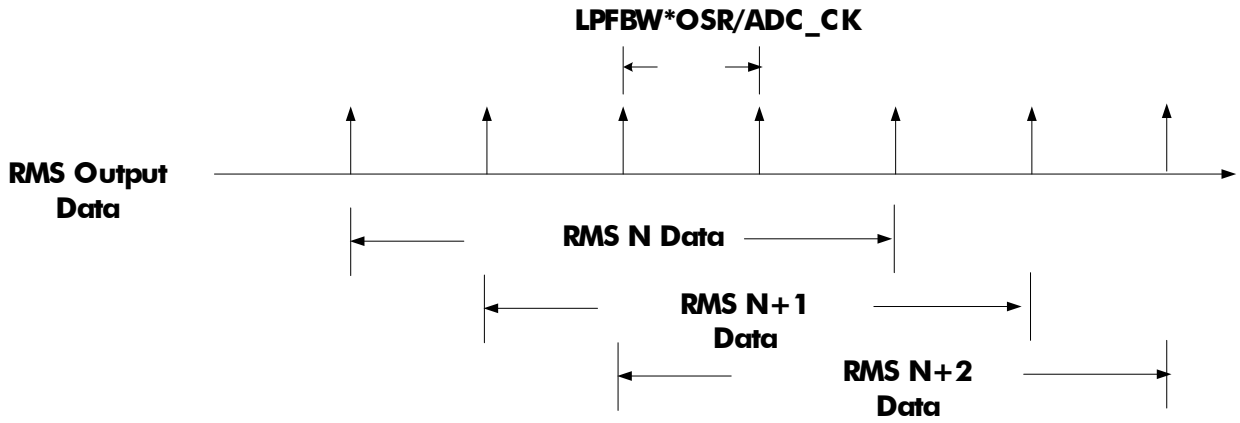


Figure 13-2 HY17P68 DSP Block Diagram

13.1. Low Pass Filter & RMS Coverter

AC measurement of HY17P6x used internal digital signal process unit to calculate its AC value when ADC is under fast output mode. Configure ENSQRE separately to calculate its true RMS value or the mean of absolute value. Before calculating its AC value, it will pass through HPF (High Pass Filter) to remove DC. If its DC signal is required, users can read LPF<18:0>. AC signal after square or absolute value, will pass through Sinc^4 Low Pass Filter to gain RMS<37:0> output. For true RMS value measurement, MCU is required to implement radical. The output time sequence is as follows:



13.2. Peak Hold (Only for HY17P68)

Peak Hold can store the Maximum /minimum ADC output value to PKHMAX and PHHMIN registers.

13.3. Register Description- DSP

"-"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								0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
025H	INTE2						RMSIE	LPFIE		.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
028H	INTF2						RMSF	LPFF		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
034H	PWRCN2			-	-	ENFIR			-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
041H	RMSCN	ENRMS	ENLPF	ENSQRE	LPFBW[1:0]		ENPKH	RSLPF	RSRMS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
055H	PKHMAXU	PKHMAX[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
056H	PKHMAXH	PKHMAX[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
057H	PKHMAXL	PKHMAX[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
058H	PKHMINU	PKHMIN[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
059H	PKHMINH	PKHMIN[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05AH	PKHMINL	PKHMIN[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05BH	RMSDATA4	RMS[37:30]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05CH	RMSDATA3	RMS[29:22]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05DH	RMSDATA2	RMS[21:14]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05EH	RMSDATA1	RMS[13:6]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
05FH	RMSDATA0	RMS[5:0]						-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
060H	LPFDATAU	LPF[18:11]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
061H	LPFDATAH	LPF[10:3]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
062H	LPFDATAL	LPF[2:0]			-	-	-	-	-	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r

Table 13-1 DSP Register

INTE0/INTE2/INTF2: Please refer to Interrupt Chapter

PWRCN2: PWR Control Register 2

Bit	Name	Description
Bit3	ENFIR	AC Frequency Compensation Enable Controller Bit <0> Disable(Default) <1> Enable

RMSCN: RMS Control Register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENRMS</td><td>RMS Converter Enable Controller Bit
<0> Disable, and clear RMS[37:0] to 0
<1> Enable</td></tr>
<tr><td>Bit6</td><td>ENLPF</td><td>Low Pass Filter Enable Controller Bit
<0> Disable, and clear LPF[18:0] to 0
<1> Enable</td></tr>
<tr><td>Bit5</td><td>ENSQRE</td><td>RMS Converter Output Data Selection Control Bit
<0> $\mathrm{RMS}[37:0]=\sum\frac{|X|}{N}$, The register value is average value(has no positive or negative) , just divide by the calibration value.
<1> $\mathrm{RMS}[37:0]=\sum\frac{X^2}{N}$, The register value is RMS, radical must be processed by MCU software (Square root), and then divide by the calibration value.</td></tr>
<tr><td>Bit4~3</td><td>LPFBW[1:0]</td><td>Set Over Sampling Ratio(OSR4) of Low Pass Filter
Low Pass Filter data output rate=data input rate/OSR4。
<table>
<tr><th>LPFBW[1:0]</th><th>OSR4</th></tr>
<tr><td>00</td><td>256</td></tr>
<tr><td>01</td><td>512</td></tr>
<tr><td>10</td><td>1024</td></tr>
<tr><td>11</td><td>2048</td></tr>
</table></td></tr>
<tr><td>Bit2</td><td>ENPKH</td><td>Peak Hold Enable Control Bit
<0> Disable, PKHMAX[18:0]=40000h, PKHMIN[18:0]=3FFFFh.
<1> Enable, ADC output to PKHMAX and PHHMIN respectively. If the result is bigger than PKHMAX, then PKHMAX=AD1. If the result is smaller than PKHMIN, then PKHMIN=AD1, or please remain the original value unchanged.</td></tr>
<tr><td>Bit1</td><td>RSLPF</td><td>Reset Low Pass Filter。
<0> Not reset
<1> Reset；Reset immediately after writing</td></tr>
<tr><td>Bit0</td><td>RSRMS</td><td>Reset RMS Low Pass Filter。
<0> Not reset
<1> Reset；Reset immediately after writing</td></tr>
</table>

RMS[37:0]: Output data register of RMS Converter

RMS data output rate=Low Pass Filter data output rate

Supposed that X= AD1 [18: 0] passes through High Pass Filter, N is the OSR of Low Pass Filter, which is configured by LPFBW [2: 0].

RMSDATA4[7:0] RMS Converter Data Register(RMS[37:30])

RMSDATA3[7:0] RMS Converter Data Register (RMS[29:22])

RMSDATA2[7:0] RMS Converter Data Register (RMS[21:14])

RMSDATA1[7:0] RMS Converter Data Register (RMS[13:6])

RMSDATA0[7:2] RMS Converter Data Register (RMS[5:0]), Must be read first when using.

PKHMAX[18:0]: Output Data Register for Peak Hold

PKHMAXU[7:0] Peak Hold Data Register (PKHMAX[18:11])

PKHMAXH[7:0] Peak Hold Data Register (PKHMAX[10:3])

PKHMAXL[7:5] Peak Hold Data Register(PKHMAX[2:0]) , Must be read first when using.

PKHMIN[18:0]: Output Data Register for Peak Hold

PKHMINU[7:0] Peak Hold Data Register (PKHMIN[18:11])

PKHMINH[7:0] Peak Hold Data Register (PKHMIN[10:3])

PKHMINL[7:5] Peak Hold Data Register (PKHMIN[2:0]) , Must be read first when using.

LPF[18:0]: Output Data Register of AD1 Low Pass Filter

Note that the data is the accumulated average value of AD1, not the low-pass filter function of AC measurement.

LPFDATAU[7:0] Low Pass Filter Data Register (LPF[18:11])

LPFDATAH[7:0] Low Pass Filter Data Register (LPF[10:3])

LPFDATAL[7:5] Low Pass Filter Data Register (LPF[2:0]) , Must be read first when using.

14. Operational Amplifier,OPAMP

The internal operational amplifier(Operational Amplifier, hereafter referred to as OPAMP) and external components can from Unity-Gain Buffer applications and Non-Inverting Amplifier applications.

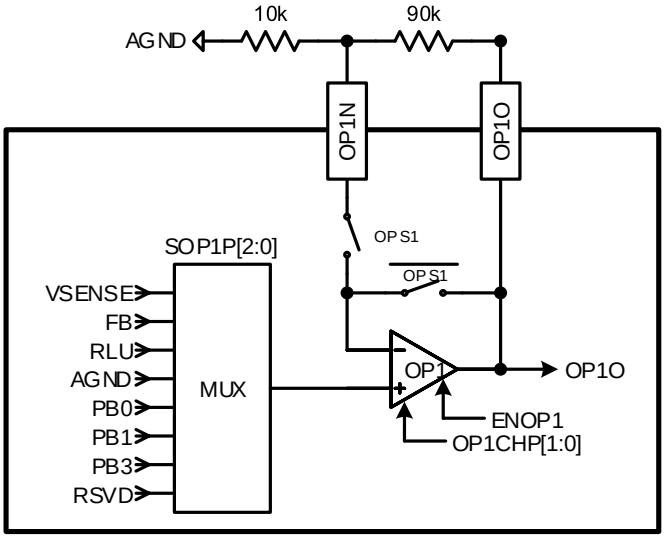


Figure 14-1 OPAMP Block Diagram

14.1. Register Description-OPAMP

"-"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**
066H	OP1CN0	ENOP1	SOP1P[2:0]			OP1CHOP[1:0]		HS	OPS1	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 14-1 OPAMP Register

OP1CN0: OPAMP1 Control Register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENOP1</td><td>OP1 Enable Control
<0> Disable
<1> Enable</td></tr>
<tr><td>Bit6~4</td><td>SOP1P[2:0]</td><td>HY17P60B :OP1 Positive Input Selector
<table>
<tr><th>SOP1P[2:0]</th><th>OP1 Positive Input</th><th>SOP1P[2:0]</th><th>OP1 Positive Input</th></tr>
<tr><td>000</td><td>-</td><td>100</td><td>PB<0></td></tr>
<tr><td>001</td><td>-</td><td>101</td><td>-</td></tr>
<tr><td>010</td><td>-</td><td>110</td><td>PB<3></td></tr>
<tr><td>011</td><td>AGND</td><td>111</td><td>-</td></tr>
</table>
HY17P68 :OP1 Positive Input Selector
<table>
<tr><th>SOP1P[2:0]</th><th>OP1 Positive Input</th><th>SOP1P[2:0]</th><th>OP1 Positive Input</th></tr>
<tr><td>000</td><td>SENSE</td><td>100</td><td>PB0</td></tr>
<tr><td>001</td><td>FB</td><td>101</td><td>PB1</td></tr>
<tr><td>010</td><td>RLU</td><td>110</td><td>PB3</td></tr>
<tr><td>011</td><td>AGND</td><td>111</td><td>RSVD</td></tr>
</table></td></tr>
<tr><td>Bit3~2</td><td>OP1CHOP[1:0]</td><td>OP1 Chopper Clock Selection Control Bit
<table>
<tr><th>OP1CHOP[1:0]</th><th>CHOP_CLK</th></tr>
<tr><td>00</td><td>Disable</td></tr>
<tr><td>01</td><td>ADC_CK ÷ 512(OP Auto Chopper Mode)</td></tr>
<tr><td>10</td><td>ADC_CK ÷256(OP Auto Chopper Mode)</td></tr>
<tr><td>11</td><td>Reserve</td></tr>
</table>
※ If the input is a higher resistance component, it is not recommended to use OP Auto Chopper, which will cause non-return to zero.
※ HY17P60 does not have OP Auto Chopper mode</td></tr>
<tr><td>Bit1</td><td>HS</td><td>OP1 High Speed Input Mode
<0> Disable
<1> Enable (This bit needs to be turned on for OPA to be used normally)</td></tr>
<tr><td>Bit0</td><td>OPS1</td><td>OP1 Negative Input Selection Control Bit
<0> Connect to OP1O pin for Unity-Gain Buffer
<1> Connect to OP1N pin for Non-Inverting Amplifier)。</td></tr>
</table>

15. Windows Comparator(only HY17P68)

Windows Comparator is composed of two analog comparators and input multiplexers,The CMPH and CMPL comparators are combined into a hysteresis window comparator.The positive input terminal VRHCMP of the CMPH and the negative input terminal VRLCMP of the CMPL are respectively the high/low comparison potentials of a comparator with a hysteresis window, which can be selected through the input multiplexer. CMPO is the output of window comparator,CMPH and CMPL respective comparator output is CMPHO and CMPLO.

Windows Comparator is mainly used for frequency measurement, short circuit test and capacitance measurement.

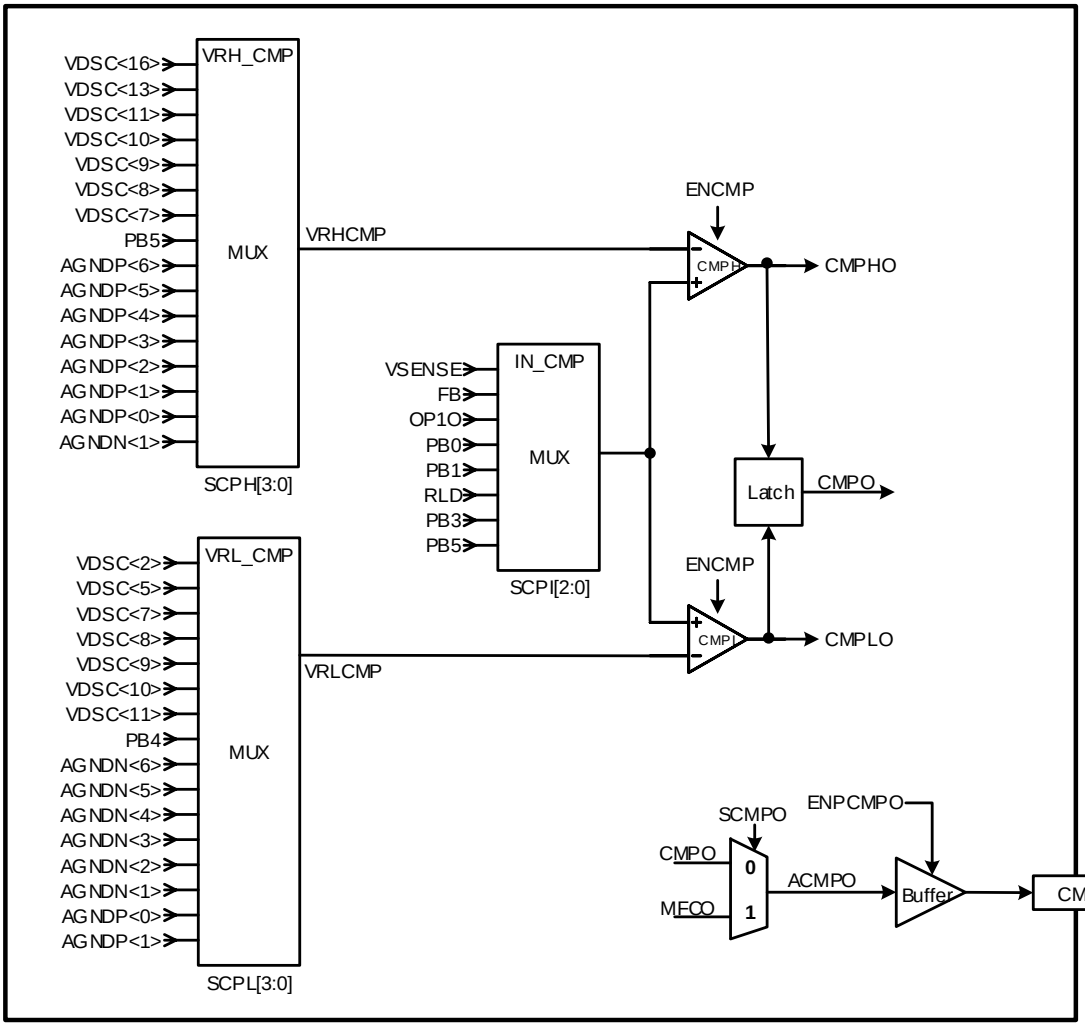


Figure 15-1 Windows Comparator Block Diagram

15.1. Register Description-Windows Comparator

"-"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								0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
025H	INTE2	-	CMPOIE	CMPHOIE	CMPLOIE	CTBOVE	RMSIE	LPFIE	BOR2IE	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
028H	INTF2	PCNTI	CMPIF	CMPHIF	CMPLIF	CTBOV	RMSF	LPFF	BOR2IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
033H	PWRCN1		ENCMP							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
044H	NET2	SCMPRH[3:0]				SCMPRL[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
045H	NET3	SCMPI[2:0]			-	CMPO	CMPHO	CMPLO	-	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 15-1 Windows Counter Control Register

INTE0/INTE1/INTF0/INTF1: Please refer to Interrupt Chapter

PWRCN1: PWR Control Register 1

Bit	Name	Description
Bit6	ENCMP	Windows Comparator Enable control <0> Disable <1> Enable

NET2: Measurement network setting control register 2

Bit	Name	Description
Bit7~4	SCMPRH[3:0]	Window Comparator VRHCMP Selection Control Bit
Bit3~0	SCMPRL[3:0]	Window Comparator VRLCMP Selection Control Bit

SCMPRH[3:0]	VRHCMP	SCMPRH[3:0]	VRHCMP
0000	VDSC<16>	1000	AGNDP<6>
0001	VDSC<13>	1001	AGNDP<5>
0010	VDSC<11>	1010	AGNDP<4>
0011	VDSC<10>	1011	AGNDP<3>
0100	VDSC<9>	1100	AGNDP<2>
0101	VDSC<8>	1101	AGNDP<1>
0110	VDSC<7>	1110	AGNDP<0>
0111	PB5	1111	AGNDN<1>

SCMPRL[3:0]	VRLCMP	SCMPRL[3:0]	VRLCMP
0000	VDSC<2>	1000	AGNDN<6>
0001	VDSC<5>	1001	AGNDN<5>
0010	VDSC<7>	1010	AGNDN<4>
0011	VDSC<8>	1011	AGNDN<3>
0100	VDSC<9>	1100	AGNDN<2>
0101	VDSC<10>	1101	AGNDN<1>
0110	VDSC<11>	1110	AGNDP<0>
0111	PB4	1111	AGNDP<1>

NET3: Measurement network setting control register 3

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~5</td><td>SCMPI[2:0]</td><td>Window Comparator INCMP Selection Control Bit
<table>
<tr><th>SCMPI[2:0]</th><th>INCMP</th><th>SCMPI[2:0]</th><th>INCMP</th></tr>
<tr><td>000</td><td>SENSE</td><td>100</td><td>PB1</td></tr>
<tr><td>001</td><td>FB</td><td>101</td><td>RLD</td></tr>
<tr><td>010</td><td>OP1O</td><td>110</td><td>PB3</td></tr>
<tr><td>011</td><td>PB0</td><td>111</td><td>PB5</td></tr>
</table></td></tr>
<tr><td>Bit3</td><td>CMPO</td><td>CMPO Status
<0> Low
<1> High</td></tr>
<tr><td>Bit2</td><td>CMPHO</td><td>CMPHO Status
<0> Low
<1> High</td></tr>
<tr><td>Bit1</td><td>CMPLO</td><td>CMPLO Status
<0> Low
<1> High</td></tr>
</table>

16. Frequency Counter

Frequency Counter includes 3 group 24bits Counters, They are CTA, CTB and CTC. CTA and CTC input clock source is SYSCLK,The signal to be tested is the CTB input clock, When the test signal is high, CTC will count.Frequency Counter signals can be chose by ENCNTI as DMM Comparator Network output ACMPO or PT3.6 input PCNTI.

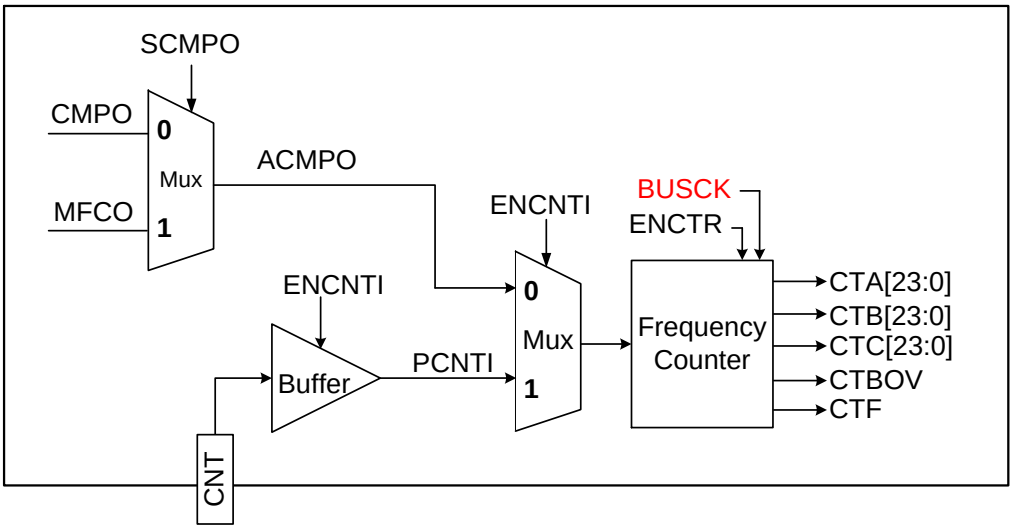


Figure 16-1 Frequency Counter Block Diagram

16.1. Frequency Counter operation calculation as follows：

- (1) Configure ENCTR=0 and CTA<7:0>, CTB<23:0>, CTC<23:0> will be configured as 0.

  (2) Write in counter initial value as CTA<23:8>, then set Gate Time=[1000000h-CTA<23:0>]/F_{SYSCLK}.

  (3) After configure ENCTR=1, start counting when the first positive edge of standby signals come up until CTA<23:0> overflow generated and the first positive edge of standby signals stop counting. The start and stop of counter is the completed cycle of standby signal numbers. The interrupt signal will be appeared when the counter stop counting.

  (4) Read CTA<23:0>, CTB<23:0>, CTC<23:0> and CTBOV.

  (5) If CTBOV=1, it means Gate Time was set to too long and standby signal frequency is too high, so CTB<23:0> overflow generated. If so, the information produced this time is useless. The procedure has to be restarted from step (1) and reset Gate Time, count again.

  (6) If CTBOV=0, it means the information produced this time is useful. The frequency of standby signal, Duty Cycle can be calculated by produced information.

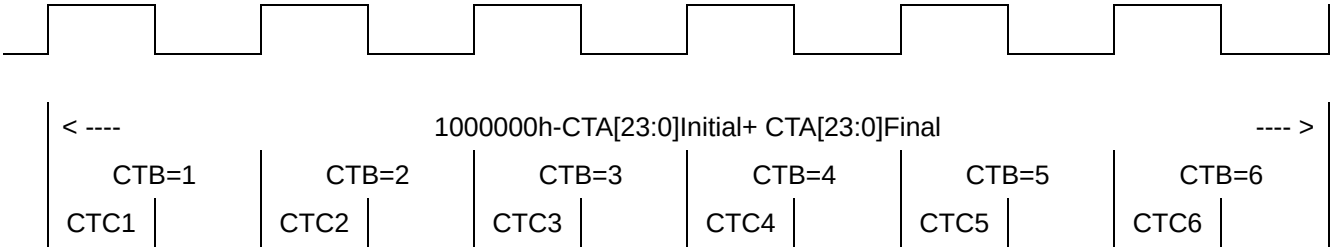
  T (Count time)=[1000000h-CTA<23:0>$_{Initial}$+ CTA<23:0>$_{Final}$]/F_{SYSCLK}

  Standby signals frequency=CTB<23:0>/T

  Standby signal, Duty Cycle=CTC<23:0>/[1000000h-CTA<23:0>$_{Initial}$+ CTA<23:0>$_{Final}$]

  Within it, F_{SYSCLK} is the frequency of SYSCLK, CTA<23:0>$_{Initial}$ is the value before count operated. CTA<23:0>$_{Final}$ is the value occurred after count operated.

16.2. Example of Calculation



Calculation element description (Taking 1kHz / 50% as example)

FSYSCLK：System oscillator frequency, supposed it is 4MHz

CTA<23:0>Initial：Default value before CTA counter, CTA<23:8> program defaults C000h, and CTA<7:0>is cleared as 00h

CTA<23:0>Final : Value after CTA counter, CTA<23:0> Initial value is C00000h; it will be 000760h under 1kHz.

CTB<23:0>：Cycles within the time period, CTA<23:0> Initial value is C00000h, it will be 000419h under 1kHz

CTC<23:0>：High time sum, CTA<23:0> Initial value is C00000h, it will be 20043Ah under Duty 50%.

Count time:

T = [1000000h-CTA[23:0]Initial+ CTA[23:0]Final]/FSYSCLK

=(1000000h-C00000h +000760h)/3D0900h --- >hexadecimal

=(16777216-12582912+1888)/4000000=1.0490 --- >decimal

Standby signals frequency:

Freq = CTB[23:0]/T

= 1049/1.0490=1000 Hz

Standby signal, Duty Cycle:

Duty Cycle = CTC[23:0]/[1000000h-CTA[23:0]Initial + CTA[23:0]Final]

= 20043Ah/400760h --- >hexadecimal

= 2098234/4196192=0.5=50% --- >decimal

16.3. Register Description-Frequency Counter

"-"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					CTIE			0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
025H	INTE2	-	CMPOIE	CMPHOIE	CMPLOIE	CTBOVE	RMSIE	LPFIE	BOR2IE	.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
026H	INTF0	-					CTF			.000 0000	.uuu uuuu	*,*,*,*,*,*,*,*
028H	INTF2	MFCIF	CMPO	CMPHO	CMPLO	CTBOV	RMSF	LPFF	BOR2IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
033H	PWRCN1			ENCNTI	ENCTR					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
045H	NET3				SCMPO	-	-	-	CNTI_IF	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
04CH	CTAU	CTA[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04DH	CTAH	CTA[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04EH	CTAL	CTA[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
04FH	CTBU	CTB[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
050H	CTBH	CTB[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
051H	CTBL	CTB[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
052H	CTCU	CTC[23:16]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
053H	CTCH	CTC[15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
054H	CTCL	CTC[7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r

Table 16-1 Frequency Counter Control Register

INTE0/INTE2/INTF0/INTF2: Please refer to Interrupt Chapter

INTF2: Interrupt Event Flag Register 2

Bit	Name	Description
Bit3	CTBOV	CTB Overflow Flag <0> Not happened <1> Happened, when CTB<23:0> is overflow, CTBOV will be set as 1. If this bit is 1, it means the measurement frequency is high, and the initial value of CTA must be set higher.

PWRCN1: PWR Control Register 1

Bit	Name	Description
Bit5	ENCNTI	Frequency Counter input source selection <0> CMPO <1> PCNTI
Bit4	ENCTR	Frequency Counter Enable Controller <0> Disable, and clear CTA[23:0], CTB[23:0], CTC[23:0] and CTBOV to 0. <1> Enable

NET3: Measurement network setting control register 3

Bit	Name	Description
Bit4	SCMPO	ACMPO signal Source selection <0> CMPO。 <1> MFCO
Bit0	CNTI_IF	Frequency Counter input source <0> CMPO。 <1> PCNTI

CTA[23:0] : Counter A data register, is the total test time

When ENCTR=0, CTA[7:0] will be cleared to 0. CTA[23:8] will not be cleared to 0, only the value can be written through the MCU.

When ENCTR=1, CTA[23:8] can only be written by Frequency Counter.

CTB [23:0]: Counter B data register, the number of cycles within the test time

When ENCTR=0, CTB[23:0] will be cleared to 0.

When ENCTR=1 and interrupt occurred by the time count finished, CTB[23:0] will record the numbers of full cycle of standby signals. The numbers can be used to calculate the frequency of signals.

CTC [23:0]: Counter C data register, which is the sum of High time during the test time

When ENCTR=0, CTC[23:0] will be cleared to 0.

When ENCTR=1 and interrupt occurred by the time count finished, CTC[23:0] will record SYSCLK numbers under the situation of standby signals is high. The numbers can be used to calculate the Duty Cycle of signals.

17. Timer-A1,TMA1

Timer-A1(Timer-A1, hereafter referred to as TMA1) is designed in 8-bit frame. TMA1 can function in Run Mode and Idle Mode.

- Ascending counter
- 8-step overflow value select
- 8Bit overflow comparator
- Overflow generated interrupt event
- Counter value is readable

TMA1 Related Registers：

INTE0	GIE, TA1CIE
INTE1	TA1IE
INTF0	TA1CIF
INTF1	TA1IF
TMA1CN	ENTMA1[0], TMACL1[0], TMAS1[0], DTMA1[2:0]
TMA1R	TMA1R[7:0]
TMA1C	TMA1C[7:0]

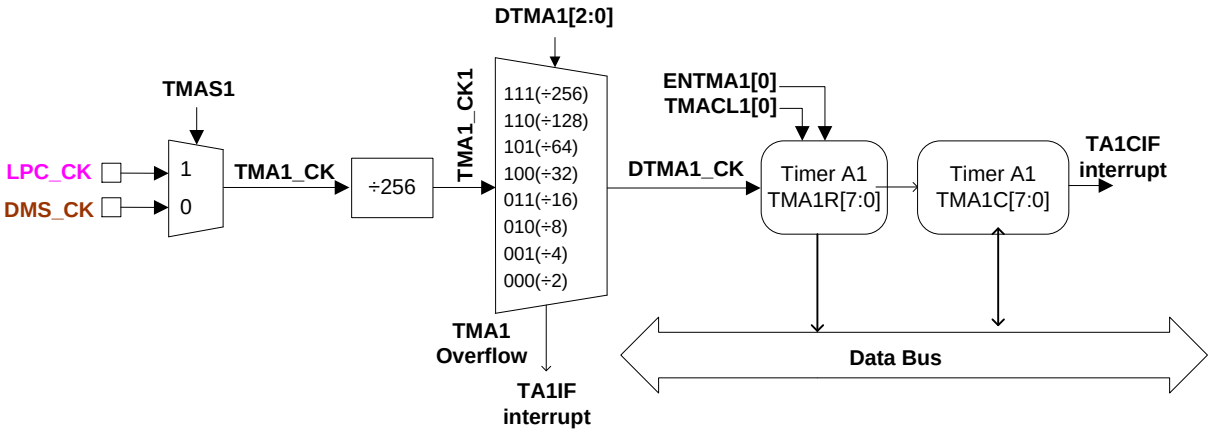


Figure 17-1 TMA1 Block Diagram

- TMA1 Manual：

  Set TMAS1[0] to select the frequency of TMA1_CK, The frequency is reduced by the pre-divider 256 and then input to the DTMA frequency divider.

  Set ENTMA1[0] to <1> to enable TMA1; on the contrary, set <0> is closed and clear TMA1R [7: 0].When the DTMA1[2:0] counting condition is established, an interrupt event is generated and TMA1R[7:0] is incremented by 1.

  When TMA1 interrupt event occurs, TMA1IE[0] must be set to <1> and GIE[0] is set to <1> to have interrupt service TMA1IF[0]. Reading TMA1R[7:0] will not reset the TMA1 counter to zero.

  After the user clears all counters of TMA1 by setting TMA1CL[0] to <1>, TMA1CL[0] is automatically set to <0>.

  TMA1R[7:0] can read the value of the TMA accumulation counter, and can write any value to clear the count value of TMA1R[7:0].

- TMA1 Comparator Operation Instructions:

  Set TMAS1 to select the frequency of TMA1_CK. After pre-dividing 256, it is the frequency source of TMA1_CK and then input the DTMA1 frequency divider.

  Set ENTMA1 <1> to enable TMA1, and clear counters such as TMA1_CK, DTMA1_CK, TMA1R, etc., and start counting from 0; otherwise, set <0> to turn off TMA1.

  The DTMA1[2:0] counting condition is established, an interrupt event (TAxIF) is generated, and TMA1R[7:0] is incremented by 1.

  TMA1 interrupt event TAxIF must be set in TAxIE <1> and GIE set <1> to have interrupt service.

  Reading TMA1R[7:0] will not reset the TMA1 counter to zero.

  After the user clears the TMA1_CK and DTMA1_CK counters by setting TMACL1 <1>, TMACL1[0] is automatically set to <0> by the hardware.

  TMA1R[7:0] can read the value of the TMA1 accumulation counter, and can be regarded as clearing the TMA1R[7:0] count value with a write action, and recalculate from TMA1R[7:0]=0.

  TMA1C[7:0] is the compare point register of TMA1, which can be read and written.

  When BOR/POR occurs, TMA1C[7:0] will be set to 0. When the value of TMA1R[7:0] is accumulated to the same value as TMA1C[7:0], the TAxCIF flag is set to 1. Similarly, TA1CIF must be set in TA1CIE <1> and GIE set <1> to have interrupt service.

  When designing TA1IF as a 1-second interrupt, if the requirement is 60-second interrupt wake-up, you can set TA1CIF as a 60-second interrupt, and enable TA1CIE interrupt request to wake up the chip.

17.1. 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**
068H	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]					0000 0000	u0uu uuuu	*,rw1,*,* *,*,*,*
069H	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
06AH	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0

Table 17-1 TMA1 Control Register

INTE0/INTF0：Please refer to the Interrupt chapter

TMA1CN: TMA1 Control Register

Bit	Name	Description
Bit7	ENTMA1	Timer-A1 Enable Controller <0> Disable <1> Enable，and clear counters (TMA1_CK、DTMA1_CK、TMA1R)
Bit6	TMACL1	TMA1 Counter Control <0> TMA1 divider counter <1> TMA1 divider count is reset to zero. TMACL1 set to <1> , and clear countrs such as TMA1_CK、DTMA1_CK,TMACL1 is automatically set to <0> by the hardware.
Bit5	TMAS1	TMA1_CK　operating frequency selector <0> DMS_CK <1> LPC_CK
Bit4~2	DTMA1[2:0]	TMA1_CK1 operating frequency selector

DTMA[2:0]	Frequency division selector	DTMA[2:0]	Frequency division selector
000	TMA1_CK ÷ 2	100	TMA1_CK ÷ 32
001	TMA1_CK ÷ 4	101	TMA1_CK ÷ 64
010	TMA1_CK ÷ 8	110	TMA1_CK ÷ 128
011	TMA1_CK ÷ 16	111	TMA1_CK ÷ 256

Frequency division selector

The calculated interruption 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)

TMA1_CK(kHz)	TMA1_CK1(kHz)	DTMA1[2:0]	DTMA1_CK(kHz)	TA1IF(msec)
1000	3.90625	000b	1.953125	0.512
		100b	0.122070313	8.192
		111b	0.015258789	65.536
14	0.0546875	000b	0.02734375	36.57142857
		100b	0.001708984	585.1428571
		111b	0.000213623	4681.142857

TMA1R: The incremental counter of TMA1 can be read but not written.

The writing action will be regarded as clearing the count value of TMA1R[7:0] and restarting the count.

TMA1C: The comparison point of the TMA1 counter can be read and written.

18. 16-bit TimerB,TMB（16-bit TimerB）

Timer B (hereinafter referred to as TMB) has 2 PWM output, which is PWMA0/1 respectively. Each TMB has 4 types of operation mode. All timers of each mode have special function design to satisfy different application method.

TMB Related Registers:

INTE0	GIE, TMBIE
INTF0	TMBIF
OSCCN1	DTMB[1:0], TMBS
TB1Flag	PWM7A, 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]
TC1CN0	TC1S[1:0]

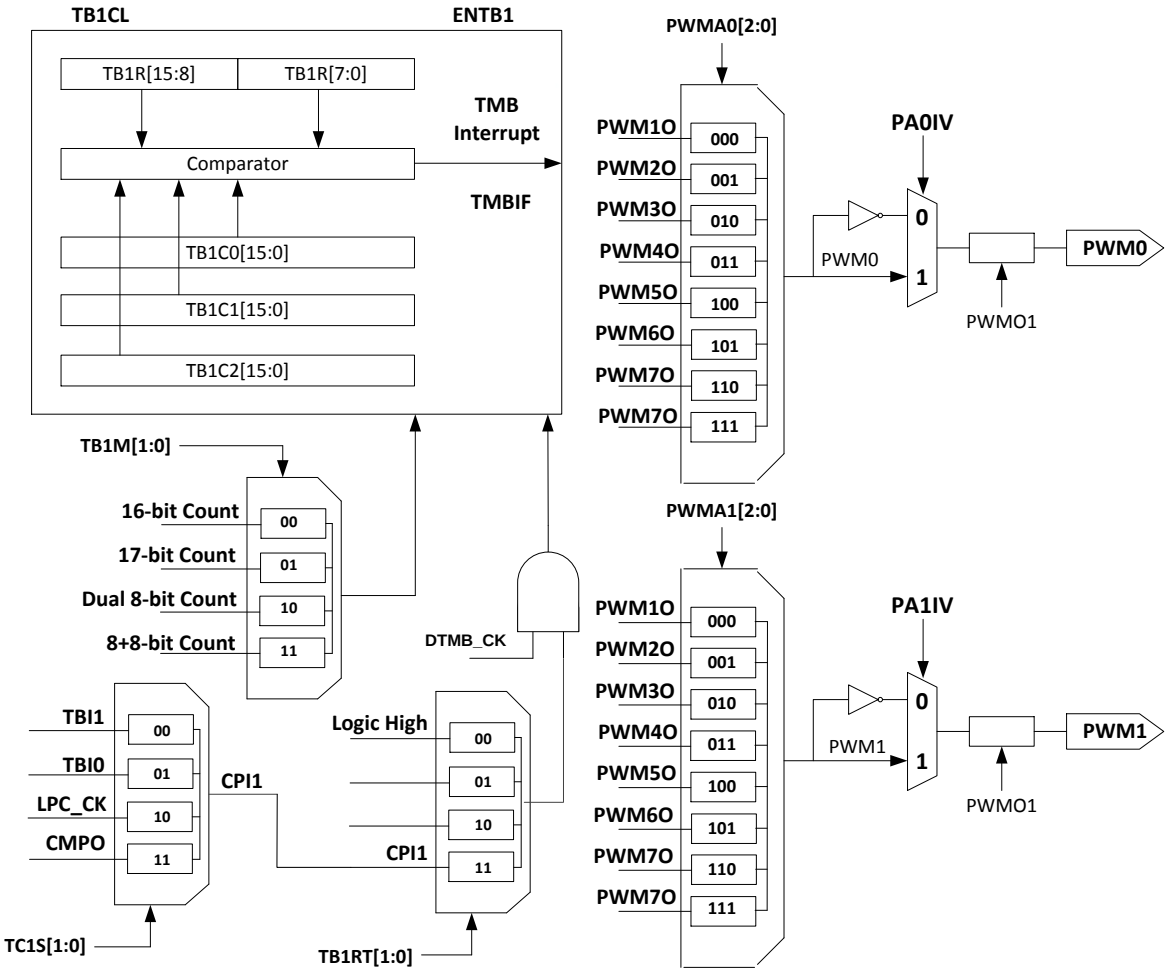


Figure 18-1 TMB Block Diagram

- Timer Registers of TMB are

  Increment / decrement timer TB1R[15:0]

  Overflow event condition controller TB1C0[15:0]

  PWMA condition controller TB1C1[15:0]

  PWMA condition controller TB1C2[15:0]

  Enable controller ENTB1[0]

  Mode controller TB1M[1:0]

  Trogger controller TB1RT[1:0]

  Zeroing controller TB1CL[0]

  PWM0 output waveform selector PWMA0[2:0]

  PWM0 output inverter controller PA0IV[0]

  PWM1 output waveform selector PWMA1[2:0]

  PWM1 output inverter controller PA1IV[0]

  Operation frequency source selector TMBS[0]

  Operation frequency pre-eliminator DTMB[1:0]

- 4 types of counting modes of TMB

  16-bit timer

  17-bit timer

  Dual 8bit timers

  8+8-bit timer

- System power consumption operation of TMB

  Operation Mode

  Idle Mode

  Sleep Mode

- TB1R[15:0] Zeroing and Re-counting Condition

  Read TMB related register, and it will not make TB1R[15:0] zero and re-count.

  Write to TB1R[15:0](read only), TB1C0[15:0]、TB1C1[15:0] and TB1C2 [15:0] and it will not make TB1R[15:0] zero and re-count.

  Write TB1CN0 and TB1CN1 control register will not make TB1R[15:0] zero and re-count.

  When TB1R[15:0] progressive counting is up to more than TB1C0[15:0], it will make TB1R[15:0] zero and re-count.

  After user set TB1CL[0] as and clear TB1R[15:0] timer, TB1CL[0] is set as automatically.

18.1. four types of counting modes of TMB

The four counting methods of TMB can be selected through the counting mode selector TB1M[1:0]. Each counting mode has different overflow and interrupt event modes. This chapter explains the operation methods of the four counting modes.

In addition, different counting modes and PWM condition selector can produce seven different PWM waveforms. They will be described one by one in subsequent chapters.

18.1.1. 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.

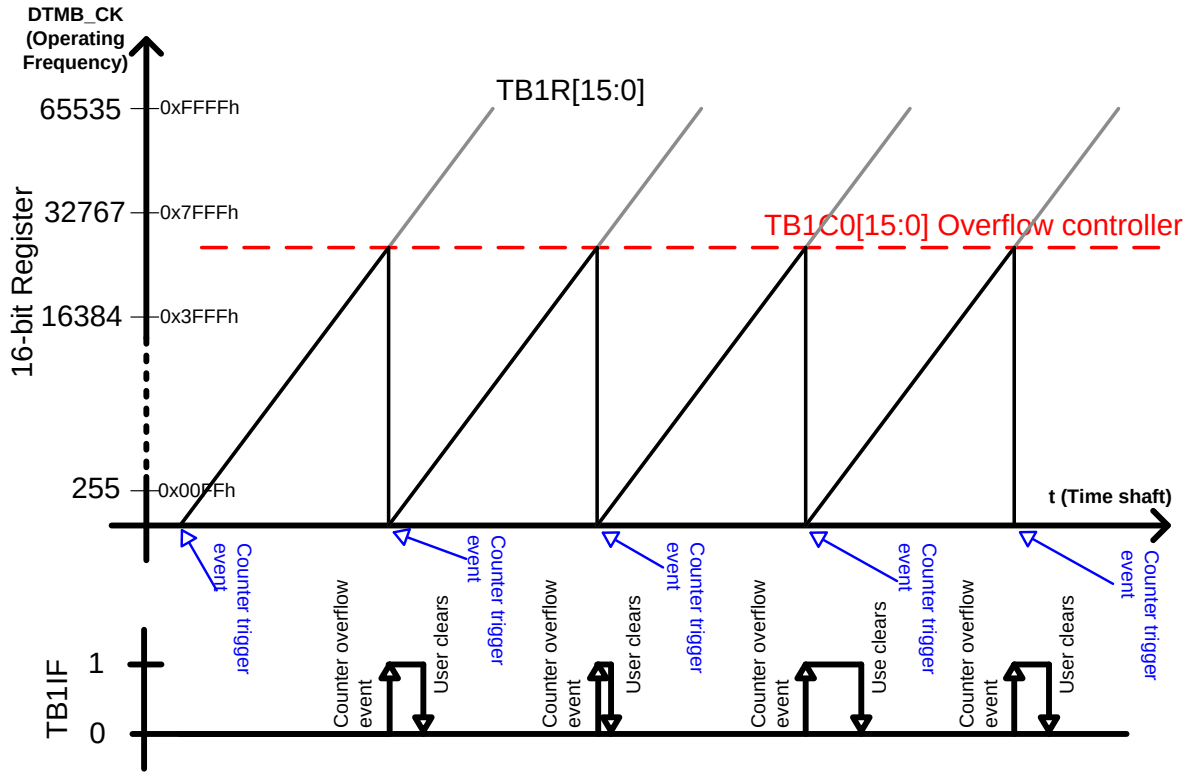


Figure 18-2 16-bit Timer Waveform and Using Schematic Diagram

- Operation Description of 16-bit Counting Mode
  - Initial Configuration
    - Set TMBS[1:0] to select the operating frequency source of TMB, and set DTMB[1:0] to determine the operating frequency of TMB.
    - Set TB1M[1:0] as<00> to plan TMB1 as 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 TB1R[15:0] counting value is equal to TB1C0[15:0], it will produce overflow event and make TB1IF[0] set as <1>, and it is zeroed and re-taken increment counting. At the moment, it will produce interrupt event service when TB1IE[0] is set as <1> .
  - During counting process, the user can set counting zeroing controller TB1CL[0] as to recount, and TB1CL[0] will be set as automatically.
- Set ENTB1[0] as <0> to disable the timer.

18.1.2. 17-bit timer

Set the counting mode selector TB1M[1:0] as <01> to make TMB operate under 17-bit counting mode. Under this mode, it has the following features:

- 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 down-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.

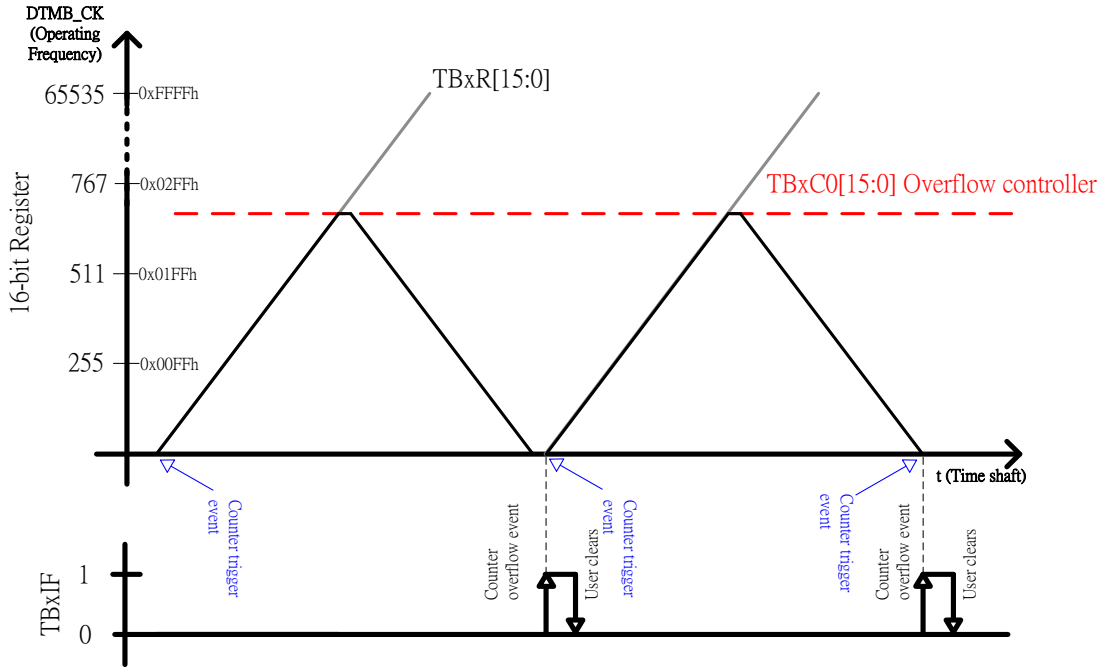


Figure 18-3 17-bit Timer Waveform and Using Schematic Diagram

- Operation description of 17-bit counting mode
  - Initialization
    - Set TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] to determine the operating 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.

18.1.3. Dual 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 dual 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.

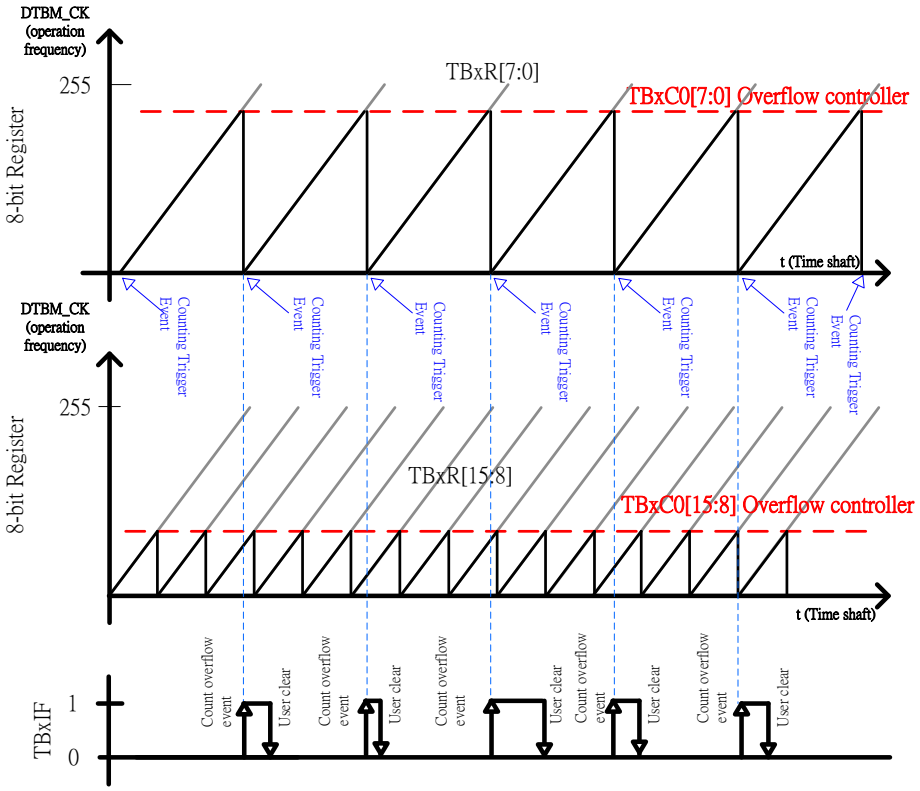


Figure 18-4 Waveform and usage schematic view of dual 8-bit

- Operation description of dual 8-bit counting modes
  - Initialization
    - Set TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] to determine the operating 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.

18.1.4. 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:

- The counting start of 8+8-bit counters TB1R[15:8] and TB1R[7:0] can be triggered by different events set by 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.

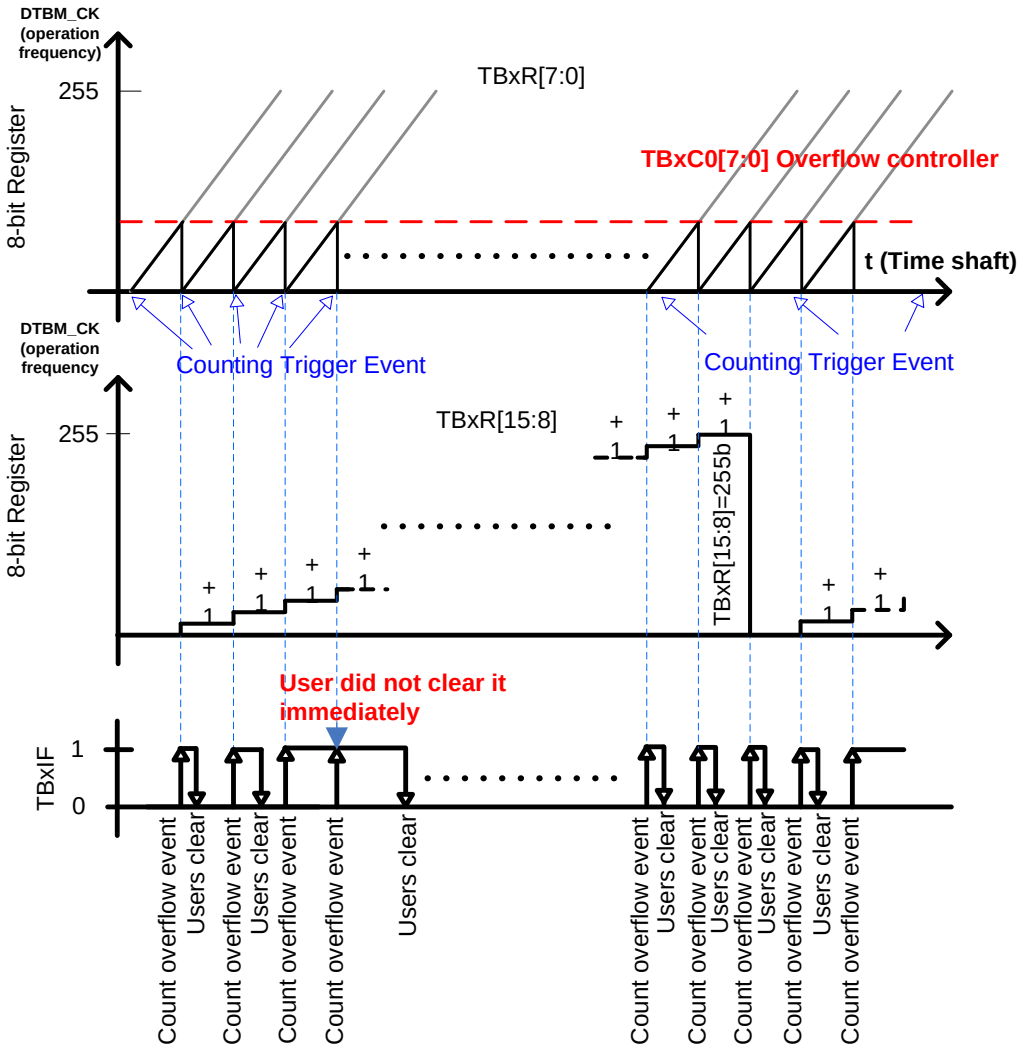


Figure 18-5 Waveform and usage schematic view of two 8+8-bit timer

- Operation description of 8+8-bit timer mode
  - Initialization
    - Set TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] to determine the operating 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.

18.2. 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.
  - Through the PWM output inverters PA0IV[0] and PA1IV[0], you can set whether the actual output waveforms of PWMA0 and PWMA1 are inverted.
  - The outputs of PWMA0 and PWMA1 can be performed from the pins, PT3.4 and PT3.5.
- 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.

18.2.1. PWM1O waveform（16-bit PWM）

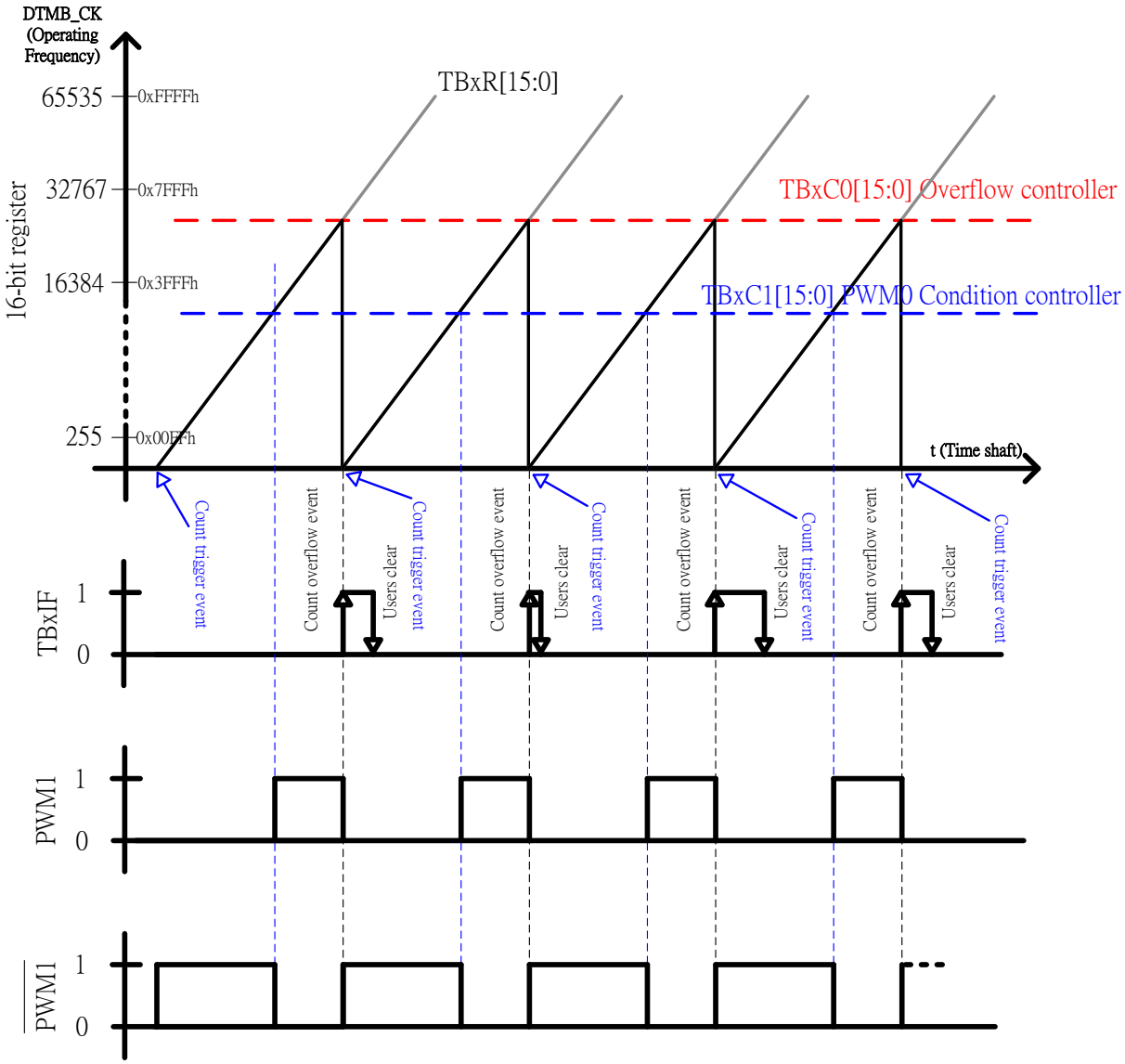


Figure 18-6 Waveform and usage schematic view of PWM1O

- ◆ Operation description of PWM1O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating 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 PWMO1[0] as <1> to enable PWM Mode.
- Set PM11.3[0]/PM11.2[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:

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

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

18.2.2. PWM2O waveform（16-bit PWM）

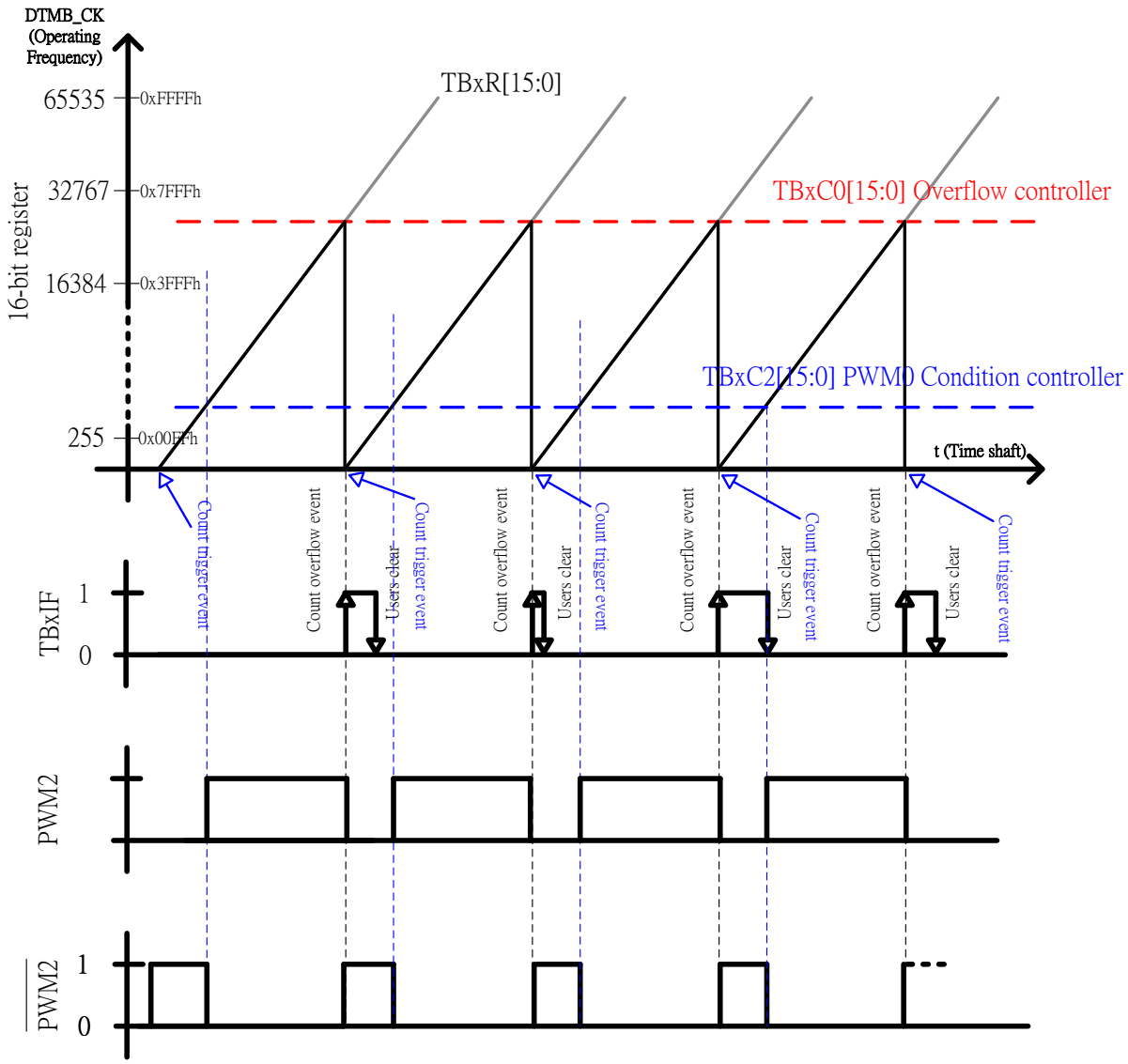


Figure 18-7 Waveform and usage schematic view of PWM2O

- ◆ Operation description of PWM2O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - ● Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating 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 PWMO1[0] as <1> to enable PWM Mode.
- Set PM11.3[0]/PM11.2[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:

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

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

18.2.3. PWM3O waveform（8-bit PWM）

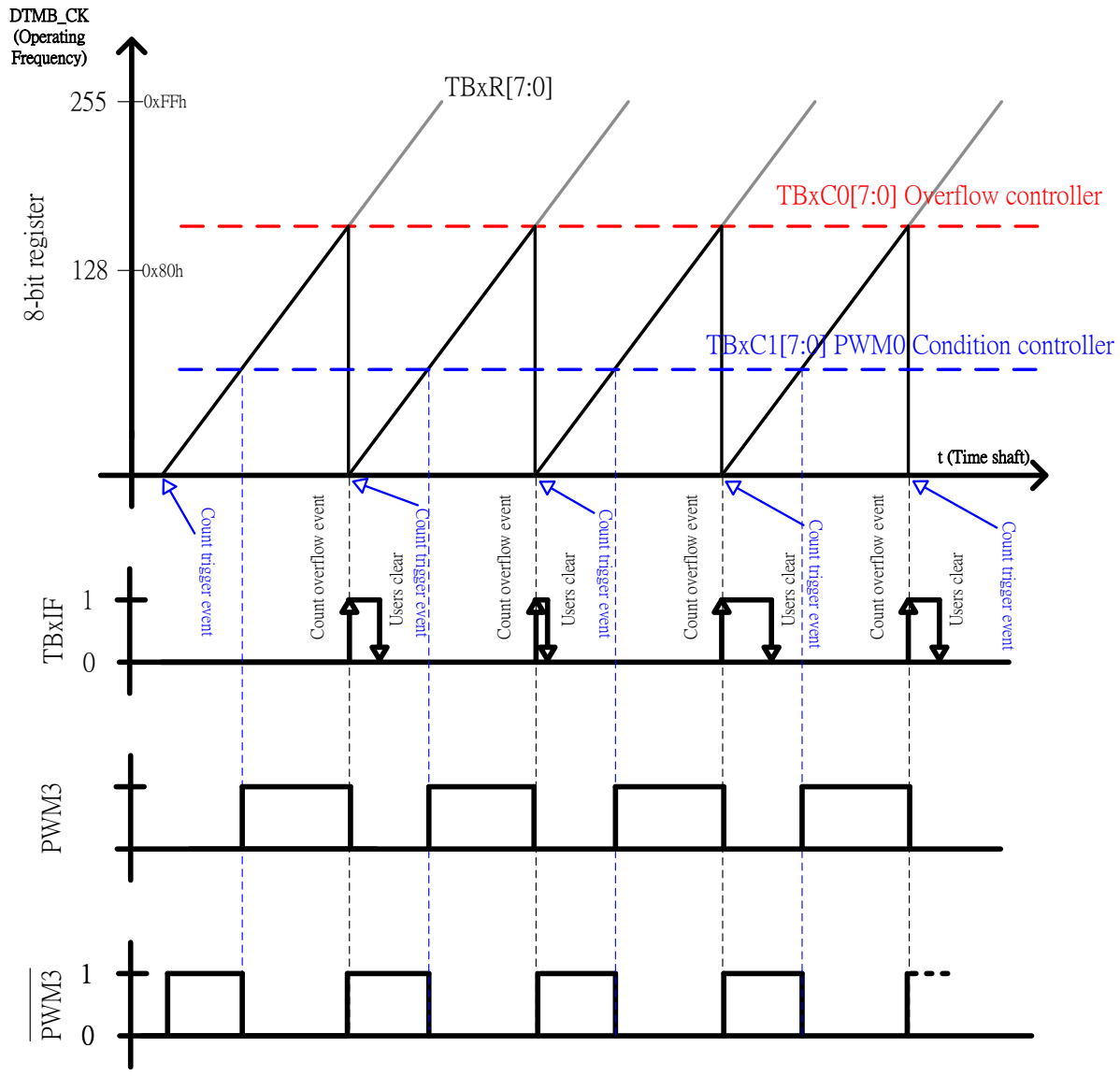


Figure 18-8 Waveform and usage schematic view of PWM3O

- ◆ Operation description of PWM3O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating frequency of TMB.
  - Set TB1M[1:0] as <10> to plan TMB1 as dual 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 PWMO1[0] as <1> to enable PWM Mode.
  - Set PM11.3[0]/PM11.2[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:

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

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

18.2.4. PWM4O waveform（8-bit PWM）

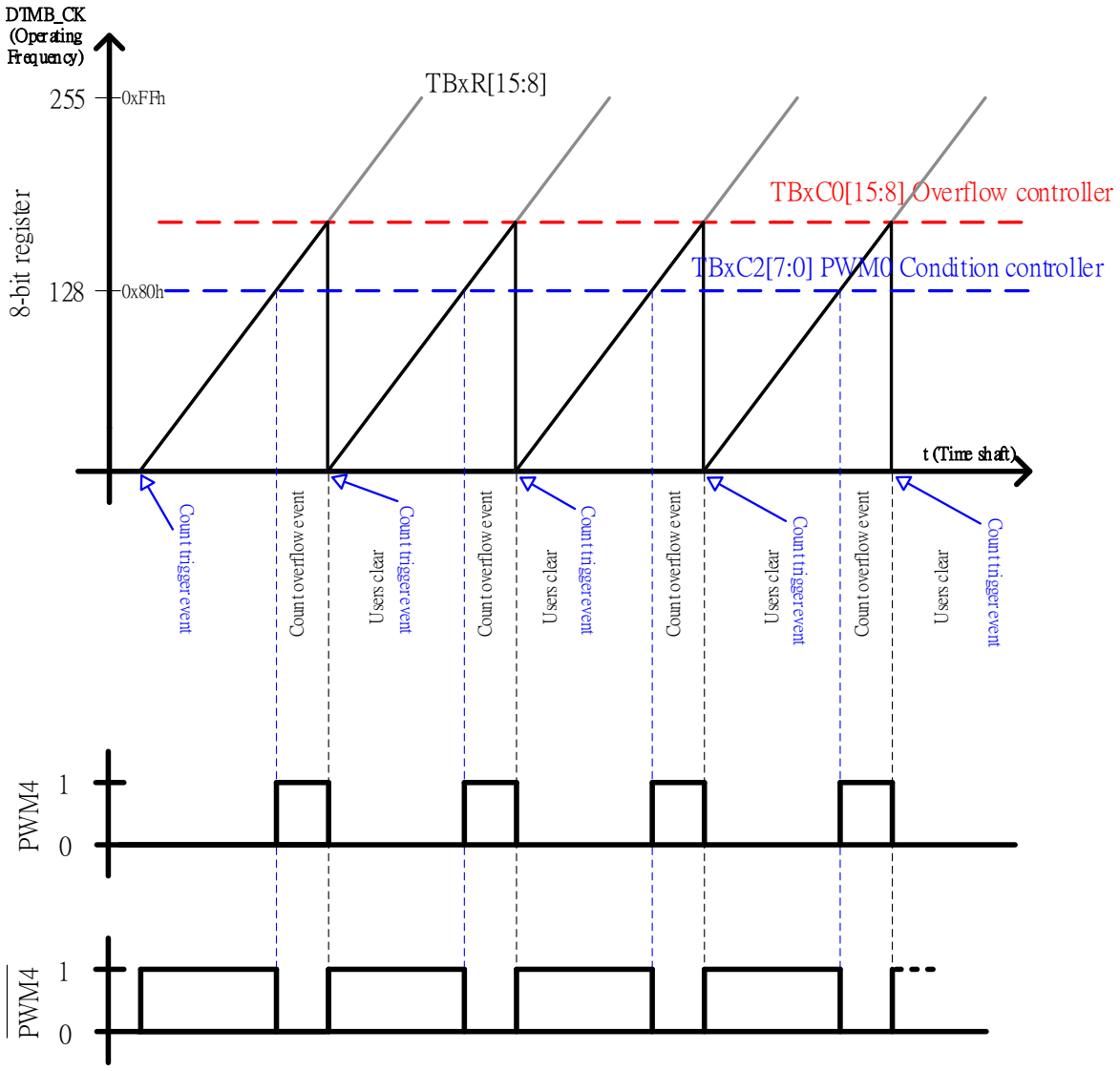


Figure 18-9 Waveform and usage schematic view of PWM4

- ◆ Output operation description of PWM4O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating frequency of TMB.
  - Set TB1M[1:0] as <10> to plan TMB1 as dual 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 PWMO1[0] as <1> to enable PWM Mode.
  - Set PM11.3[0]/PM11.2[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:

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

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

18.2.5. 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:

The description of the duty cycle trimmer TB1C2[7:0], where N is the width of the duty cycle (*PS: N = TB1C0[7:0] – TB1C1[7:0]*).

- Basic type

Fine-tuning of duty cycle of PWM		**Description**
TB1C2[7:0]	Weighted quantity	
80h	1/2	2^1 waveforms are a set of cycles, of which there are (2^1-1) waveforms with a width of N+1
40h	1/4	2^2 waveforms are a set of cycles, of which there are (2^2-1) waveforms with a width of N+1
20h	1/8	2^3 waveforms are a set of cycles, of which there are (2^3-1) waveforms with a width of N+1
10h	1/16	2^4 waveforms are a set of cycles, of which there are (2^4-1) waveforms with a width of N+1
08h	1/32	2^5 waveforms are a set of cycles, of which there are (2^5-1) waveforms with a width of N+1
04h	1/64	2^6 waveforms are a set of cycles, of which there are (2^6-1) waveforms with a width of N+1
02h	1/128	2^7 waveforms are a set of cycles, of which there are (2^7-1) waveforms with a width of N+1
01h	1/256	2^8 waveforms are a set of cycles, of which there are (2^8-1) waveforms with a width of N+1

Table 18-1 Configuration table of duty cycle trimmer

- Logic operation OR superposition type
  - When TB1C2[7:0] is not only set to 1bit, the weighted quantity is added up,which is the total weighted quantity. It is also the number of pulse widths (N,N+1) in a set of cycles.

$$weighted\ quantity = \frac{\alpha}{\beta}$$

α= The number of waveforms with pulse width N in a set of cycles
β= The total number of waveforms in a set of cycles

The following example uses random values to illustrate the law.

- When TB1C2[7:0] is set to C0h (80h+40h), it will cause the PWM duty cycle to produce 3/4 (1/2+ 1/4) weight quantity change. The waveform changes are based on 4 output cycles as a group, in which there will be 3 waveforms with a width of N and 1 (4-3) waveforms with a width of (N+1).
- When TB1C2[7:0] is set to A0h (80h+20h), it will cause the PWM duty cycle to produce 5/8 (1/2+ 1/8) weight quantity change. The waveform changes are based on 8 output cycles as a group, in which there will be 5 waveforms with a width of N and 3(8-5) waveforms with a width of (N+1).
- When TB1C2[7:0] is set to 57h (40h+10h+04h+02h+01h), it will cause the PWM duty cycle to produce 87/256 (1/4+ 1/16+ 1/64+ 1/128+ 1/256) weight quantity change. The waveform changes are based on 256 output cycles as a group, in which there will be 87 waveforms with a width of N and (256-87) waveforms with a width of (N+1).
- When TB1C2[7:0] is set to 86h (80h+04h+02h), it will cause the PWM duty cycle to produce 67/128 (1/2+ 1/64+ 1/128) weight quantity change. The waveform changes are based on 128 output cycles as a group, in which there will be 67 waveforms with a width of N and (128-67) waveforms with a width of (N+1).
- When TB1C2[7:0] is set to FFh (80h+40h+20h+10h+08h+04h+02h+01h), it will cause the PWM duty cycle to produce 255/256 weight quantity change. The waveform changes are based on 256 output cycles as a group, in which there will be 255 waveforms with a width of N and (256-255) waveforms with a width of (N+1).

◆ Table 18-2, Figure 18-10 and Figure 18-10 partly list the 8+8-bit PWM waveform changes of TB1C2[7:0] under different settings for users' reference.

Type	TB1C2 [7:0]	weighted quantity	0	1	2	3	4	5	6	7	8	9	~	127	128	129	~	254	255
Basic waveform	00h	-	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
	80h	1/2	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N
	40h	1/4	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N	N+1
	20h	1/8	N+1	N+1	N+1	N+1	N	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1
	10h	1/16	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
	08h	1/32	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
	04h	1/64	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
	02h	1/128	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	
	01h	1/256	N+1	N+1	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
Logic operation superimposed type	C0h	3/4	N+1	N	N	N	N+1	N	N	N	N+1	N	~	N	N+1	N	~	N	N
	A0h	5/8	N+1	N	N+1	N	N	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N
	E0h	7/8	N+1	N	N	N	N	N	N	N	N+1	N	~	N	N+1	N	~	N	N
	F0h	15/16	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N
	F8h	31/32	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N
	FCh	63/64	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N
	FEh	127/128	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N
	FFh	255/256	N+1	N	N	N	N	N	N	N	N	N	~	N	N	N	~	N	N
	57h	87/256	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	~	N+1	N	N+1	~	N	N+1
	86h	67/128	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N
	32h	25/128	N+1	N+1	N+1	N+1	N	N+1	N+1	N+1	N	N+1	~	N+1	N+1	N+1	~	N+1	N+1

Table 18 -2 PWM5O output waveform table

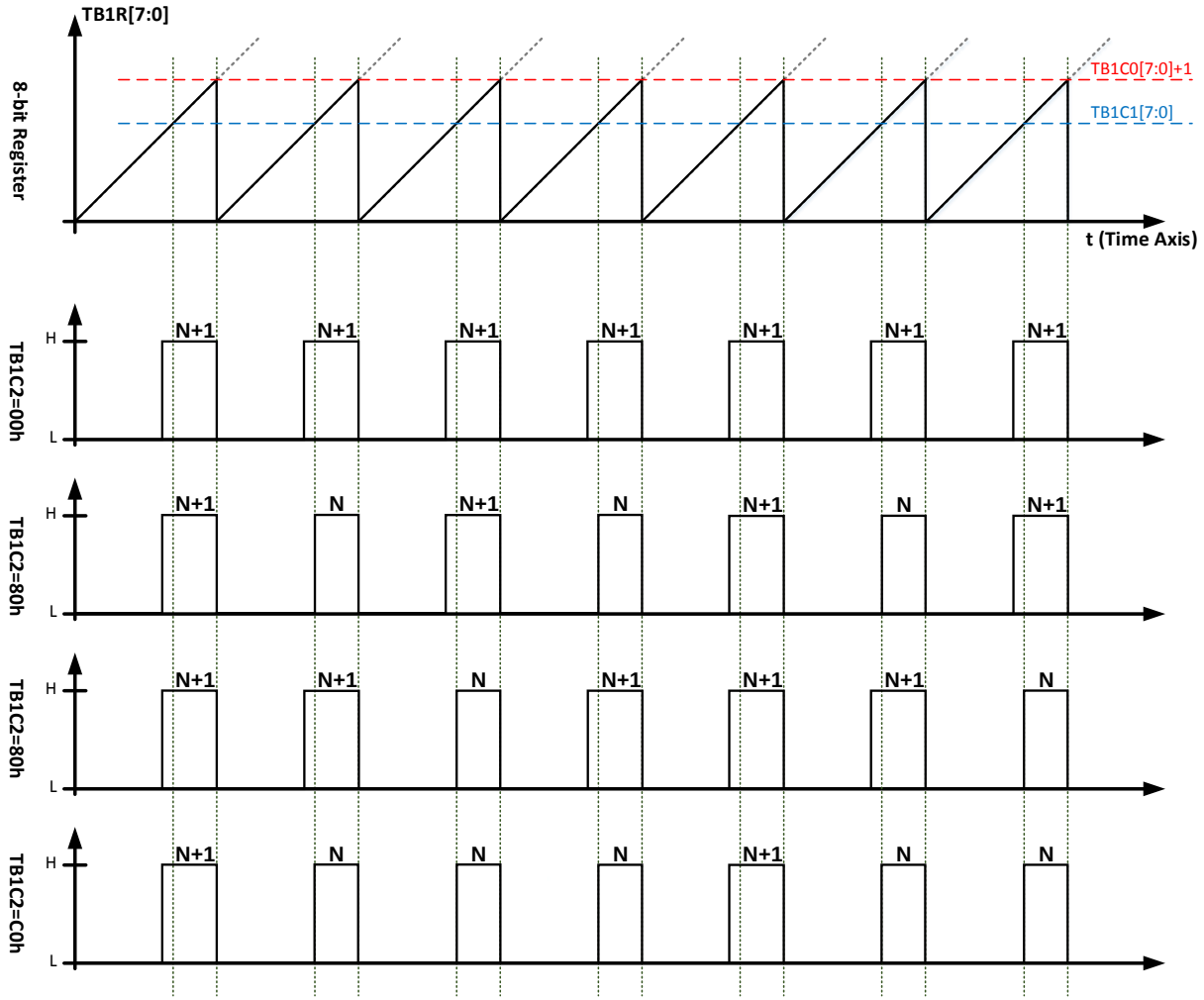


Figure 18-10 PWM5O output waveform table

- ◆ Operation description of PWM5O output
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating 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 19-1, where N = TB1C1[7:0] - TB1C0[7:0].

■ Output control of PWM
- Set PWMO1 as <1> to enable PWM Mode.
- Set PM11.3[0]/PM11.2[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:

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

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

18.2.6. 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.

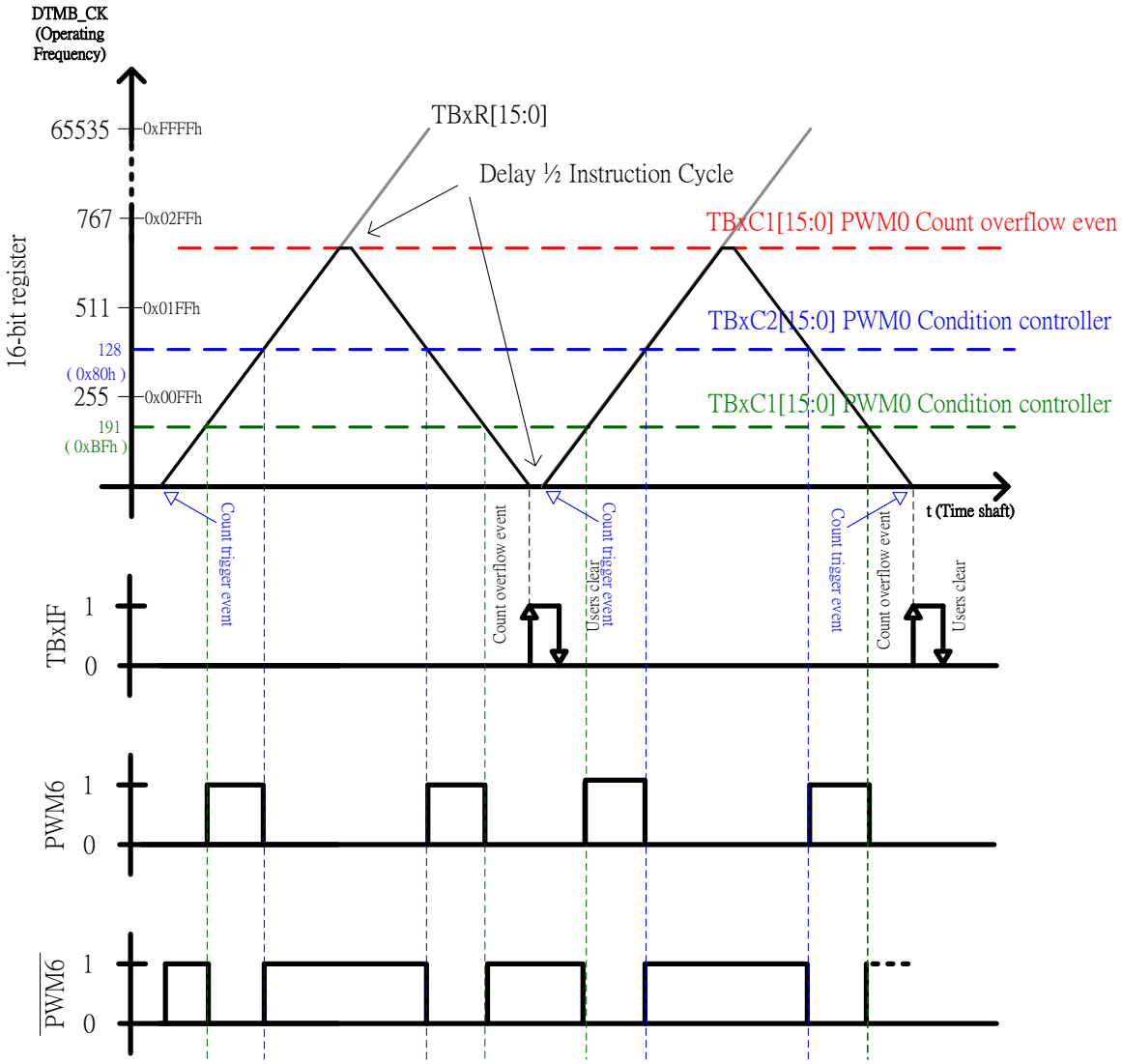


Figure 18-11 Waveform and usage schematic view of PWM6O

- ◆ Operation description of 17-bit PWM output
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating 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 TB1C2[15:0], the status of PWM6O is changed from 0➔1.
  - ✓ 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 PWMO1[0] as <1> to enable PWM Mode.
  - Set PM11.3[0]/PM11.2[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 PWM6O duty cycle calculation is due to the special waveform generated. It is not described here.

18.2.7. 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.

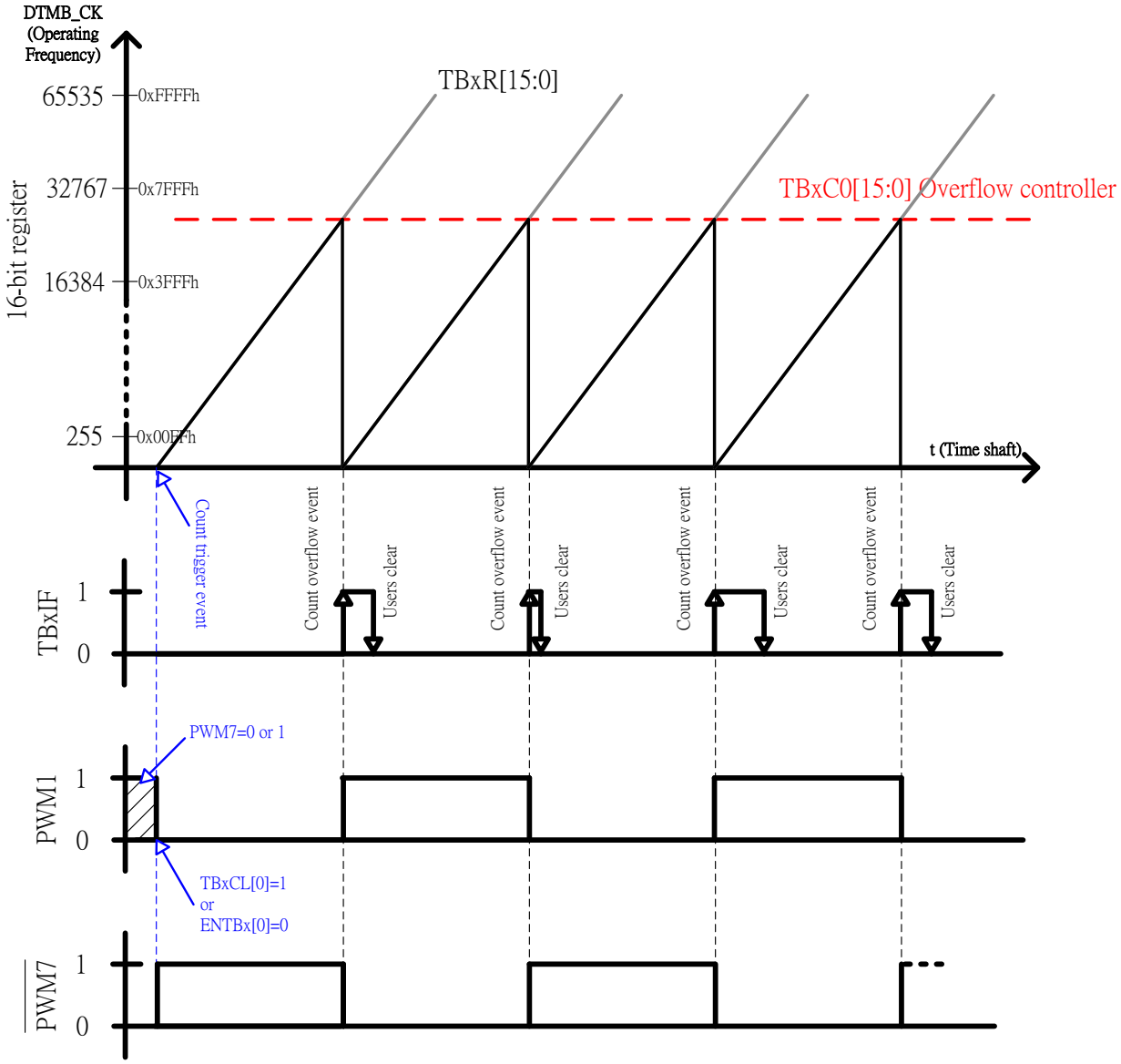


Figure 18-12 Waveform and usage schematic view of PWM7O

- ◆ Operation description of PWM7O
- ■ Initialization (the frequency and duty cycle configuration of PWM)
  - Setting TMBS[1:0] can select the operating frequency source of TMB; setting DTMB[1:0] can determine the operating frequency of TMB.
  - Set TB1M[1:0] as <00> to plan TMB1 as a 16-bit timer.
  - Set PWMA0/1[2:0] as <111> 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 PWMO1[0] as <1> to enable PWM Mode.
  - Set PM11.3[0]/PM11.2[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:

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

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

18.3. 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**
1D2H	TB1Flag	-	PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu	-,-,r,r r,r,r,r
1D3H	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu	*,*,*,* *,rw1,*,*
1D4H	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu	*,*,*,* *,*,*,*
1D5H	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
1D6H	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
1D7H	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
1D8H	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D9H	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
1DAH	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
1DBH	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
1DCH	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*,*,*,* *,*,*,*
1DDH	TC1CN0	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	uuuu uuuu

Table 18-3 Registers related to TMB1

BSRCN: Please refer to Memory chapter

INTE0/INTF0: Please refer toInterrupt chapter

OSCCN0/OSCCN1/OSCCN2: Please refer to "Oscillator, clock source and power consumption management" chapter

TB1Flag: PWM waveform state flag generated by TMB1 timer

Bit	Name	Description
Bit6~0	PWMxA	PWMx waveform state · 1≦x≦7 <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> Dual 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.)
Bit1	PWMO1	PWM1 pin output controller <0> Disable <1> Output
Bit0	PWMO0	PWM0 pin output controller <0> Disable <1> Output

TB1CN1: TMB1 Timer control register 1

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>PA1IV</td><td rowspan="2">Pin PWMAx waveform output phase（0≦x≦1）
<0> Reversed phase.
<1> Co-phase.</td></tr>
<tr><td>Bit3</td><td>PA0IV</td></tr>
<tr><td>Bit6~4</td><td>PWMA1[2:0]</td><td rowspan="2">Pin PWMAx waveform output selector（0≦x≦1）
<table>
<tr><th>PWMAx[2:0]</th><th>Output selector</th><th>PWMAx[2:0]</th><th>Output selector</th></tr>
<tr><td>000</td><td>PWM1O</td><td>100</td><td>PWM5O</td></tr>
<tr><td>001</td><td>PWM2O</td><td>101</td><td>PWM6O</td></tr>
<tr><td>010</td><td>PWM3O</td><td>110</td><td>PWM7O</td></tr>
<tr><td>011</td><td>PWM4O</td><td>111</td><td>PWM7O</td></tr>
</table></td></tr>
<tr><td>Bit2~0</td><td>PWMA0[2:0]</td></tr>
</table>

TB1CN2: TMB1 Timer control register 2

Bit	Name	Description
Bit6~5	TC1S[1:0]	Counter TC1 event input selector <00> TBI1 is input from GPIO port. (Default) <01> TBI0 is input from GPIO port. <10> Low frequency clock source LPC_CK <11> CMPO window comparator output.

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]	TB1C0RL[7:0] must be written last to be effective.

TB1C1: PWMA condition control 1

Bit	Name	Description
Bit15~8	TB1C1RH[7:0]	PWMA condition control 1
Bit7~0	TB1C1RL[7:0]	TB1C1RL[7:0] must be written last to be effective.

TB1C2: PWMA condition control 2

Bit	Name	Description
Bit15~8	TB1C2RH[7:0]	PWMA condition control 2
Bit7~0	TB1C2RL[7:0]	TB1C2RL[7:0] must be written last to be effective.

19. Serial Peripheral Interface, SPI(only for HY17P68)

Serial Peripheral Interface(Serial Peripheral Interface, hereafter referred to as SPI) has the following functions:

- SPI module allows synchronous sending and receiving of 8-bit data
- Serial interface that can be used for communication with other devices, most of which are EEPROM, shift register, etc.
- For master mode and slave mode
- The pin configuration of Master mode is as follows, it must be set to the corresponding pin input and output when using

  Serial data output SDO (PT1.5)

  Serial data input SDI(PT1.0)

  Serial clock source SCK (PT1.6)
- Slave mode synchronize selection pin SCE(PT1.1)

SPI Related Registers：

PT1M2	PM1.6, PM1.5
SSPCON1	SSPEN[0],CKP[0],CKE[0],SMP[0],SSPM[1:0]
SSPSTA	SSPBUY[0],SSPOV[0],BF[0]
SSPBUF	SSPBUF[7:0]

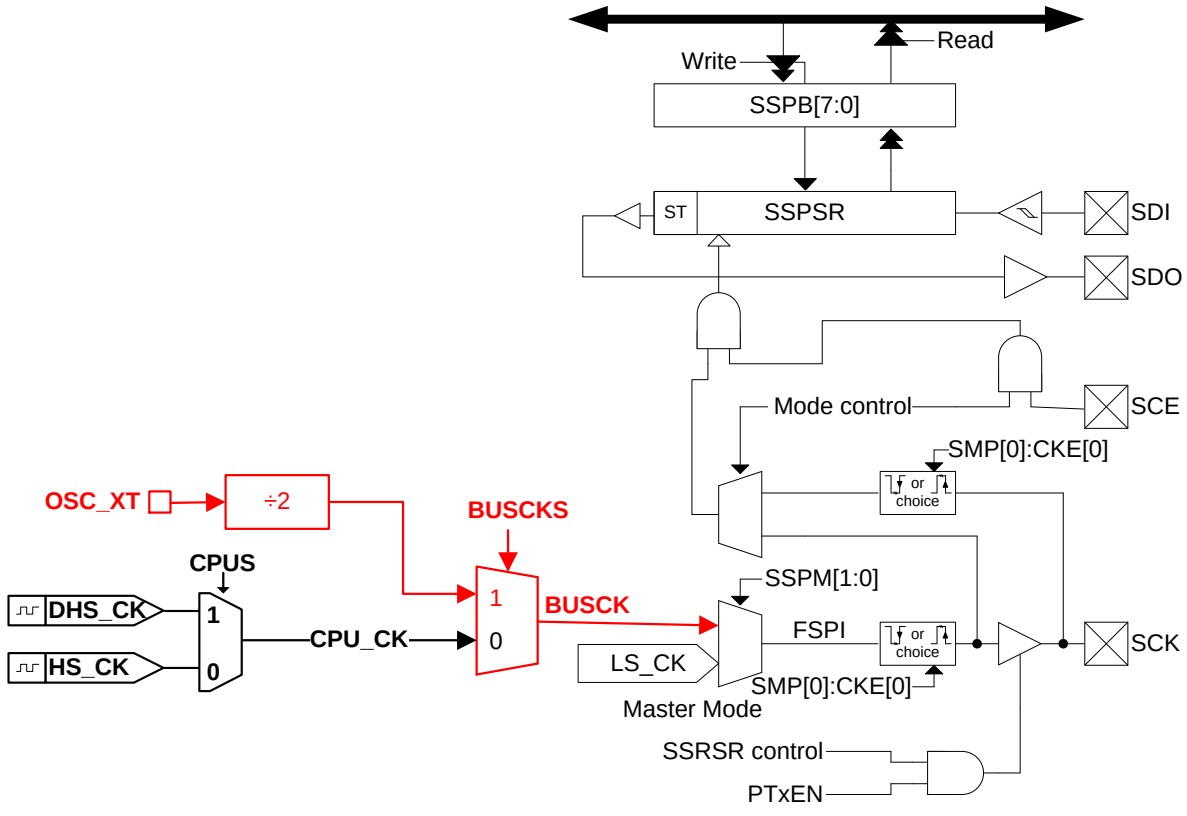


Figure 19-1 SPI Block Diagram

19.1. SPI using instructions

When using it, in addition to setting the corresponding pins to be input-output pin-out, the pin-out is different as the master and slave selection.

- Master mode: the used pins include SCK(clock output), SDI(data input), SDO(data output)
- Slave mode: The used pins include SCK(clock input), SDI(data input), SDO(data output) and SCE (Synchronous data receiving enable)

The control bit, such as SPI, can be set through being configured the register of SSPCON1:

- Set the bit of SSPEN, which can be used to enable communicate model of SPI
- Set the bit of CKP, so as to determine that the SCK is in level state after transmission
- Set the bit of CKE, so as to determine that the transmitted data is in SCK rising edge or failing edge
- Set the bit of SMP, so as to determine the sampling time of input data.(sampling in the middle or end of clock)
- Set the byte of SSPM[1:0], so as to determine that whether the master mode SCK frequency source or slave mode SCE pin is enabled.
- Through configuring the register of SSPBUF to determine to transmit or receive data, and the register of SSPSTA will reflect the Transmit and receive status.
- The register of SPI Transmit and receive is consist of SSPBUF and SSPSR.
- SSPBUF will keep the last write-in SSPSR data, till next received data are prepared. When the 8-bit data are received, the data will be transmitted to register SSPBUF, and the BF bit in register SSPSTA and the SSPIF bit in register INTF2 will be set to 1.
- The double register can allow to receive next data in register SSPSR at the same time of reading this data (Read SSPBUF). When receiving the data, the first is to judge that the BF bit in register SSPSTA is set to 1, if it is set to 1, that means there is the received data, which is not read by users, thus the users should read SSPBUF firstly, then the BF bit will be automatically cleated to be 0 by hardware. If the users did not read SSPBUF when BF is set to be 1, and new data is received, the SSPOV bit of register SSPSTA will be set to be 1, and this data will be lost and not be written in the register SSPBUF.
- When the data is transmitted, the SSPBUY bit of register SSPSTA is set to be 1, any action of writing in register SSPBUF will be ignored. And after transmitting the data, SSPBY bit will be automatically cleared to be o.
- When SPI is master mode, it cannot be pay attention to the value received by register SSPBUF, it only needs to write-in the data to be transmitted.

19.2. SPI master mode

When SPI is set to be master mode, the action of data transmission can be enabled anytime. The data will be output through pin SDO matches clock source SCK only when the data is written in register SSPBUF.

Meanwhile, if the receive data model is SPI model, the SDO of SPI slave mode can be configured to be input pin-out, there will no invalid data transmitted to master mode; if the receive model must transmit the data to master mode, the SOD of slave mode can be configured to be output pin-out, thus the SDI of master mode can receive the transferred data continuously, after receiving, the data will be written in the register SSPBUF, and the BF bit of corresponding register SSPSTA and the SPPIF bit in register INTF2 will be set to be 1.

Meanwhile, through the byte SSPM[1:0] of register SSPCON1 to set the frequency source of master mode, and through the set value of bit of CKP and CKE to determine the polarity if clock source.

Master mode configuration instructions：

Pin setting: The pins used are SCK (clock output), SDI (data input), SDO (data output)

First, correctly set the input and output functions of the I/O pins, and the SCK and SDO output module functions.

- Configure SSPCON1 register to reach SPI function control bit
- Set the bit of CKP, so as to determine that the SCK is in level state after transmission
- Set the bit of CKE, so as to determine that the transmitted data is in SCK rising edge or failing edge
- Set the bit of SMP, so as to determine the sampling time of input data.(sampling in the middle or end of clock)
- Set the SSPM[1:0] byte to determine the SCK frequency source in the master mode.
- Setting the SSPEN bit can be used to start the SPI communication module.
- Through configuring the register of SSPBUF to determine to transmit data, and the register of SSPSTA will reflect the Transmit status.
- The register of SPI Transmit and receive is consist of SSPBUF and SSPSR.

When the data is written into SSPBUF, the hardware action will move the SSPBUF data to the SSPSR register and send the SSPSR data along with the SCK clock source.

When the data is sent out, the SSPSR will receive the data input on the SDI pin synchronously, and after the reception is completed, the hardware action will move the data in the SSPSR to the SSPBUF. The BF flag can be used to determine whether the reception is complete.

SSPBUF will keep the data received last time from SSPSR until the next data received is ready.

When the 8-bit data is received, the data will be moved into the SSPBUF register, and the SSPIF bit in the INTF2 register will be set to 1 after the 8 bits of data have been sent by SCK. The BF bit in the SSPSTA register will be set to 1 after the data is received and moved to SSPBUF by the hardware.

When data is being transferred, the SSPBUY bit of the SSPSTA register will be set to 1, and any writing to the SSPBUF register will be ignored. After the data is sent, the SSPBUY bit will be automatically cleared to 0.

When the SPI is in master mode, you can ignore the value received by the SSPBUF register and just write the data to be sent. If the value received by SSPBUF is available, the user should move the data by himself before writing the data to be sent.

The relevant configuration can be seen in the following figure:

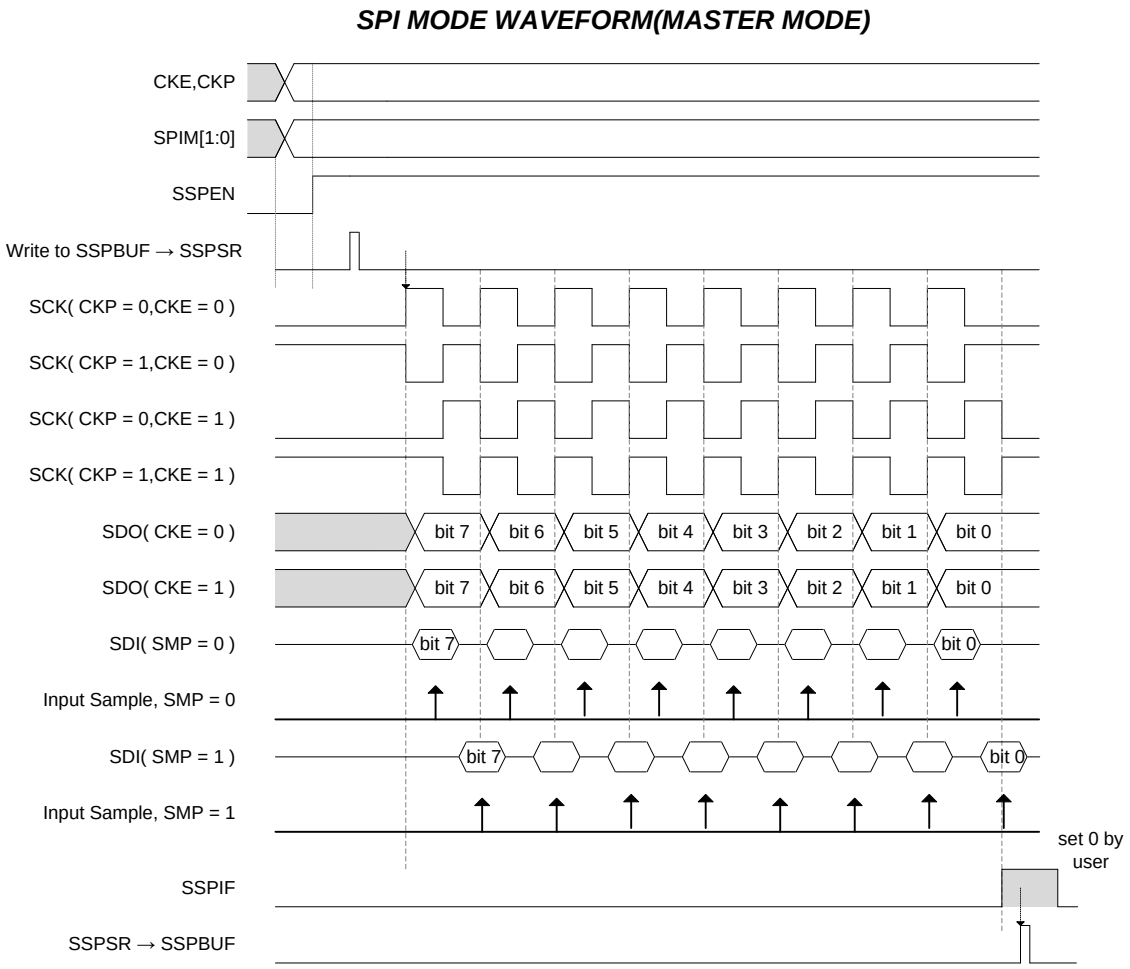


Figure 19-2 SPI master module timing waveform

19.3. SPI slave mode

For the slave mode, the clock source SCK is provided by external, and the pin of SCK muse be the idle status, the pin of SCK must be configured to be input pin-out. When matching the setting of clock source polarity of master mode, it can be determine the polarity if master mode clock source through the set value of CKE and CKP bit.

Slave mode configuration instructions：

- Pin setting：The pins used are SCK (clock output), SDI (data input), SDO (data output), SCE (synchronous data receive enable)
- First, correctly set the input and output functions of the I/O pins, and SDO output module functions.
- Configure SSPCON1 register to reach SPI function control bit
- Set the bit of CKP, so as to determine that the SCK is in level state after transmission
- Set the bit of CKE, so as to determine that the transmitted data is in SCK rising edge or failing edge
- Set the bit of SMP, so as to determine the sampling time of input data.(sampling in the middle or end of clock)
- Set SSPM[1:0] bytes to determine whether the slave mode SCE pin is enabled.
- Setting the SSPEN bit can be used to start the SPI communication module.
- By configuring the SSPBUF register to decide to receive and send data synchronously, the SSPSTA register reflects the receiving status
- The register of SPI Transmit and receive is consist of SSPBUF and SSPSR.
- Before receiving data, write the data to be sent synchronously into SSPBUF, and wait for the master clock source to arrive.
- Even if you don't want to send data synchronously, you still need to complete the action of writing SSPBUF, and it is recommended to write 0FFh data.
- The time between writing the synchronous sending data and before the clock source of the master control terminal arrives requires a delay of 5 instruction cycles to facilitate the hardware to correctly move the SSPBUF data to the SSPSR register.
- When the master SCK clock source is input, the slave module will not only capture the SDI input pin data, but also synchronously output the SSPSR data from the SDO pin to the master terminal.
- SSPBUF will keep the data received last time from SSPSR until the next data received is ready.
- When the 8-bit data is received, the data will be moved into the SSPBUF register, and the SSPIF bit in the INTF2 register will be set to 1 after the 8 bits of data have

been sent by SCK. The BF bit in the SSPSTA register will be set to 1 after the data is received and moved to SSPBUF by the hardware.

- The double register can allow to receive next data in register SSPSR at the same time of reading this data (Read SSPBUF). When receiving the data, the first is to judge that the BF bit in register SSPSTA is set to 1, if it is set to 1, that means there is the received data, which is not read by users, thus the users should read SSPBUF firstly, then the BF bit will be automatically cleated to be 0 by hardware. If the users did not read SSPBUF when BF is set to be 1, and new data is received, the SSPOV bit of register SSPSTA will be set to be 1, and this data will be lost and not be written in the register SSPBUF.

The relevant configuration can be seen in the following figure:

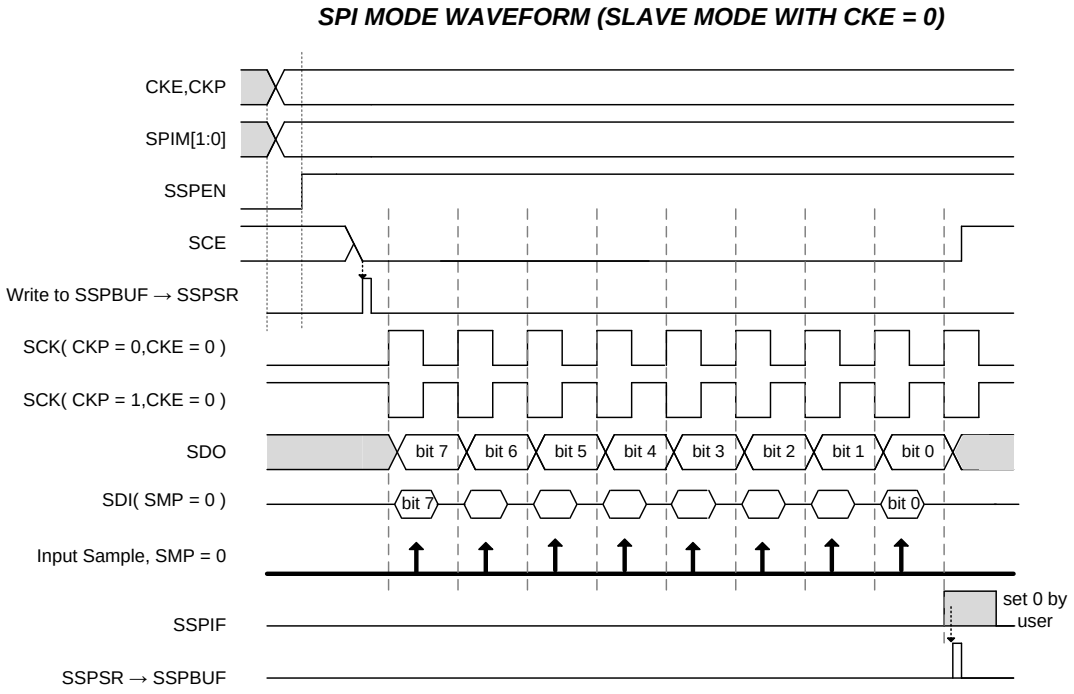


Figure 19-3 SPI slave mode waveform (CKE=0)

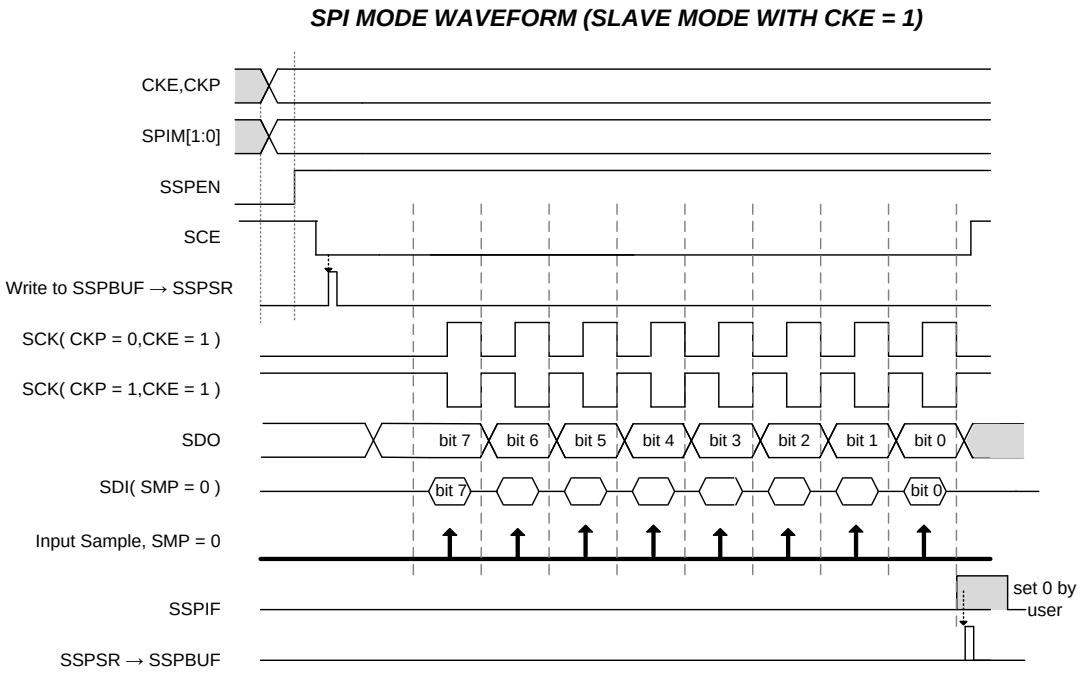


Figure 19-4 SPI slave mode waveform (CKE=1)

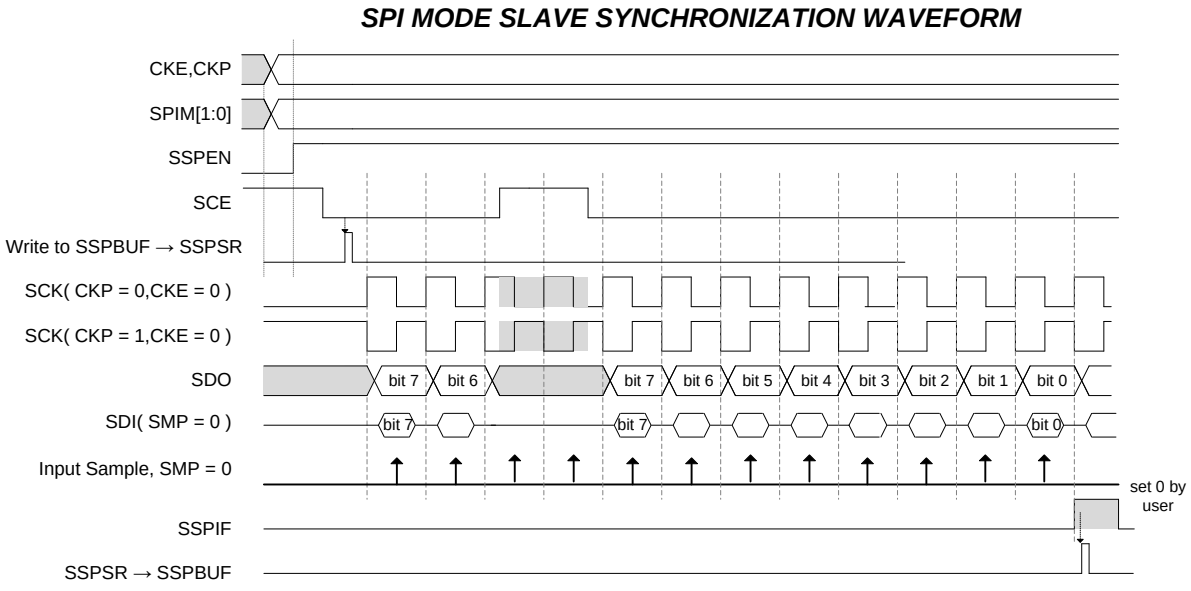


Figure 19-5 SPI mode slave synchronization waveform

Under the SLP mode, if the SSPIE bit interrupt enable of register INTE2 has been enabled, a completed 8-bit data is received, the chip will be waken up.

The slave mode can control another pin SCE, and the configuration of pin SCE is allowed to reach the slave synchronous mode, which can reach this pin setting through the byte SSPM[1:0] of register SSPCON1.

When the pin SCE is in low level, the data can conduct normal transmission and receiving action, meanwhile, the pin SDO can be normally driven. And is SCE is in high level, the output on SDO will be hanged and not be driven.

19.4. SPI master and slave module transmission mode

The following figure shows the master and slave connection of SPI modules in the two Hycon processors:

- ◆ Master mode will transmit the data of register SSPBUF through the shift register SSPSR, and output through the data transmit pin SDO. When transmitting the data, it also can receive the data transmitted by slave mode in the shift register SSPSR, after receiving the data, which will be written in the register of SSPBUF.
- ◆ The slave mode will save the received data in the shift register SSPSR temporarily, after receiving, the data will be written in register SSPBUF.

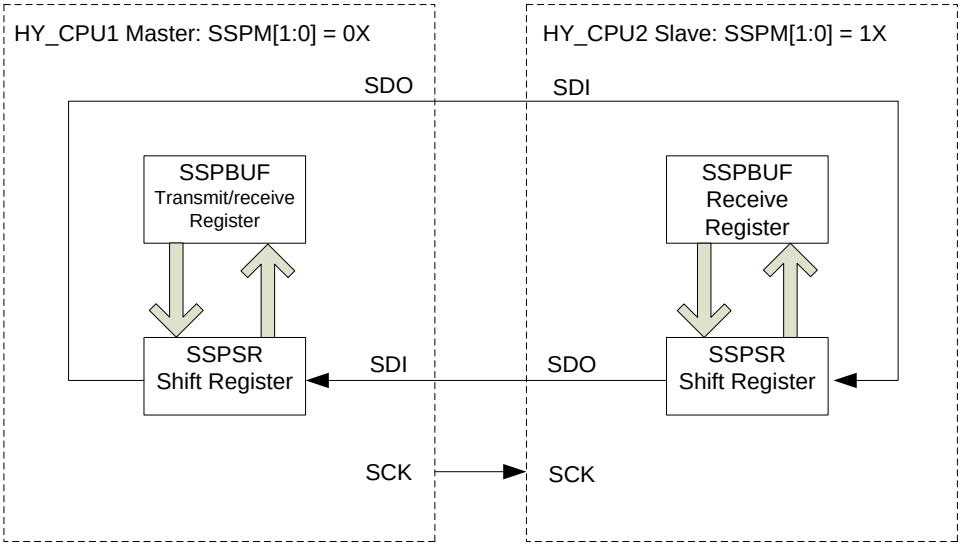


Figure 19-6 SPI master and slave ling way of two processors

19.5. Register Description-SPI

"-"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	CTIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024H	INTE1		SPIIE							0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
027H	INTF1		SPIIF							0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
1C0H	SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	*,*,*,* -,-,*,*
1C1H	SSPSTA	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ...0	uu.. ...u	*,*,-,- -,-,-,*
1C2H	SSPBUF	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 19-4 SPI Register

INTE0/INTE1/INTF1：Please refer to the Interrupt chapter

OSCCN0/OSCCN1/OSCCN2: Please refer to "Oscillator, clock source and power consumption management" chapter

SSPCN0: SPI control register 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENSSP</td><td>SPI Enable and disable controller
<0> Disable
<1> Enable</td></tr>
<tr><td>Bit6</td><td>CKP</td><td>Operating frequency polarity controller
<0> Low potential is IDLE
<1> High potential is IDLE</td></tr>
<tr><td>Bit5</td><td>CKE</td><td>Data send controller
<0> send when the Operating frequency is changed from idle to effective
<1> send when the Operating frequency is changed from effective to idle</td></tr>
<tr><td>Bit4</td><td>SMP</td><td>Data send controller
<table>
<tr><th>SET</th><th>SPI master mode</th><th>SPI slave mode</th></tr>
<tr><td>0</td><td>sampling the input data at the middle of time of data output</td><td>slave mode, uses must set SMP Bit to <0></td></tr>
<tr><td>1</td><td>sampling the input data at the end of time of data output</td><td>Can not be used</td></tr>
</table></td></tr>
<tr><td>Bit1~0</td><td>SSMP[1:0]</td><td>Mode selection byte
<table>
<tr><th>SET</th><th>Operation mode</th></tr>
<tr><td>00</td><td>SPI master mode, clock source =LS_CK</td></tr>
<tr><td>01</td><td>SPI master mode, clock source =CPU_CK</td></tr>
<tr><td>10</td><td>SPI slave mode, when the clock source =SCK pin, the pin control function of SCE is disables, the status of SCE pins are I/O usage.</td></tr>
<tr><td>11</td><td>SPI slave mode, when the clock source =SCK pin, the pin control function of SCE enables</td></tr>
</table></td></tr>
</table>

SSPSTA: SPI control register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>SSPBY</td><td>Write-in conflict bit detection(only for the usage of delivering data)
<0> no conflict
<1> When the data are in the deliver status</td></tr>
<tr><td>Bit6</td><td>SSPOV</td><td>Receive overflow flag bit
<table>
<tr><th>SET</th><th>SPI slave mode</th><th>SPI master mode</th></tr>
<tr><td>0</td><td>No overflow status</td><td>the bit of SSPOV will not be set to 1, as it needs to write in registers of SSPB for every send (receive) data</td></tr>
<tr><td>1</td><td>SSPB holds the last data, and receive new data, once the SSPSR occurs overflow, the data is the SSPSR register will be lost. SSPOV only will occurs in slave mode, and just be used to send data. Users must read the SSPB, in order to avoid that SSPOV is set to 1.(instruction clear can be used)</td><td>Not occur</td></tr>
</table></td></tr>
<tr><td>Bit0</td><td>BF</td><td>Full status bit of buffer(only for the usage of receiving data)
<0> Receive is not finished, SSPBUF is empty
<1> Receive completed, SSPBUF is full</td></tr>
</table>

SSPBUF[7:0] : Receive buffer register or Transmit Buffer register

20. Inter-Integrated Circuit Serial Interface,I²C

The I²C 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 I²C package format to the I²C 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 I²C 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.

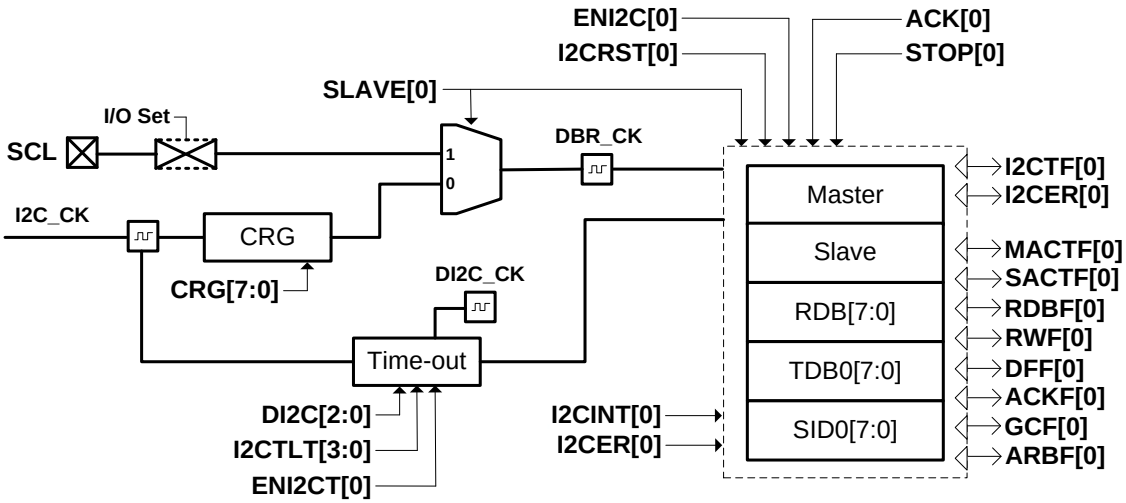


Figure 20-1 I²C system structure diagram

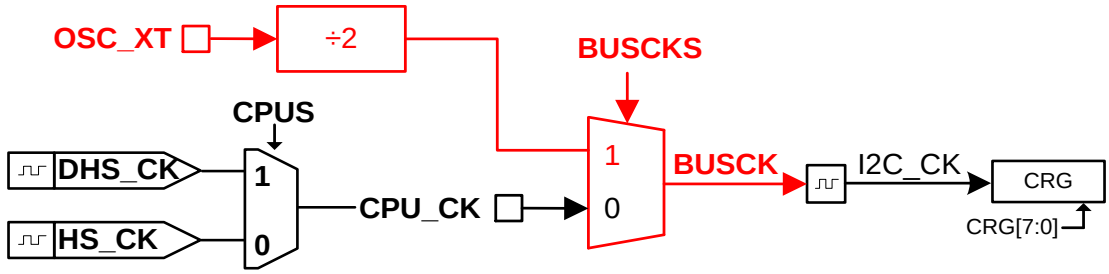


Figure 20-2 I²C operating clock configuration diagram

- Function characteristics of I²C serial interface:
  - The standard I²C 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 I²C 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 I²C.
  - The data transmission between the master and the slave is dual-direction.
  - I²C allows quite high working voltage range.
  - The reference design of I²C uses a 7-bit address space, but keeps 16 addresses, so a set of buses can have at most 112 nodes for communication.

    The reserved addresses are shown in the table below, refer to I²C-bus specification and user manual。

Slave address	R/W bit	Description
0000 000	0	general call address[1]
	1	START byte[2]
0000 001	X	CBUS address[3]
0000 010	X	reserved for different bus format[4]
0000 011	X	reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	1	device ID
1111 0XX	X	10-bit slave addressing

X = don't care; 1 = HIGH; 0 = LOW.

[1] The general call address is used for several functions including software reset.

[2] No device is allowed to acknowledge at the reception of the START byte.

[3] The CBUS address has been reserved to enable the inter-mixing of CBUS compatible and I²C-bus compatible devices in the same system. I²C-bus compatible devices are not allowed to respond on reception of this address.

[4] The address reserved for a different bus format is included to enable I²C and other protocols to be mixed. Only I²C-bus compatible devices that can work with such formats and protocols are allowed to respond to this address.

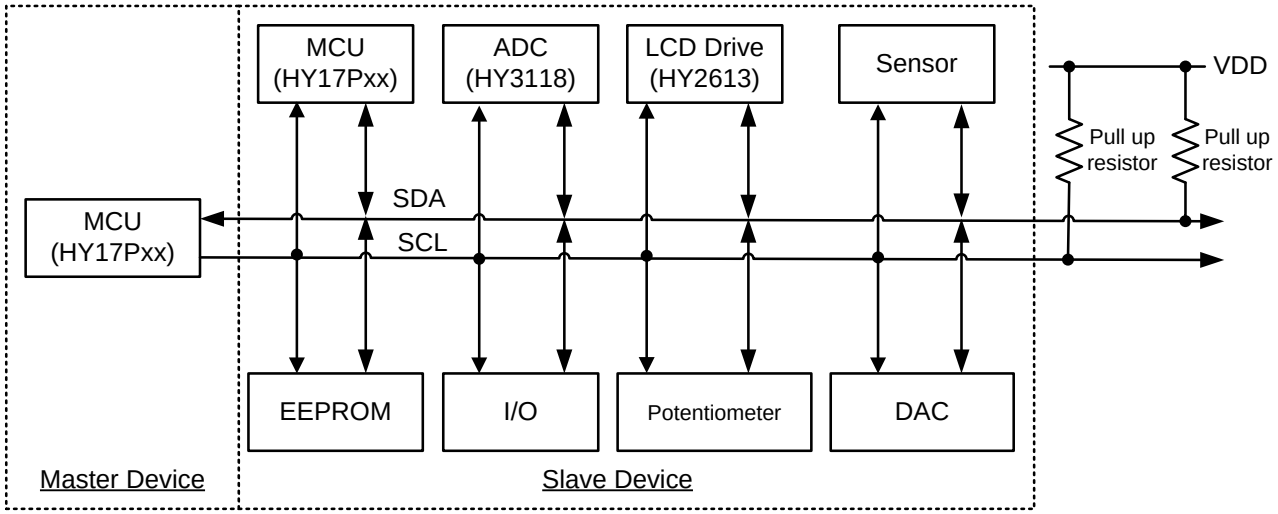


Figure 20-3 Schematic view of I²C communication connection

- 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.

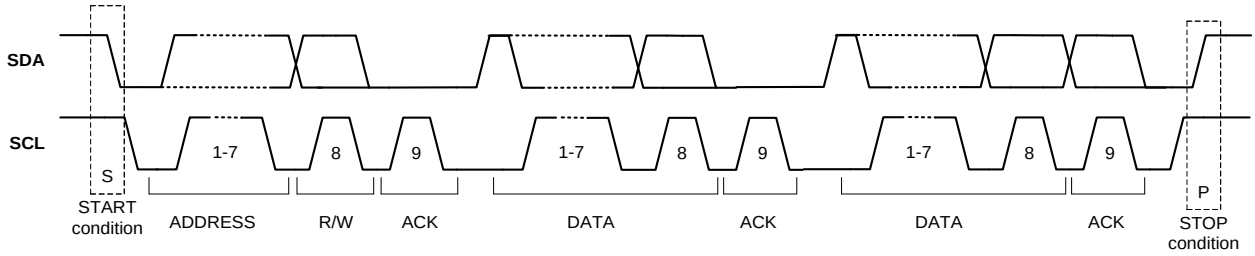


Figure 20-4 Timing diagram of I²C bus

20.1. Data transmission speed calculation

◆ Master Mode

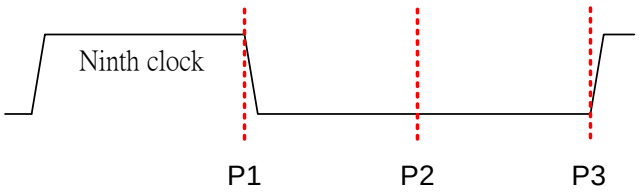
The internal register CRG[7:0] of I²C can control the data transmission speed of the master mode; the value of CRG[7:0] generates the pin signal of SCL of the host machine via the internal timer, so the data transmission speed can be calculated by the following equation according to the frequency of the clock source I2C_CK of I²C:

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

※Note: The size of the pull-up resistor will affect the Data BuardRate. The chip waits until High before calculating the High cycle.

◆ 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)

20.2. Time-Out function

- The time-out control is to avoid that the I2C controller locks the I2C communication bus; so as to provide enough time for MCU to deal with the requirements of the I2C controller during the operation process of I2C, the I2C 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 I2C controller for any reason, SCL of the I2C 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^2C 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.

20.3. Communication flow diagram of I²C serial interface

- Terms of I²C 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 I²C 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 20-6 to show the status of the I²C interface:

    Gray-background circular frame: stands for the I²C status with the interrupt flag.

    White-background circular frame: stands for the I²C status without the interrupt flag and needed by to actively read by MCU.

    Rectangular frame: means that MCU should set the instruction to I²C.

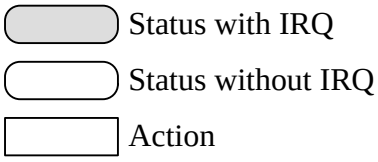


Figure 20-5 Symbols of flow chart

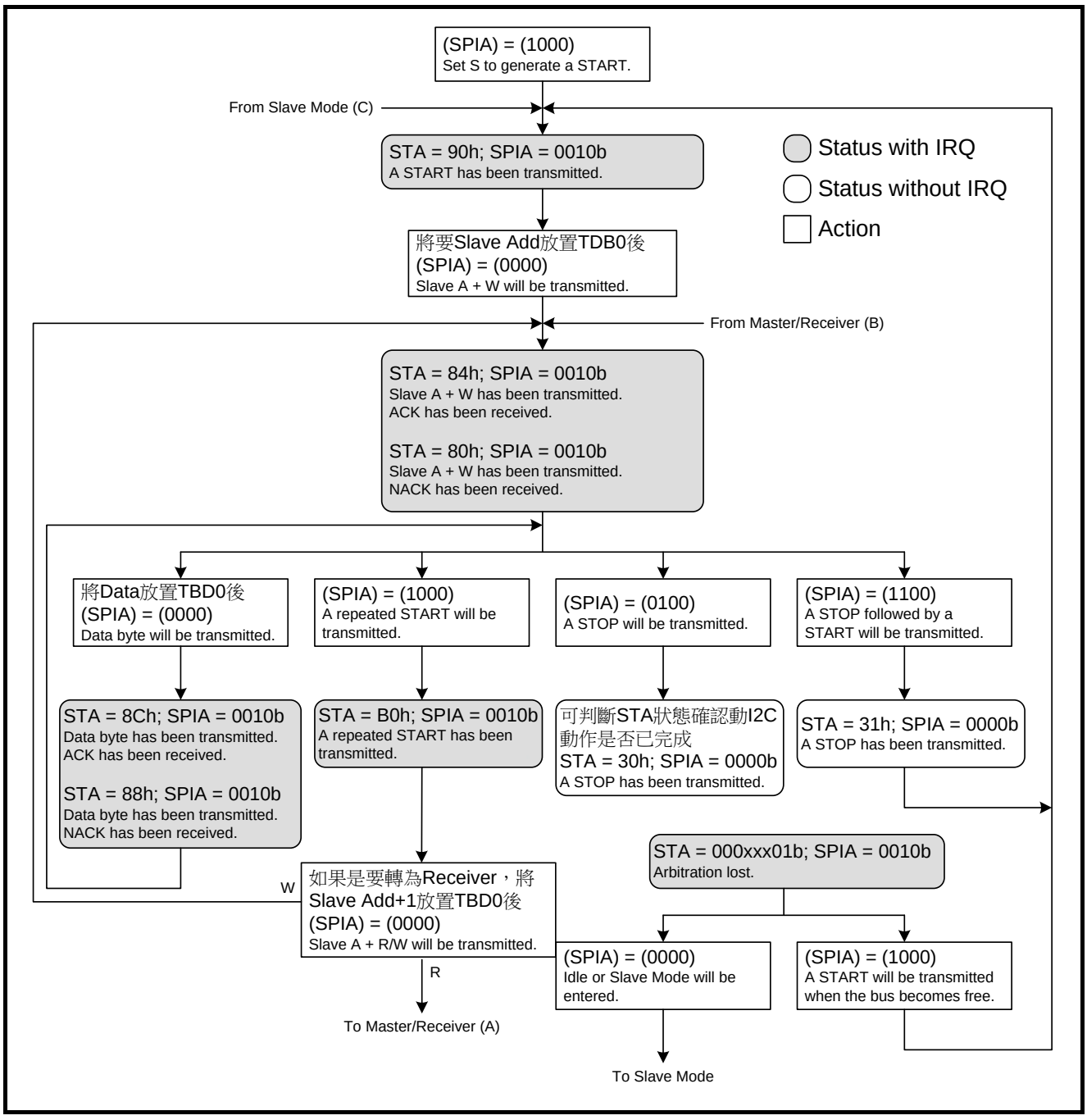


Figure 20-6 Master Transmitter Mode

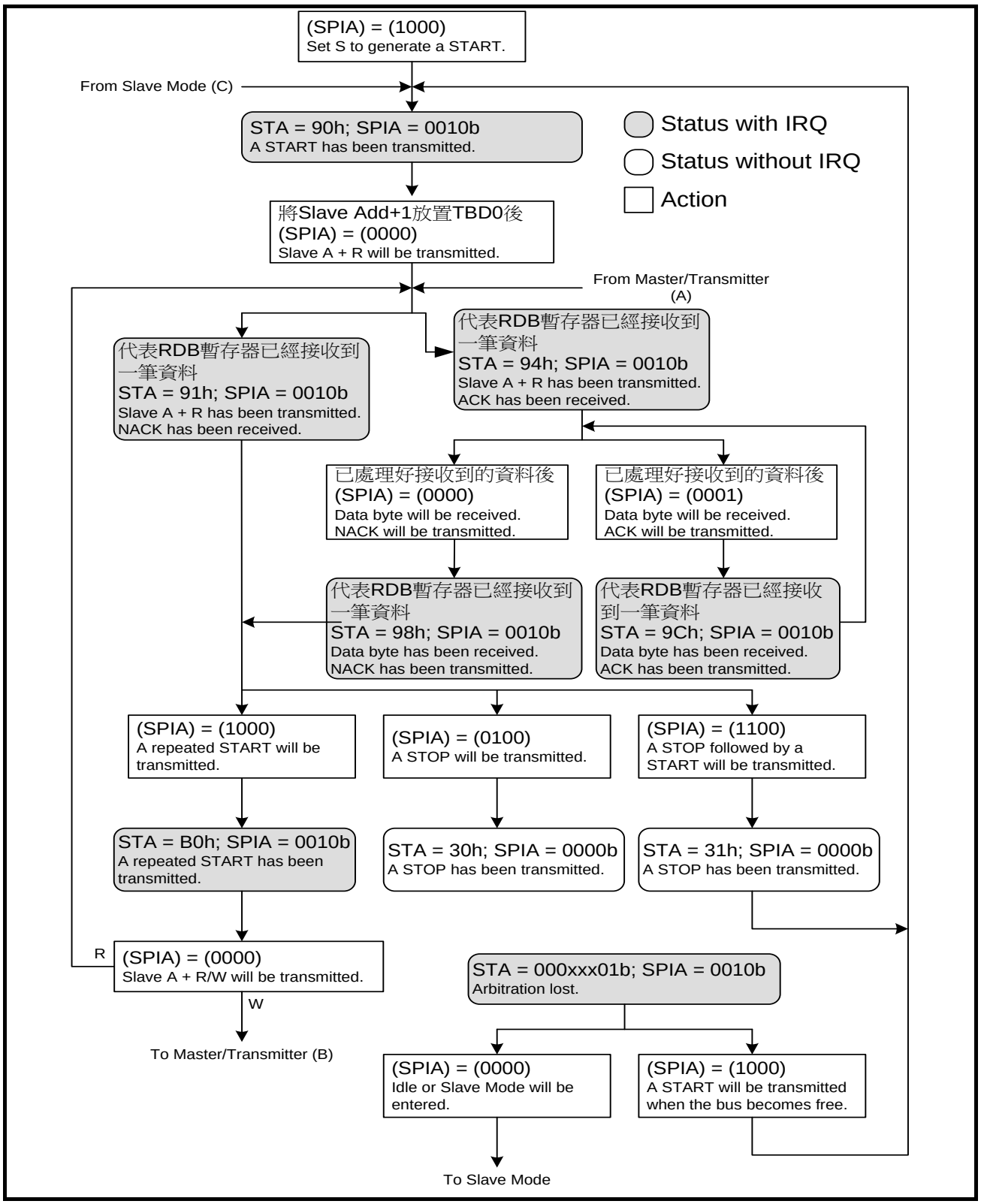


Figure 20-7 Master Receiver Mode

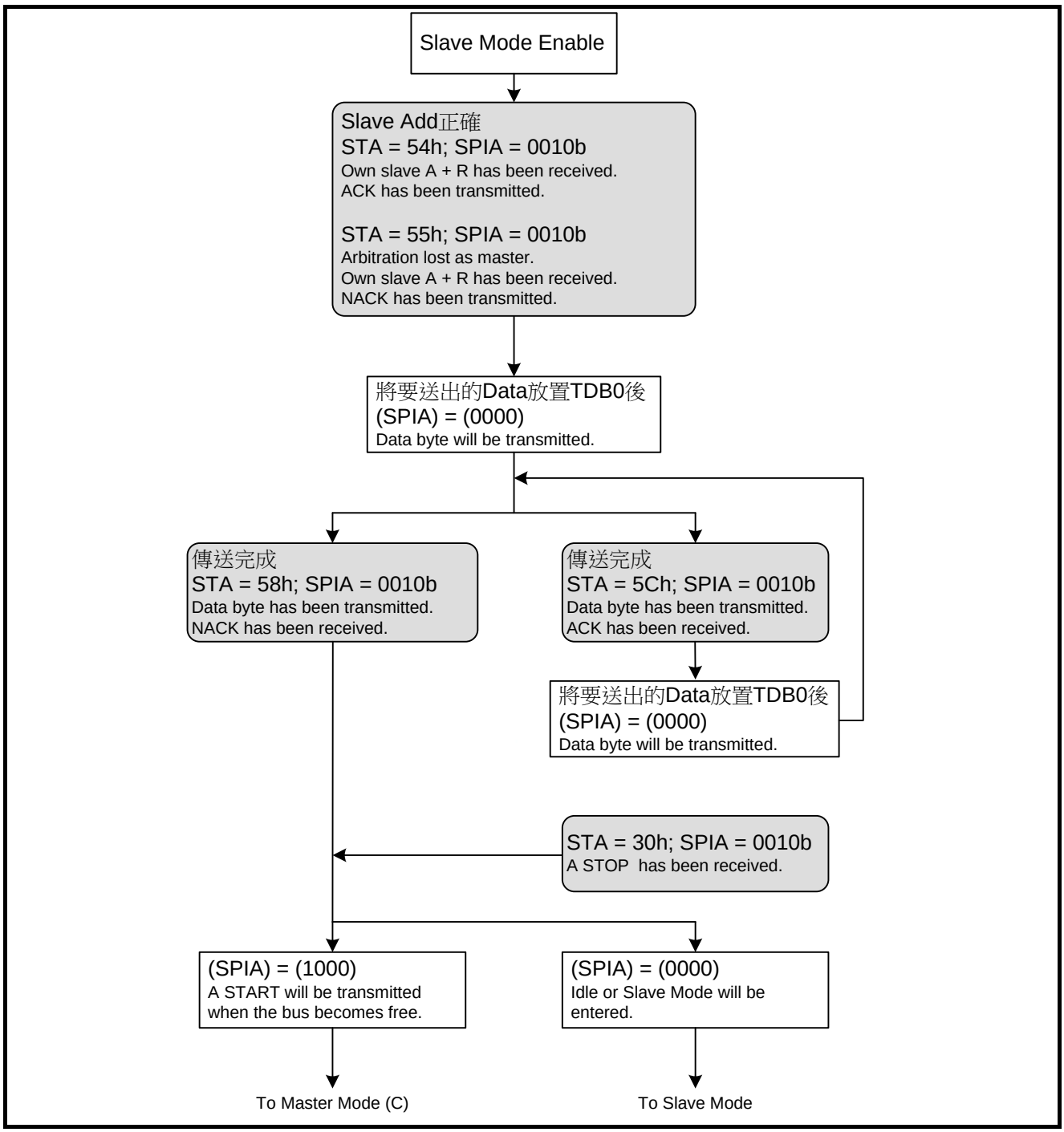


Figure 20-8 Slave Transmitter Mode

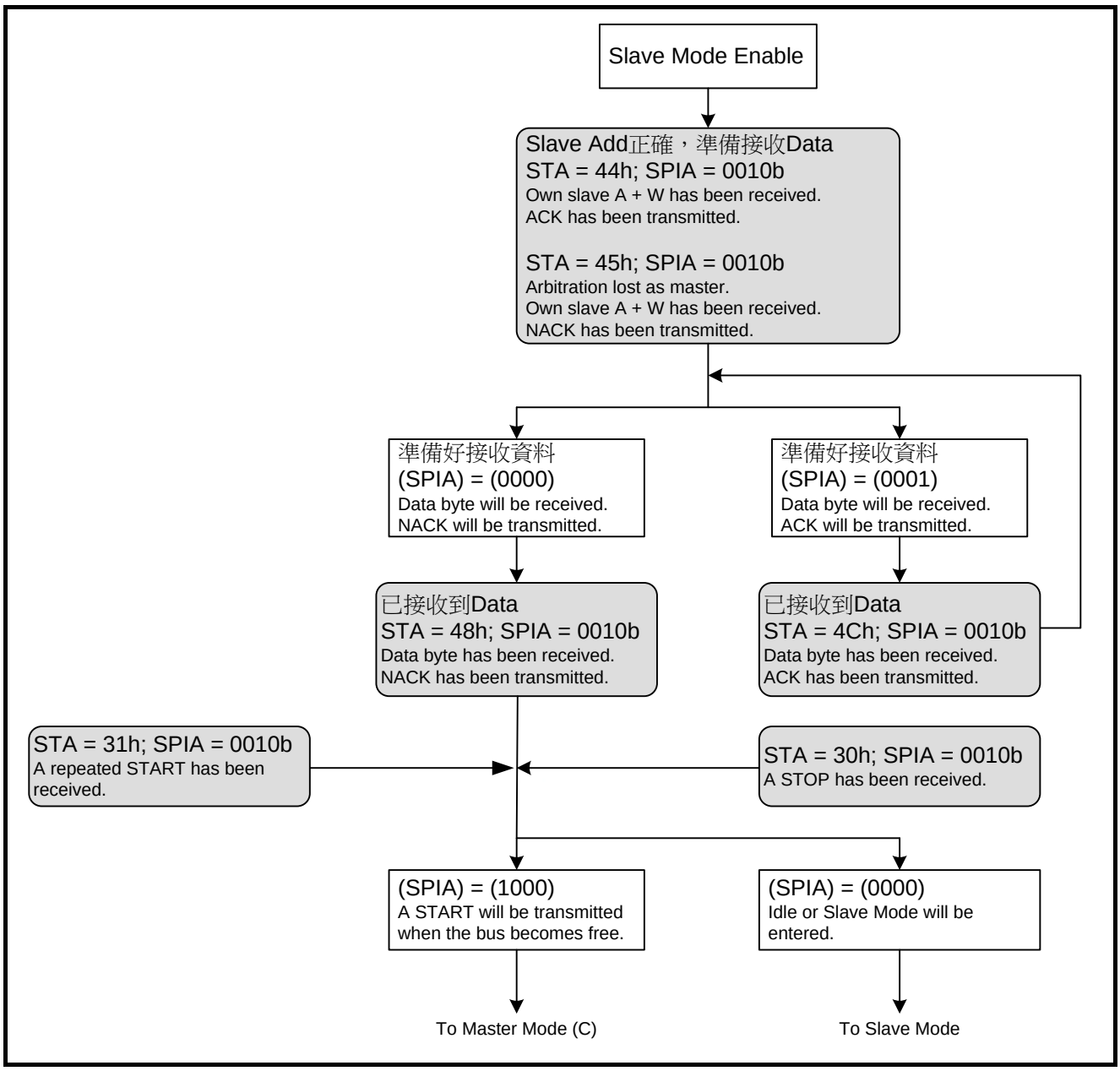


Figure 20-9 Slave Receiver Mode

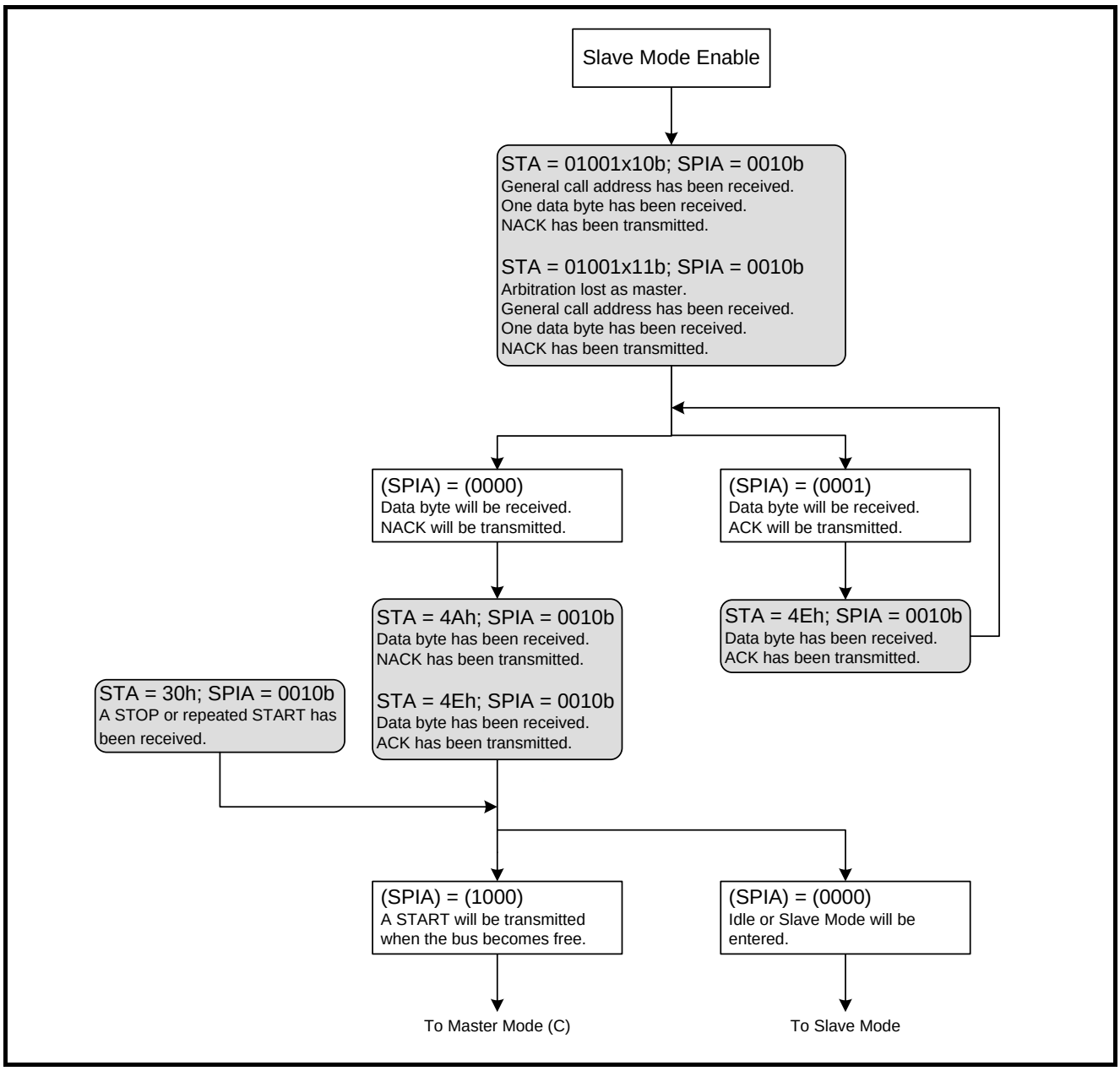


Figure 20-10 General Call Mode

20.1. 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	CTIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024H	INTE1					I2CERIE	I2CIE			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
027H	INTF1					I2CERIF	I2CIF			0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
1C3H	CFG0	Rsv.					GCRst	ENI2CT	ENI2C	…. .000	…. .uuu	-,-,-,- -,*,*
1C4H	ACT0	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C5H	STA0	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C6H	CRG0	CRG[7:0]								0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C7H	TOC0	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
1C8H	RDB0	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1C9H	TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1CAH	SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*

Table 20-1 I²C register

INTE0/INTE1/INTF1：Please refer to the chapter Interrupt for more information

CFG0：I²C Configuration Register

Bit	Name	Description
Bit2	GCRst	I²C general call reset enablement control <0>Disable <1>Enable ※ After the I²C slave mode and the GCRST function are enabled at the same time, if the I²C 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 I²C bus.
Bit1	ENI2CT	Enable I²C time-out monitoring function bit <0>Disable <1>Enable the I²C Time-out monitoring function
Bit0	ENI2C	Enable I²C function control bit <0>Disable <1>Enable the I²C communication interface ※ Notice: When ENI2C is disabled, the internal clock of I²C will be disabled; only the configuration register can perform the write operation, but other registers cannot be written by data.

ACT0: Action Register

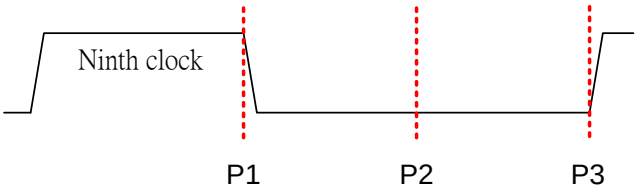
Bit	Name	Description
Bit7	SLAVE	Slave enablement control <0> Disable <1> Enable
Bit4	I2CER	Error interrupt flag <0> Normal; writing 0 will clear the error interrupt flag to make I²C proceed to the next state. <1> Error interrupt occurs.
Bit3	START	Start command bit <0> Normal <1> Generate the "Start" signal on the I²C bus.
Bit2	STOP	Stop command bit <0> Normal <1> Generate the "Stop" signal on the I²C bus.
Bit1	I2CINT	Interrupt flag <0> Normal; writing 0 will clear the interrupt flag to make I²C proceed to the next state. <1> The I²C interrupt occurs.
Bit0	ACK	ACK(Acknowledge) acknowledge bit <0> Reply ACK or reply NACK. <1> ACK has been replied.

STA0: I²C 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

CRG0: I²C 時脈控制暫存器

位元	名稱	描述
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 <0> Normal <1> I²C Bus Clock Stretching Time-out

<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 = I2C_CK / 1</td><td>100</td><td>CLKPS = I2C_CK / 16</td></tr>
<tr><td>001</td><td>CLKPS = I2C_CK / 2</td><td>101</td><td>CLKPS = I2C_CK / 32</td></tr>
<tr><td>010</td><td>CLKPS = I2C_CK / 4</td><td>110</td><td>CLKPS = I2C_CK / 64</td></tr>
<tr><td>011</td><td>CLKPS = I2C_CK / 8</td><td>111</td><td>CLKPS = I2C_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>

RDB0: 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 registe

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)

Bit	Name	Description
Bit0	SIDV[0]	Effective control of slave ID code <0> the slave ID code is invalid. <1> the slave ID code is valid.

21. Enhanced Universal Asynchronous Receiver Transmitter,EUART

The peripherals of the enhanced universal asynchronous receiver transmitter(Enhanced Universal Asynchronous Receiver Transmitter, hereafter referred to as 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 operation and debug modes: :

- The full-duplex asynchronous mode has the following functions:
  Baud transmission rate generator
- Debug mode
  Frame error detection[4]
  Overrun error detection[5]
  Hardware parity check code
- Data transmission and reception
  Asynchronous transmission (8 bits or 9 bits)
  Asynchronous reception (8 bits or 9 bits)
- Byte reception automatic wake-up function

EUART registers include：

UR0CN	ENSP[0],ENTX[0],TX9[0],TX9D[0],PARITY[1:0]
UR0STA	RC9D [0],PERR[0],FERR[0],OERR[0],RCIDL[0],TRMT[0],ABDOVF[0]
BA0CN	ENCR[0],RC9[0],ENADD[0],ENABD[0]
BR0GR[15:0]	BG0RH[7:0], BG0RL[7:0]
TX0R	TX0R[7:0]
RC0REG	RC0REG[7:0]

4 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.

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

21.1. Usage description of EUART

21.1.1. Transmission configuration steps of asynchronous data

- 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 ENSP bit of the UR0CN register to enable the EUART serial I/O module.
- 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)

21.1.2. Reception configuration steps of asynchronous data

- Configure the INTE1, the RCIE bit of the register and the GIE bit of the INTE0 register to determine whether to allow receiving the interrupt enablement.
- 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 9^{th} 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 9^{th} bit of the received data (when RC9 is configured), and then determine whether an error occurs during the receiving process.
- Read the RC0REG 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.

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

- 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 detect function.
- Configure the ENCR bit of the BA0CN register to enable the data receiving function.
- Configure the INTE1, the RCIE bit of the register and the GIE bit of the INTE0 register to determine whether to allow receiving the interrupt enablement. When data reception is complete RCIF bit will be set.
- 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 RC0REG 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 detect function, and make the next data reception.

21.2. Baud 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 21-1 is the serial transmission baud rate calculation equation, but only can be applied to the master mode.

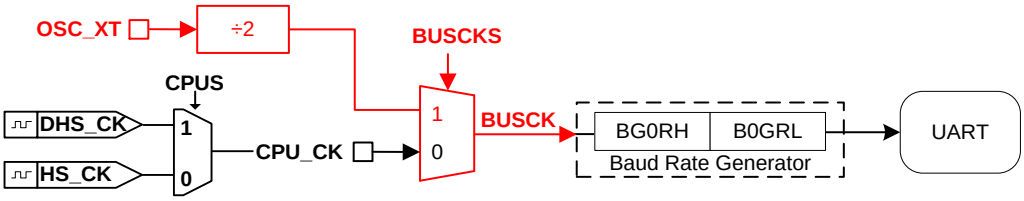


Figure 21-11 UART operating clock configuration diagram

Under the condition that the targeted serial transmission baud rate is given and the operating clock is CPU_CK or OSC_XT÷2, the equation of Table 21-1 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 21-1 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	UART_CK ÷[4 (n + 1)]
UART_CK = CPU_CK(For HY17P60； UART_CK = BUSCK(For HY17P68)； n = BGxRH:BGxRL The correct value of the register	

Table 21-1 Serial transmission baud rate equation

When operating under the asynchronous mode, its operating frequency is CPU_CK (4.9152MHz), but the targeted serial transmission baud rate is 9600bps. Calculate BG0R[15:0]=< ? > and then BG0RH[7:0]:BGxRL[7:0]=< ? >

Known equation:

targeted serial transmission baud rate = UART_CK÷(4(<BG0R[15:0]>+1)):

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

= ((4915200÷9600)÷4)–1

= 127

then, BG0RH[7:0]=<00> · BG0RL[7:0]=<7F>;

PS: 7F is hexadecimal system.

In fact, the Calculation result of BRG is:

actual serial transmission baud rate = 4915200÷4(127+1)) = 9600

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

= (9600-9600)/9600

= 0.00%

Example 21-1 Calculation of error of serial transmission baud rate

21.2.1. 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 operation under a different frequency. It may need to adjust the value of the BG0R[15:0] register.

21.2.2. 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.

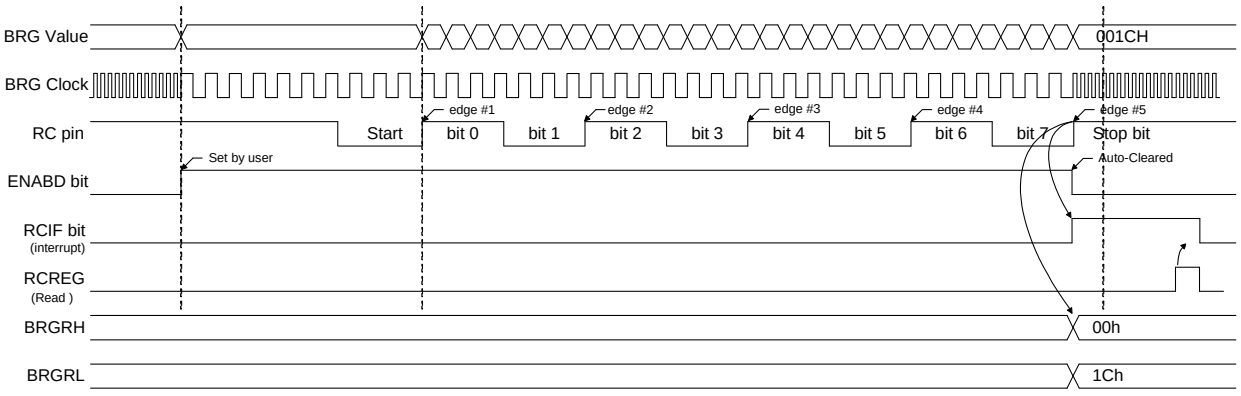
21.2.3. Automatic baud rate

The EUART module supports the automatic detect and calibration functions, which is also called automatic baud rate. The automatic buad 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 detect function begins (the received value should be 055H). After the automatic detect and calibration are finished, the calculated result will be written in BG0RH[4:0] and BG0RL[7:0]; the relevant timing is as shown in Figure 21-12.

When the calculation of BG0R[13:0] 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 21-13.

Automatic BAUD rate calculation



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

Figure 21-12 Waveforms of automatic baud rate calculation

BRG Overflow Sequence

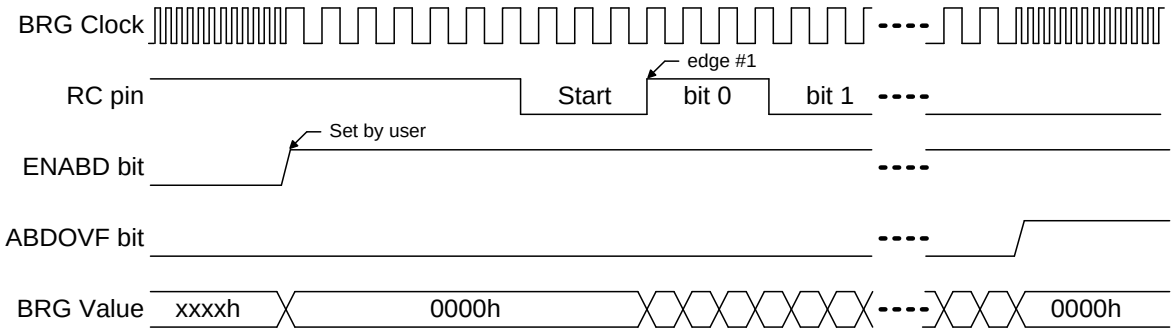


Figure 21-13 Waveforms of automatic baud rate overflow (ABDOVF)

21.3. 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 21-2

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 21-2 Parity check state table

21.4. 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 operating 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.

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

Figure 21-3 is the block 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 21-4 and Figure 21-5 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

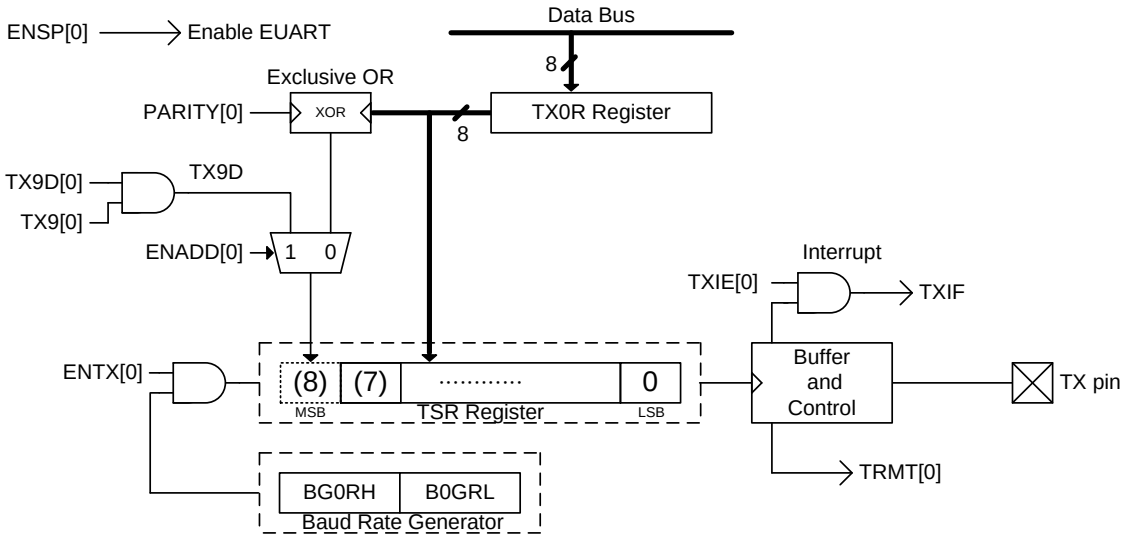


Figure 21-4 EUART transmission block diagram

ASYNCHRONOUS TRANSMISSION

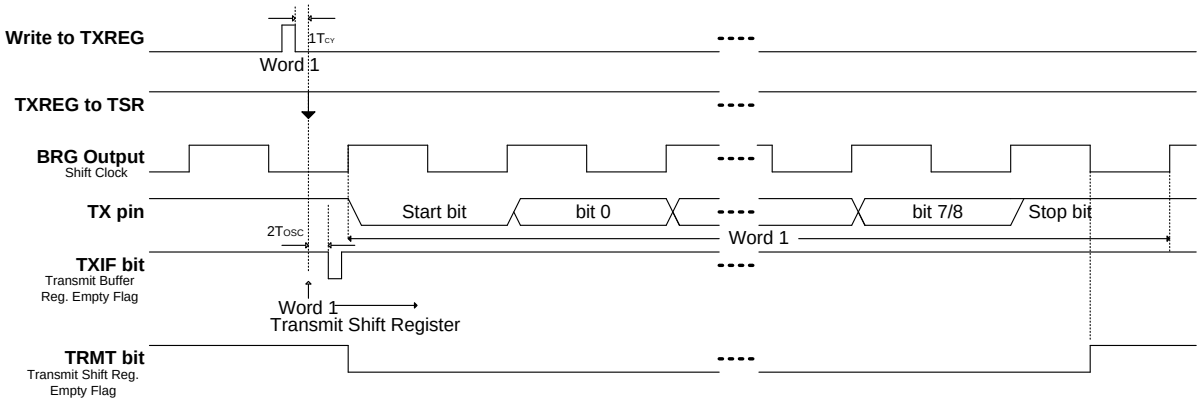


Figure 21-5 Timing diagram of asynchronous transmission

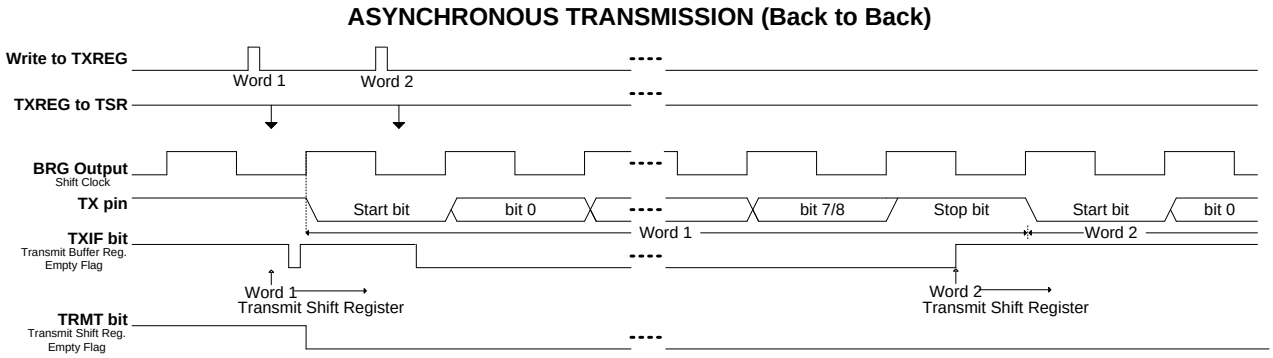


Figure 21-6 Asynchronous transmission timing (back to back)

◆ EUART asynchronous receiver

Figure 21-7 and Figure 21-8 shows the block diagram of the receiver. Figure 21-9 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 operating frequency of 13-bit serial transmission baud rate; besides, the operating frequency of the main receiving serial shifter is equal to the baud rate. 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.

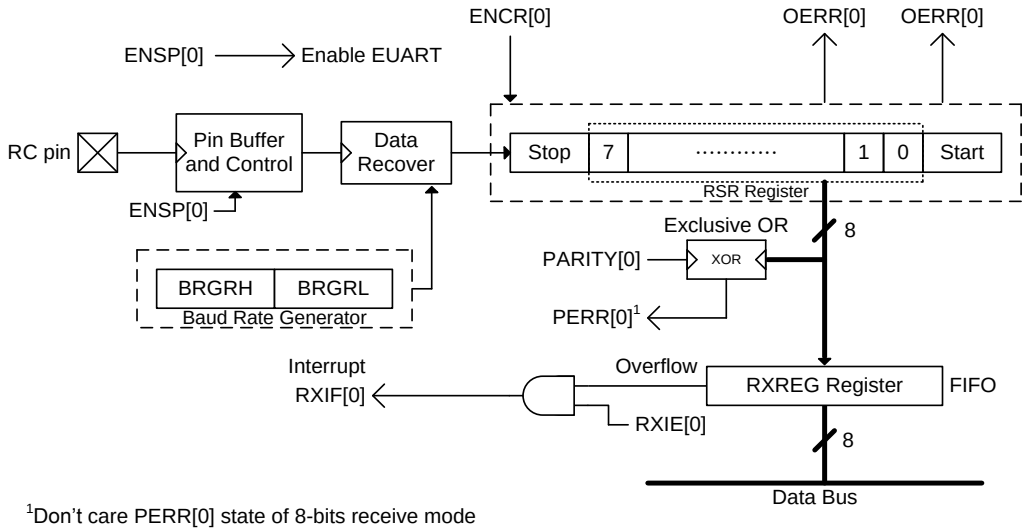
When the RC pin has received 2 pieces of complete byte data (None of the data is read from the RC0REG register), the OERR will be set as 1, it is possible to clear the OERR bit by clearing the ENCR bit.

After the complete data are completely received, the RCIF bit of the INTF1 register will be configured, and the RCIF bit cannot be cleared by instruction when being configured; executing the operation of reading the RC0REG 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 21-7 EUART 8-bits receiving block diagram

EUART 9-BITs RECEIVE BLOCK DIAGRAM

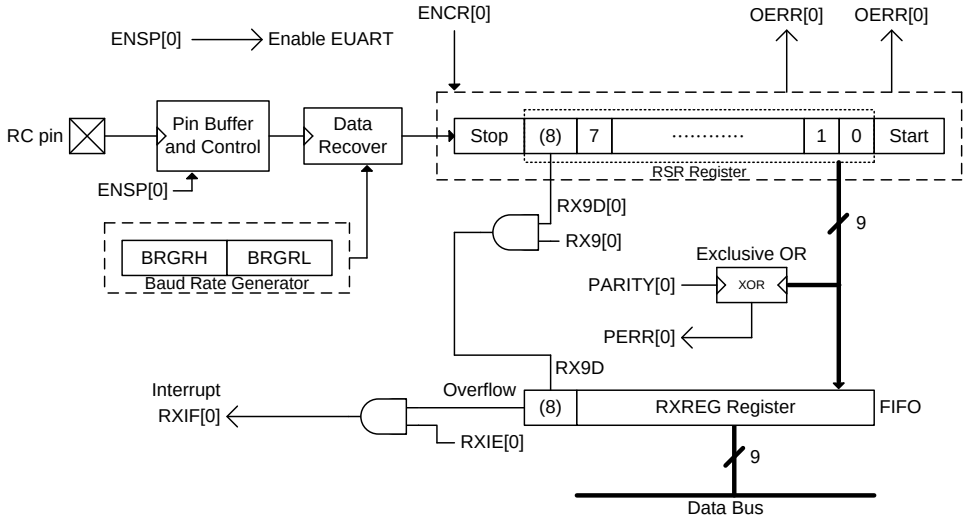


Figure 21-8 EUART 9-bits receiving block diagram

ASYNCHRONOUS RECEPTION

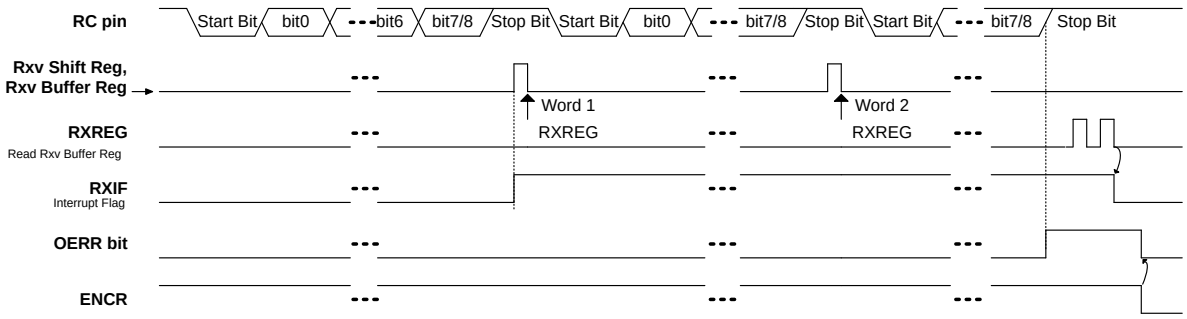


Figure 21-9 Asynchronous receiving timing

- 9-bit mode of address detect function

  The mode is usually applied to the RS-485 system. The user can configure the asynchronous reception operation with the address detect function according to the chapter of the usage description of EUART. The user can determine whether it is the address detect 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 buad 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 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 21-10; 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 21-. The interrupt condition can be cleared by reading the RC0REG 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.

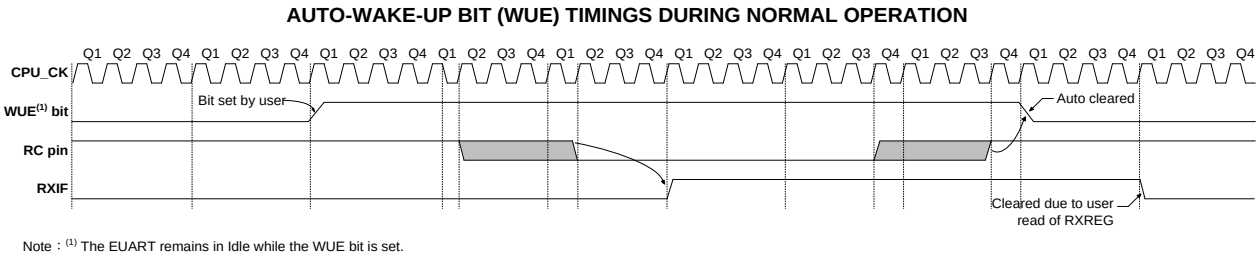


Figure 21-10 Automatic wake-up timing under normal mode

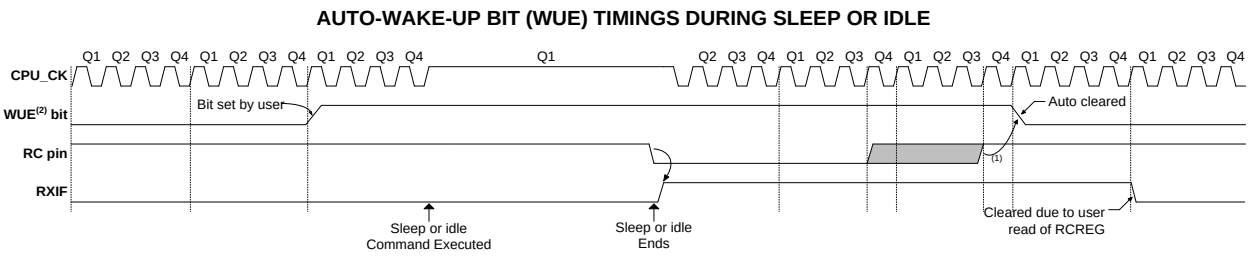


Figure 21-11 Automatic wake-up timing under sleep mode or standby mode

- 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. For standard RS-232 IC, this can be 00h (8 bits).

  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 RC0REG register. Generally speaking, the data in the awaken RC0REG 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 RC0REG 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.

21.5. Regiseter 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	TA1CIE	ADIE	WDTIE	TB1IE	CTIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*,*,*
024h	INTE1			TXIE	RCIE					0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
027h	INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	*,*,*,* r,r,*,*
1CBH	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	*,*,*,*,*,-,-,*
1CCH	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r r,r,r,rw0
1CDH	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD	 0000	.... uuuu	-,-,-,- *,*,*,*'
1CEH	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-,-,-,*,*,*,*,*
1CFH	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D0H	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
1D1H	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r

Table 21-3 UART Register

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

UR0CN: UART control register

Bit	Name	Description
Bit7	ENSP	UART port function enable 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 enable bit <0> Disable <1> Enable
Bit5	TX9	Transmit 9th bit function enable <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 enable 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	2 data unprocessed status flags have been received <0> Not occur <1> Occur
Bit2	RCIDL	Whether the response is the receiving status flag <0> under the reception state <1> not under the reception state
Bit1	TRMT	Indicates the status flag of the 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 enable bit <0> Disable <1> Enable
Bit2	RC9	Receiving 9th bit function enable bit <0> Disable <1> Enable
Bit1	ENADD	Address detect bit <0> Disable <1> Enable
Bit0	ENABD	Automatic baud rate enable bit <0> Disable <1> Enable

BR0RH/BR0RL: Baudrate control register

TX0R: UART data transmission register

RC0REG: UART data receiving register

22. LCD

LCD driving circuit is suitable for LCD production process like TN-LCD and STN-LCD, and it has the following features：

- Regulated charge pump
- Multi-section adjustable drive voltage level
- Support 4-duty, 1/3 bias LCD waveform operation mode
- Selectable input clock source and output frequency
- Equips with Blinking capability

LCD Related Registers：

LCDCN1	ENLCP, LCDV[2:0], ENLB, SELPCLK, LCDBL, ENLCD
LCDCN3	SCM3[1:0], SCM2[1:0], SCM1[1:0], SCM0[1:0]
LCDCN4	SSG21, SSG20, SSG19, SSG18, SSG17, SSG16, SSG15, SSG14
LCDCN5	SSG41, SSG40, SSG39, SSG38, SSG37, SSG36, SSG35, SSG34
LCDCN6	SSG5[1:0], SSG4[1:0], SSG3[1:0], SSG02[1:0]
LCDCN7	SSG9[1:0], SSG8[1:0], SSG7[1:0], SSG6[1:0]
LCDCN8	SSG13[1:0], SSG12[1:0], SSG11[1:0], SSG10[1:0]
LCDCN9	SSG25[1:0], SSG24[1:0] , SSG23[1:0], SSG22[1:0]
LCDCN10	SSG29[1:0], SSG28[1:0] , SSG27[1:0], SSG26[1:0]
LCDCN11	SSG33[1:0], SSG32[1:0] , SSG31[1:1], SSG30[1:0]
LCD[159:0]	LCD0[7:0]~LCD20[7:0]

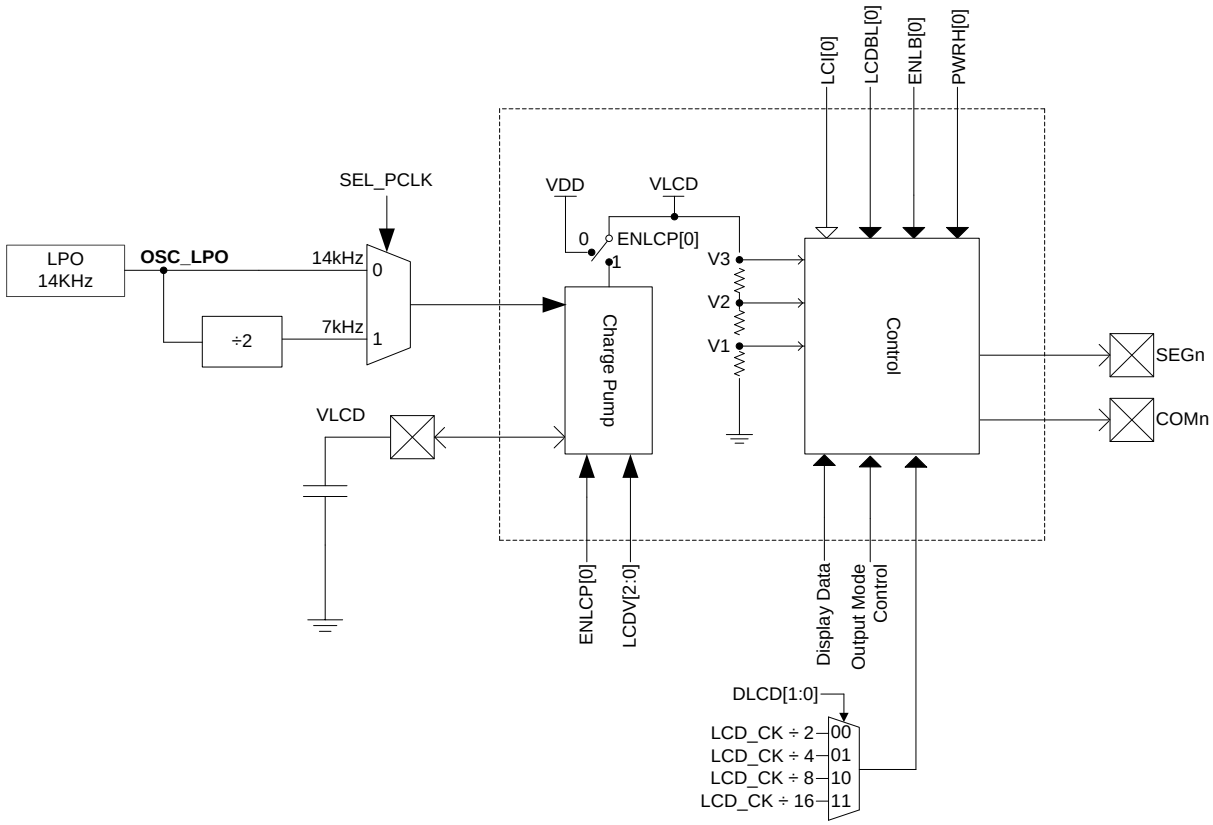


Figure 22-1 LCD Block Diagram

22.1. LCD Manual

22.1.1. Operating Frequency and Output Frame Frequency Configuration

The source of the operating frequency is selected by LCDS. After frequency division by the operating frequency pre-divider DLCD[1:0], an appropriate operating frequency is provided to the LCD to output the frame frequency.

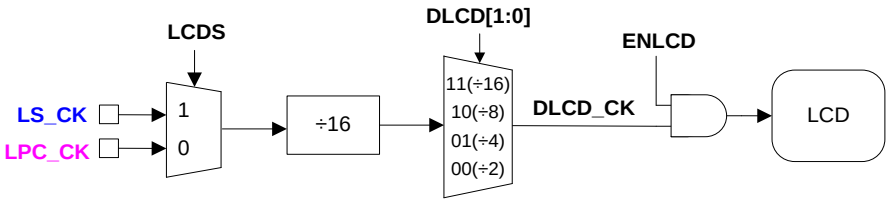


Figure 22-2 LCD operating frequency and frame frequency

22.1.2. Charge pump circuit and LCD operating voltage configuration

There are two ways to generate LCD operating voltage, VLCD:

- VLCD is input externally, ENLCP [0] must be configured as <0> and charge pump must be disabled. Voltage comes through external VLCD pin in order to determine LCD operating voltage. When using external input, the configuration of LCDV[2:0] will not influence LCD operating voltage.
  - When driving over-sized or over-loaded LCD monitor, LCD output buffer, ENLB[0] can be set up as <1> and can initiate buffer to increase LCD driving ability. On the contrary, if ENLB[0] is configured as <0>, LCD consumed current will decrease as the buffer being shut off.
- VLCD is generated by internal charge pump. By setting up charge pump circuit controller, ENLCP[0] as <1> and configure charge pump voltage state controller, LCDV[2:0] , VLCD will be produced to supply LCD power that prevents it against IC operating voltage changes.
  - LCDV[2:0] can set different operating voltage and must be valid when charge pump is enabled. Charge pump circuit may influence analog-to-digital convertor, ΣΔADC performance in high resolution conversion.
  - Using internal charge pump circuit to produce VLCD, LCD buffer will be started automatically by internal hardware circuit.

22.1.3. Blinking Configuration

Blinking effect enables LCD to switch from display status to off status or switching it back to display status. This cycling process only needs digit blink controller, LCDBL[0] to configured as <1> for off or by configuring <0> for full display. Hence, configuring LCDBL[0] as <1>, LCD monitor will not lighten up any digit. That is to say, if LCDBL[0] is set as <0>, LCD monitor will light up according to digit register's configuration of LCD0[7:0]~ LCD19[7:0].

22.1.4. LCD Digit Register

Every digit register, LCDn[7:0] controls two digit pins SEGn, and each digit has 4-bit control bit SEGn[3:0].

22.2. LCD Output Waveform

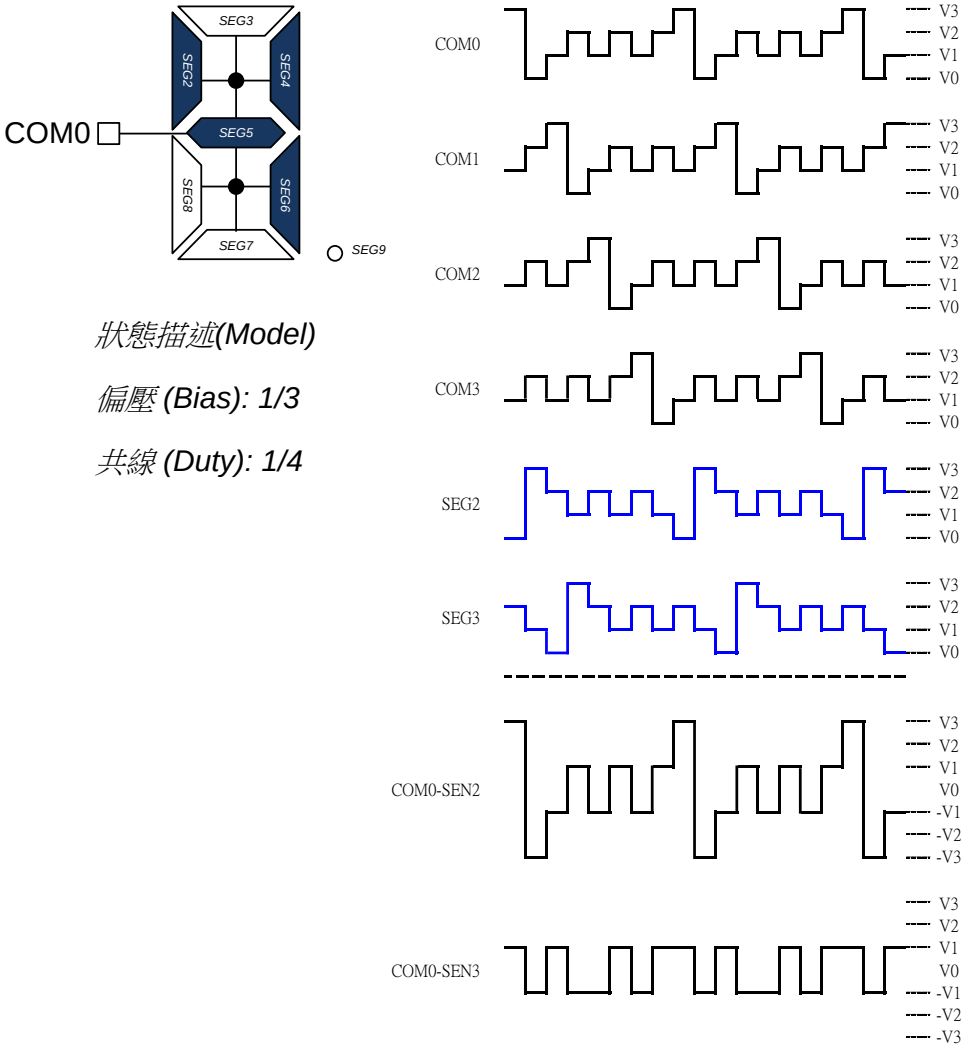


Figure 22-3 Output waveform-4-Mux

22.3. Register Description-LCD

"-"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**
02AH	BSRCN	-	-	-	-	-	BSR[2:0]			 xxxx	.... uuuu	-,-,-,- *,*,*
035H	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
036H	OSCCN1								LCDS	0000 0000	uuuu uuu.	*,*,*,*,*,*,*,-
037H	OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*,*,*,*,*,*,*,r
180H	LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELPCLK	LCDBL	ENLCD	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
181H	LCDCN2									0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
182H	LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
183H	LCDCN4	SSG21	SSG20	SSG19	SSG18	SSG17	SSG16	SSG15	SSG14	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
184H	LCDCN5					SSG37	SSG36	SSG35	SSG34	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
185H	LCDCN6	SSG5[1:0]		SSG4[1:0]		SSG3[1:0]		SSG02[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
186H	LCDCN7	SSG9[1:0]		SSG8[1:0]		SSG7[1:0]		SSG6[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
187H	LCDCN8	SSG13[1:0]		SSG12[1:0]		SSG11[1:0]		SSG10[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
188H	LCDCN9	SSG25[1:0]		SSG24[1:0]		SSG23[1:0]		SSG22[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
189H	LCDCN10	SSG29[1:0]		SSG28[1:0]		SSG27[1:0]		SSG26[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
18AH	LCDCN11	SSG33[1:0]		SSG32[1:0]		SSG31[1:0]		SSG30[1:0]		0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*
18BH	LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18CH	LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18DH	LCD2	LCD SEG7[4:7] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18EH	LCD3	LCD SEG9[4:7] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
18FH	LCD4	LCD SEG11[4:7] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
190H	LCD5	LCD SEG13[4:7] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
191H	LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
192H	LCD7	LCD SEG17[4:7] data				LCD SEG16[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
193H	LCD8	LCD SEG19[4:7] data				LCD SEG18[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
194H	LCD9	LCD SEG21[4:7] data				LCD SEG20[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
195H	LCD10	LCD SEG23[4:7] data				LCD SEG22[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
196H	LCD11	LCD SEG25[4:7] data				LCD SEG24[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
197H	LCD12	LCD SEG27[4:7] data				LCD SEG26[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
198H	LCD13	LCD SEG29[4:7] data				LCD SEG28[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
199H	LCD14	LCD SEG31[4:7] data				LCD SEG30[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19AH	LCD15	LCD SEG33[4:7] data				LCD SEG32[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19BH	LCD16	LCD SEG35[4:7] data				LCD SEG34[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
19CH	LCD17	LCD SEG37[4:7] data				LCD SEG36[3:0] data				xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 22-1 LCD Register

BSRCN: Please refer to Memory Chapter

OSCCN0/OSCCN1/OSCCN2: Please refer to Oscillator, Clock Sources and Power Managed Modes Chapter.

LCDCN1: LCD control register 1

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7</td><td>ENLCP</td><td>LCD charge pump circuit controller
<0> Disable；VLCD is inputted from external pin
<1> Enable；VLCD is generated from internal IC</td></tr>
<tr><td>Bit6~4</td><td>LCDV[2:0]</td><td>Charge pump voltage state select controller (Test Condition : VDD=3.0V, ENLCP[0]=1, C_{VLCD}=4.7uF)
<table>
<tr><th>LCDV [2:0]</th><th>VLCD output voltage</th><th>LCDV [2:0]</th><th>VLCD output voltage</th></tr>
<tr><td>000</td><td>5.05V</td><td>100</td><td>3.0V</td></tr>
<tr><td>001</td><td>4.5V</td><td>101</td><td>2.8V</td></tr>
<tr><td>010</td><td>4.05V</td><td>110</td><td>2.6V</td></tr>
<tr><td>011</td><td>3.3V</td><td>111</td><td>2.4V</td></tr>
</table></td></tr>
<tr><td>Bit3</td><td>ENLB</td><td>LCD output buffer
<0> Disable
<1> Enable, it is recommended to set to '1' in LCD output mode.</td></tr>
<tr><td>Bit2</td><td>SELPCLK</td><td>LCD charge pump clock source control
<0> 14kHz (Default)
<1> 7kHz, slower speed can save power</td></tr>
<tr><td>Bit1</td><td>LCDBL</td><td>LCD digit blink controller
<0> LCD normal display
<1> LCD off display</td></tr>
<tr><td>Bit0</td><td>ENLCD</td><td>LCD Enable control bit
<0> Disable(Default)
<1> Enable</td></tr>
</table>

LCDCN3: LCD control register 3

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit7~6</td><td>SCM3[1:0]</td><td rowspan="4">Pin multiplexing function selection (x=0~3)
<00> Default
<11> LCD COMx pin setting</td></tr>
<tr><td>Bit5~4</td><td>SCM2[1:0]</td></tr>
<tr><td>Bit3~2</td><td>SCM1[1:0]</td></tr>
<tr><td>Bit1~0</td><td>SCM0[1:0]</td></tr>
</table>

LCDCN4: LCD control register 4

Bit	Name	Description
Bit7	SSG21	Pin function selection（x=14~21）
Bit6	SSG20	<0> PT8 digital pin function. Need to set the TRISC register. (Default)
Bit5	SSG19	<1> LCD SEGx pin setting
Bit4	SSG18	
Bit3	SSG17	
Bit2	SSG16	
Bit1	SSG15	
Bit0	SSG14	

LCDCN5: LCD control register 5

Bit	Name	Description
Bit7	-	Pin function selection（n=4~7,x=34~37）
Bit6	-	<0> PT10.n digital pin function. Need to set the TRISC register. (Default)
Bit5	-	<1> LCD SEGx pin setting
Bit4	-	
Bit3	SSG37	
Bit2	SSG36	
Bit1	SSG35	
Bit0	SSG34	

LCDCN6: LCD control register 6

Bit	Name	Description
Bit7~6	SSG5[1:0]	Pin multiplexing function selection（n=4~7 ,x=2~5）
Bit5~4	SSG4[1:0]	<00>PT6.n digital pin function. Need to set the TRISC register. (Default)
Bit3~2	SSG3[1:0]	<01> Rsv.
Bit1~0	SSG2[1:0]	<10> Rsv.
		<11> LCD SEGx pin setting

LCDCN7: LCD control register 7

Bit	Name	Description
Bit7~6	SSG9[1:0]	Pin multiplexing function selection（n=0~3,x=6~9）
Bit5~4	SSG8[1:0]	<00>PT7 digital pin function. Need to set the TRISC register. (Default)
Bit3~2	SSG7[1:0]	<01> Rsv.
Bit1~0	SSG6[1:0]	<10> Rsv.
		<11> LCD SEGx pin setting

LCDCN8: LCD control register 8

Bit	Name	Description
Bit7~6	SSG13[1:0]	Pin multiplexing function selection（n=4~7, x=10~13）
Bit5~4	SSG12[1:0]	<00>PT7.n digital pin function. Need to set the TRISC register. (Default)
Bit3~2	SSG11[1:0]	<01> Rsv.
Bit1~0	SSG10[1:0]	<10> Rsv.
		<11> LCD SEGx pin setting

LCDCN9: LCD control register 9

Bit	Name	Description
Bit1~0	SSG22[1:0]	Pin multiplexing function selection（n=0~3,x=22~25）
		<00> PT9.n digital pin function. Need to set the TRISC register. (Default)
		<01> Rsv.
		<10> Rsv.
		<11> LCD SEGx pin setting

LCDCN10: LCD control register 10

Bit	Name	Description
Bit7~6	SSG29[1:0]	Pin multiplexing function selection（n=4~7,x=26~29）
Bit5~4	SSG28[1:0]	<00> PT9.n digital pin function. Need to set the TRISC register. (Default)
Bit3~2	SSG27[1:0]	<01> Rsv.
Bit1~0	SSG26[1:0]	<10> Rsv.
		<11> LCD SEGx pin setting

LCDCN11: LCD control register 11

Bit	Name	Description
Bit7~6	SSG33[1:0]	Pin multiplexing function selection（n=0~3,x=30~33）
Bit5~4	SSG32[1:0]	<00>PT10.n digital pin function. Need to set the TRISC register. (Default)
Bit3~2	SSG31[1:0]	<01> Rsv.
Bit1~0	SSG30[1:0]	<10> Rsv.
		<11> LCD SEGx pin setting

LCD0~LCD17：LCD Byte Data Register

HY17P60B(4COM*20SEG) => COM0~COM3,SEG2~SEG21

名稱	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCD0	Segment SEG3				Segment SEG2			
LCD1	Segment SEG5				Segment SEG4			
LCD2	Segment SEG7				Segment SEG6			
LCD3	Segment SEG9				Segment SEG8			
LCD4	Segment SEG11				Segment SEG10			
LCD5	Segment SEG13				Segment SEG12			
LCD6	Segment SEG15				Segment SEG14			
LCD7	Segment SEG17				Segment SEG16			
LCD8	Segment SEG19				Segment SEG18			
LCD9	Segment SEG21				Segment SEG20			

HY17P68(4COM*36SEG) => COM0~COM3,SEG2~SEG37

名稱	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCD0	Segment SEG3				Segment SEG2			
LCD1	Segment SEG5				Segment SEG4			
LCD2	Segment SEG7				Segment SEG6			
LCD3	Segment SEG9				Segment SEG8			
LCD4	Segment SEG11				Segment SEG10			
LCD5	Segment SEG13				Segment SEG12			
LCD6	Segment SEG15				Segment SEG14			
LCD7	Segment SEG17				Segment SEG16			
LCD8	Segment SEG19				Segment SEG18			
LCD9	Segment SEG21				Segment SEG20			
LCD10	Segment SEG23				Segment SEG22			
LCD11	Segment SEG25				Segment SEG24			
LCD12	Segment SEG27				Segment SEG26			
LCD13	Segment SEG29				Segment SEG28			
LCD14	Segment SEG31				Segment SEG30			
LCD15	Segment SEG33				Segment SEG32			
LCD16	Segment SEG35				Segment SEG34			
LCD17	Segment SEG37				Segment SEG36			

23. Build-In EPROM, BIE

Build-In EPROM (Build-In EPROM, hereinafter referred to as 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 or use low voltage programming control circuit, and use the programming instruction to uses the function.HY17P6x series stored address range is 64 words (equivalent to 128 bytes) in the Information block 00H~3FH and the last 1k Words (equivalent to 2048 bytes) of the OTP of the Main block. The size of the Build-In EEPROM varies depending on the chip.

When use external VBIE power supply (8.5V) to program BIE area, must call the programming subprogram (EXT17P6xWR3) to perform programming; able to program one byte (word) one time data in BIE area through order.

When using the low-voltage programming control circuit, the BIE block can be programmed without an external VBIE power supply, but the programming subprogram (LV17P6xWR3) must be called to perform programming; each time the programming subprogram is called the programming action can only program one word data, and it takes about 500msec.

BIE Related Registers：

BIECN BLKSEL[0], ENBVD[0], VPPHV[0], ENBCP[0], BIEWR[0], BIERD[0]

BIEARH BIE_ADDR[13:8]

BIEARL BIE_ADDR[7:0]

BIEDRH BIE_DATA[15:8]

BIEDRL BIE_DATA[7:0]

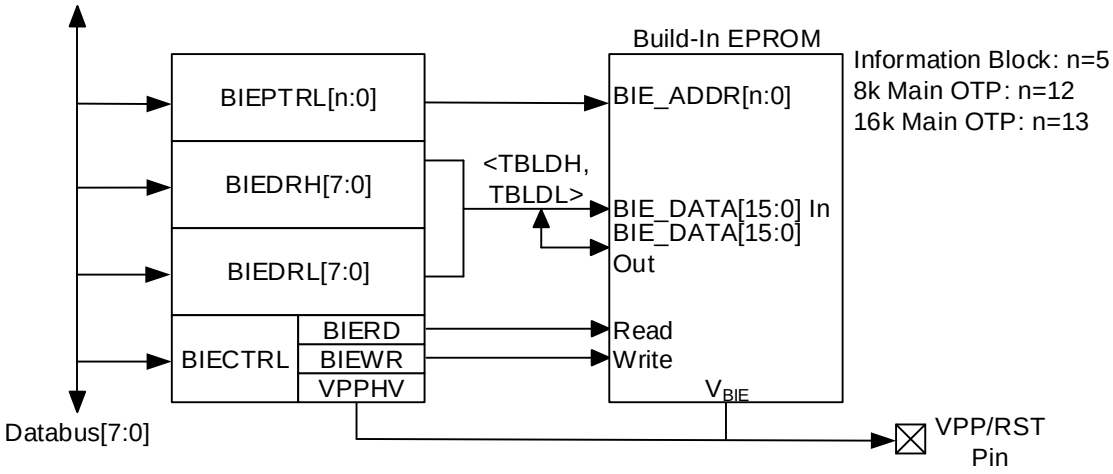


Figure 23-1 BIE Block Diagram

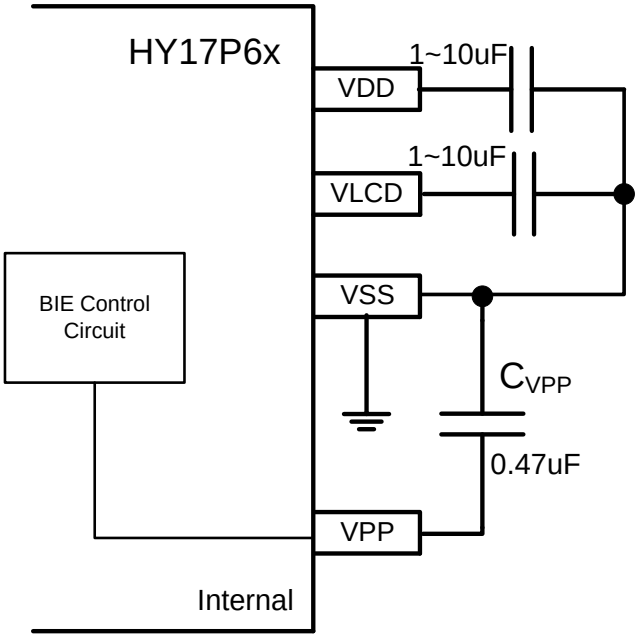


Figure 23-2 BIE internal charge pump Block Diagram

23.1. 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**
02DH	BIECN	1	BLKSEL	-	ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1… $000	1... $uuu	r1,-,-,- r,*,*,*
02EH	BIEARH	-	-	BIE High Byte Address Register as BIEA[13:8]						0... xxxx	u... uuuu	*,-,-,- *,*,*,*
02FH	BIEARL	BIE Low Byte Address Register as BIEA[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
030H	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
031H	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 23-1 BIE Register

BIECN: BIE control register

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>Bit6</td><td>BLKSEL</td><td>BIE block selection register
<0> Select Information block (00H~3FH total 64 words)
<1> Select the Main block (the last 1k Word of OTP)
8k Word OTP products: BIEAR=1C00H ~ 1FFFH
16k Word OTP products: BIEAR=3C00H ~ 3FFFH</td></tr>
<tr><td>Bit4</td><td>ENBVD</td><td>BIE voltage detection control register
<0> Disable
<1> Enable</td></tr>
<tr><td>Bit3</td><td>VPPHV</td><td>VPP voltage condition flag
<table>
<tr><th>VPPHV</th><th>ENBCP=0
(VPP pin external input)</th><th>ENBCP=1
(VPP pin internal charge pump)</th></tr>
<tr><td><0></td><td>VPP is not connected to the programming power supply 8.5V or the voltage is lower than 8.5V</td><td>VPP voltage less than 8.5V</td></tr>
<tr><td><1></td><td>VPP external programming voltage has reached 8.5V</td><td>VPP power supply has reached 8.5V
※ENLCD, ENLCP, ENBCP must be set to "1" at the same time, and the VLCD output voltage must be set to 5.05V (LCDV [2:0]=<000>)</td></tr>
</table></td></tr>
<tr><td>Bit2</td><td>ENBCP</td><td>BIE charge pump enable control register
<0> Disable
<1> Enable</td></tr>
<tr><td>Bit1</td><td>BIEWR</td><td>Write in the EPROM control bit
<0> Cannot be written in
<1> Can be written in</td></tr>
<tr><td>Bit0</td><td>BIERD</td><td>Read EPROM control bit
<0> Cannot be read</td></tr>
</table>

Bit	Name	Description
		<1> Can be read

BIEARH: EPROM High Byte address definition

Bit	Name	Description
Bit5~0	BIE_ADDR[13:8]	OTP address(Please refer to BLKSEL control bit description for range)

BIEARL: EPROM Low Byte address definition

BIE_ADDR[7:0]: OTP/BIE address(Please refer to BLKSEL control bit description for range)

BIEDRH: EPROM High Byte data definition

BIEDRL: EPROM Low Byte data definition

24. Revision Record

version	Page	Date	Summary
V01	All	2021/10/20	First edition
V02	All	2022/12/16	Modify Figure 14-1 OPA Block Diagram