HYCON TECHNOLOGY

HY11P Series Family

User's Guide

Mixed Signal Microcontroller

Table of Contents

1. READING GUIDANCE 6

1.1. Terms and Definition 7

2. CPU 9

2.1. CPU Core 9

2.2. Memory 10

3. OSCILLATOR, CLOCK SOURCES AND POWER MANAGED MODES 24

3.1. Oscillator 24

3.2. CPU and Peripheral Circuit Clock Sources 26

3.3. Register Description-Operating Clock Source Controller 30

3.4. Power Managed and Operation Mode 33

4. RESET 37

4.1. Reset Events Description 38

4.2. Status Registers 39

4.3. Register List-Data Memory Reset Status 42

5. INTERRUPT 44

5.1. Register Description-Interrupt 45

6. HARDWARE MULTIPLIER 51

7. INPUT/OUTPUT PORT, I/O 52

7.1. PORT Related Register Introduction 53

7.2. Buzzer 54

7.3. I/O Port 1 55

7.4. Input/Output Port 2, I/O Port2 58
7.5. Input/Output Port 3, I/O Port3 61
7.6. Input/Output Port 4, I/O Port4 63
7.7. Input/Output Port 5, I/O Port5 65
8. LOW VOLTAGE DETECT 67
8.1. Low Voltage Detect Manual 68
8.2. Register Description-LVD 70
9. WATCH DOG TIMER 71
9.1. WDT Manual 71
9.2. Register Description-WDT 73
10. TIMER-A 74
10.1. TMA Manual 75
10.2. Register Description-TMA 76
11. TIMER-B 77
11.1. Timer-B Manual 78
11.2. Register Description-TMB 80
12. TIMER-C 81
12.1. Timer-C Manual 82
12.2. Register Description-TMC 83
13. CAPTURE/COMPARE MODE 85
13.1. Capture Mode Manual 86
13.2. Compare Mode Manual 88

13.3. Register Description-Capture/Compare 90

14. FREQUENCY GENERATOR, PWM/PFD 91

14.1. PFD Mode Manual 92

14.2. PWM Mode Manual 93

14.3. Register Description-PFD/PWM 101

15. POWER SYSTEM 103

15.1. VDDA Manual 104

15.2. ACM Manual 104

15.3. Register Description-PWR 105

16. ENHANCED COMPARATOR 106

16.1. ECPA Manual 107

16.2. Register Description-ECPA 110

17. LOW NOISE OPAMP1 113

17.1. LNOP1 Manual 114

17.2. Register Description-LNOP1 115

18. LOW NOISE OPAMP2 116

18.1. LNOP2 user Instruction 117

18.2. Register Description-LNOP2 118

19. ANALOG-TO-DIGITAL CONVERTER SD18,ΣΔADC 119

19.1. SD18 Manual 121

19.2. Analog Channel Input Characteristics 126

19.3. Absolute Temperature Sensor, TPS 129

19.4. Register Description-SD18 131

20. LCD 136

20.1. LCD Manual 138

20.2. LCD Output Waveform 141

20.3. Register Description-LCD 146

21. SERIAL PERIPHERAL INTERFACE 148

21.1. SPI Manual 149

21.2. SPI Master Mode 150

21.3. SPI Slave Mode 151

21.4. SPI Master Module Transmission Way 154

21.5. Register Description-SPI 157

22. ENHANCED UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER 159

22.1. EUART Manual 160

22.2. Baud Rate Generator, BRG 161

22.3. Hardware Parity Check 163

22.4. EUART Asynchronous Mode 163

23. BUILT-IN EPROM 170

23.1. BIE Manual 171

23.2. Register Description-BIE 186

24. REVISIONS 187

1. Reading Guidance

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.

1.1. Terms and Definition

1.1.1. 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
SD18	Sigma-Delta ADC
SR	Special Register
SRAM	Static Random Access Memory
STK	Stack
WDT	Watch Dog Timer
WREG	Work Register

1.1.2. 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 (H08) 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. Furthermore, to strengthen user's design competitiveness, core processing methods are divided into two versions, H08A and H08B.

CPU features include：

- Program memory and data memory independent design architecture, making the instruction execution speed and improve CPU efficiency.
- Maximum address ability: 1MW for program memory and 4096KB for data memory.
- At most 67 instructions including 16-bit look-up-table, 8x8 hardware multiplier and program memory 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 divided into 2 versions, namely H08A and simplified H08B 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[5:0],PCLATH[5:0],PCLATL[7:0]
TOS[13:0]	TOSH[5:0],TOSL[7:0]
FSRx[9:0]	FSRxH[9:8],FSRxL[7:0]
INDFx	INDF0[7:0],INDF1[7:0]
POINCx	POINC0[7:0], POINC1[7:0]
PODECx	PODEC0[7:0], PODEC1[7:0]
PRINCx	PRINC0[7:0], PRINC1[7:0]
PLUSWx	PLUSW0[7:0], PLUSW1[7:0]
STKCN	STKFL[0],STKOV[0],STKUN[0],STKPRT[4:0]
PSTATUS	SKERR[0]
BSRCN	BSR[3:0]

2.2.1. Program Memory

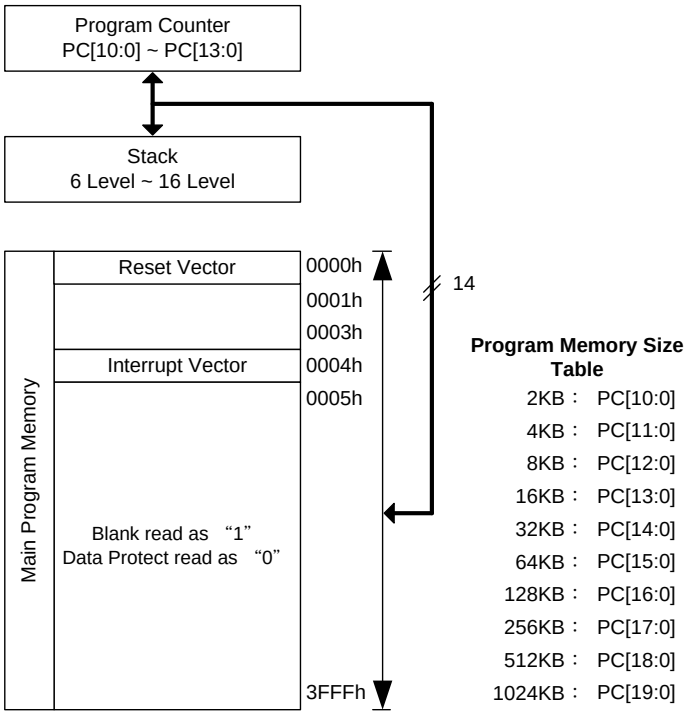


Figure 2-1 Program Memory Flame

2.2.1.1. Main Program Memory, MPM

The frame of main program memory is as follows：

- Interrupt Vector
- Reset Vector

Maximum Address ability[1] starting from 0x00000h to 0xFFFFFh, the entire capacity is 1048576 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.

[1] Program memory address ability varies from every product scheme. Common capacitiy is 2KB(0x7FFh), 4KB(0xFFFh), 8KB(0x1FFFh) and 16KB (0x3FFFh, HY11S14 emulation IC capacity).

2.2.1.2. Program Counter, PC

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

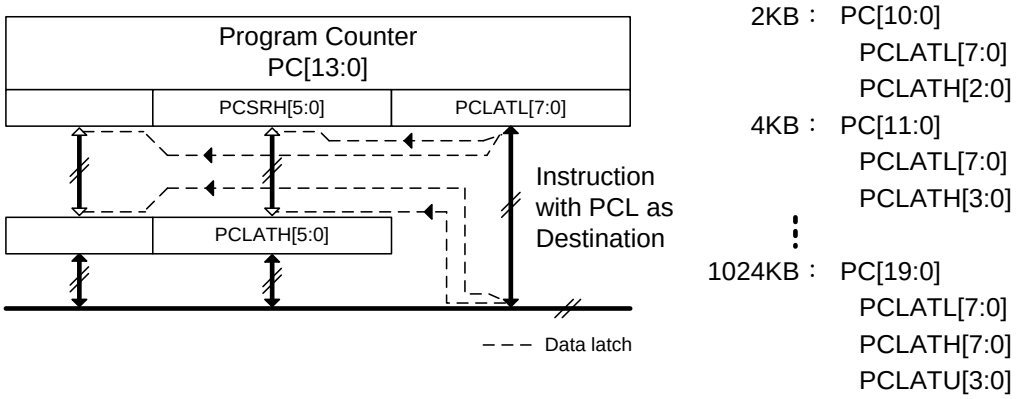


Figure 2-2 Program Counter Frame

PC [13:0][2] of the ICE equips with 14 bit data length and is composed by two registers: PCSRH [5:0] and PCLATL [7:0]. PCLATL [7:0] and PCLATH [5:0] can be directly Read/Written but PCSRH [5:0] cannot. Buffer register, PCLATH [5:0] 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[5:0] in order to obtain correct data. Any reverse order may result in incorrect data.
- To write PC[13:0], PCLATH[5:0] must be written first then to write PCLATL[7:0]. Any reverse order may result in incorrect data.

[2] Program memory address ability varies from every product scheme. Common capacities are 2KB (0x3FFh), 4KB (0xFFFh), 8KB (0x1FFFh) and 16KB (0x3FFFh, HY11S14 emulation IC capacity).

```
ORG  0000
     JMP   START
ORG  0004H
     RETI
     ...
START:                    ;jump to 0109h
     MVFF PCLATL,B1
     INF   PCLATH,F,ACCE
     MVL   2
     ADDF B1,W,ACCE
     MVF   PCLATL,F,ACCE
     ...
ORG  0109H
     NOP   ...
```

Example 2-1 Read/Write PCLAT Example Program

2.2.1.3. Stack, STK

Stack, STK is mainly composed by Stack Index Control Register (STKCN), Top-of-Stack Register (TOSx), Stack Layer Register (STKn3), Stack Error Flag Bit (SKERR) and Stack Error Reset Controller (SKRST[0]). As presented in Figure 2-3.

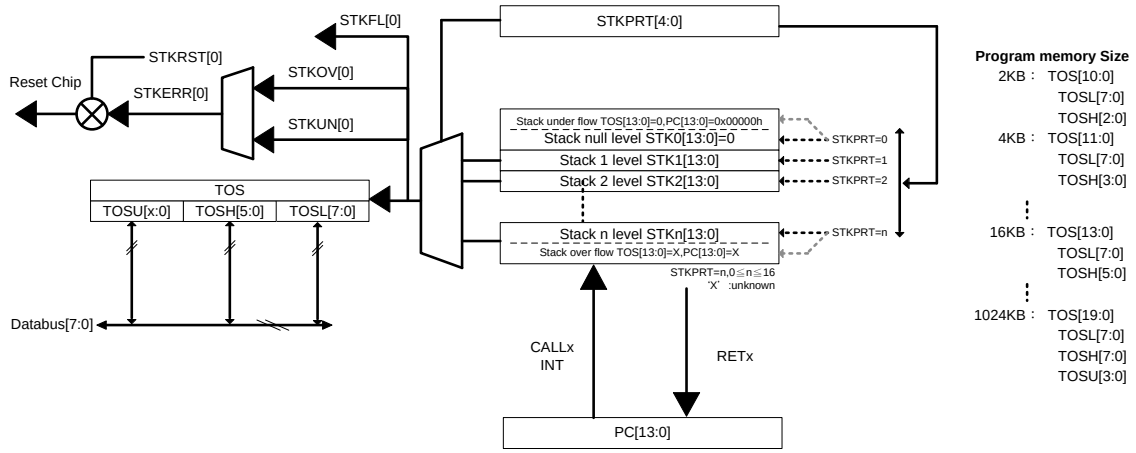


Figure 2-3 Stack architecture diagram

Top-Of-Stack register, TOS[13:0] is 14 bit wide and is constituted by two registers, TOSH[5:0] and TOSL [7:0]. When STKPRT[4:0]=<0> TOS[13:0]=<0> null, program executes CALL instruction or an interrupt (INT) is acknowledged, stack pointer STKPRT[4:0] will add 1 and write the PC address into current register, TOS[13:0]. When program executes instruction RETx, STKPRT[4:0] will subtract 1. Before subtracting 1, TOS[13:0] data will be written to PC[13:0]. After completion, STKPRT[4:0] will subtract 1 and change current TOS[13:0] value.

- There is no special regulation of reading register, TOS[13:0]. It can be read directly.
- CALL instruction or interrupt (INT) can be utilized to write PC[13:0] data to register, TOS[13:0]. POP instruction can discard current TOS[13:0] data and may result in 1 decrement of STKPRT[4:0] and may load in new TOS[13:0] data.

STKFL[0] (Stack full), STKOV[0](Stack overflow) or STKUN[0](Stack underflow) may happen during stack operation processes. Stack full is the early warning flag of stack overflow, executes POP instruction this time can discard current TOS[13:0] data and STKPRT[4:0] may subtract 1 and rewrite the newly appointed stack layer data into TOS[13:0]. Users must be aware that when STKPRT[4:0]=<0>, executes POP instruction may not lead to stack underflow, STKPRT[4:0] data can still be <0>. Therefore, users must determine if it is blank stack.

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

[3] Stack layer register, STKn：Every stack layer has the same length data register as that of top-of-stack register, TOS. When stack index STKPRT being designated, the content of data register will be sent to TOS.

program development, stack reset control bit, SKRST[0]4 can be configured as <1> through software. When stack overflow or stack underflow take place, reset signal will be generated and SKERR[0] will be set as <1> to restart the IC.

- Stack Full: Configure STKFL[0] as <1>, PC[13:0] is not influenced.
- Stack Underflow: Configure STKUN[0] as <1>, PC[13:0] moves to 0x00000h, STKPRT points to 0 Level. If SKRST[0] is set as <1>, reset signal will be aroused after stack underflow and SKERR[0] may be configured as <1>, STKUN[0] will be <0> after reset.
- Stack Overflow: Configure STKOV[0] as <1>, PC[13:0] is not influenced but STKPRT 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[0] is configured as <1>, reset signal may be generated after stack overflow and SKERR[0] may be set as <1>. STKOV[0] will be set as <0> after reset.
- Error: Configure SKERR[0] as <1>, stack error occurred. If SKRST[0] is configured <1>, reset signal will be generated after stack overflow and SKERR[0] will be placed <1>. STKUN[0] and STKOV[0] will be configured as <0> after reset.
- If stack overflow that resulted from ignorance of stack full situation and stack underflow that caused by continuously execute POP instruction happened, STKFL[0], STKOV[0] and STKOV[0] must be configured as <1> in the same time. It is recommended to implement flag clearance action in order to prevent program misjudgment.

To ignore the known stack overflow status when writing program, it is suggested to use POP instruction to erase stack overflow flag, then to continue implementing program. Otherwise, the Interrupt/Call instruction that generated from stack overflow may cape current TOS[13:0] data.

4 SKRST[0] is the generated reset signal control bit of stack error. Instead of direct read or write, it only can be set by developing software at the program development stage. That is to say, whether to generate stack error reset signal must be determined at program developing stage. If reset is chosen, after IC on powered, SKRST [0] is set as 1, the opposite situation is set as 0.

2.2.1.4. Register Description-Program Memory Controller

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
16H	TOSH			TOS[13]	TOS[12]	TOS[11]	TOS[10]	TOS[9]	TOS[8]	..00 0000	..00 0000
17H	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000	0000 0000
18H	STKPTR	STKFL	STKUN	STKOV	STKPRT[4]	STKPRT[3]	STKPRT[2]	STKPRT[1]	STKPRT[0]	0000 0000	0000 0000
1AH	PCLATH			PC[13]	PC[12]	PC[11]	PC[10]	PC[9]	PC[8]	..00 0000	..00 0000
1BH	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
2CH	PSTATUS	PD	TO	IDLEB	BOR		SKERR			000d .0..	uduu .d..

Table 2-1 Program Memory Control Register

TOSU/TOSH/TOSL：Top-OF-Stack Register

TOSH：TOS[13:8]

TOSL：TOS[7:0]

STKPTR：Stack Controller

STKFL：Stack full flag

1：Happened

0：Not happened

STKUN：Stack underflow flag

1：Happened

0：Not happened

STKOV：Stack overflow flag

1：Happened

0：Not happened

STKPRT[4:0]：Stack pointer register

10000：The 16th layer

01111：The 15th layer

⋮

00000：The 0 layer, TOS[13:0]=0x0000h

PCLATU/PCLATH/PCLATL：Program Counter, PC[13:0]

PCLATH：PC[13:8]

PCLATL：PC[7:0]

PSTATUS：Status Register

SKERR：Stack error generated reset flag

1：Happened

0：Not happened

2.2.2. Data Memory, DM

Data Memory comprises Special Register (SR) and General Purpose Register (GPR), every 256byte is a segment. Segment 0 and segment 1 in particular, include 128byte SR and 128byte GPR respectively. Other segments contain 256byte GPR, as illustrated in Figure 2-4.

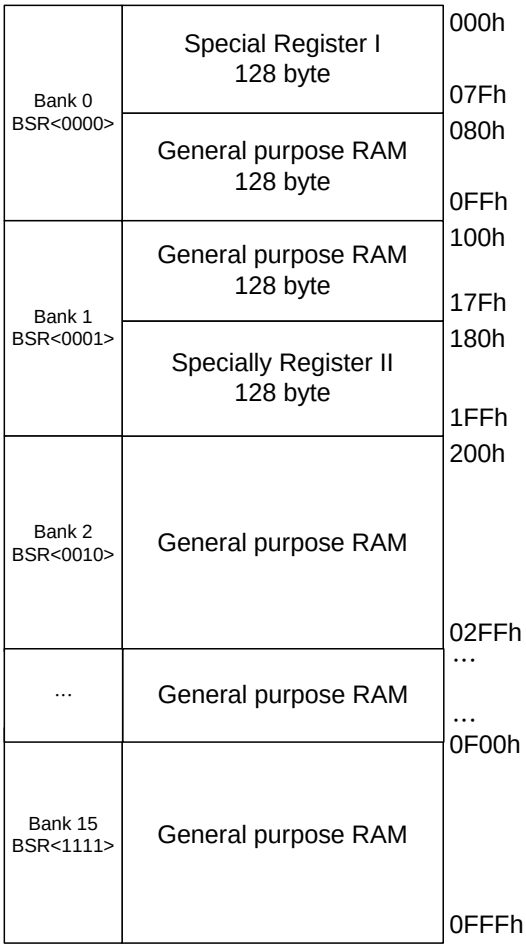


Figure 2-4 Data Memory architecture diagram

2.2.2.1. Memory and Instruction

H08 instruction set can be divided into A and B version. Both are quite different in their memory application, 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

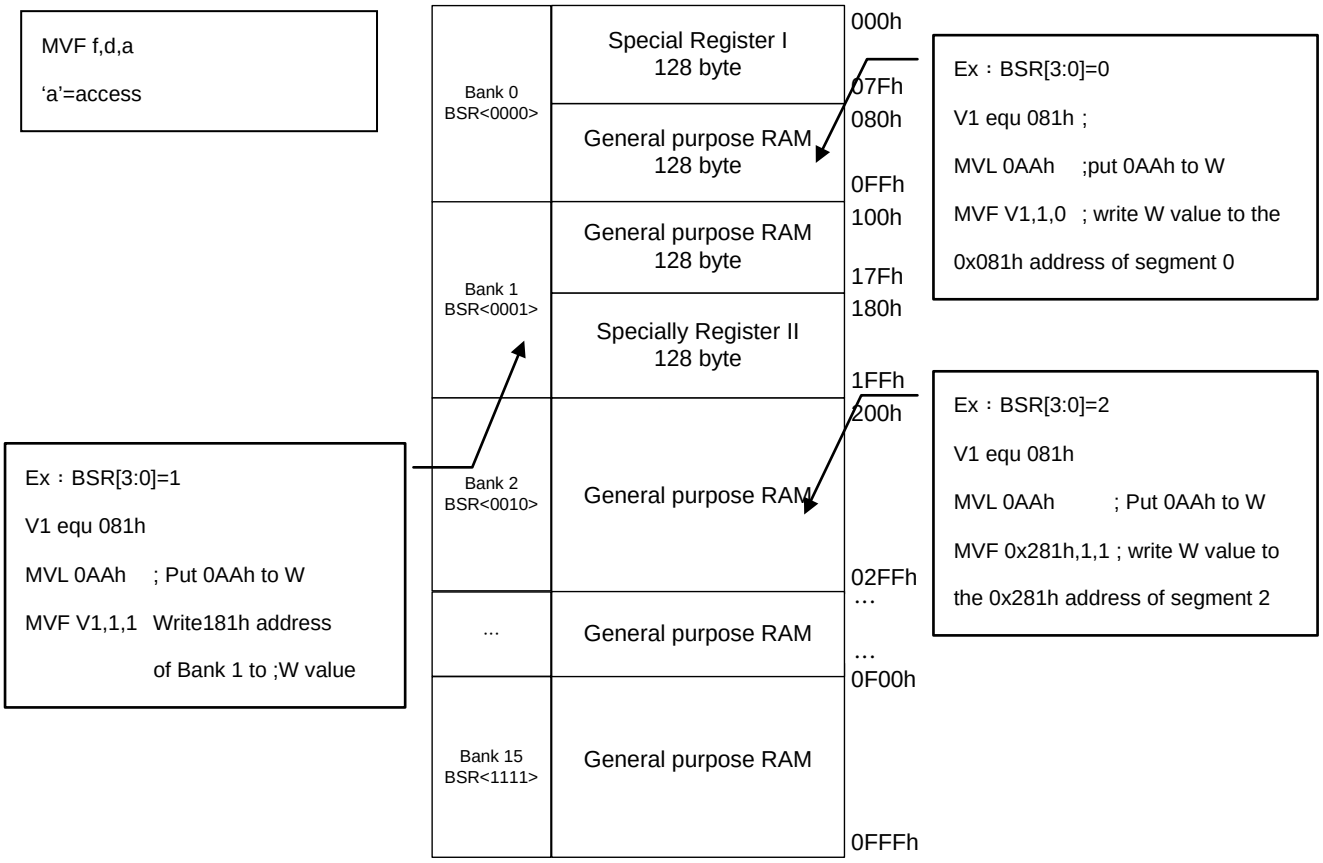
"d" is data storage place after operation. d=0 is saved in WREG register, d=1 is saved in Data Memory Register.

"a" is the designated memory operation segment；a=0 is operated in segment 0, a=1 is operated in designated segment of BSR[3:0].

2.2.2.2. Segment Select Control Register

Every 256 byte of data memory is set to be one segment (000h～0FFh). To read or write address 0FFh register data, it is necessary to set correct segment control register BSR [3:0] and instruction argument "a". Description is as follows:

- When a = 0, no matter register BSR [3:0] appoints to which segment, data memory read/write instruction will only show in segment 0.
- When a = 1, read/write instruction of H08A CPU Core to data memory will be in compliance with the assigned segment of BSR[3:0]; read/write instruction of H08B CPU Core to data memory will be in segment 0.



Example 2-2 Segment selector example program and data memory relationship

2.2.2.3. 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 whilst 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.

2.2.2.3.1. Working Register, WREG

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

2.2.2.3.2. File Select Register, FSR and INDF

File select register, FSR includes instruction register FSR0 [9:0], FSR1 [9:0] and index register, INDF0 [7:0] and INDF1 [7:0]. Because of function similarity, only FSR0 is explained in this chapter.

FSR0[9:0] can be separated into two registers, FSR0H[1:0] and FSR0L[7:0]. There is no need to set up BSR [4:0] to address different segments. Through special instruction, only applying one instruction can write 16-bit data.

INDF0[7:0] is index register that can read FSR0[9:0] appointed address data of data memory.

H08A 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 [9:0] pointed to will be sent back first.
  - Then pointer register, FSR0[9:0] value will add 1 and points to the next instruction.
- PODEC0[7:0]：Events that followed by read/write PODEC0 [7:0] register by instruction.
  - The address value that FSR0 [9:0] pointed to will be sent back first.
  - Then pointer register, FSR0[9:0] 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[9:0] value will add 1 and points to the next address.
  - Current value of FSR0[9:0] 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[9:0] value together with working register, W value.
  - Send back current FSR0[9:0] appointed address value. The register W value will be±128d.

2.2.2.3.3. General Purpose Register, GPR

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

2.2.2.4. Register Description- Data Memory Controller

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
00H	INDF0	Contents of FSR0 to address data memory value of FSR0 not changed								N/A	N/A	*,*,*,*,*,*,*,*
01H	POINC0	Contents of FSR0 to address data memory value of FSR0 post-incremented								N/A	N/A	*,*,*,*,*,*,*,*
02H	PODEC0	Contents of FSR0 to address data memory value of FSR0 post-decremented								N/A	N/A	*,*,*,*,*,*,*,*
03H	PRINC0	Contents of FSR0 to address data memory value of FSR0 pre-incremented								N/A	N/A	*,*,*,*,*,*,*,*
04H	PLUSW0	Contents of FSR0 to address data memory value of FSR0 offset by W								N/A	N/A	*,*,*,*,*,*,*,*
05H	INDF1	Contents of FSR1 to address data memory value of FSR0 not changed								N/A	N/A	*,*,*,*,*,*,*,*
06H	POINC1	Contents of FSR1 to address data memory value of FSR0 post-incremented								N/A	N/A	*,*,*,*,*,*,*,*
07H	PODEC1	Contents of FSR1 to address data memory value of FSR0 post-decremented								N/A	N/A	*,*,*,*,*,*,*,*
08H	PRINC1	Contents of FSR1 to address data memory value of FSR0 pre-incremented								N/A	N/A	*,*,*,*,*,*,*,*
09H	PLUSW1	Contents of FSR1 to address data memory value of FSR0 offset by W								N/A	N/A	*,*,*,*,*,*,*,*
0FH	FSR0H							FSR0[9]	FSR0[8]	xx	.... ..uu	-,-,-,- -,-,*,*
10H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
11H	FSR1H							FSR1[9]	FSR1[8]	xx	.... ..uu	-,-,-,- -,-,*,*
12H	FSR1L	Indirect Data Memory Address Pointer 1 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
29H	WREG	Working Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
2AH	BSRCN						BSR[2]	BSR[1]	BSR[0]	..... .000	.... .000	-,-,-,- -,*,*,*

Table 2-2 Data Memory Control Register

INDF0/POINC0/PODEC0/PRINC0/PLUSW0： Different Functional Index Register

INDF0[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

POINC0[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

PODEC0[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

PRINC0[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

PLUSW0[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

FSR0：File Select Register

FSR0H[1:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

FSR0L[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

FSR1：File Select Register

FSR1H[1:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

FSR1L[7:0]：Please refer to 2.2.2.3.2 File Select Register, FSR and INDF Description.

WREG：Working Register

WREG[7:0]：Please refer to 2.2.2.3.1 Working Register, WREG Description.

BSRCN：Memory Segment Read/Write Control Register

BSR[3:0]：Memory read/write segment pointer register

1111：Segment 15, address 0xF00h~0xFFFh.

1110：Segment 14, address 0xE00h~0xEFFh.

⋮

0001：Segment 1, address 0x100h~0x1FFh.

0000：Segment 0, address 0x000h~0xFFFh.

2.2.3. 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"omly write 1
"."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
00H	INDF0	Contents of FSR0 to address data memory- value of FSR0 not changed								N/A	N/A
01H	POINC0	Contents of FSR0 to address data memory- value of FSR0 post-incremented								N/A	N/A
02H	PODEC0	Contents of FSR0 to address data memory- value of FSR0 post-decremented								N/A	N/A
03H	PRINC0	Contents of FSR0 to address data memory- value of FSR0 pre-incremented								N/A	N/A
04H	PLUSW0	Contents of FSR0 to address data memory- value of FSR0 offset by W								N/A	N/A
05H	INDF1	Contents of FSR1 to address data memory- value of FSR0 not changed								N/A	N/A
06H	POINC1	Contents of FSR1 to address data memory- value of FSR0 post-incremented								N/A	N/A
07H	PODEC1	Contents of FSR1 to address data memory- value of FSR0 post-decremented								N/A	N/A
08H	PRINC1	Contents of FSR1 to address data memory- value of FSR0 pre-incremented								N/A	N/A
09H	PLUSW1	Contents of FSR1 to address data memory- value of FSR0 offset by W								N/A	N/A
0FH	FSR0H							FSR0[9]	FSR0[8]	xx	.... ..uu
10H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu
11H	FSR1H							FSR1[9]	FSR1[8]	xx	.... ..uu
12H	FSR1L	Indirect Data Memory Address Pointer 1 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu
16H	TOSH			TOS[13]	TOS[12]	TOS[11]	TOS[10]	TOS[9]	TOS[8]	..00 0000	..00 0000
17H	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000	0000 0000
18H	STKPTR	STKFL	STKUN	STKOV	STKPRT[4]	STKPRT[3]	STKPRT[2]	STKPRT[1]	STKPRT[0]	0000 0000	0000 0000
1AH	PCLATH			PC[13]	PC[12]	PC[11]	PC[10]	PC[9]	PC[8]	..00 0000	..00 0000
1BH	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
1DH	TBLPTRH			TBLPTR[13]	TBLPTR[12]	TBLPTR[11]	TBLPTR[10]	TBLPTR[9]	TBLPTR[8]	..00 0000	..00 0000
1EH	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000	0000 0000
1FH	TBLDH	Program Memory Table Latch High Byte								0000 0000	0000 0000
20H	TBLDL	Program Memory Table Latch Low Byte								0000 0000	0000 0000
21H	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu
22H	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..
29H	WREG	Working Register								xxxx xxxx	uuuu uuuu
2AH	BSRCN				BSR[4]	BSR[3]	BSR[2]	BSR[1]	BSR[0]	..... .000	..... .000
2BH	STATUS				C	DC	N	OV	Z	...x xxxx	...u uuuu
2CH	PSTATUS	PD	TO	IDLEB	BOR		SKERR			000d .0..	uduu .d..
2DH	LVDCN		LVDFG	LVD	LVDON	VLDX[3:0]				.000 0000	.000 uuuu
2EH	SBMSET1	SKRST		HAOTR[5:0]						x.xx xxxx	u.uu uuuu
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
34H	CPACN1	ENCPA	CPIST	CPIX	CPIH[1:0]		CPIL[2:0]			0000 0000	0000 0000
35H	CPACN2		CPOX	CPOFR	CS1	CPAT	CPVCS[1:0]			.000 000.	.000 000.
36H	CPACN3				CS2	CPVRX[3:0]				...0 0000	...0 0000
37H	OPCN1	ENOP	OPM[1:0]		OPP[1:0]		OPN[2:0]			0000 0000	0000 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
41H	TMACN	ENTMA	TMACK	TMAS[1:0]		ENWDT	WDTS[2:0]			0000 0000	0000 0000
42H	TMAR	TimerA data register								xxxx xxxx	uuuu uuuu
43H	TMBCN	ENTMB	TMBCK	TMBS[1:0]		TMBSYC	TMBR2R			0000 00..	0000 00..
44H	TMBRH	TimerB High Byte data register								xxxx xxxx	uuuu uuuu
45H	TMBRL	TimerB Low Byte data register								xxxx xxxx	uuuu uuuu
46H	TMCCN	ENTMC	TMCCK[1:0]		TMCS1[2:0]			TMCS0[1:0]		0000 0000	0000 0000
47H	PRC	TimerC programmable register								1111 1111	1111 1111
48H	TMCR	TimerC register								0000 0000	0000 0000
49H	CCPCN	CCP1M[3:0]				CCP0M[3:0]				0000 0000	0000 0000
4AH	CCP0RH	CCP0 High Byte data register								xxxx xxxx	uuuu uuuu
4BH	CCP0RL	CCP0 Low Byte data register								xxxx xxxx	uuuu uuuu
4CH	CCP1RH	CCP1 High Byte data register								xxxx xxxx	uuuu uuuu
4DH	CCP1RL	CCP1 Low Byte data register								xxxx xxxx	uuuu uuuu
4EH	PASC	PASF		PASCF[1:0]		PSSCN0[1:0]		PSSCN1[1:0]		0.00 0000	0.00 0000
4FH	PWMCN	ENPWM	ENPFD	PWMRL[1:0]		PWMCG[1:0]		PWMM[1:0]		0000 0000	0000 0000
50H	PDBD	ENPRS	DBDC[6:0]							0000 0000	0000 0000
51H	PWMR	PWM MSB Byte register								xxxx xxxx	uuuu uuuu

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"omly write 1											
"."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**
52H	LCDCN1	ENLCD	LCDPR	VLCDX[1:0]		LCDBF	LCDBI[1:0]			0000 000.	0000 000.
53H	LCDCN2	LCDBL	LCDMX[1:0]							000.	000. ....
54H	LCD0	Segment SEG2@[3:0] and SEG3@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
55H	LCD1	Segment SEG4@[3:0] and SEG5@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
56H	LCD2	Segment SEG6@[3:0] and SEG7@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
57H	LCD3	Segment SEG8@[3:0] and SEG9@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
58H	LCD4	Segment SEG10@[3:0] and SEG11@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
59H	LCD5	Segment SEG12@[3:0] and SEG13@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
5AH	LCD6	Segment SEG14@[3:0] and SEG15@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
5BH	LCD7	Segment SEG16@[3:0] and SEG17@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
5CH	LCD8	Segment SEG18@[3:0] and SEG19@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
5DH	LCD9	Segment SEG20@[3:0] and SEG21@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
5EH	SSPCON1	SSPEN	CKP	CKE	SMP			SSPM<1:0>		0000 ..00	uuuu ..uu
60H	SSPSTA	SSPBUY	SSPOV						BF	00.. ...0	00.. ...0
61H	SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
63H	URCON	ENSP	ENTX	TX9	TX9D	PARITY			WUE	0000 0..0	0000 0..0
64H	URSTA		RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0110	.000 0110
65H	BAUDCON					ENCR	RC9	ENADD	ENABD	 0000	 0000
66H	BRGRH				Baud Rate Generator Register High Byte					...x xxxx	...u uuuu
67H	BRGRL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu
68H	TXREG	UART Transmit Register								xxxx xxxx	uuuu uuuu
69H	RCREG	UART Receive Register								xxxx xxxx	uuuu uuuu
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
6DH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu
6EH	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4					0000	0000 ....
6FH	PT1DA						DA1.2	DA1.1	DA1.0	.... .000	.... .000
70H	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	0000 0000
71H	PT1M1					INTEG1[1:0]		INTEG0[1:0]		 0000	 0000
72H	PT1M2		PM1.7[0]		PM1.6[0]		PM1.5[0]		PM1.4[0]	.0.0 .0.0	.0.0 .0.0
73H	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2			0000 00..	0000 00..
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000
76H	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000
78H	PT2M1		PM2.3[0]	PM2.2[1]	PM2.2[0]					.000	.000 ....
79H	PT2M2		PM2.7[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	.0.0 0000	.0.0 0000
7AH	PT3	PT3.7	PT3.6	PT3.5	PT3.4	PT3.3	PT3.2	PT3.1	PT3.0	xxxx xxxx	uuuu uuuu
7BH	TRISC3	TC3.7	TC3.6	TC3.5	TC3.4	TC3.3	TC3.2	TC3.1	TC3.0	0000 0000	0000 0000
7DH	PT3PU	PU3.7	PU3.6	PU3.5	PU3.4	PU3.3	PU3.2	PU3.1	PU3.0	0000 0000	0000 0000
80H ~ FFH	GPR0	General Purpose Register as 128Byte								xxxx xxxx	uuuu uuuu
100H~17FH	GPR1	General Purpose Register as 128Byte								xxxx xxxx	uuuu uuuu
180H	LCD10	Segment SEG22@[3:0] and SEG23@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
181H	LCD11	Segment SEG24@[3:0] and SEG25@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
182H	LCD12	Segment SEG26@[3:0] and SEG27@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
183H	LCD13	Segment SEG28@[3:0] and SEG29@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
184H	LCD14	Segment SEG30@[3:0] and SEG31@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
185H	LCD15	Segment SEG32@[3:0] and SEG33@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
186H	LCD16	Segment SEG34@[3:0] and SEG35@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
187H	LCD17	Segment SEG36@[3:0] and SEG37@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
188H	LCD18	Segment SEG38@[3:0] and SEG39@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
189H	LCD19	Segment SEG40@[3:0] and SEG41@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA					DA5.3	DA5.2	DA5.1	DA5.0	 1111	 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000
1FDH	WREGSDW	shadow of WREG,reserve for ICE only									
1FEH	BSRSDW	shadow of BSR,reserve for ICE only									
1FFH	STASDW	shadow of STATUS,reserve for ICE only									
200H ~ 2FFH	GPR2	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu
300H ~ 3FFH	GPR3	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu
400H ~ 4FFH	GPR4	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu

Table 2-4 Data Memory List (continued)

3. Oscillator, Clock Sources and Power Managed Modes

HY11P Series has three clock sources, HAO, LPO and XT as shown in Table 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：

MCKCN1 ADCS[2:0],ADCCK[0],XTHSP[0],XTSP[0],ENXT[0],ENHAO[0]

MCKCN2 LSCK[0],HSCK[0],HSS[1:0],CPUCK[1:0]

MCKCN3 LCDS[2:0],PERCK[0],BZS[2:0]

Oscillator			Oscillation Way
Symbol	Attributes	Frequency	
HAO	Internal	2MHz	RC
LPO		28KHz	RC
XTL/S/H	External	32768Hz ~ 8MHz	crystal/ RC

Table 3-5 IC Clock Sources

3.1. Oscillator

3.1.1. XT External Crystal/ Resonator Oscillator

Clock sources control register, MCKCN1[7:0] must be set up in accordance with external operating frequency of oscillator, as Table 3-2 presents. Figure 3-1 depicts the block diagram for HY11P Series external oscillator.

Using external oscillator, I/O must be set in input mode and cannot use internal pull high resistance. MCKCN1 register control bit have to be configured in compliance with its oscillation frequency in order to determine external crystal/resonator operating frequency.

As Figure 3-1 illustrates, C1, C2 and R1 value changes according to different external crystal oscillator or resonator frequency. What is more, even with the same oscillator, capacitance value will have small discrepancy resulted from different PCB layout. Table 3-2 lists the value for design guidance, not optimized value.

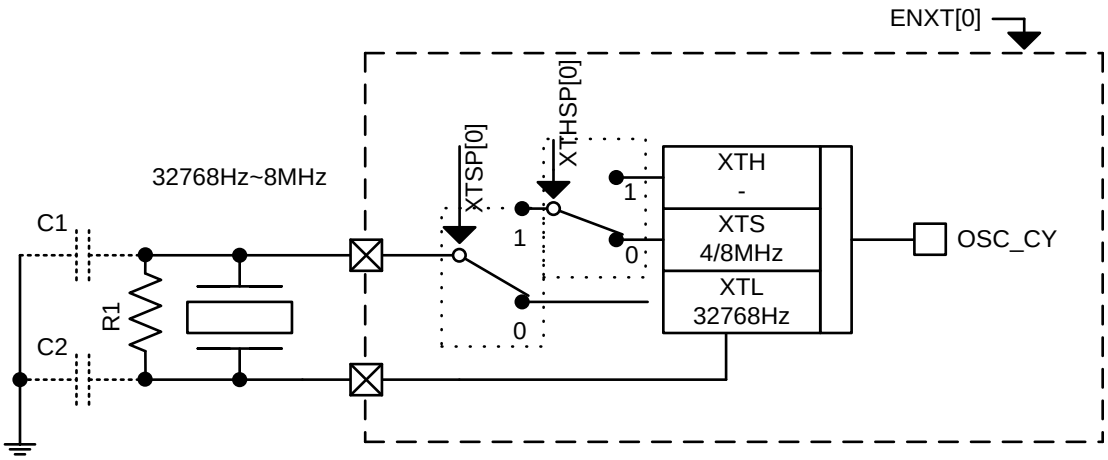


Figure 3-5 Block Diagram of External Oscillator

External oscillator is sorted into three oscillation frequency level, each level has diverse driving current and thus, different configuration must be made on the apparatus, as Table 3-2 exhibits. When external oscillator is applied, ENXT [0] must be set as 1. If the external oscillator frequency is higher than 4MHz, XTSP [0] must be set up as 1 as well. On the contrary, if no changes have been made to XTSP [0] based on different external oscillation frequencies; external oscillator may not be started-up and may cause huge power consumption.

Besides, high speed oscillator controller, XTHSP[0] is inactive, please set XTHSP[0]as <0>. Set XTHSP[0] as <1> will increase power consumption.

Under the configuration of connecting external oscillator, 32768Hz and R1=10MΩ, the ideal oscillation time of oscillator is 1.3 second. If external oscillator 8MHz is connected and R1=1MΩ, the ideal oscillation time of oscillator is 30m second.

"x"：ignore, resistor unit:Ω,capacitor unit:F

Symbol	Frequency	Ceramic Resonator			Crystal Oscillator			MCKCN1 Configuration		
		C1	C2	R1	C1	C2	R1	ENXT	XTSP	XTHSP
XTL	455Hz							1	0	0
	32768Hz			10M	20p	20p	10M	1	0	0
XTS	4.0MHz			1M	20p	20p	1M	1	1	0
XTH	8.0MHz			1M	20p	20p	1M	1	1	0

Table 3-6 Oscillator Optimized Capacitor Value and MCKCN1 Register Set Up

3.1.2. HAO Oscillator

HAO is internal high speed RC oscillator; the typical output frequency is 2.0MHz. Compared to external XT oscillator, internal HAO has the characteristics of fast start-up and better anti-jamming, thus, HAO is called restated CPU operation clock source.

When CPU of HY11P Series utilizes other oscillator as operating clock source, HAO oscillator can be turned off by setting ENHAO[0] as <0>.

3.1.3. LPO Oscillator

LPO is internal low speed RC oscillator, the typical output frequency is 28KHz. LPO consumes 0.7uA current; therefore, 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 HY11P 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

Three sets of oscillator output（OSC_XT、OSC_HAO、OSC_LPO）will firstly pass through pre-operating clock divider to initiate start/stop, switch and prescaling, then to CPU and other peripheral configurations of the IC. Pre-operating clock divider can produce four different kinds of clock source frequency. As Figure 3-2 indicates, based on their interdependence and frequency speed, the permutation is shown as:

HS_CK≧HSS_CK≧HS_DCK or LS_CK.

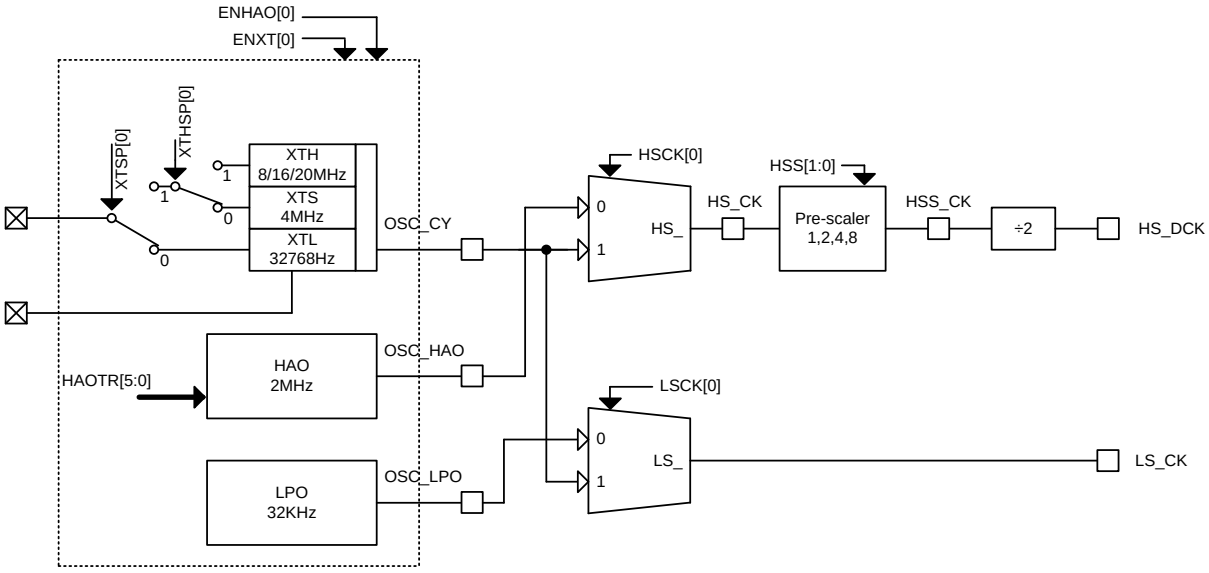


Figure 3-6 Pre-Operating Clock Divider

Example program：

Setting up the internal secondary oscillator HAO as primary clock source, HS_DCK outputs 500 KHz frequency. Setting LS_CK as internal LPO outputs 28 KHz frequency.

```
MVL     01h             ；Starting HAO
MVF     MCKCN1,1,0      ；
MVL     00000100b       ；Set up HS_CK clock source as OSC_HAO
MVF     MCKCN2,1,0      ；Set up LS_CK clock source as OSC_LPO
                          First, HSS[1:0] prescale 2
                          Then, internal prescale 2, can get HS_DCK= 500 KHz
```

Example 3-3 HS_DCK Outputs 500KHz Example Program

Example program：

Configure external 8MHz oscillator as the primary clock source. HS_DCK outputs 4MHz frequency and LS_CK is configured as internal LPO output frequency 28 KHz.

```
CLRF    TRISC2,0        ; Set up PT2.0, PT2.1 for external oscillator input signal
CLRF    PT2PU,0
MVL     047h            ；Set up external 8MHz oscillation, ADC_CK=250KHz
MVF     MCKCN1,1,0      ；
CALL    DELAY           ；DELAY LOOP is subroutine of delay time
                        ；Reserve 30msec delay time for oscillation
MVL     00010001b       ；Set up CPU_CK as HS_DCK and switches to external oscillator
MVF     MCKCN2,1,0      ；HS_DCK clock source is OSC_CY
                            ；LS_CK clock source is OSC_LPO
                            ；Instruction cycle is INTR_CK=8M/2/4=1MHz
MVL     046h            ；Turn off internal OSC_HAO frequency source for power-saving
MVF     MCKCN1,1,0      ；
NOP
```

Example 3-2 HS_DCK Outputs 4MHz (External Oscillator) Example Program

```
Example program：
Configure external 32768Hz oscillator as the primary clock source.
CLRF    TRISC2,0        ; Set up  PT2.0, PT2.1for external oscillator input signal
CLRF    PT2PU,0
MVL     003h            ; Set up external  32768Hz oscillation
MVF     MCKCN1,1,0      ;
CALL    DELAY           ; DELAY LOOP is subroutine of delay time
                        ; Reserve  1sec delay time for oscillation
MVL     00100010b       ; Set up  CPU_CK as  LS_CK and switches to external oscillator
MVF     MCKCN2,1,0      ; LS_CK clock source is OSC_CY= external  32768Hz
                        ; Instruction cycle  INTR_CK=32768Hz/4=8192Hz
MVL     002h            ; Turn off internal  OSC_HAO frequency source for power-saving
MVF     MCKCN1,1,0      ;
NOP
```

Example 3-3 LS_CK Outputs 32768Hz (External Oscillator) Example Program

3.2.2. CPU Clock Source

There are four clock sources of IC CPU core operating frequency namely, HS_CK, HSS _CK, HS_DCK and LS_CK.

- When using high performance SD18, it is suggested to use HS_DCK as CPU operating clock source so as to reduce interference caused by digital circuit to ADC.
- Instruction execution cycle is CPU_CK/4 and PERA_CK is the main clock source of peripherals, as Figure 3-3 exhibits.

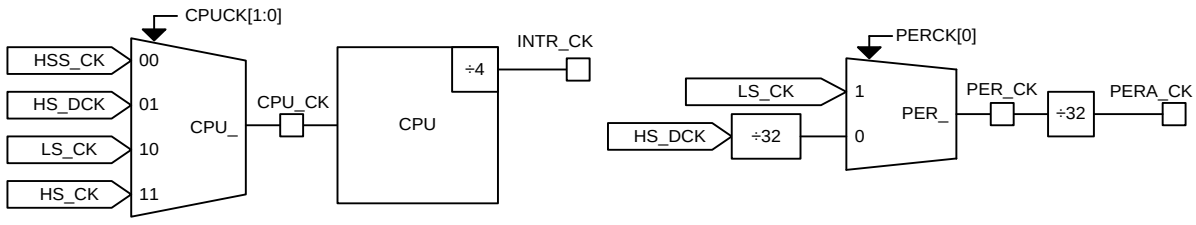


Figure 3-7 CPU and Peripheral Operating Clock Sources

Operation Clock Source HAO, LPO		CPU Operation Frequency CPU_CK	Instruction Operation Cycle INST_CK
HSS_CK	2MHz	2MHz	2us
HS_DCK	1MHz	1MHz	4us
LS_CK	28KHz	28KHz	142us
HS_CK	2MHz	2MHz	2us

Table 3-7 CPU Operating Frequency and Instruction Execution Cycle

3.2.3. CPU Peripheral Circuit Operating Clock Source

Clock source of HY11P Series peripheral circuit are configured by different distribution controller and prescaler. The configuration will be fully illustrated in each peripheral unit; hence Figure 3-4 only presents the peripheral clock sources configuration.

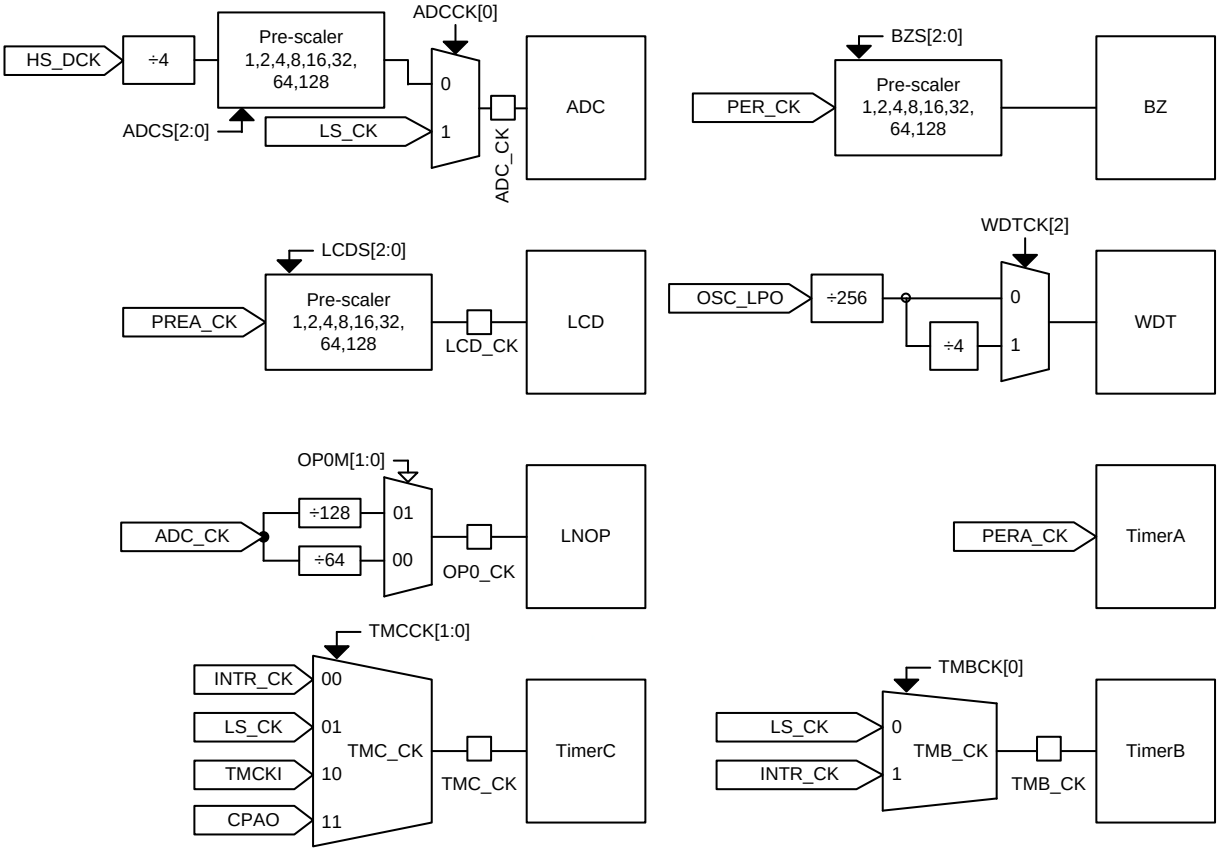


Figure 3-8 Configuration of Peripheral Operating Clock Source

3.3. 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"omly write 1												
"."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
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001	*,*,*,*,*,*,*,*
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000	-,-,*,*,*,*,*,*
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000	*,*,*,-,*,*,*,*
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2							TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 3-8 Operating Clock Source Control Register

MCKCN1：Operating Clock Source Controller 1

ADCS[2:0]：Prescaler of SD18 peripheral operating frequency

111：ADC_CK/128

110：ADC_CK/64

101：ADC_CK/32

100：ADC_CK/16

011：ADC_CK/8

010：ADC_CK/4

001：ADC_CK/2

000：ADC_CK/1

ADCCK：Selector of SD18 peripheral operating clock source

1：LS_CK

0：HS_DCK

XTHSP：Control bit of external oscillator frequency selection

1：Reserved；Inactive (Configure <1> will result in bigger IC operating current).

0：XTS mode

XTSP：Control bit of external oscillator frequency selection

1：XTS mode

0：XTL mode

ENXT：Control bit of starting external crystal/resonator

1：Active

0：Inactive

ENHAO：Control bit of starting internal HAO (2MHz)

1：Active

0：Inactive

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001	*,*,*,*,*,*,*,*
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000	-,-,*,*,*,*,*,*
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000	*,*,*,-,*,*,*,*
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2							TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*

MCKCN2：Operating Clock Source Controller 2

LSCK：Control bit of low speed clock source selector

When ENXT = 1

1：OSC_CY

0：OSC_LPO

When ENXT = 0

1：Cannot be configured

0：OSC_LPO

HSCK：Control bit of high speed clock source selector

When ENXT = 1

1：OSC_CY

0：OSC_HAO

When ENXT = 0

1：Cannot be configured

0：OSC_HAO

HSS[1:0]：Prescaler of high speed clock source

11：HS_CK/8

10：HS_CK/4

01：HS_CK/2

00：HS_CK/1

CPUCK[1:0]：Control bit of CPU operating clock selection

11：HS_CK

10：LS_CK

01：HS_DCK

00：HSS_CK

Notice:

If any re-write or read action took place in register" MCKCN2(032h)" after ADC started, it must be switched off and enable again to prevent AD counts drifts.

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001	*,*,*,*,*,*,*,*
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000	-,-,*,*,*,*,*,*
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000	*,*,*,-,*,*,*,*
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2							TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*

MCKCN3：Operating Clock Source Controller 3

LCDS[2:0]：Prescaler of LCD peripheral operating frequency

111：PERA_CK/128

110：PERA_CK/64

101：PERA_CK/32

100：PERA_CK/16

011：PERA_CK/8

010：PERA_CK/4

001：PERA_CK/2

000：PERA_CK/1

PERCK：Control bit for peripheral clock source selection

1：HS_DCK/32。

0：LS_CK。

BZS[2:0]：Prescaler of BZ peripheral operating frequency

111：PER_CK/128。

110：PER_CK/64。

101：PER_CK/32。

100：PER_CK/16。

011：PER_CK/8。

010：PER_CK/4。

001：PER_CK/2。

000：PER_CK/1。

3.4. Power Managed and Operation Mode

HY11P Series CPU provides three operating modes for users to acquire the best managed circumstance between operating efficiency and power economy. These modes are run mode, idle mode and sleep mode.

3.4.1. Run Mode

Run mode means that CPU dealt with every occurred event in accordance with the clock source. At this time, all peripherals operate normally and consume maximum power within the same clock source.

3.4.2. Idle Mode

The chip enters Idle Mode through IDLE instruction. IDLE Mode means that CPU stops running to wait for awakening as long as it accesses into energy efficient status. The IDLEB [0] flag of PSTATUS reset register is set as 1. At this time, peripherals still operate normally. When interrupt event occurs, CPU[5] will then be awakened. Moreover, under this mode, signals produced when WDT counter ceased, are part of interrupt signals instead of reset signals. Example program can be referred to Example 3-4.

CPU pauses under IDLE instruction in IDLE Mode. Its internal oscillator is not affected by this instruction and has not been turned off. If users would like to achieve more energy efficient status, it is recommended to turn off relative resources, for example, peripherals or oscillators according to real application condition. The chip will leave IDLE Mode by external interrupt sources or other peripheral interrupt signals.

If the chip encountered with interrupt source and leave the IDLE Mode, it will take 2 instruction cycles time to retrieve the location of interrupt vector, 04H. If the ideal CPU frequency source is internal 2MHz, then 1 instruction cycle time is 2usec. Thus, it will take 4usec for awakening, then the program will retrieve back to interrupt vector location. If the ideal CPU frequency source is internal 28KHz, the 1 instruction cycle time is 143usec. It will cost 286usec for awakening, then the program will retrieve back to interrupt vector location.

Under IDLE Mode, if CPU frequency source is internal 28KHz and internal 2MHz oscillator has been turned off, after starting 2MHz oscillator, it will take 2 28KHz instruction cycle awakening time and 128 2MHz instruction cycle oscillation time (approx. 542usec) to fully start internal 2MHz oscillator.

3.4.3. Sleep Mode

The chip enters Sleep Mode through SLP instruction. The chip stops operation in Sleep Mode. CPU, internal oscillator (HAO & LPO) and external oscillator (XT) also ceases operation. Peripherals that use HAO, LPO, and XT…related clock sources will stop operation and PD flag bit of PSTATUS reset register will be configured as <1>. Under this

[5] When CPU was awoken by interrupt signals, PC will switch to interrupt vector address (0x004h). For detail PSTATUS reset register and interrupt service vector description, please refer to *Reset, Interrupt* chapter.

mode, the chip will be awakened by interrupt event for partial peripherals still retaining in open status despite stop operation. In order to maintain the lowest power consumption under Sleep Mode, peripherals that do not related to awaking the chip must be further turned off. Example program can be founded in Example 3-5.

If CPU frequency source is from HAO (2MHz) before entering into Sleep Mode, restarting HAO frequency source will take 256 instruction time (approx. 512usec to restart HAO) when encountered interrupt source and left Sleep Mode. If CPU frequency source is from LPO (28KHz) before entering into Sleep Mode, restarting LPO frequency source will take 512 instruction time (approx. 73.2msec to restart LPO) when encountered interrupt source and left Sleep Mode.

```
    MVL     00000010B
    MVF     MCKCN2, F, ACCE         ; Setup internal LPO as CPU clock
    MVL     00000000B
    MVF     MCKCN1, F, ACCE         ; Turn off all external clock sources and internal HAO
    MVL     00000000B
    MVF     MCKCN3, F, ACCE
    CLRF    TMACN                   ; Turn off every clock counter module, or turn on
    CLRF    TMBCN                   ; This 3 modules and SDT based on different functional requirements
    CLRF    TMCCN                   ;
    CLRF    ADCCN1                  ; Turn off ADC function module
    CLRF    ADCCN2
    CLRF    ADCCN3
    CLRF    AINET1
    CLRF    AINET2
    CLRF    PWRCN                   ; Turn off VDDA, turn off ACM
    MVL     00000000B
    MVF     PT1DA, F, ACCE          ;
    MVL     11110000B
    MVF     TRISC1, F, ACCE         ; Set PT1.0 as external interrupt awaken input
    MVL     00001111B
    MVF     PT1PU, F, ACCE
    MVL     00000000B
    MVF     PT1, F, ACCE
    CLRF    PT1M1
        ......
        ......                      ; The function can be the same in respect of TMA, TMB, and TMC
        ......          .           ; PT1 and PT2 can be configured in accordance with the example program of TMA, TMB, and TMC
        ......                      ;
CLRF    INTF1,ACCE
MVL     10000001B
    MVF     INTE1,F,ACCE            ; Set external interrupt 1 to enable
    IDLE
    NOP
```

Example 3-4 Idle Mode Example Program

```
MVL     00000010B
MVF     MCKCN2, F, ACCE      ; Set internal LPO as CPU clock
MVL     00000000B
MVF     MCKCN1, F, ACCE      ; Turn off all external clock sources and internal HAO
MVL     00000000B
MVF     MCKCN3, F, ACCE
CLRF    LCDCN1               ; Turn off LCD drive
CLRF    TMACN                ; Turn off TIMERA function module
CLRF    TMBCN                ; Turn off TIMERB function module
CLRF    TMCCN                ; Turn off TMERC function module
CLRF    ADCCN1               ; Turn off ADC function module
CLRF    ADCCN2
CLRF    ADCCN3
CLRF    AINET1
CLRF    AINET2
CLRF    PWRCN                ; Turn off VDDA and ACM
MVL     00000000B
MVF     PT1DA, F, ACCE       ;
MVL     11110000B
MVF     TRISC1, F, ACCE      ; Set PT1.0 as external interrupt awakening input
MVL     00001111B
MVF     PT1PU, F, ACCE
MVL     00000000B
MVF     PT1, F, ACCE
CLRF    PT1M1
CLRF    PT2PU
MVL     11111111B
MVF     TRISC2, F, ACCE
CLRF    PT2
CLRF    PT2M1
CLRF    PT2M2
CLRF    INTF1,ACCE
MVL     10000001B
MVF     INTE1,F,ACCE         ; Set external interrupt 1 to enable
SLP
NOP
```

Example 3-5 Sleep Mode Example Program

4. RESET

HY11P Series reset circuit includes the following four events to trigger reset signal. Reset block diagram is as Figure 4-1.

- **BOR** Power interference reset.
- **RST** External reset input pin.
- **WDT** watch dog timer reset.
- **SKERR** Stack error reset (determined by users)

Operating Status Register：

PSTATUS PD[0],TO[0],IDLEB[0],BOR[0],LVD[0],SKERR[0]

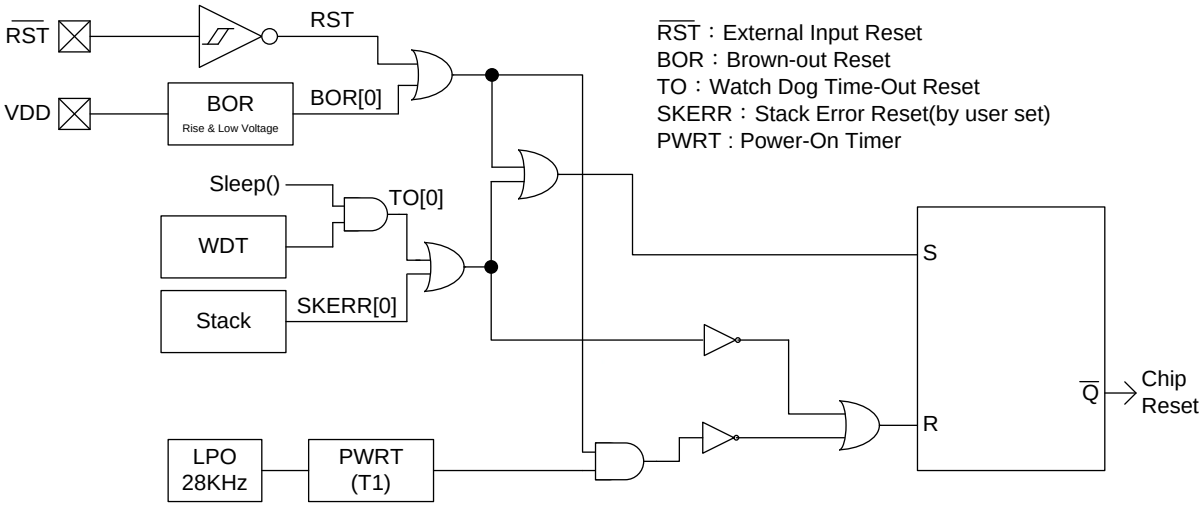


Figure 4-9 Reset Block Diagram

These reset events can be divided into hardware reset and software reset as depicted in 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 4-9Reset 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.

When BOR reset occurred, BOR[0] flag of register PSTATUS[7:0] will be configured as <1> to record the occurred event.

BOR circuit of HY11P Series will generate approximately 0.6uA current consumption; there is no other program or configuration methods can shut it off.

4.1.2. RST External Reset Input

Reset signal will be produced when external RST pin voltage is lower than default value6, at this time, the IC will enter into restart status. The IC will return to normal operating mode till RST voltage regain to default arguments.

4.1.3. Watch Dog Timer Reset, WDT

The IC will enter into quick restart status when reset signal is produced after WDT operating mode ceased. As soon as WDT reset happens, TO [0] flag of register, PSTATUS [7:0] is set as 1 in order to record the occurred event.

Please be noticed that there are two kinds of signals which happened after WDT ceased. Reset signal is produced when the IC is under RUN Mode. If it is in IDLE Mode, CPU will be awakened by the signals generated by interrupt events. Detailed instruction description can be referred to in Watch Dog Timer, WDT chapter.

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 [0] flag of register, PSTATUS[7:0] is set as 1 while SKERR stack error reset happened. Detailed instruction description will be elaborated on **Memory** chapter.

6 The pin includes two other different functions. One is that when RST input voltage conforms to V_{IU} specification, the IC will get into OTP programming mode, the other one is that when RST input voltage matches to V_{IL} specification, the IC will enter into current leakage detect mode.

4.2. Status Registers

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

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

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

Table 4-10 Interrelation of Reset Status Flags

4.2.1. Reset Status Sequence

Figure 4-2 presents the time frame from hardware reset signal happened to IC accesses into operating status. As for the time frame from different reset signals occurred to IC enters into operating status, please refer to Figure 3-2(b).

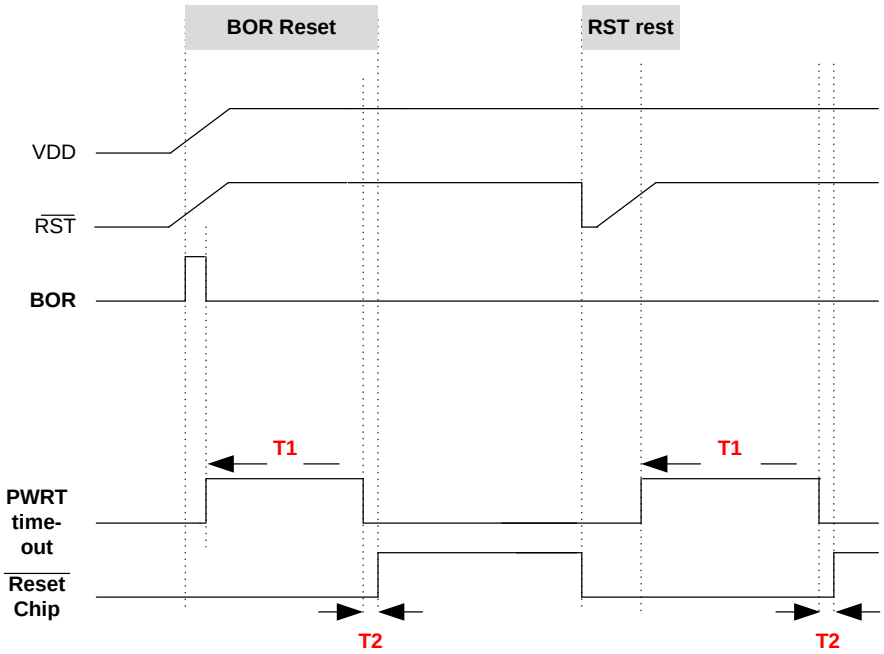


Figure 4-10 Reset Status Sequence

"-"：no definition

Reset Signal	Delay Time			Operating Status		
	Symbol	T17	T28	Run	Idle	Sleep
BOR	t_{RST}	T1＋T2		Valid	Valid	Valid
RST		T1＋T2		Valid	Valid	Valid
WDT9	-	-		Valid	Valid	Invalid
SKERR	-	-		Valid	Invalid	Invalid

Table 4-11 Interrelation of Reset Status Delay Time and Operating Status

[7] T1：2048 oscillation delay cycle，using LPO(28KHz) clock source.

[8] T2：1024 oscillation delay cycle, using HAO(2MHz) clock source.

[9] Two signals will be generated after WDT ceased, please refer to ***Watch Dog, WDT*** chapter for detailed instruction description.

4.2.2. 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"omly write 1											
"."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**
2CH	PSTATUS	PD	TO	IDLEB	BOR		SKERR			000d .0..	uduu .d..

Table 4-12 Reset Register

PSTATUS：Status Register

PD：Sleep Mode flag

1：Set up <1> to execute SLEEP instruction

0：Cleared through BOR, RST or instruction

TO：WDT flag

1：Set up <1>, when WDT ceased

0：Cleared through BOR, RST or instruction

IDLEB：Idle Mode flag

1：Set up <1> to execute IDLE instruction

0：Cleared through BOR, RST or instruction

BOR：Power interference reset flag,

1：Set up <1> to operate BOR.

0：Cleared through instruction

SKERR：Stack error reset flag

1：Stack error set up <1>.

0：Cleared through BOR, RST or instruction

4.3. Register List-Data Memory Reset Status

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1
"."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
00H	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								N/A	N/A	********,********
01H	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								N/A	N/A	********,********
02H	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								N/A	N/A	********,********
03H	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								N/A	N/A	********,********
04H	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								N/A	N/A	********,********
05H	INDF1	Contents of FSR1 to address data memoryvalue of FSR0 not changed								N/A	N/A	********,********
06H	POINC1	Contents of FSR1 to address data memoryvalue of FSR0 post-incremented								N/A	N/A	********,********
07H	PODEC1	Contents of FSR1 to address data memoryvalue of FSR0 post-decremented								N/A	N/A	********,********
08H	PRINC1	Contents of FSR1 to address data memoryvalue of FSR0 pre-incremented								N/A	N/A	********,********
09H	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR0 offset by W								N/A	N/A	********,********
0FH	FSR0H							FSR0[9]	FSR0[8]	xx	.... ..uu	-,-,-,- -,-,*,*
10H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	********,********
11H	FSR1H							FSR1[9]	FSR1[8]	xx	.... ..uu	-,-,-,- -,-,*,*
12H	FSR1L	Indirect Data Memory Address Pointer 1 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	********,********
16H	TOSH			TOS[13]	TOS[12]	TOS[11]	TOS[10]	TOS[9]	TOS[8]	..00 0000	..00 0000	-,-,******,******
17H	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000	0000 0000	********,********
18H	STKPTR	STKFL	STKUN	STKOV	STKPRT[4]	STKPRT[3]	STKPRT[2]	STKPRT[1]	STKPRT[0]	0000 0000	0000 0000	r,rw 0,rw 0,r r,r,r,r
1AH	PCLATH			PC[13]	PC[12]	PC[11]	PC[10]	PC[9]	PC[8]	..00 0000	..00 0000	-,-,******,******
1BH	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	********,********
1DH	TBLPTRH			TBLPTR[13]	TBLPTR[12]	TBLPTR[11]	TBLPTR[10]	TBLPTR[9]	TBLPTR[8]	..00 0000	..00 0000	-,-,******,******
1EH	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000	0000 0000	********,********
1FH	TBLDH	Program Memory Table Latch High Byte								0000 0000	0000 0000	********,********
20H	TBLDL	Program Memory Table Latch Low Byte								0000 0000	0000 0000	********,********
21H	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
22H	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	********,********
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	**,-,- ****,****
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	******,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,******,*******
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	**,-,- ****,****
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	******,-,-,-
29H	WREG	Working Register								xxxx xxxx	uuuu uuuu	********,********
2AH	BSRCN				BSR[4]	BSR[3]	BSR[2]	BSR[1]	BSR[0]	..... .000	.... .000	-,-,-,- -,-,*,*
2BH	STATUS				C	DC	N	OV	Z	...x xxxx	...u uuuu	-,-,-,******,*****
2CH	PSTATUS	PD	TO	IDLEB	BOR		SKERR			000d .0..	uduu .d..	rw 0,rw 0,rw 0,rw 0 -,rw 0,-,-
2DH	LVDCN		LVDFG	LVD	LVDON	VLDX[3:0]				.000 0000	.000 uuuu	-,*,r,r ****,****
2EH	SBMSET1	SKRST	HAOTR[5:0]							x.xx xxxx	u.uu uuuu	*,-,******
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000	****,-,-,-,-
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001	********,********
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000	-,-,******,******
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000	***,- ****,****
34H	CPACN1	ENCPA	CPIST	CPIX	CPIH[1:0]		CPIL[2:0]			0000 0000	0000 0000	********,********
35H	CPACN2		CPOX	CPOFR	CS1	CPAT	CPVCS[1:0]			.000 000.	.000 000.	-,***,r,***
36H	CPACN3				CS2	CPVRX[3:0]				...0 0000	...0 0000	-,-,-,*******
37H	OPCN1	ENOP	OPM[1:0]		OPP[1:0]		OPN[2:0]			0000 0000	0000 0000	********,********
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000	********,********
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000	********,********
3EH	ADCCN3	OSR[2:0]								000.	000.	***,- -,-,-,-
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000	********,********
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.	-,******,-
41H	TMACN	ENTMA	TMACK	TMAS[1:0]		ENWDT	WDTS[2:0]			0000 0000	0000 0000	*,***,w 1,***,***
42H	TMAR	TimerA data register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
43H	TMBCN	ENTMB	TMBCK	TMBS[1:0]		TMBSYC	TMBR2R			0000 00..	0000 00..	******,-,-
44H	TMBRH	TimerB High Byte data register								xxxx xxxx	uuuu uuuu	********,********
45H	TMBRL	TimerB Low Byte data register								xxxx xxxx	uuuu uuuu	********,********
46H	TMCCN	ENTMC	TMCCK[1:0]		TMCS1[2:0]			TMCS0[1:0]		0000 0000	0000 0000	********,********
47H	PRC	TimerC programmable register								1111 1111	1111 1111	********,********
48H	TMCR	TimerC register								0000 0000	0000 0000	r,r,r,r r,r,r,r
49H	CCPCN	CCP1M[3:0]				CCP0M[3:0]				0000 0000	0000 0000	********,********
4AH	CCP0RH	CCP0 High Byte data register								xxxx xxxx	uuuu uuuu	********,********
4BH	CCP0RL	CCP0 Low Byte data register								xxxx xxxx	uuuu uuuu	********,********

Table 4-13 Data Memory Reset Status

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

"."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
4CH	CCP1RH	CCP1 High Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
4DH	CCP1RL	CCP1 Low Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
4EH	PASC	PASF		PASCF[1:0]		PSSCN0[1:0]		PSSCN1[1:0]		0.00 0000	0.00 0000	*,-,*,*,*,*,*,*
4FH	PWMCN	ENPWM	ENPFD	PWMRL[1:0]		PWMCG[1:0]		PWMM[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
50H	PDBD	ENPRS	DBDC[6:0]							0000 0000	0000 0000	*,*,*,*,*,*,*,*
51H	PWMR	PWM MSB Byte register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
52H	LCDCN1	ENLCD	LCDPR	VLCDX[1:0]		LCDBF	LCDBI[1:0]			0000 000.	0000 000.	*,*,*,*,*,*,*,-
53H	LCDCN2	LCDBL	LCDMX[1:0]							000.	000.	*,*,*,- -,-,-,-
54H	LCD0	Segment SEG2@[3:0] and SEG3@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
55H	LCD1	Segment SEG4@[3:0] and SEG5@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
56H	LCD2	Segment SEG6@[3:0] and SEG7@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
57H	LCD3	Segment SEG8@[3:0] and SEG9@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
58H	LCD4	Segment SEG10@[3:0] and SEG11@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
59H	LCD5	Segment SEG12@[3:0] and SEG13@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5AH	LCD6	Segment SEG14@[3:0] and SEG15@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5BH	LCD7	Segment SEG16@[3:0] and SEG17@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5CH	LCD8	Segment SEG18@[3:0] and SEG19@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5DH	LCD9	Segment SEG20@[3:0] and SEG21@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5EH	SSPCON1	SSPEN	CKP	CKE	SMP			SSPM<1:0>		0000 ..00	uuuu ..uu	*,*,*,*-,-,*,*
60H	SSPSTA	SSPBUY	SSPOV						BF	00.. ...0	00.. ...0	r,r,-,- -,-,-,r
61H	SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
63H	URCON	ENSP	ENTX	TX9	TX9D	PARITY			WUE	0000 0..0	0000 0..0	*,*,*,*,-,-,*
64H	URSTA		RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0110	.000 0110	-,r,r,r r,r,r,rw 0
65H	BAUDCON					ENCR	RC9	ENADD	ENABD	 0000	 0000	-,-,-,-,*,*,*,*
66H	BRGRH				Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-,-,-,-,*,*,*,*
67H	BRGRL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
68H	TXREG	UART Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
69H	RCREG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111	*,*,*,*,*,*,*,*
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
6DH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,* r,r,r,r
6EH	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4					0000	0000	*,*,*,*-,-,-,-
6FH	PT1DA						DA1.2	DA1.1	DA1.0	.... .000	.... .000	-,-,-,- -,*,*,*
70H	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
71H	PT1M1					INTEG1[1:0]		INTEG0[1:0]		 0000	 0000	-,-,-,- *,*,*,*
72H	PT1M2		PM1.7[0]		PM1.6[0]		PM1.5[0]		PM1.4[0]	.0.0 .0.0	.0.0 .0.0	-,*,-,* -,*,-,*
73H	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2			0000 00..	0000 00..	*,*,*,*,*,-,-
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
76H	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
78H	PT2M1		PM2.3[0]	PM2.2[1]	PM2.2[0]					.000	.000	-,*,*,-,-,-,-,-
79H	PT2M2	PWMTR[1]	PWMTR[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	00.0 0000	00.0 0000	*,*,-,*,*,*,*,*
7AH	PT3	PT3.7	PT3.6	PT3.5	PT3.4	PT3.3	PT3.2	PT3.1	PT3.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
7BH	TRISC3	TC3.7	TC3.6	TC3.5	TC3.4	TC3.3	TC3.2	TC3.1	TC3.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
7DH	PT3PU	PU3.7	PU3.6	PU3.5	PU3.4	PU3.3	PU3.2	PU3.1	PU3.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
80H ~ FFH	GPR0	General Purpose Register as 128Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
100H~17FH	GPR1	General Purpose Register as 128Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
180H	LCD10	Segment SEG22@[3:0] and SEG23@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
181H	LCD11	Segment SEG24@[3:0] and SEG25@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
182H	LCD12	Segment SEG26@[3:0] and SEG27@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
183H	LCD13	Segment SEG28@[3:0] and SEG29@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
184H	LCD14	Segment SEG30@[3:0] and SEG31@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
185H	LCD15	Segment SEG32@[3:0] and SEG33@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
186H	LCD16	Segment SEG34@[3:0] and SEG35@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
187H	LCD17	Segment SEG36@[3:0] and SEG37@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
188H	LCD18	Segment SEG38@[3:0] and SEG39@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
189H	LCD19	Segment SEG40@[3:0] and SEG41@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
193H	PT5DA					DA5.3	DA5.2	DA5.1	DA5.0	 1111	 1111	*,*,*,*,*,*,*,*
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
195H	BIECTRLA					VPP_HIGH		BIEWR	BIERD	1000 d000	1000 d000	-,-,-,- *,r0,*,*
196H	BIEPTRHA	SBMSEL					BIE_ADDR[10:8]			0000 0000	0000 0000	w0,w0,W0,W0 W0,W0,W0,W0
197H	BIEPTRLA	0	0	BIE_ADDR[5:0]						0000 0000	0000 0000	w 0,w 0,*,*,*,*,*,*
198H	BIEDHA	BIE_DATA[15:8]								xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
199H	BIEDLA	BIE_DATA[7:0]								xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*,*
19AH	OPCN2	ENOP2	OPM2[1:0]		OPIS2[0]	OPP2[1:0]		OPN2[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
200H ~ 2FFH	GPR2	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
300H ~ 3FFH	GPR3	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
400H ~ 4FFH	GPR4	General Purpose Register as 256Byte								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 4-14 Data Memory Reset Status (continued)

5. Interrupt

The interrupt consists of the interrupt enable controller INTE and the 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：

INTE1 GIE[0],ADCIE[0],TMCIE[0],TMBIE[0],TMAIE[0],WDTIE[0],E1IE[0],E0IE[0]

INTE2 TXIE[0],RCIE[0],CPOIE[0],SSPIE[0],CCP1IE[0],CCP0IE[0]

INTE3 E7IE[0],E6IE[0], E5IE[0],E4IE[0], E3IE[0],E2IE[0]

NTF1 ADCIF[0],TMCIF[0],TMBIF[0],TMAIF[0],WDTIF[0],E1IF[0],E0IF[0]

INTF2 TXIF[0],RCIF[0],CPOIF[0],SSPIF[0],CCP1IF[0],CCP0IF[0]

INTF3 E7IF[0],E6IF[0], E5IF[0],E4IF[0], E3IF[0],E2IF[0]

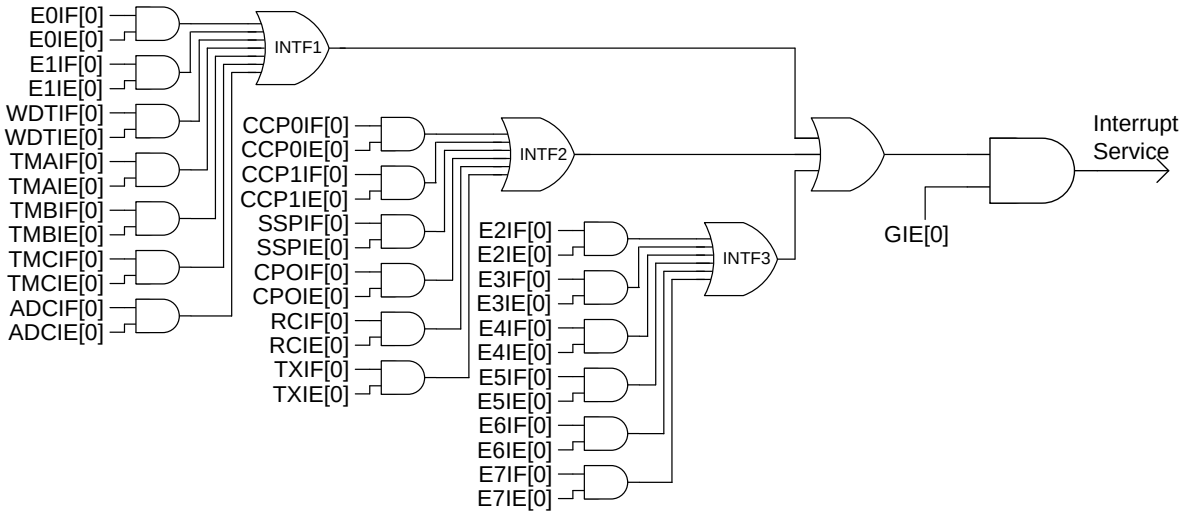


Figure 5-11 Interrupt Vector Block Diagram

There are two layers of interrupt service event controller. The top layer is global interrupt enable, GIE [0] 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 [0] that relatives to interrupt control register INTE1 [7:0] to be 1. Conversely, set up 0 to close the interrupt service.

GIE [0] 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 [0] will automatically be set up as 1. Moreover, if return instruction RET is conducted, GIE [0] 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"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,- *,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,- *,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

Table 5-15Interrupt Registers

INTE1：Interrupt Starting Control Register 1

GIE[0]：Global interrupt enable

1：Start

0：Shutoff

ADCIE[0]：ADC interrupt event starting controller

1：Start (analog-to-digital converter, SD18)

0：Shutoff

TMCIE[0]：Timer-C interrupt event starting controller

1：Start (Timer C, TMC)

0：Shutoff

TMBIE[0]：Timer-B interrupt event starting controller

1：Start (Timer B, TMB)

0：Shutoff

TMAIE[0]：Timer-A interrupt event starting controller

1：Start (Timer A, TMA)

0：Shutoff

WDTIE[0]：Watch Dog interrupt event starting controller

1：Start (WDT)

0：Shutoff

E1IE[0]：Interrupt event starting controller of input pin 1

1：Start (external input pin, PT1.1)

0：Shutoff

E0IE[0]：Interrupt event starting controller of input pin 0

1：Start (external input pin, PT1.0)

0：Shutoff

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

INTE2：Interrupt Starting Control Register 2

TXIE[0]：TX interrupt event starting controller

1：Start (Communication interface, EUART)

0：Shutoff

RCIE[0]：RX interrupt event starting controller

1：Start (Communication interface, EUART)

0：Shutoff

CPOIE[0]：CPO interrupt event starting controller

1：Start (Enhanced comparator)

0：Shutoff

SSPIE[0]：SPI interrupt event starting controller

1：Start (Communication interface, SPI)

0：Shutoff

CCP1IE[0]：Capture/Compare 1 interrupt event starting controller

1：Start (Capture/compare mode II ,CCP1)

0：Shutoff

CCP0IE[0]：Capture/Compare 0 interrupt event starting controller

1：Start (Capture/compare mode I ,CCP0)

0：Shutoff

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

INTE3：Interrupt Starting Control Register 3

E7IE[0]：Interrupt event starting controller of input pin 7

1：Start (External input pin, PT1.7)

0：Shutoff

E6IE[0]：Interrupt event starting controller of input pin 6

1：Start (External input pin, PT1.6)

0：Shutoff

E5IE[0]：Interrupt event starting controller of input pin 5

1：Start (External input pin, PT1.5)

0：Shutoff

E4IE[0]：Interrupt event starting controller of input pin 4

1：Start。(External input pin, PT1.4)

0：Shutoff

E3IE[0]：Interrupt event starting controller of input pin 3

1：Start (External input pin, PT1.3)

0：Shutoff

E2IE[0]：Interrupt event starting controller of input pin 2

1：Start (External input pin, PT1.2)

0：Shutoff

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

INTF1：Interrupt Event Flag Register 1

ADCIF[0]：ADC interrupts event flag

1：Happened (Analog-to-digital converter, SD18)

0：Not happened

TMCIF[0]：Timer-C interrupts event flag

1：Happened (Timer C, TMC)

0：Not happened

TMBIF[0]：Timer-B interrupts event flag

1：Happened (Time B, TMB)

0：Not happened

TMAIF[0]：Timer-A interrupt event flag

1：Happened (Timer A, TMA)

0：Not happened

WDTIF[0]：Watch Dog interrupt event flag

1：Happened (WDT)

0：Not happened

E1IF[0]：Interrupt event flag of input pin 1

1：Happened (External input pin, PT1.1)

0：Not happened

E0IF[0]：Interrupt event flag of input pin 0

1：Happened (External input pin, PT1.0)

0：Not happened

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

INTF2：Interrupt Event Flag Register 2

TXIF[0]：TX interrupts event flag

1：Happened (Communication interface, EUART)

0：Not happened

RCIF[0]：RC interrupts event flag

1：Happened (Communication interface, EUART)

0：Not happened

CPOIF[0]：CPO interrupts event flag

1：Happened (Enhanced comparator, ECPA)

0：Not happened

SSPIF[0]：SPI interrupt event flag

1：Happened (Communication interface, SPI)

0：Not happened

CCP1IF[0]：Capture/Compare Ⅱ interrupt event flag

1：Happened (Capture/compare mode Ⅱ, CCP1)

0：Not happened

CCP0IF[0]：Capture/Compare Ⅰ interrupt event flag

1：Happened (Capture/compare mode Ⅰ, CCP0)

0：Not happened

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2	TXIE	RCIE			CPOIE	SSPIE	CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-

INTF3：Interrupt Event Flag Register 3

E7IF[0]：Interrupt event flag of input pin 7

1：Happened (External input pin, PT1.7)

0：Not happened

E6IF[0]：Interrupt event flag of input pin 6

1：Happened (External input pin, PT1.6)

0：Not happened

E5IF[0]：Interrupt event flag of input pin 5

1：Happened (External input pin, PT1.5)

0：Not happened

E4IF[0]：Interrupt event flag of input pin 4

1：Happened (External input pin, PT1.4)

0：Not happened

E3IF[0]：Interrupt event flag of input pin 3

1：Happened ◦ (External input pin, PT1.3)

0：Not happened

E2IF[0]：Interrupt event flag of input pin 2

1：Happened (External input pin, PT1.2)

0：Not happened

6. Hardware Multiplier

H08A instruction set has 8x8 hardware multiplier processing instruction "MULF and MULL". H08B instruction set, however, does not have hardware multiplier function. 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 STATUS[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 Example 6-1 and Example 6-2 illustrates.

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

Example 6-4 Unsigned Value Operation

```
Ex 2：N1 x N2 = N, s=7, B
MVL     N1              ; Put N1 value to register W
MVF     BUF0,1,0        ; Put N1 value to memory block 1 of register BUF0
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
; Determin whether N1 is minus, if so, then set PRODH – N2
; Determin 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 6-5 Signed Value Operation

7. Input/Output Port, I/O

Every 8 I/O is a port that can be utilized as digital input/output and analog signal measurement channel. Every port is controlled by a set of registers. Small discrepancies of I/O register composition exist in different products.

I/O Related Registers：

PT PT1[7:0], PT2[7:0], PT3[7:0], PT4[7:0], PT5[7:0]

TRISC TC1[7:4], TC2[7:0], TC3[7:0],

PTDA DA1[2:0], DA2[7:0], DA4[7:0], DA5[5:0]

PTPU PU1[7:0], PU2[7:0], PU3[7:0], PU4[7:0], PU5[7:0]

PTM PM1[7:0], PM2[7:0]

PT1INT INTEG7[0], INTEG6[0], INTEG5[0], INTEG4[0], INTEG3[0], INTEG2[0],

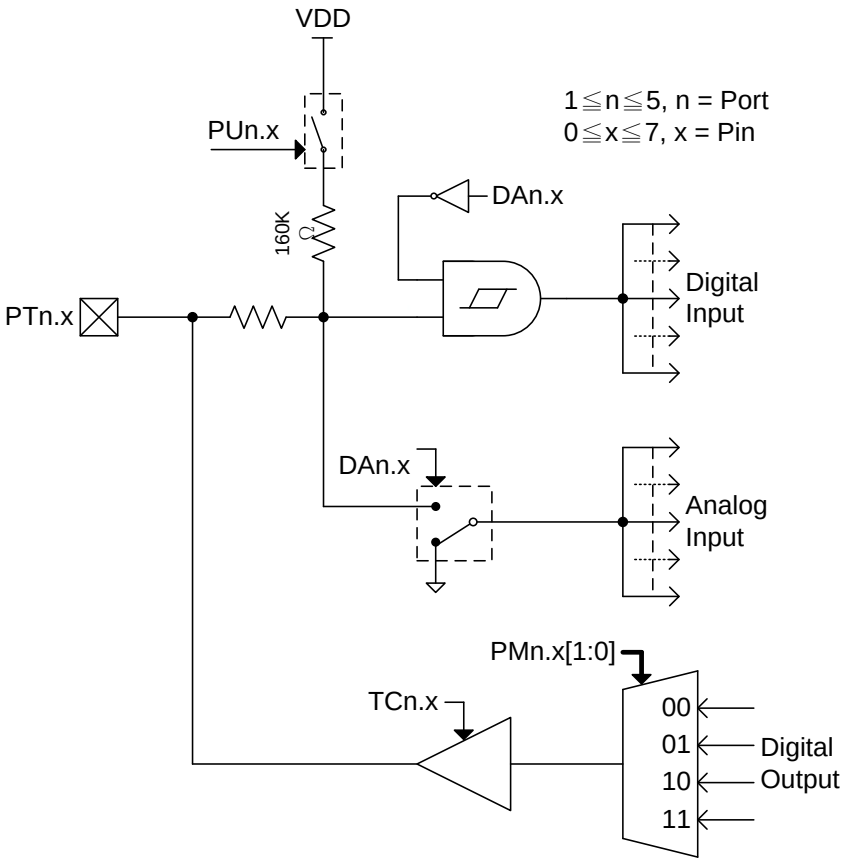


Figure 7-12 Block Diagram of I/O Frame

7.1. PORT Related Register Introduction

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

7.1.1. PT 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.

7.1.2. TRISC Input/Output Control Register

Selecting I/O to be input or output, set <1> and the I/O is in output status, set <0>, the I/O is in output 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.

7.1.3. PTDA Digital or Analog Input Control Register

Selecting I/O to be analog/digital input status, set <1> will be analog input; set <0> will be digital input. Other I/O related registers configuration status must also be taken into accounts as to avoid interference of digital/analog signal.

7.1.4. 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.

7.1.5. PTM Digital Output Mode Select Register

I/O output mode selector, PMn.x[1:0] can configure I/O output signal($1 \leqq n \leqq 3, 0 \leqq x \leqq 7$). Partial I/O equips at least 1 kind of digital peripheral circuit output signals. Thus, correctly configures PMn.x[1:0] can get expected output signal.

7.1.6. PTINT Interrupt Signal Generated Conditions

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

7.2. Buzzer

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

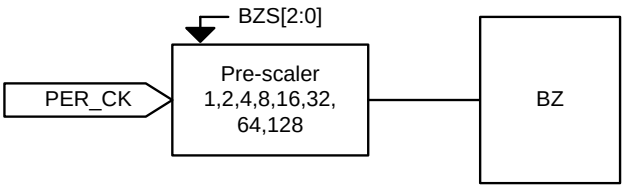


Figure 7-13 BZ Block Diagram

7.3. I/O Port 1

"i"：input, "o"：output, "a"：analog, "c"：cmos i/o, "x"：undefined, "p"：power

Pin Name	Design		Register Configuration			Description
	Type	Buffer	TC[0]	DA[0]	PM[1:0]	
PT1.0	i	c	x	0	00	Digital input pin
INT0	i	s	0	0	00	External interrupt source
CPAI6	a	a	0	1	00	Enhanced comparator signal input pin
PSCK	i	s	0	0	00	OTP programming interface SCK pin
PT1.1	i	c	x	0	00	Digital input pin
INT1	I	s	0	0	00	External interrupt source
CPAI7	a	a	0	1	00	Enhanced comparator signal input pin
SCE	I	s	0	0	00	SPI communication interface pin, SCE
PSDI	i	s	0	0	00	OTP programming SDI pin
PT1.2	i	c	x	0	00	Digital input pin
INT2	i	s	0	0	00	External interrupt source
SDI	i	s	0	0	00	SPI communication interface pin, SDI
LVDIN	a	a	0	1	00	LVD external signal input pin
PT1.3	i	c	x	0	00	Digital input pin
INT3	i	s	0	0	00	External interrupt source
TST	i	s	0	0	00	Reserved by manufacturer
RC	i	s	0	0	00	EUART communication interface RC pin
PT1.4	i/o	c	x	0	00	Digital input/output pin
INT4	i	s	0	0	00	External interrupt source
TX	o	c	1	0	01	EUART communication interface TX pin
PT1.5	i/o	c	x	0	00	Digital input/output pin
INT5	i	c	0	0	00	External interrupt source
SDO	o	c	1	0	01	SPI communication interface pin, SDO
PSDO	o	c	1	0	00	OTP programming pin, SDO
PT1.6	i/o	c	x	0	00	Digital input/output pin
INT6	i	s	0	0	00	External interrupt source
SCK	i/o	s	x	0	0x	SPI communication interface pin, SCK
PT1.7	i/o	c	x	0	00	Digital input/output pin
INT7	i	s	0	0	00	External interrupt source
FIL0	i	s	0	0	00	PWM auto shutoff trigger event
BZ	o	c	1	0	01	Buzzer output pin

Table 7-16 PORT1 Function

7.3.1. Register Description-PORT1

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
25H	INTE3	E7IE	E6IE	E5IE	E4IE	E3IE	E2IE			0000 00..	0000 00..	*,*,*,*,*,*,-,-
26H	INTF1		ADCIF	TMCIF	TMBIF	TMAIF	WDTIF	E1IF	E0IF	.000 0000	.000 0000	-,*,*,*,*,*,*,*
28H	INTF3	E7IF	E6IF	E5IF	E4IF	E3IF	E2IF			0000 00..	0000 00..	*,*,*,*,*,*,-,-
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000	*,*,*,-,*,*,*,*
6DH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,* r,r,r,r
6EH	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4					0000	0000	*,*,*,* -,-,-,-
6FH	PT1DA						DA1.2	DA1.1	DA1.0	.... .000	.... .000	-,-,-,- -,*,*,*
70H	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
71H	PT1M1					INTEG1[1:0]		INTEG0[1:0]		 0000	 0000	-,-,-,- *,*,*,*
72H	PT1M2		PM1.7[0]		PM1.6[0]		PM1.5[0]		PM1.4[0]	.0.0 .0.0	.0.0 .0.0	-,*,-,* -,*,-,*

Table 7-17 PORT1 Control Register

INTE1/INTF1：Please refer to Interrupt Chapter

MCKCN3：Please refer to Oscillator, Clock Sources and Power Managed Modes Chapter

PT1：PORT1 Status Control Register

PT1.x：External pin control bit（0≦x≦7）

1：High potential

0：Low potential

TRISC1：Input/Output Control Register

TC1.x：External input/output pin control bit（0≦x≦7）

1：Output

0：Input

PT1DA：Digital/Analog Input Control Register

DA1.x：External input pin analog/digital signal control bit（0≦x≦7）

1：Analog

0：Digital

PT1PU：Pull-Up Resistor Control Register

PU1.x：External pull-up resistor pin control bit（0≦x≦7）

1：Start

0：Shutoff

PT1Mn[1:0]：Digital Output Mode Select Register1, 2

PM1.x：I/O digital output mode（2≦x≦7）selector, please refer to 7.1.5 for further description.

"-"：Unused

PM1.X **[1:0]** / PM1.**X**		11	10	01	00
PT1	4	-	-	TX	$V_{OH/L}$
	5	-	-	SDO	$V_{OH/L}$
	6	-	-	SCK	$V_{OH/L}$
	7	-	-	BZ	$V_{OH/L}$

Table 7-18 PT1M2 Register Multiplexed Function Comparison

INTEGx[1:0]：Interrupt Signal generated condition（0≦x≦1）

11：Potential conversion (0→1 or 1→0)

10：Potential conversion (0→1 或 1→0)

01：Rising Edge (0→1)

00：Falling Edge (1→0)

PT1INT：I/O Interrupt Signal Generated Condition

INTEGx：Interrupt Signal generated condition（2≦x≦7）

1：Rising Edge (0→1)

0：Falling Edge (1→0)

7.4. Input/Output Port 2, I/O Port2

"i"：input, "o"：output, "a"：analog, "c"：cmos i/o, "x"：undefined, "p"：power

Pin Name	Design		Register Configuration			Description
	Type	Buffer	TC[0]	DA[0]	PM[1:0]	
PT2.0	i/o	c	x	0	00	Digital input/output pin
XTO	a	a	x	0	00	External oscillator pin
PT2.1	i/o	c	X	0	00	Digital input/output pin
XTI	a	a	X	0	00	External oscillator pin
PT2.2	i/o	c	X	0	00	Digital input/output pin
CPAI0	a	a	0	1	00	Enhanced comparator signal input pin
PWM0	o	c	1	0	01	PWM output pin
PFD	o	c	1	0	10	PFD output pin
PT2.3	i/o	c	X	0	00	Digital input/output pin
CPAI1	a	a	0	1	00	Enhanced comparator signal input pin
PWM1	o	c	1	0	01	PWM output pin
TMCKI	i	s	0	0	00	TIMER-C clock source input pin
PT2.4	i/o	c	X	0	00	Digital input/output pin
CPAI2	a	a	0	1	00	Enhanced comparator signal input pin
PWM2	o	c	1	0	01	PWM output pin
CCP0	i	s	0	0	10	Capture/compare mode signal pin
PT2.5	i/o	c	X	0	00	Digital input/output pin
CPAI3	a	a	0	1	00	Enhanced comparator signal input pin
PWM3	o	c	1	0	01	PWM output pin
CCP1	i	s	0	0	10	Capture/compare mode signal pin
PT2.6	i/o	c	X	0	00	Digital input/output pin
CPAI4	a	a	0	1	00	Enhanced comparator signal input pin
CPAO	o	c	1	0	01	Enhanced comparator signal output pin
PT2.7	i/o	c	X	0	00	Digital input/output pin
CPAI5	a	a	0	1	00	Enhanced comparator signal input pin
CPAO	o	c	1	0	01	Enhanced comparator signal output pin

Table 7-19 PORT2 Function

7.4.1. Register Description-PORT2

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
76H	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
78H	PT2M1		PM2.3[0]	PM2.2[1]	PM2.2[0]					.000	.000	-,*,*,* -,-,-,-
79H	PT2M2	PWMTR[1]	PWMTR[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	00.0 0000	00.0 0000	*,*,-,*,*,*,*,*

Table 7-20 PORT2 Control Register

PT2：PORT2 Status Control Register

PT2.x：External pin control bit（0≦x≦7）

1：High potential

0：Low potential

TRISC2：Input/Output Control Register

TC2.x：External input/output pin control bit（0≦x≦7）

1：Output

0：Input

PT2DA：Digital/Analog Input Control Register

DA2.x：External analog/digital signal pin control bit（0≦x≦7）

1：Analog

0：Digital

PT2PU：Pull-Up Resistor Control Register

PU2.x：External pull-up resistor pin control bit（0≦x≦7）

1：Start

0：Shutoff

PT2M1/2[1:0]：Digital Output Mode Select Register 1, 2

PM2.x：Digital output mode（0≦x≦7）

"-"：Unused

PM2.X **[1:0]** / PM2.**X**		11	10	01	00
PT2	0	-	-	-	$V_{OH/L}$
	1	-	-	-	$V_{OH/L}$
	2	-	**PFD**	**PWM0**	$V_{OH/L}$
	3	-	-	**PWM1**	$V_{OH/L}$

Table 7-21 PT2M1 Register Multiplexed Function Comparison

"-"：Unused

PM2.X [1:0] / PM2.X		11	10	01	00
PT2	4	-	**CCP0**	**PWM2**	$V_{OH/L}$
	5	-	**CCP1**	**PWM3**	$V_{OH/L}$
	6	-	-	**CPAO**	$V_{OH/L}$
	7	PWMTR [1:0]		**CPAO**	$V_{OH/L}$

Table 7-22 PT2M2 Register Multiplexed Function Comparison

7.5. Input/Output Port 3, I/O Port3

"i"：input, "o"：output, "a"：analog, "c"：cmos i/o, "x"：undefined, "p"：power

Pin Name	Design		Register Configuration			Description
	Type	Buffer	TC[0]	DA[0]	PM[1:0]	
PT3.0	i/o	c	x	0	00	Digital input/output pin
PT3.1	i/o	c	x	0	00	Digital input/output pin
PT3.2	i/o	c	x	0	00	Digital input/output pin
PT3.3	i/o	c	x	0	00	Digital input/output pin
PT3.4	i/o	c	x	0	00	Digital input/output pin
PT3.5	i/o	c	x	0	00	Digital input/output pin
PT3.6	i/o	c	x	0	00	Digital input/output pin
PT3.7	i/o	c	x	0	00	Digital input/output pin

Table 7-23 PORT3 Function

7.5.1. Register Description-PORT3

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
7AH	PT3	PT3.7	PT3.6	PT3.5	PT3.4	PT3.3	PT3.2	PT3.1	PT3.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
7BH	TRISC3	TC3.7	TC3.6	TC3.5	TC3.4	TC3.3	TC3.2	TC3.1	TC3.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
7DH	PT3PU	PU3.7	PU3.6	PU3.5	PU3.4	PU3.3	PU3.2	PU3.1	PU3.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 7-24 PORT3 Control Register

PT3：PORT3 Status Control Register

PT3.x：External pin control bit（0≦x≦7）

1：High potential

0：Low potential

TRISC3：Input/Output Control Register

TC3.x：External input/output pin control bit（0≦x≦7）

1：Output

0：Input

PT3PU：Pull-Up Resistor Control Register

PU3.x：External pull-up resistor pin control bit（0≦x≦7）

1：Start

0：Shutoff

7.6. Input/Output Port 4, I/O Port4

"i"：input, "o"：output, "a"：analog, "c"：cmos i/o, "x"：undefined, "p"：power

Pin Name	Design		Register Configuration			Description
	Type	Buffer	TC[0]	DA[0]	PM[1:0]	
PT4.0	i	c	x	0	xx	Digital input/output pin
AI0	a	a	x	1	xx	Digital input pin
PT4.1	i	c	x	0	xx	Digital input/output pin
AI1	a	a	x	1	xx	Digital input pin
PT4.2	i	c	x	0	xx	Digital input/output pin
AI2	a	a	x	1	xx	Digital input pin
PT4.3	i	c	x	0	xx	Digital input/output pin
AI3	a	a	x	1	xx	Digital input pin
PT4.4	i	c	x	0	xx	Digital input/output pin
AI4	a	a	x	1	xx	Digital input pin
PT4.5	i	c	x	0	xx	Digital input/output pin
AI5	a	a	x	1	xx	Digital input pin
PT4.6	i	c	x	0	xx	Digital input/output pin
AI6	a	a	x	1	xx	Digital input pin
PT4.7	i	c	x	0	xx	Digital input/output pin
AI7	a	a	x	1	xx	Digital input pin

Table 7-25 PORT4 Function

7.6.1. Register Description-PORT4

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111	*,*,*,*,*,*,*,*
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 7-26 PORT4 Control Register

PT4：PORT4 Status Control Register

PT4.x：External pin control bit（0≦x≦7）

1：High potential

0：Low potential

PT4DA：Digital/Analog Input Control Register

DA4.x：External analog/digital signal input pin control bit（0≦x≦7）

1：Analog

0：Digital

PT4PU：Pull-Up Resistor Control Register

PU4.x：External pull-up resistor pin control bit（0≦x≦7）

1：Start

0：Shutoff

7.7. Input/Output Port 5, I/O Port5

"i"：input, "o"：output, "a"：analog, "c"：cmos i/o, "x"：undefined, "p"：power

Pin Name	Design		Register Configuration			Description
	Type	Buffer	TC[0]	DA[0]	PM[1:0]	
PT5.0	i	c	x	0	xx	Digital input/output pin
AI8	a	a	x	1	xx	Analog input pin
PT5.1	i	c	x	0	xx	Digital input/output pin
AI9	a	a	x	1	xx	Analog input pin
PT5.2	i	c	x	0	xx	Digital input/output pin
AI10	a	a	x	1	xx	Analog input pin
PT5.3	i	c	x	0	xx	Digital input/output pin
AI11	a	a	x	1	xx	Analog input pin
PT5.4	i	c	x	0	xx	Digital input/output pin
PT5.5	i	c	x	0	xx	Digital input/output pin
PT5.6	i	c	x	0	xx	Digital input/output pin
PT5.7	i	c	x	0	xx	Digital input/output pin

Table 7-27 PORT5 Function

7.7.1. Register Description-PORT5

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA					DA5.3	DA5.2	DA5.1	DA5.0	 1111	 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

Table 7-28 PORT5 Control Register

PT5：PORT5 Status Control Register

PT5.x：External pin control bit（0≦x≦7）

1：High potential

0：Low potential

PT5DA：Digital/Analog Input Control Register

DA5.x：External analog/digital signal input pin control bit（0≦x≦3）

1：Analog

0：Digital

PT5PU：Pull-Up Resistor Control Register

PU5.x：External pull-up resistor pin control bit（0≦x≦7）

1：Start

0：Shutoff

8. Low Voltage Detect

Low voltage detect, LVD equips with operating voltage detect or external input voltage function. These functions enable users to correctly determine the voltage level when the monitored voltage produces low voltage events.

LVD includes the following functions

- VDD operating voltage and external input voltage detect function
- Able to configure 14 steps of low potential detect points of operating voltage.
- Simple external potential comparison function.

LVD Related Register：

LVDCN LVDFG[0], LVD [0],LVDON[0],VLDX[3:0]

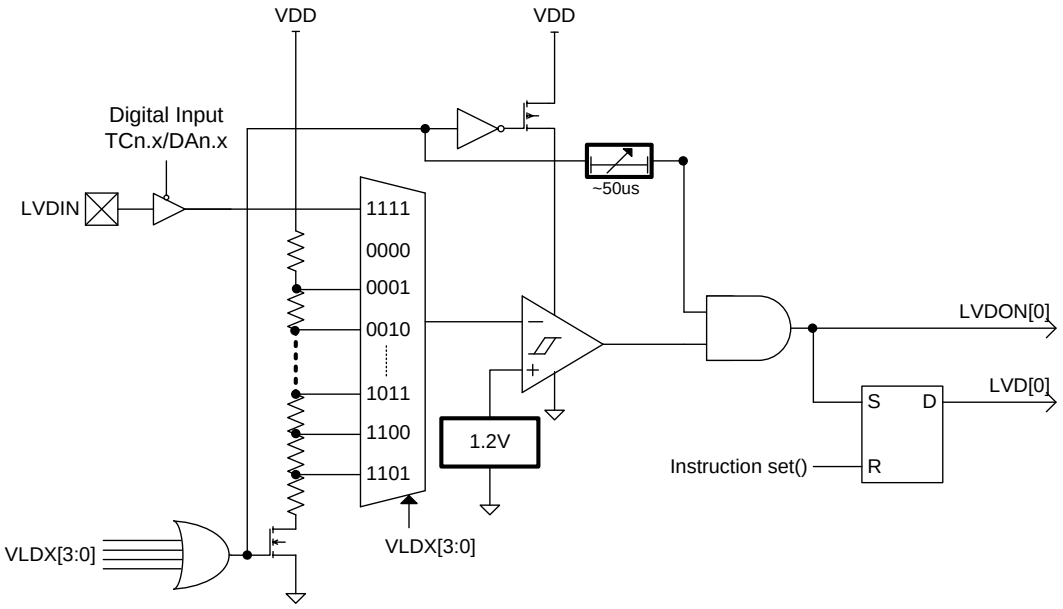


Figure 8-14 LVD Block Diagram

8.1. Low Voltage Detect Manual

8.1.1. LVD Initial Configuration

The LVD detection voltage can be determined by the operating voltage divider or the external input pin, LVDIN, by setting the monitor voltage selector VLDX [3:0].

When VLDX[3:0] is configured as operating voltage detect, 14 steps voltage detect points can be set up by matching the voltage divided resistors. If it was configured as external input, users may need to design voltage divided circuit to produce voltage signal. The signal will access to LVD comparator through LVDIN pin.

8.1.2. LVD Low Voltage Occurred Condition and Event Record

The way of LVD voltage detection is through comparing the detected voltage signal and internal voltage reference of internal comparator. If the detected voltage heightens than voltage reference, the comparator outputs 1; on the contrary, it will output 0. In order to keep low voltage detect functions normally, users must refer to stable flag, LVDON[0] status and wait 50uS delay time after modifying VLDX[3:0] configuration. After the circuit is stable, configure LVDON[0] as <1> to determine low voltage react flag, LVD[0]. Conversely, if LVDON[0] is configured as <0>, LVD[0] may be error information.

Low voltage event record flag, LVDFG[0] records instantaneously the occurred low voltage event, as Figure 8-2 illustrates. When a sudden low voltage event happened to monitored voltage, LVD[0] will be set <1>, so as LVDFG[0]. Until the monitored voltage restores to or above than the default value, configure LVD[0] as <0> and LVDFG[0] as <1> .When users configure or reset the IC, will the LVDFG[0] be set up as <0>.

8.1.3. LVD Initiation

When the configuration of VLDX[3:0] not equals to <0000>, LVD will start automatically. By contrast, if VLDX[3:0] equals to <0000>, LVD will close automatically.

Any of ENHAO/ENADC/ENACM must be enabled will LVD functions normally.

If only external 32768 oscillator and LVD function is used, the best power-saving way is to configure ENACM=1.

VDD=3V, CPUCLK=ext 32768 Hz, enable LVD, Normal Run (JMP LOOP)

Operation current (uA)

ENHAO=1 78uA

ENADC=1 60uA

ENACM=1 38uA

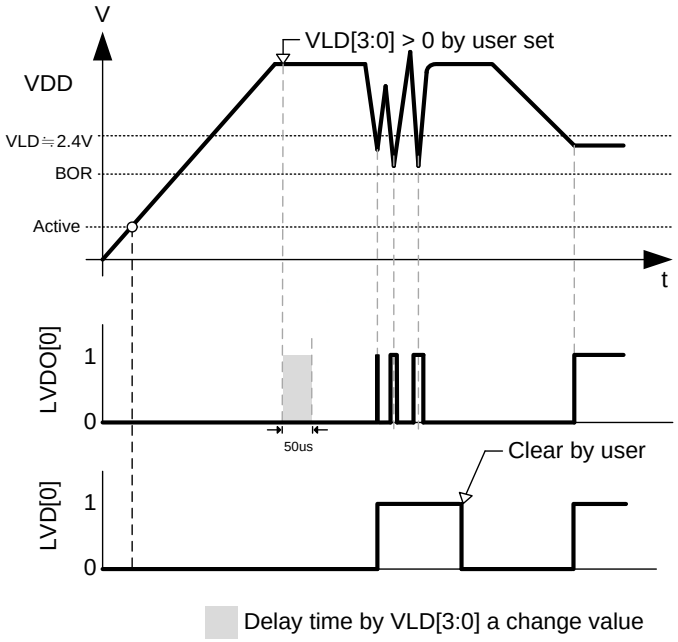


Figure 8-15 LVD Sequence

8.2. Register Description-LVD

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
2DH	LVDCN		LVDFG	LVD	LVDON	VLDX[3:0]				.000 0000	.000 uuuu	-,*,r,r *,*,*,*

Table 8-29 LVD Control Register

PSTATUS：Please Refer To RESET Chapter

PT1/PT1DA/PT1PU：Please Refer To Input/Output Port, I/O Chapter

LVDCN：LVD Control Register

LVDFG：Record flag of low voltage occurrence

1：Happened（Cleared must through software）

0：Not happened

LVD：Low voltage response flag

1：Low voltage

0：Not happened

LVDON：Low voltage detect circuit stable flag

1：Stable

0：Unstable

VLDX[3:0]：Monitored Voltage Selector

VLDX[3:0]	Monitored Potential	VLDX[3:0]	Monitored Potential	VLDX[3:0]	Monitored Potential	VLDX[3:0]	Monitored Potential
1111	1.2(LVDIN)	1011	3.01	0111	2.61	0011	2.21
1110	3.31	1010	2.91	0110	2.51	0010	2.11
1101	3.21	1001	2.81	0101	2.41	0001	2.01
1100	3.11	1000	2.71	0100	2.31	0000	LVDOFF

Table 8-30 LVD Monitored Voltage Default Value

9. Watch Dog Timer

Watch dog timer (WDT) is IC guardian that mainly being applied in awakening event.

- Run Mode
  - WDT overflow generates reset signal, IC will be restarted.
  - Timer can be cleared zero through software
- Sleep Mode
  - Shutoff WDT, it is inactive
- Idle Mode
  - WDT overflow generates interrupt event to awake IC

WDT Related Registers：

TMACN	ENWDT[0],WDTS[2:0]
PSTATUS	TO[0]
INTF	WDTIF[0]
INTE	WDTIE[0]

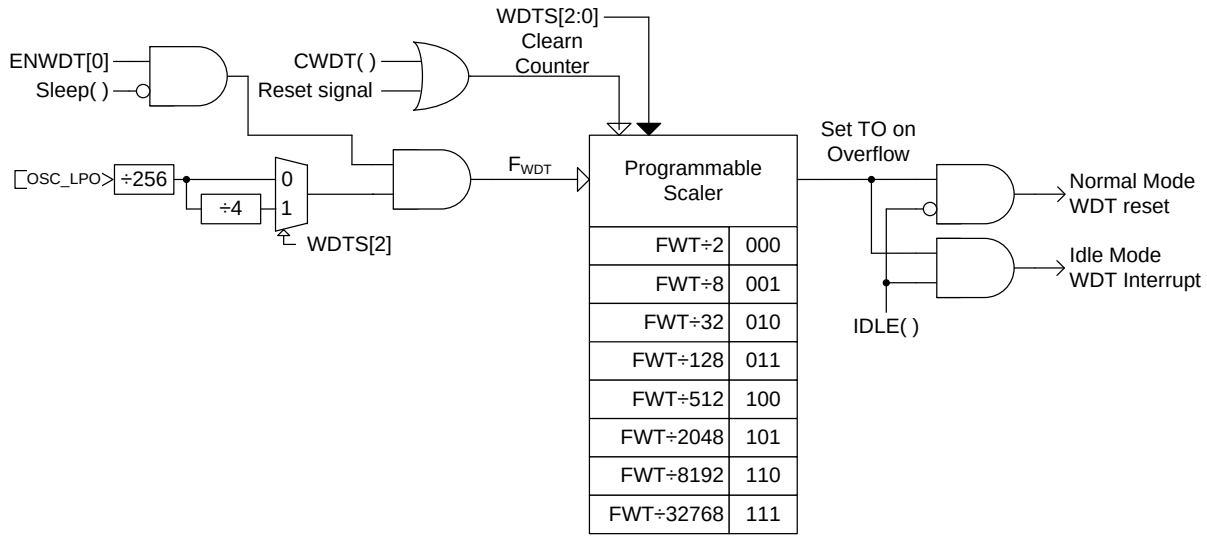


Figure 9-16 WDT Block Diagram

9.1. WDT Manual

9.1.1. WDT Initial Configuration

WDT counting controller, WDTS[2:0] can decide WDT counter operating frequency F_{WDT} and overflow. WDT reset signal TO or interrupt event, WDTIF[10] will be produced after the counter being overflowed.

[10] 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. Moreover, in Idle Mode, hardware will automatically set ENWDT[0] as 0 if WDT is awakened by interrupt event.

```
MVL 00Ah
MVF TMACN,1,0      ; Start WDT and configure WDTS[2:0] = 010b
....               ; WDT overflow time is around: 3.4Hz
CWDT               ; Zero WDT
....
```

Example 9-6 WDT Reset Event Example Program

```
MVL   00Ah
MVF   TMACN,1,0      ; Start WDT and configure WDTS[2:0] = 010b
....                 ; WDT overflow time is around: 3.4Hz
CWDT                 ; Zero WDT
IDLE                 ; Enter into Idle Mode. After wake up from Idle Mode, ENWDT will be turned
                     off; users need to restart it again.
....
Idle Interrupt :     ; Interrupt Service Program
BCF   INTF,WDTIF,0   ; Clear WDT Interrupt event flag
....
RETI                 ; Interrupt service return
```

Example 9-7 WDT Interrupt Event Example Program

9.2. Register Description-WDT

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
26H	INTF1						WDTIF			.000 0000	.000 0000	-,*,*,*,*,*,*,*
2CH	Pstaus	PD	TO	IDLEB	BOR		SKERR			000d .0..	uduu .d..	rw0,rw0,rw0,rw0 -,rw0,-,-
41H	TMACN					ENWDT	WDTS[2:0]			0000 0000	0000 0000	*,*,*,* w1,*,*,*

Table 9-31 WDT Control Register

INTE1/INTF1：Please refer to Interrupt Chapter

PSTATUS：Please refer to RESET Chapter

TMACN：Timer-A Control Register

ENWDT：WDT Starting Controller

1：Start

0：Shutoff；(cannot be configured <0> through software)

WDTS[2:0]：Configure WDT overflow time

111：FWDT/32768

110：FWDT/8192

101：FWDT/2048

100：FWDT/512

011：FWDT/128

010：FWDT/32

001：FWDT/8

000：FWDT/2

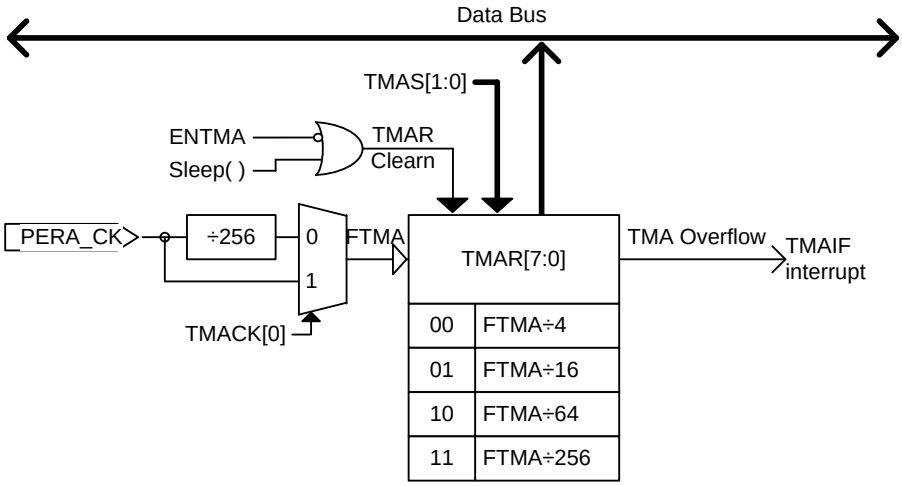
10. Timer-A

Timer-A is designed in 8-bit frame. TMA can function in Run Mode and Idle Mode.

- Ascending counter
- 4-step overflow value select
- Overflow generated interrupt event
- Counter value is readable

TMA Registers：

TMACN	ENTMA[0],TMACK[0],TMAS[1:0]
TMAR	TMAR[7:0]
INTE1	TMAIE[0]
INTF1	TMAIF[0]



Configure TMAS[1:0]=00b, when TMAR[7:0]=00000100b, first interrupt occurs,
The next time interrupt occurs TMAR[7:0]=00001000b。
So, every time interrupt occurs TMAR[7:0]=TMAR[7:0]+4.

Configure TMAS[1:0]=10b, when TMAR[7:0]=01000000b, first interrupt occurs,
The next time interrupt occurs TMAR[7:0]=10000000b。
So, every time interrupt occurs TMAR[7:0]=TMAR[7:0]+64.

Figure 10-17 Timer-A Block Diagram

10.1. TMA Manual

10.1.1. TMA Initial Configuration

TMA operating frequency is provided by PERA_CK, by configuring TMACK [0] can implement PERA_CK pre-frequency divided, F_{TMA} operating frequency is PERA_CK/256 or PERA_CK.

TMAR[7:0] is the counter of TMA, overflow or interrupt event TMAIF[0] time can be configured separately through TMAS[1:0]. TMAS [1: 0] sets TMAR [7: 0] to generate an overflow[11] for every + 4, + 16, + 64, or +256.

10.1.2. TMA Interrupt Event Service

After TMA counter TMAR[7:0] overflowed, interrupt event will be produced and TMAIF[0] will be configured as <1>. If interrupt service is needed at this stage, TMAIE[0] and GIE[0] must be configured as <1>.

- For example, if TMAS[1:0] were configured as <00>, TMAR[7:0] value will change from 00000011b to 00000100b, interrupt event will be generated at this point. When next interrupt event occurred to TMAR[7:0], it will change from 00000111b to 00001000b. If TMAS[1:0] were configured as <10>, TMAR[7:0] value will change from 00111111b to 01000000b, interrupt event will be generated at this point. Next time when interrupt event occurred, TMAR[7:0] value will change from 01111111b to 10000000b.

10.1.3. TMA Initiation

ENTMA[0] is set as <1> to start TMA, TMAR[7:0] begin counting. If it is configured as <0>, TMA will be closed and TMAR[7:0] value will be set as zero.

```
MVL   088h
MVF   INTE1,1,0          ; Configure Timer A Interrupt Service
CLRF  TMAR               ; Configure TMAR as <0>
MVL   0D0h               ; Start Timer A and configure operating frequency as PERA_CK
MVF   TMACN,1,0          ; Configure TMAS[1:0] =01b, the frequency of TMAR overflow is:
                         ; PERA_CK/16 Hz. This is to say, every generated interrupt event is: 1/ (PERA_CK/16) second
TMA Interrupt：          ; Timer A interrupt event service program
BCF   INTF1,TMAIF,0      ; Clear TMA interrupt event flag and TMAR=TMAR+16. Notice: every time TMAR overflowed
                         ; Whether interrupt event service is started TMAR=TMAR+16
          RETI           ; Interrupt service return
```

Example 10-8 TMA Interrupt Event Example Program

10.2. Register Description-TMA

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
26H	INTF1					TMAIF				.000 0000	.000 0000	-,*,*,*,*,*,*,*
41H	TMACN	ENTMA	TMACK	TMAS[1:0]		ENWDT	WDTS[2:0]			0000 0000	0000 0000	*,*,*,* w1,*,*,*
42H	TMAR	TimerA data register								xxxx xxxx	uuuu uuuu	r,r,r,r r,r,r,r

Table 10-32 TMA Control Register

INTE1/INTF1：Please refer to Interrupt Chapter

TMACN：Timer-A Control Register

ENTMA：Timer-A starting controller

1：Start

0：Shutoff；Zero counters

TMACK：Timer-A operating frequency selector

1：F_{TMA}=PERA_CK

0：F_{TMA}=PERA_CK / 256

TMAS[1:0]：Timer-A overflow controller

11：F_{TMA} / 256；Every overflow occurs interrupt event, TMAR[7:0]=TMAR[7:0]+256

10：F_{TMA} / 64；Every overflow occurs interrupt event, TMAR[7:0]=TMAR[7:0]+64

01：F_{TMA} / 16；Every overflow occurs interrupt event, TMAR[7:0]=TMAR[7:0]+16

00：F_{TMA} / 4；Every overflow occurs interrupt event, TMAR[7:0]=TMAR[7:0]+4

TMAR：TMA ascending counter, readable but not writable

11. Timer-B

Timer B is designed in 8-bit frame, supporting Capture/Compare Mode and can be operated in Run Mode and Idle Mode.

- Can configure 8-bit or 16-bit counter
- Supporting Capture Mode
- Supporting Compare Mode
- Ascending counter
- Overflow generated interrupt event
- Readable/writable TMBR counter value

TMB Registers：

TMBCN ENTMB[0], TMBCK[0],TMBS[1:0],TMBSYC[0],TMBR2R[0]

TMBR[15:0] TMBRH[7:0], TMBRL[7:0]

INTE1 TMBIE[0]

INTF1 TMBIF[0]

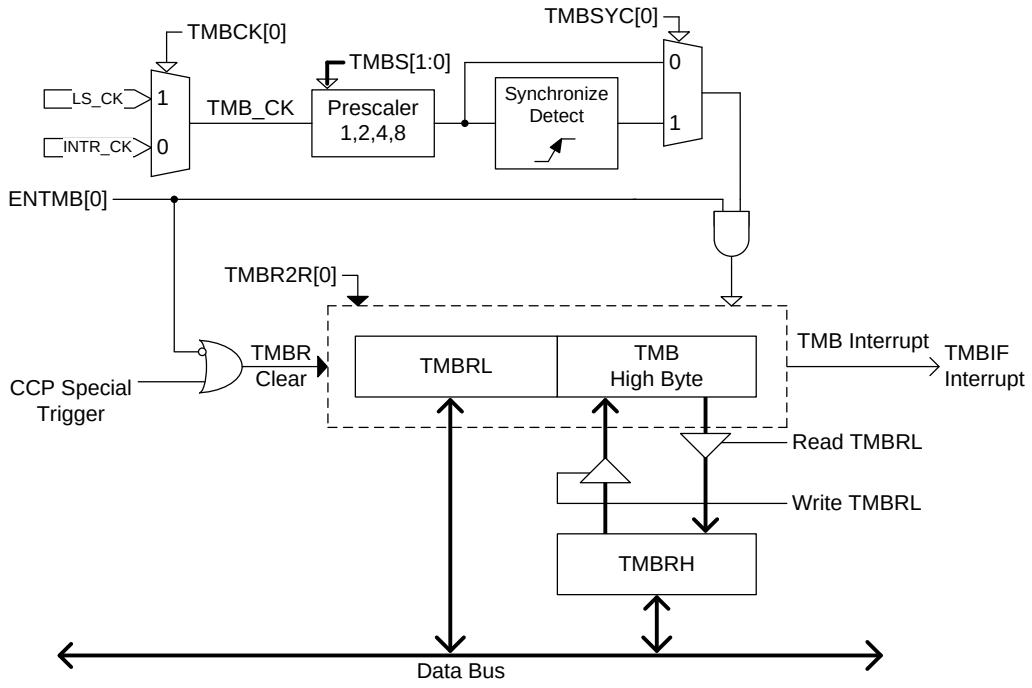


Figure 11-18 Timer-B Block Diagram

11.1. Timer-B Manual

11.1.1. TMB Initial Configuration

TMB operating frequency can be configured by operating frequency selector, TMBCK[0] [11]. It can configure TMB_CK to be INTR_CK or LS_CK operating frequency. Different configured frequencies enable TMB to be operated in Run Mode and Idle Mode.

The configuration of prescale controller, TMBS[1:0] can prescale TMB_CK. When it applies in supporting Capture/Compare Mode, prescale configuration will impact the speed of event execution.

Counter, TMBR[15:0] [12] can be configured through TMBR2R[0]. When the value is <0>, it is 8-bit counter. If TMBR2R[0] is configured as <1>, it is 16-bit counter. Using 16-bit TMBR[15:0] counter to read/write TMBRH[7:0] and TMBRL[7:0] data must follow the regulations below：

- When writing data, TMBRH[7:0] must be written first then to write TMBRL[7:0], so data can be accurately written. ENTMB[0] must be configured as <1> before writing data into counter. If ENTMB[0] is configured as <0>, writing is ineffective.
- When reading data, TMBRL[7:0] must be read first then to read TMBRH[7:0] in order to get correct counter value.

11.1.2. TMB Interrupt Event Service

After TMBR[15:0] ascending overflowed, interrupt event will be generated TMBIF[0] is configured as <1>. If interrupt event service is needed, TMBIE[0] and GIE[0] has to be set as <1>.

No matter TMB is 8-bit or 16-bit, interrupt event will be produced when counter overflowed. At this time, TMBIF[0] is set as<1>.

11.1.3. TMB Initiation

ENTMB[0] is set as <1> to start TMB and TMBR[15:0] start counting；It is configured as <0>, TMB will be shut off and TMBR[15:0] counter value will be zeroed automatically.

[11] If Timer B and CPU is operated in different clock sources, TMBSYC can be configured 1 in order to get synchronous frequency process. By so doing, it can prevent Timer-B from miss counting.

[12] Presumed that TMBR adopts 16-bit operation, TMBR[15:0]=00F0Fh. After Timer B started, overflow signal will be generated from 0x0F0Fh to 0xFFFFh for the first time only. After that, overflow signal is generated from TMBR[15:0], from 0x0000h to 0xFFFFh. Same situation happens when it is in 8-bit operation. When writing TMBRL[7:0]=0x0Fh, after Timer B started, overflow signal will be generated from 0x0Fh to 0xFFh for the first time only. After that, overflow signal is generated from TMBRL[7:0], from 0x00h to 0xFFh.

```
MVL   090h
MVF   INTE1,1,0     ; Configured Timer B interrupt service
MVL   08Ch
MVF   TMBCN,1,0     ; Start TMB and set operating frequency as CPU_CK and do not prescale,
                      TMBS[1:0] = 00b, no need to do synchronous process. TMBR is set as 16-bit mode
CLRF  TMBRH         ;
MVL   00Fh          ; Set TMBR to start counting from 0Fh, TMBRH must be written first, then TMBRL register
MVF   TMBRL,1,0     ; TimerB ceased time is around CPU_CK/ (FFF0h) Hz
....
MVF   TMBRL,0,0     ; Read TMBR counter value and save it in BUF0 and BUF1 register,
                    ; TMBRL must be read first, then TMBRH data
MVF   BUF0,1,0      ;
MVF   TMBRH,0,0     ;
MVF   BUF1,1,0
....
TMB Interrupt :     ; TMB interrupt event service program
CLRF  TMBRH         ; TMBR is configured from 000Fh to start counting. Please note the correlation between
                              instruction execution cycle and TMB_CK
MVL   00Fh          ; Otherwise it will occur counting bias problems
MVF   TMBRL,1,0     ;
BCF   INTF1,TMBIF,0 ; Clear TMB interrupt event flag
....
```

Example 11-9 Timer-B Interrupt Event Example Program

11.2. Register Description-TMB

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
26H	INTF1				TMBIF					.000 0000	.000 0000	-,*,*,*,*,*,*,*
43H	TMBCN	ENTMB	TMBCK	TMBS[1:0]		TMBSYC	TMBR2R			0000 00..	0000 00..	*,*,*,*,*,-,-
44H	TMBRH	TimerB High Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
45H	TMBRL	TimerB Low Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 11-33 TMB Control Register

INTE1/INTF1：Please refer to Interrupt Chapter

TMBCN：Timer-B Control Register

ENTMB：Timer-B starting controller

1：Start

0：Shutoff: clear zero counter value

TMBCK：Timer-B operating frequency select controller

1：LS_CK

0：INTR_CK

TMBS[1:0]：Timer-B operating frequency prescaler

11：TMB_CK/8

10：TMB_CK/4

01：TMB_CK/2

00：TMB_CK/1

TMBSYC：Timer-B operating frequency and CPU synchronous process controller

1：Synchronous process

0：Do not synchronous process

TMBR2R：TMBR register operating way

1：16-bit operating mode

0：8-bit operating mode

TMBRH/L：Timer-B ascending counter that can configure overflow value. It is readable/writable

TMBRH[7:0]

TMBRL[7:0]

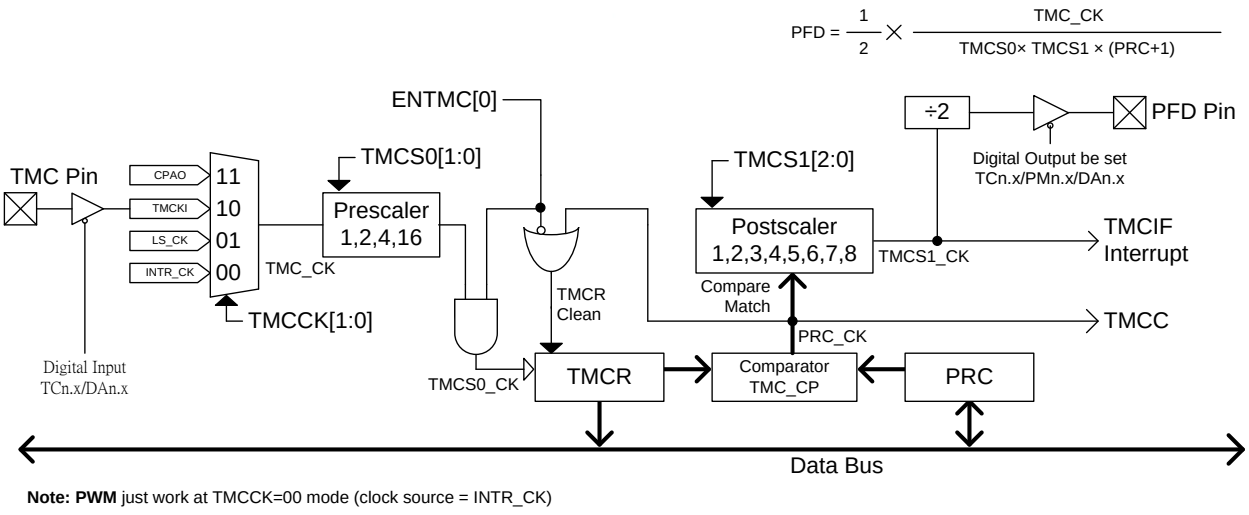
12. Timer-C

Timer C is designed in 8-bit frame. TMC counter is composed by two value registers and one comparator. The overflow event is generated by postscaler. It can be operated in Run Mode, Idle Mode and Sleep Mode.

- Equip with 8-bit frequency controller, value comparator and counter
- Ascending counter
- Value comparator
- Supporting PWM function
- Supporting PFD function
- Plan overflow value
- Overflow generated interrupt event

Timer-C Registers：

TMCCN	ENTMC[0], TMCCK[1:0],TMCS1[2:0],TMCS0[1:0]
TMCR	TMCR[7:0]
PRC	PRC[7:0]
INTE1	TMCIE[0]
INTF1	TMCIF[0]



Note: PWM just work at TMCCK=00 mode (clock source = INTR_CK)

Figure 12-19 Timer-C Block Diagram

12.1. Timer-C Manual

12.1.1. TMC Initial Configuration

TMC operating frequency is configured through operating frequency selector, TMCCK [1:0]. It can set TMC_CK as INTR_CK[13], LS_CK, TMCKI[14] or CPAO operating frequency. TMC can be operated under Run, Idle or Sleep Mode by configuring diverse frequency.

Configuration of prescale controller, TMCS0[1:0] will prescale TMC_CK to generate TMCS0_CK. The configuration of overflow controller, TMCS1[2:0] will prescale PRC_CK and TMCS1_CK will be produced.

Counter, TMCR[7:0] [15], frequency controller, PRC[7:0] and value comparator, TMC_CP compose PRC_CK signal generated scheme. That is, when TMCR[7:0] and PRC[7:0] register content is the same, PRC_CK signal will be aroused to postscaler, TMCS1[2:0].

12.1.2. TMC Interrupt Event Service

PRC_CK, after postscaler, will generate overflow interrupt event as TMCS1[2:0] configured condition is satisfied. TMCIF[0] is configured as <1>. If interrupt event service is required, TMCIE[0] and GIE[0] must be set up as <1>.

12.1.3. Timer-C Initiation

Configuring ENTMC[0] as <1> can start TMC and TMCR[7:0] will begin counting. If it is configured as <0>, TMC will be shut off and TMCR[7:0] counter value will be zeroed automatically. Thus, in order to obtain correct value, users must first write the value in PRC[7:0] then starting TMC.

```
....
MVL   0A0h
MVF   INTE1,1,0          ; Configure Timer C interrupt service
MVL   01Fh               ; Write 01Fh into PRC
MVF   PRC,1,0            ; Interrupt frequency is around: INTR_CK/(1Fh x 2h)
MVL   084h               ; Start Timer C configured operating frequency
MVF   TMCCN,1,0          ; Operating frequency is INTR_CK, not prescale but postscale, configure TMCS1[2:0] = 001b
....
TMC Interrupt :          ; TMC interrupt event service program
BCF   INTF1,TMCIF,0      ; Clear TMC interrupt event flag
```

Example 12-10 Timer-C Interrupt Event Example Program

13 When using PWM peripherals, TMC operating frequency must be configured as this argument.

14 When input frequency choses external TMCKI, I/O pin must be configured correctively, otherwise the signal may not be entered and will bring about abnormal execution. For detail register description, please refer to ***Input/Output Port, I/O*** Chapter.

15 TMCR[7:0] only can be read, any writing action to TMCR[7:0] or TMCCN[7:0] will be deemed as zeroing counter, TMCR[7:0] and the value of prescaler and postscaler will be zeroed as well.

12.2. Register Description-TMC

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
26H	INTF1			TMCIF						.000 0000	.000 0000	-,*,*,*,*,*,*,*
46H	TMCCN	ENTMC	TMCCK[1:0]		TMCS1[2:0]			TMCS0[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
47H	PRC	TimerC programmable register								1111 1111	1111 1111	*,*,*,*,*,*,*,*
48H	TMCR	TimerC register								0000 0000	0000 0000	r,r,r,r r,r,r,r

Table 12-34 TMC Control Register

INTE1/INTF1：Please refer to Interrupt Chapter

PT2/TRISC2/PT2DA/PT2PU：Please refer to Input/Output Port, I/O Chapter

TMCCN：Timer-C Control Register

ENTMC：Timer-C starting control bit

1：Start

0：Shutoff: clear zero counter value

TMCCK[1:0]：TMC operating frequency select controller

11：CPAO；This configuration does not support PWM peripheral circuits

10：TMCKI；This configuration does not support PWM peripheral circuits

01：LS_CK

00：INTR_CK

TMCS1[2:0]：Timer-C counter overflow controller

111：PRC_CK/8

110：PRC_CK/7

101：PRC_CK/6

100：PRC_CK/5

011：PRC_CK/4

010：PRC_CK/3

001：PRC_CK/2

000：PRC_CK/1

TMCS0[1:0]：Timer-C operating frequency prescaler

11：TMC_CK/16

10：TMC_CK/4；This configuration support partial PWM peripheral circuits, please refer to 14 Frequency Generator, PWM/PFD description

01：TMC_CK/2；This configuration support partial PWM peripheral circuits, please refer to 14 Frequency Generator, PWM/PFD description

00：TMC_CK/1；This configuration support partial PWM peripheral circuits, please refer to 14 Frequency Generator, PWM/PFD description

TMCR：Timer C Counter

Ascending counter of Timer-C that is can only read. Any write in action to TMCR[7:0] or TMCCN[7:0] will be deemed as zeroing TMCR[7:0].

PRC：Frequency Control Register

Frequency controller of Timer-C, TMC_CP will contrast the content of TMCR[7:0] and PRC[7:0], when the value is in equality, reversed PRC_CK status.

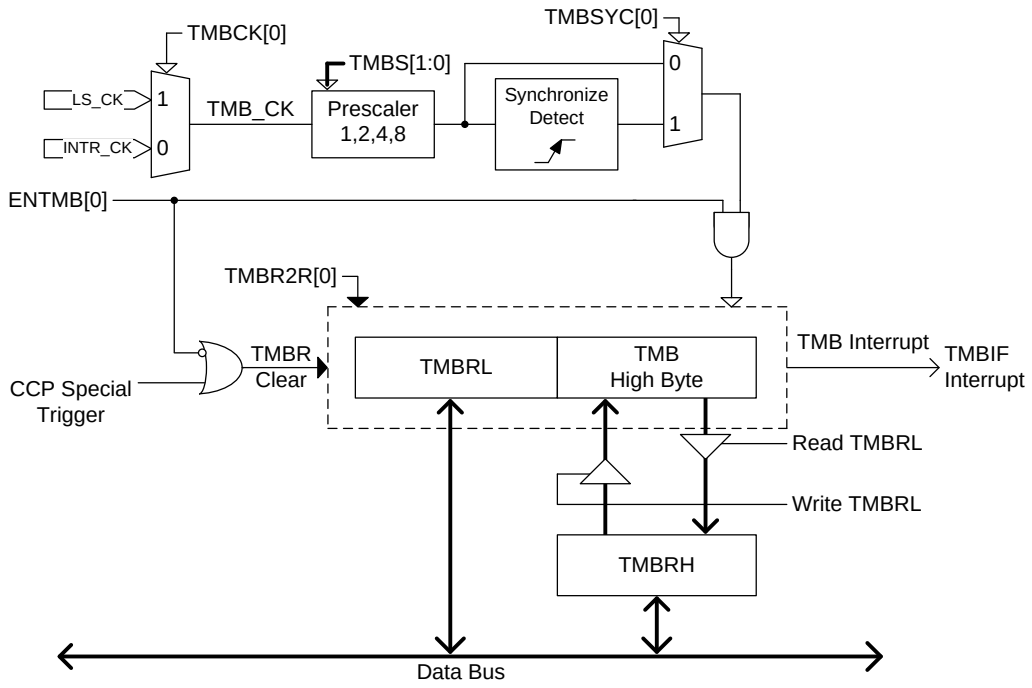
13. Capture/Compare Mode

Capture/compare mode, CCP is designed in pairs. When using Capture Mode, the Compare Mode is banned, and vice versa. Both Capture and Compare Mode are schemed out in two set, CCP0 and CCP1

- Must be used with Timer-B
- Multiplexed Capture/Compare Mode
- Able to generate interrupt event
- 16-bit data register

CCP Registers：

TMBCN	ENTMB[0], TMBCK[0],TMBS[1:0],TMBSYC[0],TMBR2R[0]
TMBR[15:0]	TMBRH[7:0],TMBRL[7:0]
CCPCN	CCP1M[3:0], CCP0M[3:0]
CCPxR[15:0]	CCP0RH[7:0],CCP0RL[7:0],CCP1RH[7:0],CCP1RL[7:0]
INTE2	CCP0IE[0],CCP1IE[0]
INTF2	CCP0IF[0],CCP1IF[0]



Timer-B Block Diagram

13.1. Capture Mode Manual

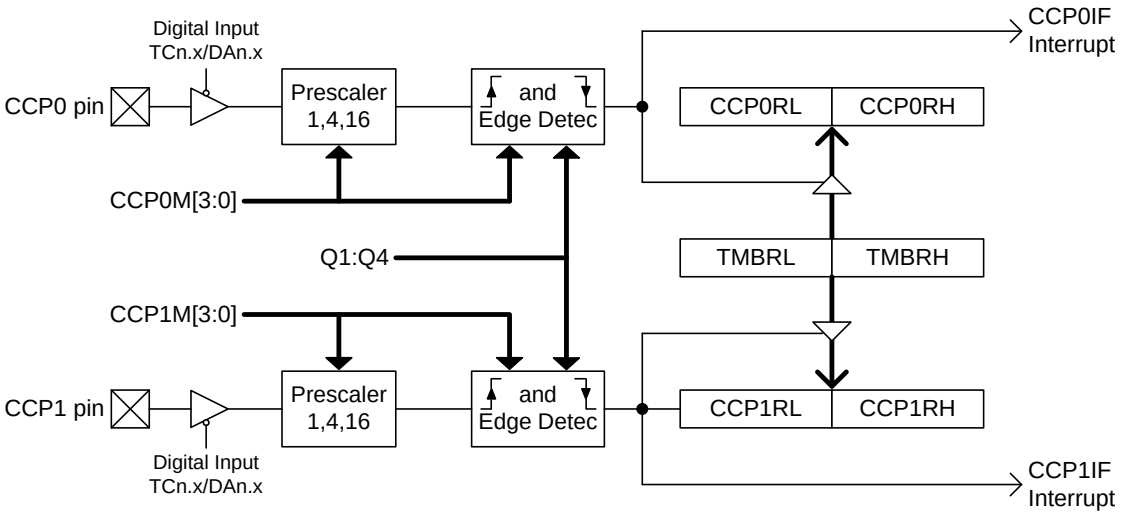


Figure 13-20 Capture Mode Block Diagram

Capture Mode captures event through external input CCPx pin. It can request interrupt service and 16-bit data register, CCPxR[15:0] to record the occurred results.

13.1.1. Capture Event Initial Configuration

Timer-B must be started first, related configuration please refer to Timer-B for detail description.

CCP event controller, CCPxM[3:0][16] can decide the occurred condition of CCP0 and CCP1 pin. The event conditions are：

- Rising edge and falling edge of every 1 signal
- Rising edge of every 4, 16 signal

CCP data register, CCPxR[15:0] is composed by CCPxRH[7:0] and CCPxRL[7:0][17]. As capture event established, TMBR[15:0] data will be moved to CCPxR[15:0] and interrupt signal will be produced, CCPxIF[0] is set up as <1>. After interrupt event happened, CCPxR[15:0] data must be read out, otherwise the data will be covered by the newly captured data of next interrupt event.

13.1.2. Capture Interrupt Event Service

When capture event condition established, capture event will generate settled signal, CPPxIF[0] is set as <1>. At this time, if interrupt event is required, CCPxIE[0] as well as GIE[0] must be configured as <1>.

16 In Capture Mode, PORT relative configurations must be set up accurately or it may result in signal input inability and abnormal executing function. For detailed register description, please refer to ***Input/Output Port, I/O*** chapter.

17 The acquired data of first established event in Capture Mode may be incorrect. Users are suggested to discard the first data.

13.1.3. Capture Mode Initiation

After CCPxM[3:0] configured capture event condition, Capture Mode is enabled automatically. Conversely, Capture Mode will close automatically if CCPxM[3:0] is configured in non-capture event.

```
0RG 04h                    ; Interrupt service
BTSZ     INTF2,CCP0IF
JMP      CCP0 interrupt
BTSZ     INTF2,CCP1IF
JMP      CCP1 interrupt
….                         ; Configure CCPx interface as digital input pin
….                         ; PORT relative configuration DAn.x, TCn.x
MVL      001h
MVF      CONT0,1,0         ; Set Capture event initial discard numbers
MVL      003h
MVF      INTE2,1,0         ; Configure CCPxIE interrupt service
MVL      084h              ; Start Timer-B and configure operating frequency as CPU_CK
                             and do not prescale
MVF      TMBCN,1,0         ; TMBS[1:0] = 00b, no need to do synchronic process and set
….                           TMBR as 16-bit operating way
MVL      066h
MVF      CCPCN,1,0         ; Set Capture event as every 4 rising edge
….
CCP0 Interrupt：           ; CCP0 interrupt event service program
BCF      INTF2,CCP0IF,0    ; Clear CCP0 interrupt event flag
MVFF     CCP0RH,BUF0       ; Safe the captured value in BUFx
MVFF     CCP0RL,BUF1
….
RETI                       ; Interrupt service return
CCP1 Interrupt：           ; CCP1 interrupt event service program
BCF      INTF2,CCP1IF0     ; Clear CCP1 interrupt event flag
MVFF     CCP1RH,BUF0       ; Safe the captured value to BUFx
MVFF     CCP1RL,BUF1
….
```

Example 13-11 Capture Event Example Program

13.2. Compare Mode Manual

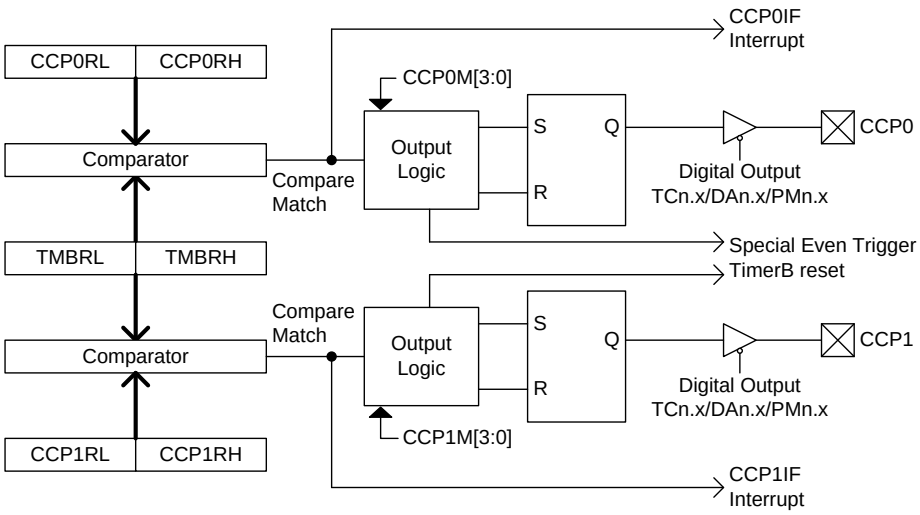


Figure 13-21 Compare Mode Block Diagram

In Compare Mode, comparator will contrast CCPxR[15:0] and TMBR[15:0] content. When two registers value equals to each other, compare event established signal will be set up.

13.2.1. Compare Event Initial Configuration

Timer-B must be initiated first, relative configuration please refer to Timer-B description.

CCP data register, CCPxR[15:0] constitutes CCPxRH[7:0] and CCPxRL[7:0][18]. Comparator will carry out ongoing comparison with the content of TMBR[15:0]. As both side data is consistent, compare event established signal will be aroused, CCPxIF[0] will be set as <1>. Event controller, CCPxM[3:0][19] can configure CCP0 and CCP1 pin output status when compare event established. There are several possible configurations：

- Pin outputs high logic level, low logic level
- Pin outputs reversed potential, remains unchanged

13.2.2. Compare Interrupt Event Service

When compare event occurred conditions are settled, the content of CCPxR[15:0] corresponds with TMBR[15:0] , comparator will produce compare event established signal, CPPxIF[0] will be configured as <1>. At this moment, if interrupt event is required, CCPxIE[0] and GIE[0] should be set as <1>.

[18] Please be noted that TMBR is an accumulated value counter. Thus, CCPxR value must be written first. After that, Timer-B can be started as to avoid first time compare event established condition error.

[19] Under Capture Mode, correct configuration of PORT relative setup is a must; any mis-setup will bring about abnormal signal and function. For detail register description, please refer to ***Input/Output Port, I/O*** Chapter.

13.2.3. Compare Mode Initiation

After CCPxM[3:0] configured the condition of compare event output pin, Compare Mode starts automatically. Reversely, when CCPxM[3:0] is configured in non-comparison event output pin, the Compare Mode will close automatically.

```
0RG 04h                  ; Interrupt service
BTSZ    INTF2,CCP0IF
JMP     CCP0 interrupt
BTSZ    INTF2,CCP1IF
JMP     CCP1 interrupt
....                     ; Set PTn.x as digital output interface and outputs CCPx signal
....                     ; PORT relative configuration DAn.x, TCn.x, PMn.x
MVL     003h
MVF     INTE2,1,0        ; Configure CCPxIE interrupt service
MVL     099h             ; Configure compare event operation way.
                           Initial CCPx interface is high logic level
MVF     CCPCN,1,0        ; Event established will generate interrupt event and
                           the interface is set up as low logic level
....                     ;
MVFF    BUF0,CCP0RH      ; Put the value in CCPxR value register
MVFF    BUF1,CCP0RL      ;
MVFF    BUF0,CCP1RH      ;
MVFF    BUF1,CCP1RL      ;
MVL 084h
MVF TMBCN,1,0            ; Start Timer-B configuration operating frequency is CPU_CK
                           and do not pre divide frequency
....                     ; TMBS[1:0] = 00b, no need to conduct synchronic and configures
                           TMBR as 16-bit operating way

CCP0 Interrupt :         ; CCP0 interrupt event service program
BCF  INTF2,CCP0IF,0      ; Clear CCP0 interrupt event flag
....
RETI                     ; Return from interrupt service
```

Example 13-12 Compare Event Example Program

13.3. Register Description-Capture/Compare

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
24H	INTE2							CCP1IE	CCP0IE	00.. 0000	00.. 0000	*,*,-,-,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*
43H	TMBCN	ENTMB	TMBCK	TMBS[1:0]		TMBSYC	TMBR2R			0000 00..	0000 00..	*,*,*,*,*,-,-
44H	TMBRH	TimerB High Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
45H	TMBRL	TimerB Low Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
49H	CCPCN	CCP1M[3:0]				CCP0M[3:0]				0000 0000	0000 0000	*,*,*,*,*,*,*
4AH	CCP0RH	CCP0 High Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
4BH	CCP0RL	CCP0 Low Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
4CH	CCP1RH	CCP1 High Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
4DH	CCP1RL	CCP1 Low Byte data register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
74H	PT2			PT2.5	PT2.4					xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*
76H	PT2DA			DA2.5	DA2.4					0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*
79H	PT2M2	PM2.7[1]	PM2.7[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	00.0 0000	00.0 0000	*,*,-,*,*,*,*

Table 13-35 CCP Control Register

INTE1/INTE2/INTF2：Please refer to **Interrupt** Chapter

TMBCN/TMBRH/TMBRL：Please refer to **Timer-B** Chapter

PT2/TRISC2/PT2DA/PT2PU：Please refer to **Input/Output Port, I/O** Chapter

CCPCN：**Capture/Compare Control Register**

CCPxM[3:0]：CCP function select control bit

11xx：Unused

1011：Compare Mode. Event established, CCPxIF[0] is configured as <1> and Timer-B is zeroed.

1010：Compare Mode. Event established, CCPxIF[0] is configured as <1>, no signal will be sent to CCPx pin.

1001：Compare Mode. CCPx pin initiation is high logic level. Event established, CCPxIF[0] is set as <1> and CCPx pin becomes low logic level.

1000：Compare Mode. CCPx pin initiation is low logic level. Event established, CCPxIF[0] is set as <1> and CCPx pin becomes high logic level.

0111：Capture Mode. Capture condition is every 16 rising edge, event established and CCPxIF[0] is set as <1>.

0110：Capture Mode. Capture condition is every 4 rising edge, event established and CCPxIF[0] is set as <1>.

0101：Capture Mode. Capture condition is 1 rising edge, event established and CCPxIF[0] is set as <1>.

0100：Capture Mode. Capture condition is 1 falling edge, event established and CCPxIF[0] is set as <1>.

0011：Unused

0010：Compare Mode. Event established, CCPxIF[0] is configured as <1> and CCPx pin outputs reversed logic level.

0001：Unused

0000：Close Capture/Compare Mode function

CCPxRH/L：**Capture/Compare Mode Value Register**

CCPxRH[7:0]

CCPxRL[7:0]

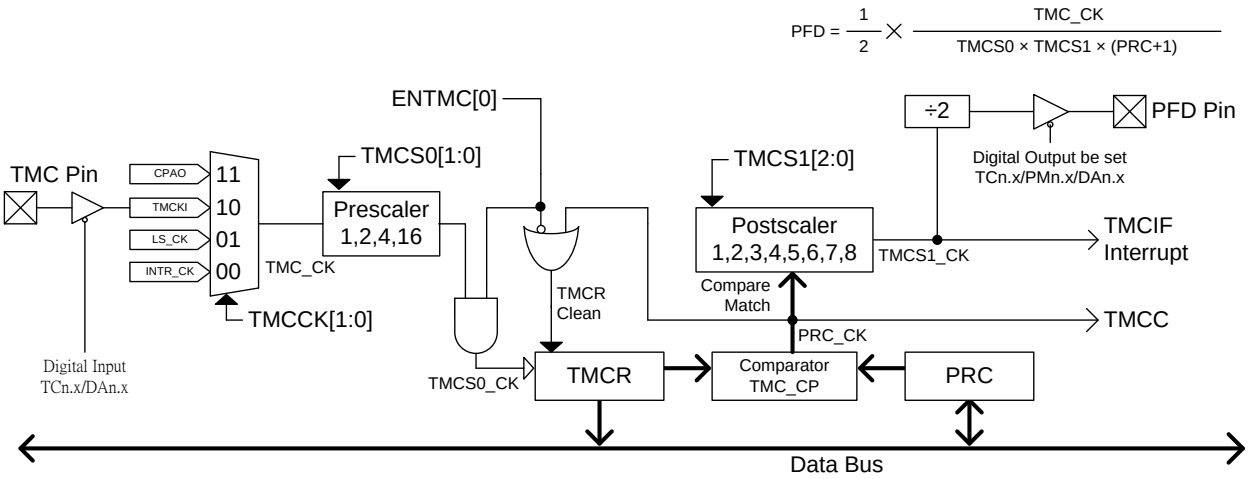
14. Frequency Generator, PWM/PFD

Frequency generator contains two modes, one is Pulse Width Modulation, PWM, and the other is Pulse Divider, PFD.

- Must be used in collocation with Timer-C
- Multiplexed PWM and PFD Mode
- 10-bit frequency controller

PFD and PWM Registers：

TMCCN	ENTMC[0], TMCCK[1:0],TMCS1[2:0],TMCS0[1:0]
TMCR	TMCR[7:0]
PRC	PRC[7:0]
PASC	PASF[0],PASCF[2:0],PSSCN0[1:0], PSSCN1[1:0]
PWMCN	ENPWM[0], ENPFD[0],PWMRL[1:0],PWMCG[1:0],PWMM[1:0]
PDBD	ENPRS[0], DBDC[6:0]
PWMR[9:0]	PWMRH[7:0] ,PWMRL[1:0]



Note: PWM just work at TMCCK=00 mode (clock source = INTR_CK)

Timer-C Block Diagram

14.1. PFD Mode Manual

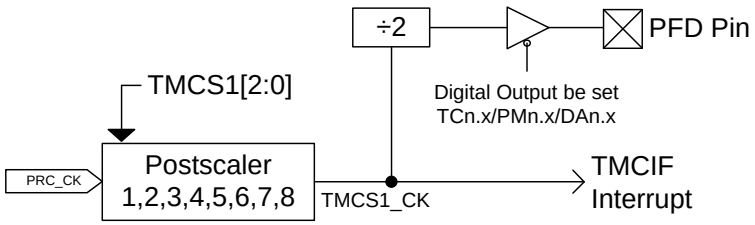


Figure 14-22 PFD Block Diagram

Frequency modulation, PFD Mode must be utilized with Timer-C supportability; configuration of output frequency modulation must through Timer-C. PFD frequency output must be set as digital output and the output signal must select PFD.

14.1.1. PFD Mode Initial Configuration

Timer-C has to be started first, relative configuration please refer to Timer-C description.

PFD operating frequency is PRC_CK, through TMC overflow controller configuration, TMCS1[2:0], can change PFD pin[20] output frequency, PFD frequency modulation formula is as Equation 14-1.

$$PFD = \frac{1}{2} \times \frac{TMC\_CK}{TMCS0 \times TMCS1 \times (PRC+1)}$$

PFD Modulation Frequency Equation

14.1.2. PFD Mode Initiation

Configure ENPFD[0] as <1> to start PFD Mode. When ENPFD[0] is set as <0> will shut off PFD Mode.

```
....                ; Configure PTn.x as digital output interface and output PFD signal
....                ; PORT relative configuration DAn.x, TCn.x, PMn.x
MVL 01Fh            ; Write 01Fh to PRC
MVF PRC,1,0         ; Start Timer C, configure operating frequency as
MVL 084h            ; INTR_CK, do not prescale but postscale configured TMCS1[2:0] = 001b
MVF TMCCN,1,0       ; So PFD frequency is: INTR_CK/(1Fh x 2h)
BSF PWMCN,6,0       ; Start PFD
....
```

Example 14-13 PFD Output Example Program

[20] PFD Mode must correctly configure I/O PORT setup while using or the signal may not be output and PFD Mode will have abnormal function. Detailed description please refers to ***Input/Output Port,* I/O** Chapter.

14.2. PWM Mode Manual

Pulse width modulation, PWM equips with the following functions：

- PWM Output
  - Single output
  - Dual output
  - Quad-output
- Must have Timer-C support
- Automatic shutoff and start
- Can program dead zone delay function

14.2.1. PWM Single Output Manual

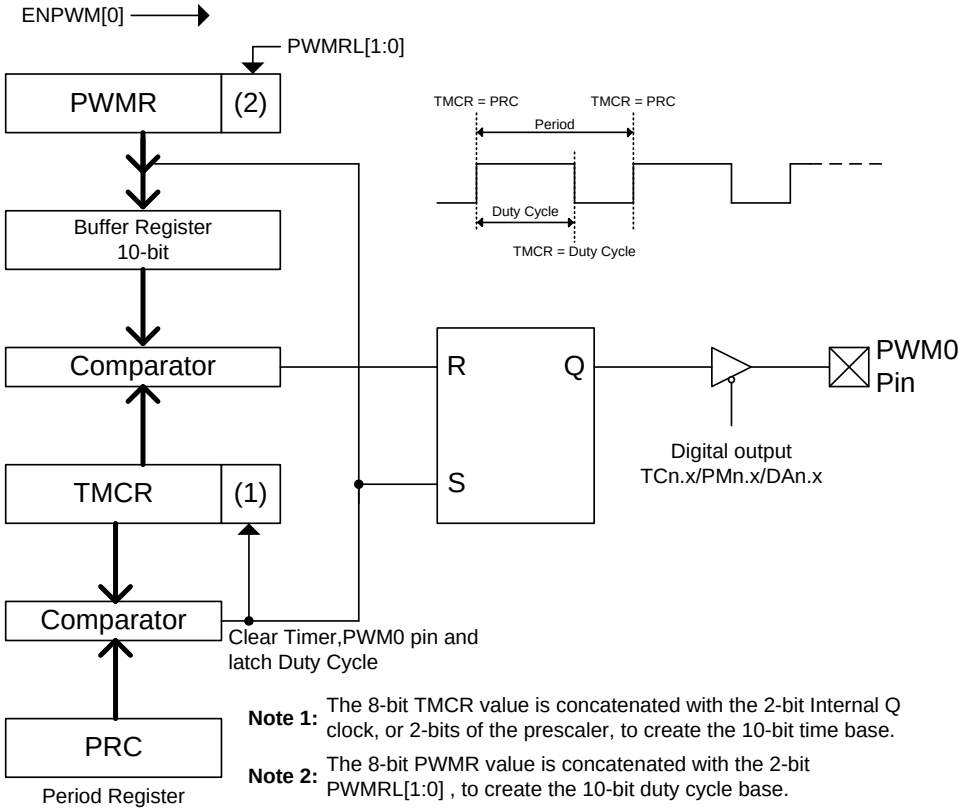


Figure 14-23 PWM Single Output Block Diagram

PWM single output pulse width modulation signal has single output PWM0 pin[21].

Before using, Timer-C must be configured first to scheme PWM frequency and duty cycle.

21 PWM Mode must correctly configure PORT related configuration while using, or the signal may not be output and PFD Mode will have abnormal function. Detailed register description please refer to ***Input/Output Port, I/O*** chapter.

14.2.1.1. PWM Single Output Initial Configuration

14.2.1.1.1. Frequency (Period) Configuration

Frequency controller, PRC[7:0][22] has 8-bit length, change its configuration parameter can determine PWM period (frequency). The equation is shown in Equation 14-2：

Equation 14-2 (a)

$$\text{PWM Period} = \frac{\text{TMCS0} \times (\text{PRC}+1)}{\text{TMC\_CK}}$$

PWM Period Equation

Equation 14-2(b)

$$\text{PWM Frequency} = \frac{1}{\text{PWM Period}}$$

PWM Frequency Equation

14.2.1.1.2. Duty Cycle Configuration

Pulse width modulation controller, PWMR[9:0][23] has 10-bit length and is composed by PWMRH[7:0] and PWMRL[1:0]. PWM's pulse width is determined by changing its configuration parameters. As Equation 14-3 shown.

Equation 14-3

$$\text{PWM Duty Cycle} = \frac{\text{TMCS0} \times (\text{PWMR[9:0]} + 4)}{\text{TMC\_CK} \times 4}$$

PWM Duty Cycle Equation

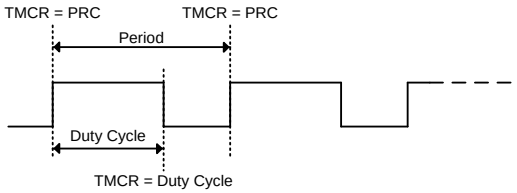


Figure 14-24 PWM Waveform Chart

[22] When the value of TMCR[7:0] and PRC[7:0] is equal, it will bring about two results：a. PWMx output pin will be configured as 1. (If PWMR[9:0]=0，PWMx output pin will not be set as 1). b. TMCR will be cleared zero.

[23] When PWM is in operating status, PWMR will show the written value until the end of this period. If PWMR value is bigger than PRC value, PWMx output pin will not be configured as 0.

14.2.1.1.3. Resolution Configuration

TMC operating frequency, TMC_CK and PWM output frequency must be taken into account while computing PWM greatest resolution, as shown in Equation 14-4.

Equation 14-4

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{\text{TMC\_CK} \times 4}{\text{TMCS0} \times \text{PWM Frequency}}\right)}{\log(2)}$$

PWM Resolution Equation

14.2.1.1.4. Timer-C Configuration

PWM must adopt Timer-C to generate frequency and duty cycle. Therefore, when Timer-C supports PWM Mode, its operating frequency selector, TMCCK[1:0] and prescaler, TMCS0[1:0] configuration will be restrained.

- If CPU operating frequency equals to or heightens than 2MHz, PWM's operating frequency selector, TMCCK[1:0] and prescaler, TMCS0[1:0] configuration is as follows:
  - In order to provide operating frequency, TMC for PWM utilization, TMCCK[1:0] can only be configured as <00> through INTR_CK or to be configured as <01> through LS_CK.
  - 4 prescaling arguments of TMCS0[1:0] configuration can be used.
- If CPU operating frequency equals to or lowers than 28KHz, PWM's operating frequency selector, TMCCK[1:0] and prescaler, TMCS0[1:0] configuration is as follows:
  - Configure TMCCK[1:0] as <00>, INTR_CK provides operating frequency to TMC. Under this configuration, 4 prescaling arguments of TMCS0[1:0] configuration can be used.
  - Configure TMCCK[1:0] as <01>, LS_CK provides operating frequency to TMC. Under this configuration, only TMCS0[1:0] <11> is applicable.

Detailed description please refers to **Register Description-TMC** Chapter.

14.2.1.2. PWM Single Output Initiation

Start controller, ENPWM[0] is configured as <1> to initiate PWM Mode. When ENPWM[0] is set as <0>, PWM Mode will be shut off.

14.2.2. PWM Dual and Quad-Output Manual

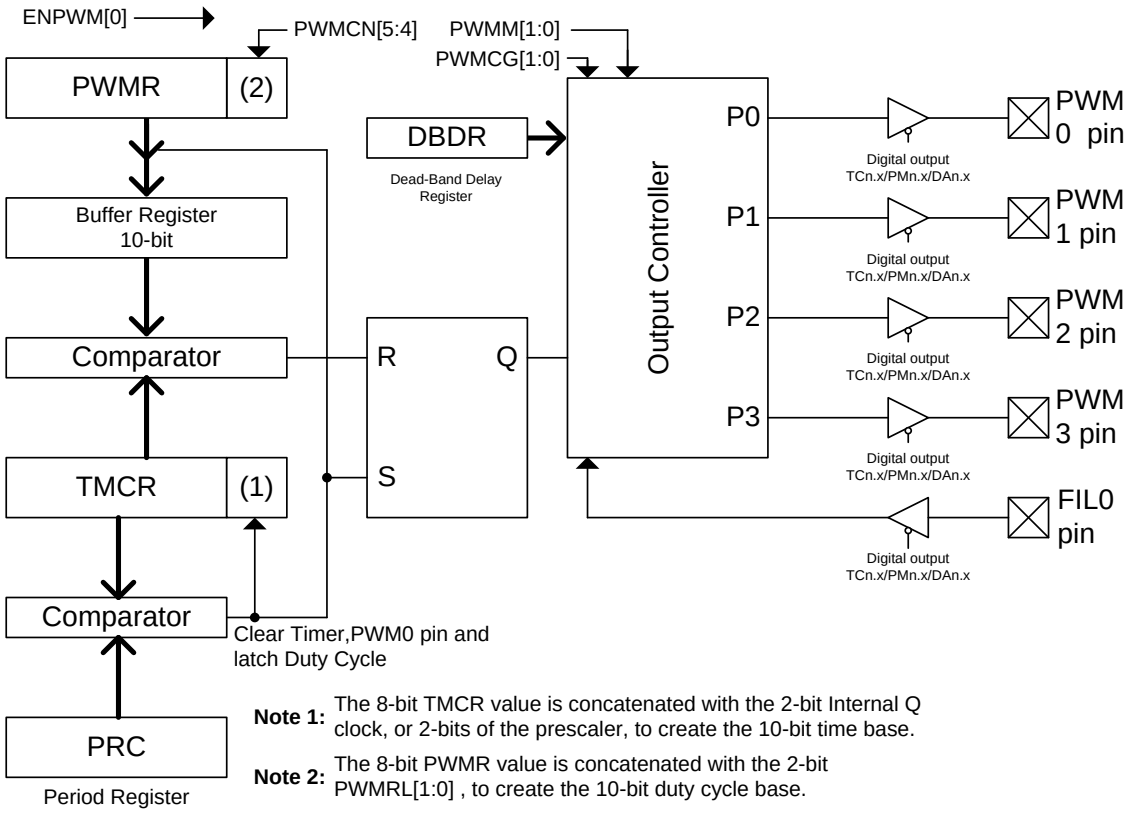


Figure 14-25 PWM Dual and Quad-Output Block Diagram

Dual and Quad-output pulse width modulation signal has PWM0, PWM1 and PWM0, PWM1, PWM2, PWM3 pin combination. Before using, Timer-C must be configured first to scheme PWM frequency and duty cycle.

14.2.2.1. PWM Dual and Quad-Output Initial Configuration

14.2.2.1.1. PWM Dual and Quad-Output Initial Configuration

Please refer to the way of PWM Single Output Initial Configuration.

14.2.2.1.2. PWM Dual and Quad-Output Auto Shutoff and Start Configuration

Dual and Quad-output PWM has auto shutoff and start function, it is mainly used in auto stop or start PWM. This function allows users to define auto shutoff trigger events, pin out status and auto start conditions after auto shutoff.

When PWMx pin is in auto shutoff event status and it detects the condition that matches to auto start, PWM will output signal in next period. Detailed description, please refer to **Auto Shutoff and Start** Chapter.

14.2.2.2. PWM Dual and Quad-Output Set Configuration

PWM output set configuration can be separated into two parts：

- One is PWM output controller, PWMCG[1:0]. It can be configured as single output, dual output or quadric output, as shown in Figure 14-6.
- The effective logic level selector of the other output pin, PWMM[1:0] can configure the output signal to be effective high or low logic level, as presented in Figure 14-6.

14.2.2.3. Dead Band Delay time

Dead band delay primarily is adopted to prevent huge current that resulted from sudden short power of MOS switch. Dead band delay controller, DBDC[6:0] can be configured as to determine the delay time. Detailed descriptions please refer to **Dead Band Delay Description** Chapter.

14.2.2.4. PWM Dual and Quad-Output Initiation

Configure starting controller, ENPWM[0] as <1> to initiate PWM Mode. Reversely, when ENPWM[0] is set as <0>, PWM Mode will be shut off.

14.2.3. Auto Shutoff and Start

Auto shutoff trigger event selector, PASCF[1:0] can configure auto shutoff trigger source as CPAO and FIL0 or CPAO and FIL0. When PASCF[1:0] is set as <00>, auto shutoff and start function will be closed. If the value is not configured as 00, auto shutoff and start function will enabled.

After start, if auto shutoff trigger event is established, PWMx pin will accord with pin controller, PASF[0] configuration, as shown in Figure 14-5.

Pin out definition controller, PSSCN0[1:0] and PSSCN1[1:0] can configure PWMx pin to have six different outputs that composed of high state, low state and other combination state, as shown in Figure 14-6.

After auto shutoff, ENPRS[0] can set PWM auto start trigger event through automatic hardware or user configuration.

- Configure ENPRS[0] as <1>, after auto shutoff trigger event established, hardware will place PASF[0] as <1> and place PASF[0] as <0> later. Configure ENPRS[0] as <0>, after auto shutoff trigger event established, hardware will place PASF[0] as <1> and later, user may configure PASF[0] as <0>[24] through software, as Figure 14-5 illustrates.
- When PASF[0] is configured as <1>, PWMx pinout change from PWM regular modulation output to auto shutoff pinout definition. If PASF[0] is set as <0>, PWMx

[24] Users must be cautious that if auto shut off trigger event is effective, PASF[0] is forbid writing.

pinout will change from auto shutoff pinout definition to PWM regular modulation in next period, as Figure 14-5.

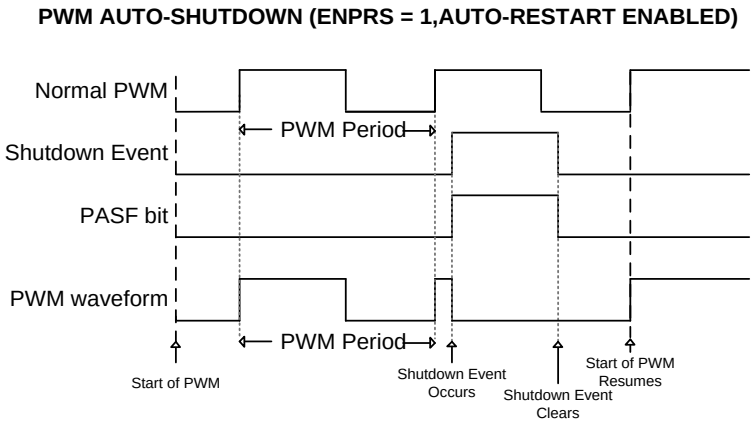


Figure 14-26 (a) PWM Auto Shutoff Event (ENPRS=1)

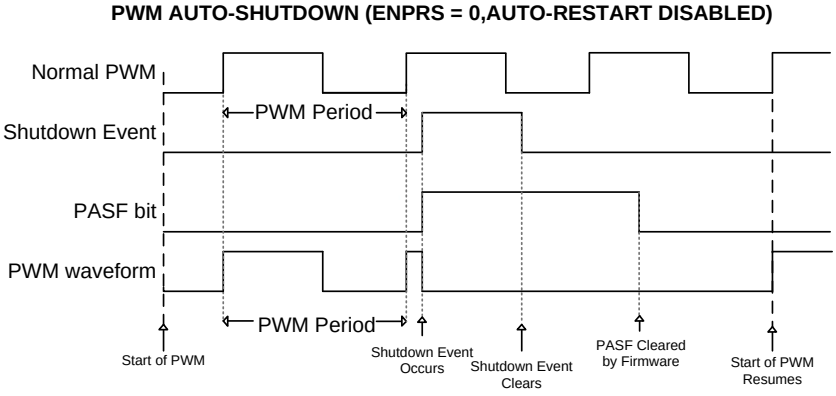


Figure 14-5 (b) PWM Auto Shutoff Event (ENPRS=0)

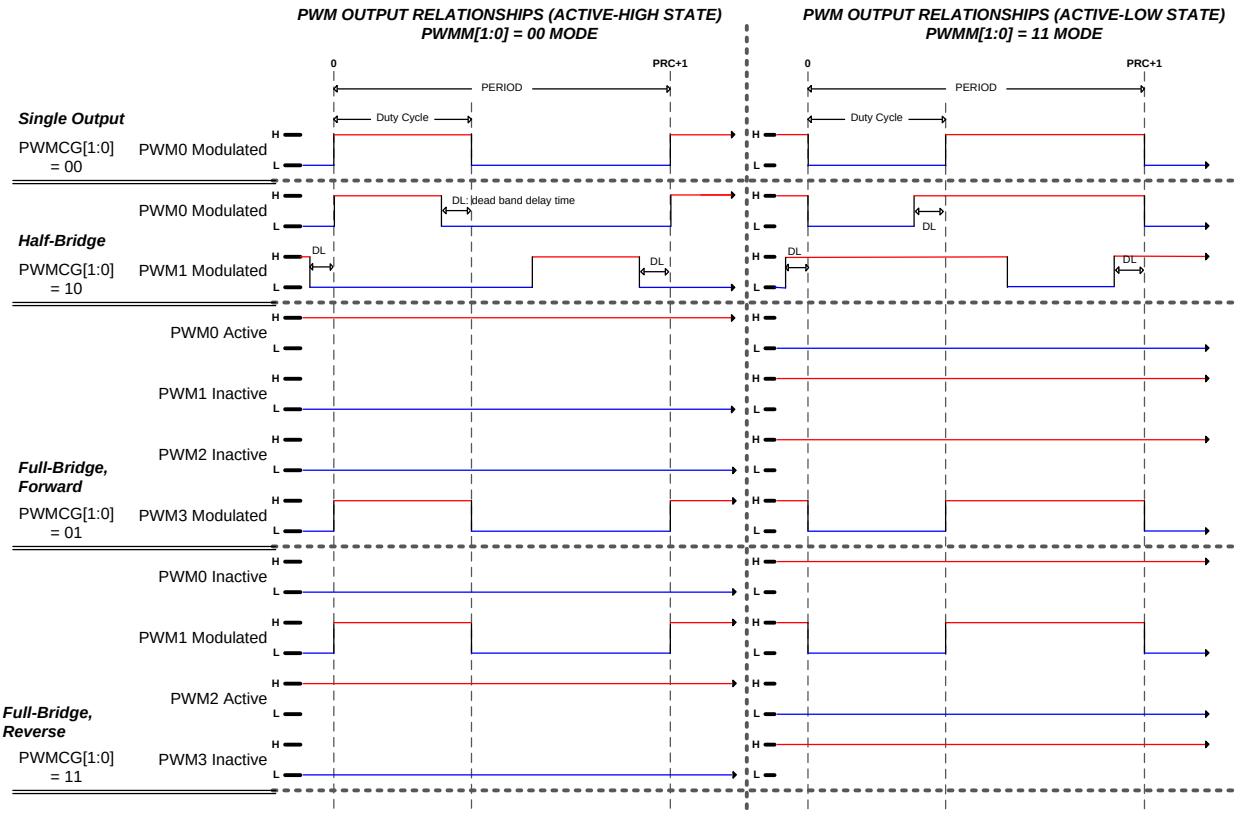


Figure 14-27 Interrelation Chart of PWM Output Set Active High State or Active Low State

14.2.4. Dead Band Delay Description

In dual or quadric output application, PWM often is being utilized to drive MOS power switch signal. In driving process, time difference exists between MOS open and close. Sudden short circuit usually occurs within this time difference to cause huge current and device damage. Hence, including dead band delay in dual or quadric output PWM Mode design can improve this problem.

Dead band delay time controller, DBDC[6:0] is employed to avoid sudden short circuit that result from MOS open and close. The practice is to retain the power switch, MOS_1 in close status and wait until another power switch, MOS_2 is completely closed. After that, open power switch MOS_1. By so doing, it can hinder MOS_1 and MOS_2 from being in the same open status and bring about shout current.

If dead band delay time occurs in inactive status to active status switch delay time, the dead band delay time (DL) is presented in Equation 14-5.

Equation 14-5

$$DL = \frac{DBDC[6:0]}{TMC\_CK}$$

Dead-band Delay Time

Dead Band Delay Time Equation

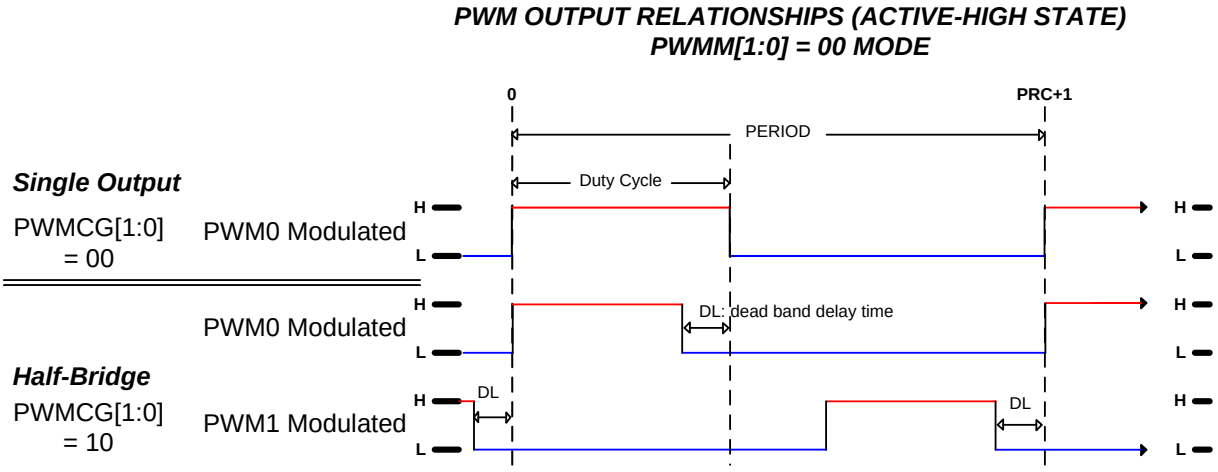


Figure 14-28 PWM Dead Band Delay Sequence

```
....                        ;
MVL     03CH                ; Configure PTn.x as digital output interface and output PWMx signal port
MVF     TRISC2,1,0          ; Related configuration DAn.x, TCn.x, PMn.x
MVL     0FFh                ; Write 0FFh to PRC, determine PWM Period
MVF     PRC,1,0             ; PWM Period=1*(255)/500khz =512us
MVL     07FH                ; PWM Duty cycle (07FH)
MVF     PWMR,1,0            ; PWM duty cycle=1*127/2mhz =254us
BSF     PWMCN,5,0           ; High duty percentage=49.61%
BSF     PWMCN,4,0
MVL     011H                ; Auto shutoff event source is FIL0, auto shutoff status PWM0, PWM2
MVF     PASC,1,0            ; Drive is 0, PWM1 and PWM3 drive is 1
MVL     013H                ; Enable restart auto shutoff status
MVF     PDBD,1,0            ; Delay time= 3* (1/2mhz) =1.5us
MVL     084h
MVF     TMCCN,1,0           ; Start TMC  counter, configure the operating frequency as INTR_CK
....                        ; Do not prescale but configure postscale: TMCS1[2:0] = 001b
MVL     0B0H                ; Start PWM, PWW0 enabled output, PWM1~PWM3 is I/O
MVF     PWMCN,1,0           ; Configure PWM0~PWM3 active output status as high
....
```

Example 14-14 PWM Output Example Program

14.3. Register Description-PFD/PWM

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
46H	TMCCN	ENTMC	TMCCK[1:0]		TMCS1[2:0]			TMCS0[1:0]		0000 0000	0000 0000
47H	PRC	TimerC programmable register								1111 1111	1111 1111
48H	TMCR	TimerC register								0000 0000	0000 0000
4EH	PASC	PASF		PASCF[1:0]		PSSCN0[1:0]		PSSCN1[1:0]		0.00 0000	0.00 0000
4FH	PWMCN	ENPWM	ENPFD	PWMRL[1:0]		PWMCG[1:0]		PWMM[1:0]		0000 0000	0000 0000
50H	PDBD	ENPRS	DBDC[6:0]							0000 0000	0000 0000
51H	PWMR	PWM MSB Byte register								xxxx xxxx	uuuu uuuu
74H	PT2			PT2.5	PT2.4	PT2.3	PT2.2			xxxx xxxx	uuuu uuuu
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000
76H	PT2DA			DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000
78H	PT2M1		PM2.3[0]	PM2.2[1]	PM2.2[0]					.000	.000 ....
79H	PT2M2	PWMTR[1]	PWMTR[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	00.0 0000	00.0 0000

Table 14-36 PFD/PWM Register

TMBCCN：Please refer to **Timer-C** Chapter

PT2/TRISC2/PT2DA/PT2PU：Please refe**r Input/Output Port, I/O** Chapter

PRC：PWM Period Controller, TMC Frequency controller

PWM relative equation：

PWM Period = (PRC+1) × (1/TMC_CK) × TMCS0

PWM Duty Cycle = TMCS0 x PWMR ÷ (TMC_CK x 4)

PWM Resolution = log(TMC_CK / PWM Frequency) / log(2)

Dead-band Delay Time = DBDC / TMC_CK

PASC：PWM Auto Shutoff and Start Control Register

PASF：PWM auto shutoff event pin controller

1：Auto shutoff pinout definition；PWMx auto shutoff status pinout

0：PWM normal modulation output

PASCF[1:0]：Auto shutoff trigger event selector

11：CPAO (Enhanced Comparator output)

10：FIL0 or CPAO (Enhanced Comparator output)

01：FIL0

00：Auto shutoff event

PSSCN0[1:0]：PWM0 and PWM2 Pins Auto shutoff pinout defined controller

1x：PWM0 and PWM2 in auto shutoff, pinout is other combination status.

01：PWM0 and PWM2 in auto shutoff, pinout is high state

00：PWM0 and PWM2 in auto shutoff, pinout is low state

PSSCN1[1:0]： PWM1 and PWM3 Pins auto shutoff defined controller

1x：PWM1 and PWM3 auto shutoff, pinout is other combination status.

01：PWM1 and PWM3 in auto shutoff, pinout is high state

00：PWM1 and PWM3 in auto shutoff, pinout is low state

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
46H	TMCCN	ENTMC	TMCCK[1:0]		TMCS1[2:0]			TMCS0[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
47H	PRC	TimerC programmable register								1111 1111	1111 1111	*,*,*,*,*,*,*,*
48H	TMCR	TimerC register								0000 0000	0000 0000	r,r,r,r r r,r,r,r
4EH	PASC	PASF	PASCF[2:0]			PSSCN0[1:0]		PSSCN1[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
4FH	PWMCN	ENPWM	ENPFD	PWMRL[1:0]		PWMCG[1:0]		PWMM[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*
50H	PDBD	ENPRS	DBDC[6:0]							0000 0000	0000 0000	*,*,*,*,*,*,*,*
51H	PWMR	PWM MSB Byte register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
74H	PT2			PT2.5	PT2.4	PT2.3	PT2.2			xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
76H	PT2DA			DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
78H	PT2M1		PM2.3[0]	PM2.2[1]	PM2.2[0]					.000	.000	-,*,*,* -,-,-,-
79H	PT2M2	PM2.7[1]	PM2.7[0]		PM2.6[0]	PM2.5[1]	PM2.5[0]	PM2.4[1]	PM2.4[0]	00.0 0000	00.0 0000	*,*,-,*,*,*,*,*

PWMCN：PWM Control Register

ENPWM：PWM start controller

1：Start

0：Shutoff

PWMRL[1:0]: PWMR[9:0] low bit

PWMR[9:0] = PWMRH[7:0] + PWMRL[1:0]

PWMCG[1:0]: PWM output controller

11：Full bridge reversed output mode；Modulation output, PWM2 active output state, PWM0 and PWM3 are inactive output state.

10：Half bridge output mode；PWM0 and PWM1 dead band delay control modulated output, PWM2 and PWM3 configured as I/O pin.

01：Full bridge positive output mode；PWM3 modulated output, PWM0 active state output; PWM1 and PWM2 are inactive state output.

00：Single output；PWM0 modulated output and PWM1, PWM2 and PWM3 is configured as I/O pin.

PWMM[1:0]: PWM pinout active state selector

11：PWM0 and PWM2 is active low state, PWM1 and PWM3 are active low state

10：PWM0 and PWM2 is active low state, PWM1 and PWM3 are active high state

01：PWM0 and PWM2 is active high state, PWM1 and PWM3 are active low state

00：P PWM0 and PWM2 is active high state, PWM1 and PWM3 are active high state

PDBD：PWM Dead Band Delay Controller

ENPRS：Auto start condition controller

1：Hardware automatically configures PASF as 0. PWMx modulated output will restart in next period

0：Users employ software to configure PASF as 0. PWMx modulated output will restart in next period

DBDC[6:0]：PWM dead band delay time controller

Delay time definition is PWM signal transfer from inactive state to active state

PWMR：PWM Duty Cycle High Bit Register

PWM duty cycle high byte [9:2]

15. Power System

Power system, PWR has one linear regulator, VDDA and analog circuit common ground power source, ACM. It provides power to IC analog peripheral circuits and can be used to drive external circuits.

- VDDA linear regulator
  - 4 steps adjustable voltage design
  - External bias voltage design
  - Low temperature drift parameters
- ACM internal analog circuit common ground power source
  - Voltage output 1.0V, 1.2V
  - Low temperature drift parameters

PWR Registers：

PWRCN	ENVDDA[0],VDDAX[1:0],ENACM[0]

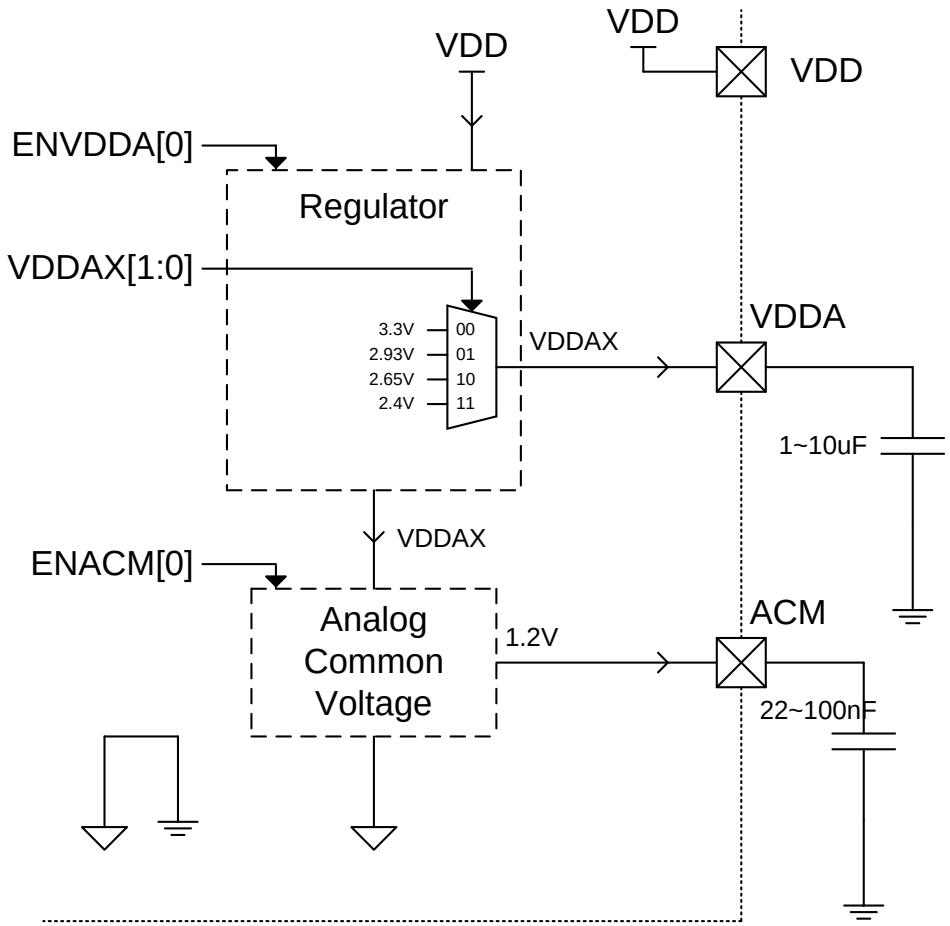


Figure 15-29 Power System Block Diagram

15.1. VDDA Manual

15.1.1. VDDA Initial Configuration

Regulated selector, VDDAX[1:0] can configure VDDA pinout voltage as 3.3V, 2.9V, 2.6V and 2.4V. VDDA is a linear regulated power; users must pay attention to VDD operating voltage so as to avoid the value lowers than that of VDDA output voltage to trigger unexpected circuit miss-action.

15.1.2. VDDA Using External Bias

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 ENVDDA[0] must be configured as 0. Moreover, this method may impact analog circuit performance, so it should be dealt with extra caution.

15.1.3. VDDA Initiation

Configure ENVDDA[0] as <1> to initiate VDDA regulator. Oppositely, if ENVDDA[0] is configured as <0>, VDDA will be shut off. To start VDDA, LNOP and SD18 cannot in enabled status. It must wait after VDDA voltage is stabilized then to start LNOP and SD18. When external 1uF(10uF) regulated capacitor is connected, it requires 500uS(5mS) to stabilize.

15.2. ACM Manual

15.2.1. ACM Initial Configuration

When using ACM, VDDA must be started first. ACM internal generated voltage output is fixed at 1.0V.

15.2.2. ACM Using External Bias

ACM can be designed by external input voltage, if users would like to provide alternative reference voltage, the voltage must input from ACM pin. When using this method, ACM must be shut off, which means ENACM[0] must be configured as 0. Moreover, this method may impact analog circuit performance, so it should be dealt with extra caution. Detailed descriptions please refer to Low Noise PGA and Input Buffer Configuration.

15.2.3. ACM Initiation

Configure EN ACM[0as <1> to start ACM reference voltage source. On the contrary, if EN ACM[0] is configured as <0>, ACM will be shut off.

```
....
MVL   0F0h                    ; Start VDDA and ACM and configure VDDA voltage output as 2.4V
MVF   PWRCN,1,0               ;
....
```

Example 15-15 VDDA and ACM Example Program

15.3. Register Description-PWR

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000	*,*,*,* -,-,-,-

Table 15-37 PWR Register

PWRCN：Power System Control Register

ENVDDA：VDDA start controller

1： Start

0： Shutoff

VDDAX[1:0]：VDDA regulated selector

11：2.4V

10：2.6V

01：2.9V

00：3.3V

ENACM：ACM start controller

1： Start

0： Shutoff

16. Enhanced Comparator

Enhanced comparator, ECPA is composed by analog front end input network, comparator, CPA and voltage generator Ⅰ & Ⅱ. It can be applied in integral analog to digital converter and input voltage comparator etc.

Analog Input Channel,CPAI0~7

- High input impedance, low input current
- 8 input channels
- Input channel short

Comparator, CPA

- Reversed input, output signal
- Equip with low pass filter
- Can be used as Timer- C operating frequency
- Can generate interrupt event

Voltage Generator Ⅰ, Voltage Source Ⅰ

- Output power source has 0.25 time or 0.5 time operating voltage generator
- Support CPA output auto transformation and can change voltage source amplifications Built-in temperature sense circuit

Voltage Generator Ⅱ, Voltage Source Ⅱ

- 15 steps programmable output
- Optimum voltage output VCP

ECPA Registers：

CPACN1	ENCPA[0],CPIST[0],CPIX[0],CPIH[1:0],CPIL[2:0]
CPACN2	CPOX[0],CPOFR[0],CS1[0],CPAT[0],CPVCS[1:0]
CPACN3	CS2[0],CPVRX[3:0]
INTE2	CPOIE[0]
INTF2	CPOIF[0]

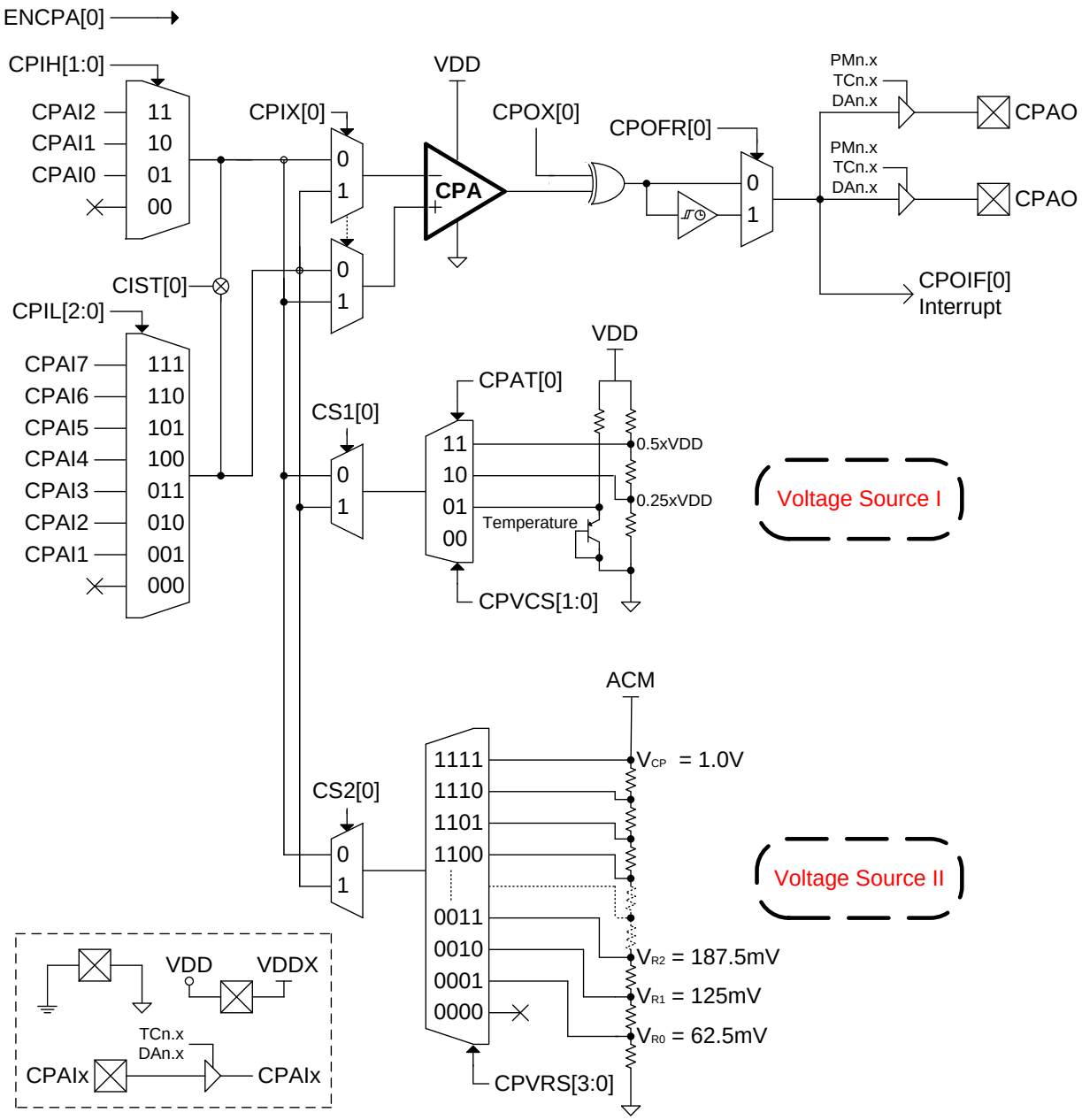


Figure 16-30 ECPA Block Diagram

16.1. ECPA Manual

16.1.1. ECPA Initial Configuration

ECPA input channel selector, CPIH[1:0] and CPIL[2:0] 25 can separately select external input pin to set CPA"+" and "-". High impedance status will appear when it was configured as <00> and <000>. After signal entered, through input signal converter, CPIX[0], IC internal signal will enter into CPA "+" and "-" edge. Thus, the status will be reversed. Input channel, CPIST[0] can make the specified channel to be short circuit.

25 CPAI has 8 channels. While using ECPA, I/O PORT related configurations must be correctly setup, otherwise the signal may not be entered and ECPA may have abnormal function. Detailed register description, please refers to ***Input/Output Port,* I/O** Chapter.

Output signal, CPOX[0] can reverse CPA output status. When CPAO signal is configured from 0 to1, signal will be produced and CPOIF[0] will be set as <1>. Moreover, if the output signal buffer, CPOFR[0] is configured as <1>, CPA output signal will first pass through low pass filter. After filtering the surge, which is smaller than 2uS and output it from CPAO pin. On the other hand, if CPOFR[0] is configured as <0>, CPAO pin may react CPA status instantly.

Two built-in voltage sources respectively are VR Ⅰ and R Ⅱ. CS1[0], CS2[0], CPVCS[1:0] and CPVRX[3:0] can select output voltage to CPA "+" , "-" edge and output potential. Please be noted, ACM must be initiated first when using VR II. ACM。

- When CPVCS[1:0] and CPVRX [3:0] are configured as <00> and <0000>, the output is in high impedance status.
  - CPVCS[1:0] can configure output signal as x 0.25 times or x0.5 times operating voltage, temperature sensor voltage.
  - CPVRX [3:0] can configure signals from 62.5mV～1.0V
- Through CS1[0] and CS2[0] configuration, voltage source can switch to CPA"+" or "-"input edge.

16.1.2. Special Auto Switch Function

Voltage source, VR I has special auto switch output function, CPAT[0] is configured as <1>. When CPVCS[1:0] is set as <1x>, it will output 0.5 times operating voltage signal to corresponding CPAO output high state. When it output 0.25 times operating voltage signal, the corresponding CPAO output is in low state.

When CPAO increases to high state, output reference voltage will shift automatically to 0.25 times operating voltage, changing CPA input signal voltage state. This time, CPAO will descend 0.25 times to low state operating voltage. If CPAO descends to low state status, output voltage reference will switch automatically to 0.5 times operating voltage. By so doing, CPA input signal voltage state will change, making CPAO ascends 0.5 time to high state operating voltage, as illustrated in Figure 16-2.

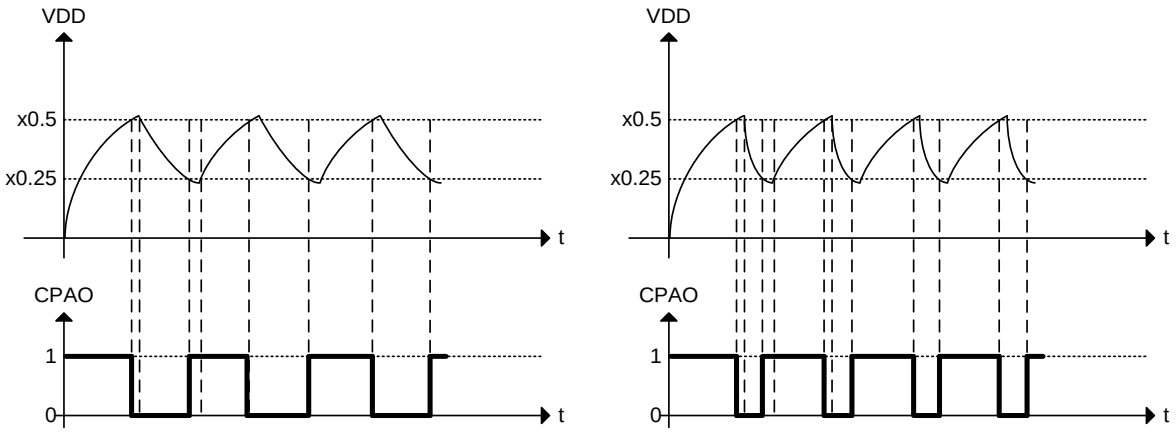


Figure 16-31 ECPA Auto Switch Waveform

16.1.3. ECPA Interrupt Event Service

When CPAO, CPA output signal of ECPA, moves from L➔H, interrupt event signal will be aroused. At this stage, CPOIF[0] will be placed as <1>. If the interrupt service is needed at this time, CPOIE[0] and GIE[0] must be configured as <1>

16.1.4. ECPA Initiation

Configure ENCPA[0] as <1> to initiate ECPA Mode. When ENCPA[0] is set as <0>, ECPA Mode will be closed.

```
0RG   04h                ; Interrupt
BTSZ  INTE2,CPOIF
JMP   CPO interrupt
....                     ; Monitored whether the operating voltage is smaller than 3.0V . Thus, do not
                           configure the interface
MVL   0F0h               ; Start VDDA and ACM. Configure VDDA input voltage as 2.4V
MVF   PWRCN,1,0          ;
MVL   0A4h               ; Configure voltage reference generator I to output x0.25 operating voltage to
                           comparator "+" edge, comparator outputs
MVF   CPACN2,1,0         ; not reversed but filtered, CAPO generates interrupt event through L→H
MVL   01Bh               ;
MVF   CPACN3,1,0         ; Configuring voltage reference generator II to output 750mV to comparator "-"
                           edge
MVL   080h               ;
MVF   CPACN1             ; Start ECAP
....
CPO Interrupt :          ;  ECAP interrupt event service program
```

Example 16-16 ECPA Example Program

16.2. Register Description-ECPA

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2					CPOIE				00.. 0000	00.. 0000	*,*,-,- *,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP2IF	00.. 0000	00.. 0000	*,*,-,- *,*,*,*
34H	CPACN1	ENCPA	CPIST	CPIX	CPIH[1:0]		CPIL[2:0]			0000 0000	0000 0000	*,*,*,*,*,*,*,*
35H	CPACN2		CPOX	CPOFR	CS1	CPAT	CPVCS[1:0]			.000 000.	.000 000.	-,*,*,* r,*,*,*
36H	CPACN3				CS2	CPVRX[3:0]				...0 0000	...0 0000	-,-,-,* *,*,*,*
6DH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,* r,r,r,r
6EH	TRISC1									0000	0000	*,*,*,* -,-,-,-
6FH	PT1DA						DA1.2	DA1.1	DA1.0	.... .000	.... .000	-,-,-,- -,*,*,*
70H	PT1PU							PU1.1	PU1.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2			0000 0000	0000 0000	*,*,*,*,*,*,*,*
76H	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2			0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 16-38 ECPA Register

INTE1/INTE2/INTF2：Please refer to **Interrupt** Chapter

PT1/PT1DA/PT1PU/ PT2/TRISC2/PT2DA/PT2PU：Please refer to **Input/Output Port, I/O** Chapter

CPACN1：ECPA Control Register 1

ENCPA：ECPA start controller

1：Start (Start Restriction：ECPA is powered by VDD. When starting ECPA, not only ENCPA bit must be enabled but also any of ENHAO, ENVDDA or ENACM must be placed 1 to normally start ECPA.)

0：Shutoff

CPIST：Input channel short

1：Short

0：Open

CPIX：Input signal converter

1：Switch

0：Not switch

CPIH[1:0]：Comparator input channel selector

11：CAPI2

10：CAPI1

01：CAPI0

00：High impedance

CPIL[2:0]：Comparator input channel selector

111：CAPI7

110：CAPI6

101：CAPI5

100：CAPI4

011：CAPI3

010：CAPI2

001：CAPI1

000：High impedance

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2					CPOIE				00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,-,*,*,*,*
34H	CPACN1	ENCPA	CPIST	CPIX	CPIH[1:0]		CPIL[2:0]			0000 0000	0000 0000	*,*,*,*,*,*,*,*
35H	CPACN2		CPOX	CPOFR	CS1	CPAT	CPVCS[1:0]			.000 000.	.000 000.	-,*,*,* r,*,*,*
36H	CPACN3				CS2	CPVRX[3:0]				...0 0000	...0 0000	-,-,-,*,*,*,*'
6DH	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	uuuu uuuu	*,*,*,* r,r,r,r
6EH	TRISC1									0000	0000	*,*,*,* -,-,-,-
6FH	PT1DA						DA1.2	DA1.1	DA1.0	.... .000	.... .000	-,-,-,- -,*,*,*
70H	PT1PU							PU1.1	PU1.0	0000 0000	0000 0000	*,*,*,*,*,*,*,*
74H	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
75H	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2			0000 0000	0000 0000	*,*,*,*,*,*,*,*
76H	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2			0000 00..	0000 00..	*,*,*,*,*,-,-
77H	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2			0000 0000	0000 0000	*,*,*,*,*,*,*,*

CPACN2：ECPA Control Register 2

CPOX：Reverse CPA input signal

1：Reverse

0：Not reverse

CPOFR：CPA output signal processor

1：Output after filtered

0：Output without filtered

CS1：Voltage reference generator Ⅰ outputs signal to comparator input "+" or "-" edge

When CPIX is configured as 1

1：Comparator "+" edge

0：Comparator "-" edge

When CPIX is configured as 0

1：Comparator "-" edge

0：Comparator "+" edge

CPAT：Use voltage reference generator Ⅰ, comparator outputs state auto transformation control bit

1：Auto transformation

0：Regular operation

CPVCS[1:0]：voltage reference generator Ⅰ signal output select control bit

11：0.5 times operating voltage

10：0.25 times operating voltage

01：Reference temperature

00：High impedance

CPACN3：Enhanced Comparator Control Register 3

CS2：Voltage reference generator II outputs signal to comparator "+" or "-" edge control bit

When CPIX is configured as 1

1：Comparator "+" edge

0：Comparator "-" edge

When CPIX is configured as 0

1：Comparator "-" edge

0：Comparator "+" edge

CPVRX[3:0]：Voltage reference generator II outputs voltage control bit (ACM=1V configuration)

CPVRX[3:0]	VR(mV)	CPVRX[3:0]	VR(mV)	CPVRX[3:0]	VR(mV)	CPVRX[3:0]	VR(mV)
1111	1000	1011	750	0111	437.5	0011	187.5
1110	937.5	1010	687.5	0110	375	0010	125
1101	875	1001	625	0101	312.5	0001	62.5
1100	812.5	1000	562.5	0100	250	0000	High impedance

Table 16-39 VR II Output Voltage

17. Low Noise OPAMP1

Low noise amplifier, LNOP1 can be solely used or collocated with Σ Δ ADC. It can be employed in small current or high output impedance signal source measurement application.

- Measurement application of high output impedance or low output current
- Multi-channel input and input short circuit design
- Able to measure pin current leakage
- Input signal forward/reverse offset or chopper
- Push/Pull drive ability
-

LNOP1 Registers：

OPCN1 ENOP[0],OPM[1:0],OPP[1:0],OPN[2:0]

AINET1 OPIS[0]

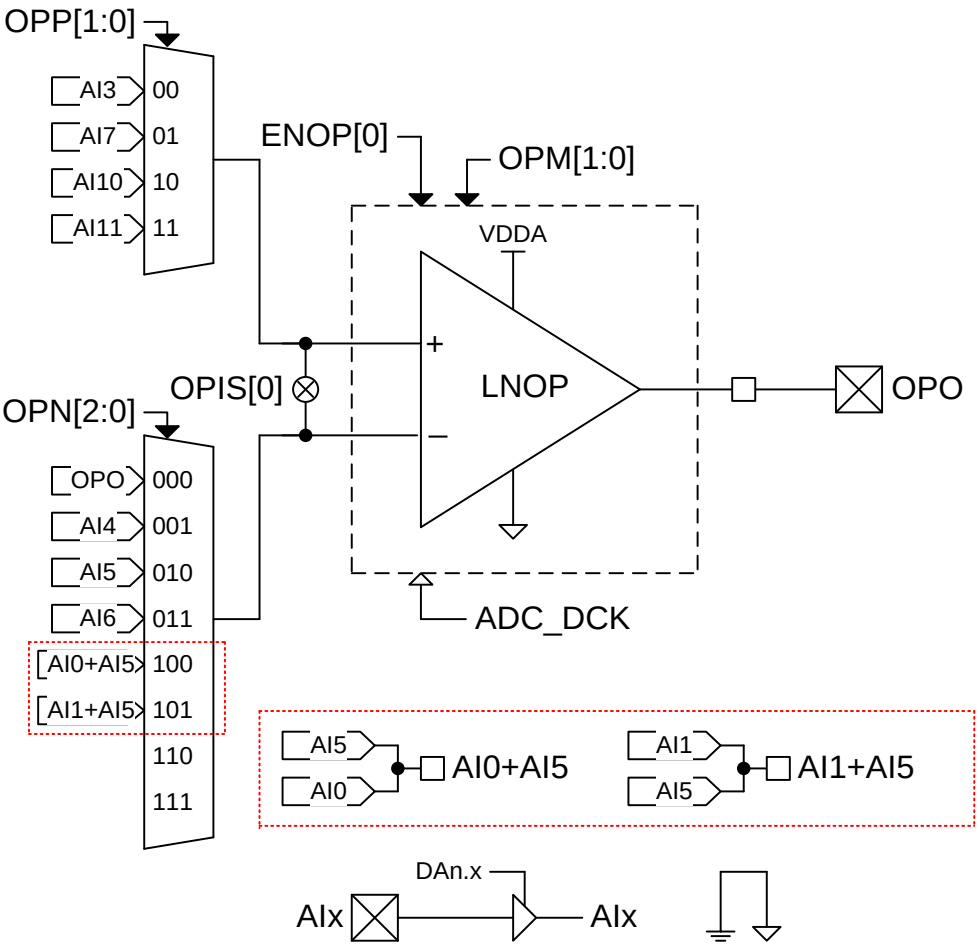


Figure 17-32 LNOP Block Diagram

17.1. LNOP1 Manual

17.1.1. LNOP1 Initial Configuration

Input channel selector, OPP[1:0] and OPN[2:0]26 can determine LNOP "+" and "-" end respectively being input from which external pin. When OPN is configured as <11x>, the value will display high impedance status, after signal entered, OPIS[0] can make the selected channel to be short circuit.

Input processor, OPM[1:0] can configure LNOP1 input signal to conduct forward offset, reverse offset or chopper process.

One special design of LNOP1 input channel is that OPN[2:0] enables input pin to have AI0-AI5 or AI1-AI5, so as to carry out ultra-low current leakage measurement through analog channel pin, as shown in Example 17- 1.

17.1.2. LNOP1 Initiation

Configure ENOP[0] as <1> to initial LNOP1 Mode. Conversely, when ENOP[0] is set up as <0>, LNOP1 will be closed. LNOP1 is powered by VDDA and ACM is its internal common mode voltage reference. Therefore, VDDA and ACM must be started before LNOP1 initiation.

[26] While using LNOP1, I/O PORT related configuration must be correctively setup, otherwise the signal may not be entered and LNOP1 may function abnormally. Detailed register description, please refers to ***Input/Output Port, I/O*** Chapter.

17.2. Register Description-LNOP1

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000	*,*,*,* -,-,-,-
37H	OPCN1	ENOP	OPM[1:0]		OPP[1:0]		OPN[2:0]			0000 0000	0000 0000	*,*,*,*,*,*,*,*
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 17-40 LNOP Related Register

PWRCN：Please refer to **Power System** Chapter

OPCN1：Low Noise OPAMP Control Register 1

ENOP：LNOP start controller

1：Start

0：Shutoff

OPM[1:0]：LNOP input processor

11：Input reverse offset

10：Input forward offset

01：Chopper, Frequency as ADC_CK/128

00：Chopper, Frequency as ADC_CK/64

OPP[1:0]：LNOP "+" end input channel selector

11：AI11

10：AI10

01：AI7

00：AI3

OPN[2:0]：LNOP"-" end input channel selector

11x：Unused

101：AI1-AI5；AI1and AI5 short but not short with AI0

100：AI0-AI5；AI0 and AI5 short but not short with AI1

011：AI6

010：AI5

001：AI4

000：OPO

AINET1：AI Network Control Register 1

OPIS：LNOP input channel short

1：Short

0：Open

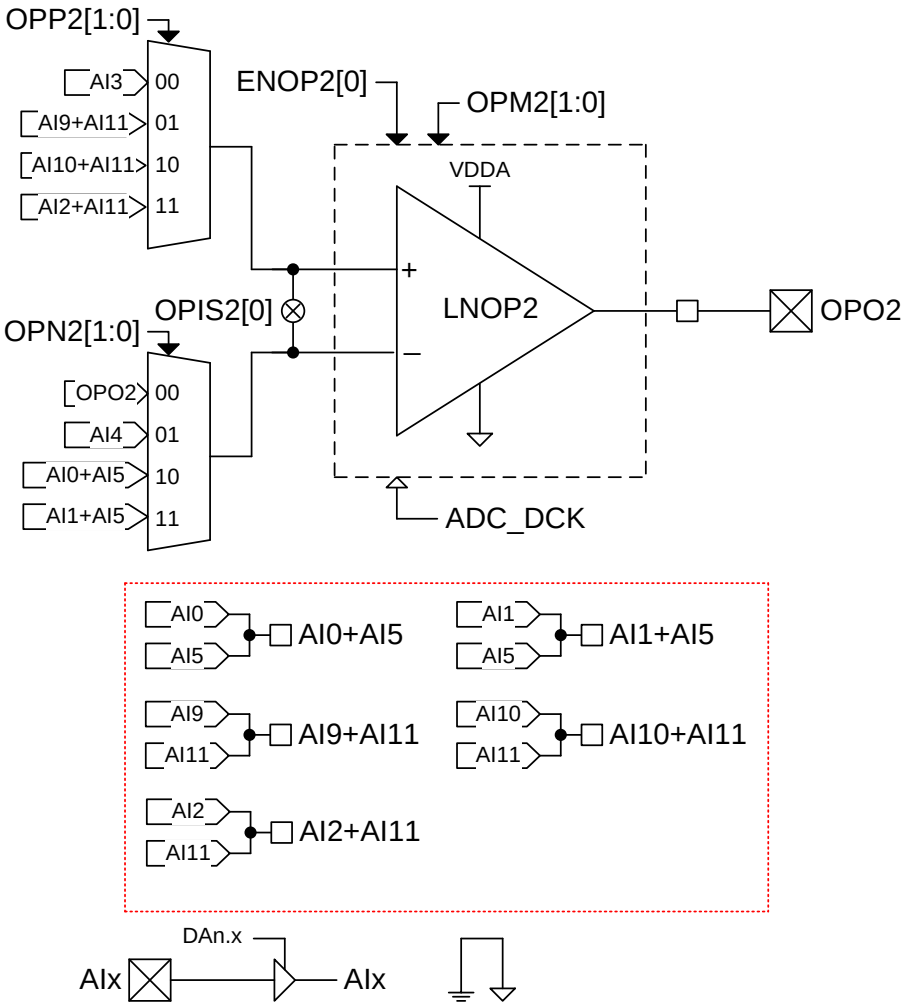
18. Low Noise OPAMP2

Low Noise OPAMP2 can use separately or with ΣΔADC, it is compatible for small current or high output value resistor signal measurement application.

- High output value impedance or low output current.
- Multi-channel input and output short circuit design.
- With the ability to measure the leakage of pins.
- Input positive signal, reverse offset value or Chopper
- Push/Pull drive ability
- Body fat scales dedicated test network

LNOP2 register description：

OPCN1 ENOP2[0],OPM2[1:0],OPIS2[0],OPP2[1:0],OPN2[1:0]



18-33 LNOP2 Diagram

18.1. LNOP2 user Instruction

18.1.1. LNOP2 Initial setting：

Input channel selector OPP2[1:0] and OPN2[1:0]27 can separately choose LNOP's "+"and"-" is input port by specified external pin.

Input processor OPM2[1:0] can Set the input signal to LNOP2 make a positive offset, reverse offset or Chopper process.LNOP2 input channels having the 10GΩ input impedance and common mode input capacitance of 90pF design, the use of a more special place is OPN2 [1: 0] can make input pins AI9-AI1, AI10-AI1, AI2-AI11, AI0- AI5, AI1-AI5 configuration of, in order to achieve pin analog channel measurement of body fat scales application, as shown in 17-1.

18.1.2. LNOP2 Activation

ENOP2 [0] set <1> to Activate LNOP2 mode. Conversely, when ENOP2 [0] Set <0> will turn off LNOP2. LNOP2 power is to use VDDA, and use the ACM for internal common-mode voltage reference point, it must be activated after start VDDA and ACM LNOP2.

When using LNOP2, correct I / O PORT-related settings is required, otherwise it will cause signal not enter phenomenon and lead to LNOP2 function execution does not function properly. Detailed register description, see input / output ports, I / O section

18.2. Register Description-LNOP2

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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
30H	PWRCN	ENVDDA	VDDAX[1:0]=11		ENACM			ENLEDP		0xx0 ..0.	0xx0 ..0.	*,-,-,* -,-,-,-
19AH	OPCN2	ENOP2	OPM2[1:0]		OPIS2[0]	OPP2[1:0]		OPN2[1:0]		0000 0000	0000 0000	*,*,*,*,*,*,*,*

Table 18-1 LNOP2 related register

PWRCN：For detail please check Chapter **Power System**

OPCN2：Low Noise OPAMP Control Register 2

ENOP2：LNOP2 activate control

1：Start

0：Shutoff

OPM2[1:0]：LNOP2 input processor

11：Input reverse offset

10：Input forward offset

01：Chopper, Frequency as ADC_CK/128

00：Chopper, Frequency as ADC_CK/64

OPIS2：LNOP2 input channel short

1：Short

0：Not Short

OPP2[1:0]：LNOP2"+"pin input channel selector

11：AI2+AI11；AI2 and AI11Short

10：AI10+AI11；AI10 and AI11 Short

01：AI9+AI11；AI9 and AI11 Short

00：AI3

OPN2[1:0]：LNOP2"-" pin input channel selector

11：AI1-AI5；AI1 and AI5 short but not short with AI0

10：AI0-AI5；AI0 and AI5 short but not short with AI1

01：AI4

00：OPO2

19. Analog-To-Digital Converter SD18, Σ Δ ADC

SD18 is a high resolution over sampling sigma delta analog-to-digital converter that equips with multi-channel 24 bit output. SD18 consists of four main categories, multi-functional input multiplexer, input buffer and pre-low noise programmable gain amplifier (PGA), Sigma Delta Modulator(Σ Δ AD) and comb filter.

- Multi-Functional Input Multiplexer
  - Can switch to diversified set of input channels, single IC can execute several measurements
  - Input channel can conduct reserve and short, eliminating ADC zero point drift
  - Built-in temperature sensor circuit voltage output
- Input buffer and pre-low noise programmable gain amplifier
  - Pre-low noise PGA has 2, 4 and 8 amplifications and input noise can reach 0.1uV(RMS Noise @ OSR=32768)
  - Without low noise PGA, input buffer can be used to reach high impedance input
  - Without input buffer, input signal common range(CMR) can reach -0.2V~VDDA
- ΣΔ Modulator
  - Adjustable input voltage amplification: 1/4, 1/2, 1, 2, 4, 8 and 16 amplifications
  - Selectable reference voltage amplification: 1 or 1/2
  - 3 bit direct current input bias configuration
  - Adjustable sampling frequency of modulator: 31.5kHz~250kHz
- Comb Filter
  - Can adjust OSR(Over Sampling Ratio)= 256~32768，ADC output rate:1kHz~8Hz (sampling rate =250kHz)
  - Can produce interrupt event

SD18 Registers：

ADCR[23:0]	ADCRH[7:0], ADCRM[7:0], ADCRL[7:0],
ADCCN1	ENADC[0],ENHIGN[0],ENCHP[0],PGAGN[1:0],ADGN[2:0]
ADCCN2	INBUF[0],VRBUF[0],VREGN[0],DCSET[2:0]
ADCCN2	OSR[2:0],CPIST[0],CPIX[0],CPIH[1:0],CPIL[2:0]
AINET1	INH[2:0],INL[2:0],INIS[0],OPIS[0]
AINET2	VRH[1:0],INX[1:0],VRL[1:0]

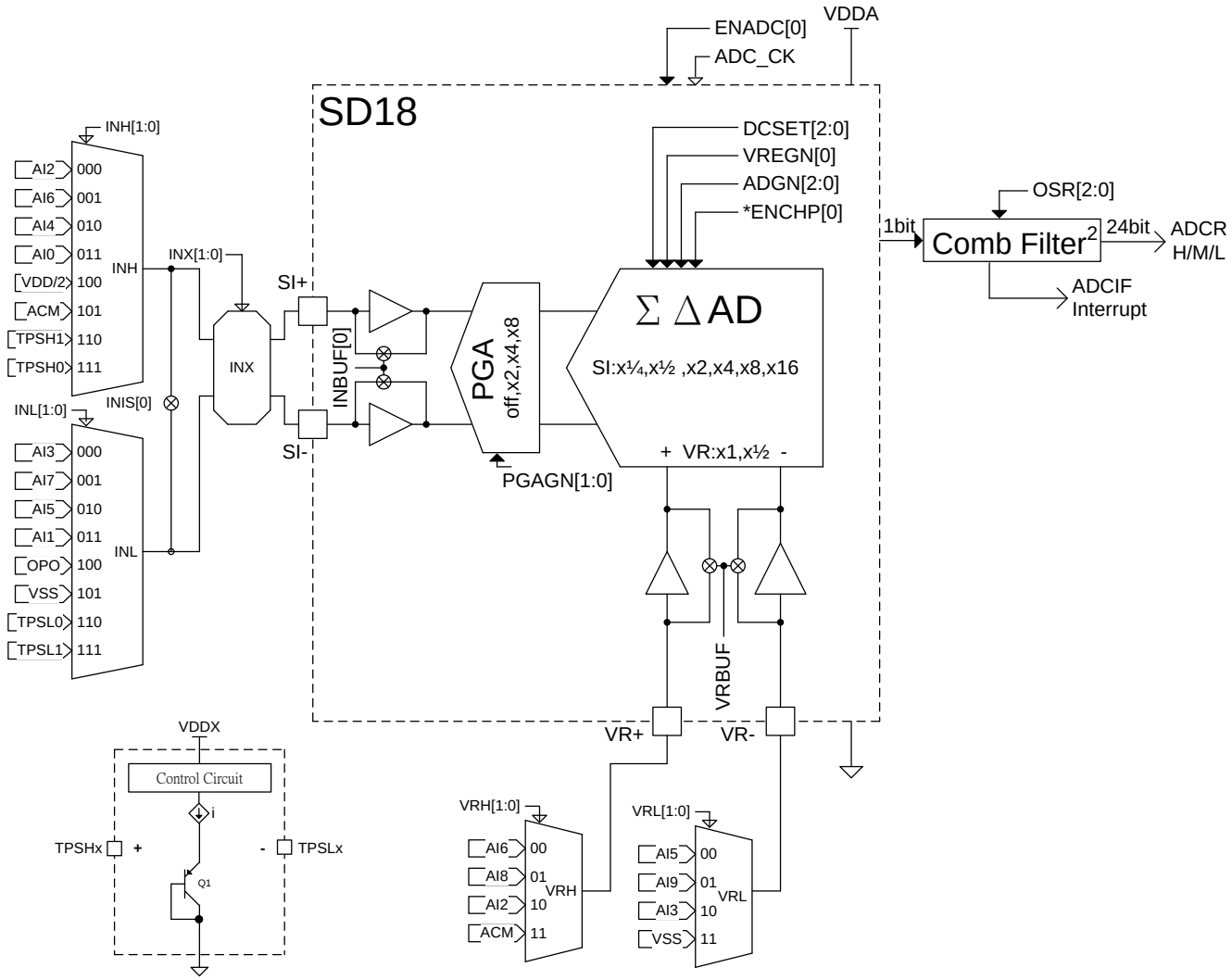


Figure 19-34 SD18 Block Diagram

19.1. SD18 Manual

19.1.1. SD18 Initial Configuration

19.1.1.1. Operating Frequency Configuration

SD_ADC sampling frequency can be determined by sampling frequency selector, ADCCK[0] that can configure whether the SD_ADC operating frequency is offered by HS_DCK or LS_CK. The optimum sampling frequency cannot over than 300 KHz and the minimum sampling frequency cannot less than 25 KHz. In the same output rate, faster sampling frequency can acquire better resolution but it will lower the input impedance (please refers to: 19.2 Analog Channel Input Characteristics). When HS_DCK frequency exceeds the maximum permitted value, frequency adjustment must be made through sampling frequency prescaler, ADCS[2:0], as shown in Table 19-1.

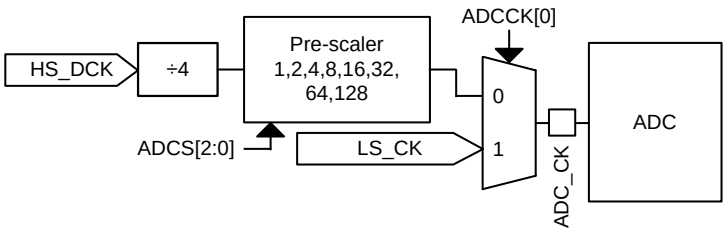


Figure 19-35 SD18 Operating Frequency Block Diagram

"-"：SD18 Inactive Operating Frequency (Unit：KHz)

Configuration / HS_DCK	ADCS[2:0]							
	000	001	010	011	100	101	110	111
4000	-	-	250	125	62.5	31.2	-	-
2000	-	250	125	62.5	31.2	-	-	-
1000	250	125	62.5	31.2	-	-	-	-
500	125	62.5	31.2	-	-	-	-	-

Table 19-41 SD18 Operating Frequency Configuration

19.1.1.2. Configuration of Multi-Functional Input Multiplexer

Multi-functional input multiplexer can generate two sets of differential input signals, unmeasured signal SI+ and SI- and voltage reference VR+ and VR-.

- SI± input signal selector, INH[2:0] and INL[2:0] and SI± input signal converter, INX[1:0] can deliver the external input signal to SI+ or SI- end through the following routes, as Figure 19-3 and Table 19-2(a)；
  - AI0~AI7 pin through INH and INL channel
  - LNOP output signal, OPO
  - Voltage reference source, ACM
  - Internal temperature sensor signal, TPS
- VR± voltage signal selector, VRH[1:0] and VRL[1:0] can determine SD18 voltage reference to be delivered to VR+ or VR- end through the following routes, as shown in Table 19-2(b).
  - AI2~AI3, AI6~AI5, AI8~AI9 pin pass through VRH and VRL channel
  - Voltage reference source, ACM
  - Operating voltage source, VSS
- If SI± input signal, INIS[0] is configured as <1>, INH and INL channel is short. Reversely, INH and INL channel is not short when INIS[0] is configured as <0>.

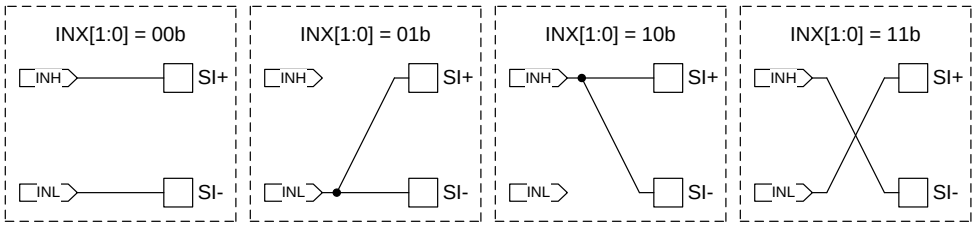


Figure 19-36 4 Combinations of INX Input Signal Convertor

Configuration / Unmeasured Signal	INH[2:0],INL[2:0]							
	000	001	010	011	100	101	110	111
SI+	AI2	AI6	AI4	AI0	VDD/2	ACM	TPSH1	TPSH0
SI-	AI3	AI7	AI5	AI1	OPO	VSS	TPSL0	TPSL1

Table 19-42 (a) S±Input Selector

Configuration / Input	VRH[1:0],VRL[1:0]			
	00	01	10	11
VR+	AI6	AI8	AI2	ACM
VR-	AI5	AI9	AI3	VSS

Table 19-2 (b) VR±Input Selector

19.1.1.3. Low Noise PGA and Input Buffer Configuration

Low noise PGA utilizes capacitance switch amplifying method to distinguish any unmeasured signal that is smaller than 1uV. After that, low noise PGA will be implemented to amplify signal.

PGA amplification can be configured by amplification adjuster, PGAGN[1:0] as presented in Table 19-3.

Configuration / Input	PGAGN[1:0]			
	00	01	10	11
PGA Gain	off	x2	x4	x8

Table 19-3 PGA Configuration

Input buffer includes unmeasured signal buffer and voltage reference buffer that are activated by configuring INBUF[0] and VRBUF[0] respectively. When low noise PGA is enabled, unmeasured signal buffer configuration will be invalid. That is to say, the internal hardware circuit will close the unmeasured signal buffer automatically. Moreover, when unmeasured signal buffer is activated, amplification of ΣΔ modulator must lower than or equal to 4 amplifications. Measurement deviation may be brought about when signal amplification is higher than 4 amplifications.

SD18 has high input impedance while using input buffer. Measurement for high output impedance signal can avoid deviation that caused by impedance matching problem. However, the buffer itself exists in zero point deviation and restriction on input signal common range. As for the relation of input common mode voltage range and buffer, please refer to Table 19-4.

Configuration / Input	Active Buffer	Inactive Buffer
VR+	VDDA-0.2~VSS+1	VDDA~VSS
VR-, SI+,SI-	VDDA-1~VSS+0.2	VDDA~VSS-0.2

Table 19-4 Input Signal CMR

19.1.1.4. Σ Δ Modulation Configuration

SD18 adopts two-stage ΣΔ modulator, unmeasured signal and voltage reference can conduct amplification and bias adjustment through below configurations.

- Configuring ΔVR± amplification modulator, VREGN[0] as <1>, voltage reference signal must adjust to 1/2 amplification. Input signal ΔSI± = (SI+ − SI-) and ΔVR± = (VR+ − VR-) ratio will change accordingly. If VREGN[0] is configured as <0>, reference voltage signal will be adjusted to 1 amplification.
- Amplification modulator, ADGN[2:0] can configure input signal. It has optimum 16 times signal amplifying gain, as shown in Table 19-5(a).
- Configuring ENCHP[0] as <0> enables signal to pass through chopper to reduce frequency noise. Conversely, set ENCHP[0] as <1>, input signal will bypass chopper.
- Through direct current input bias modulator, DCSET[2:0], input signal SI± can adjust input signal zero point address in order to increase measurement range. Bias method adopts weighted reference signal VR± amplification value, as shown in Table 19-5(b).
- When measuring signals, please mind the matching problems between external input signal impedance and ADC. Detailed description please refers to 19.2 Analog Channel Input Characteristics.

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

Table 19-5 (a) ADGN[2:0] Amplification Configuration

Configuration / Input	DCSET[2:0]							
	000	001	010	011	100	101	110	111
SI±	+0	+1/4	+1/2	+3/4	+0	-1/4	-1/2	-3/4

Unit：VR±

Table 19-5 (b) SI± Input Signal Weighted Voltage Reference Chart

After pre-PGA and modulator amplification bias adjustment, ΣΔ modulator's equivalent unmeasured signal, ΔSI_I and equivalent reference voltage, ΔVR_I is given by：

Equation 19-6

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

Equation 19-7

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

Please be noted that in order to make ΣΔ modulator output has higher resolution and linearity, equivalent voltage reference, ΔVR_I is suggested to fall between ΔVR_I=0.8V~1.2V. Equivalent unmeasured signal, ΔSI_I is recommended to be operated between ΔSI_I=±0.9 x ΔVR_I.

19.1.1.5. Comb Filter Configuration

ΣΔ Modulator outputs 1-bit data to two-stage comb filter. Comb Filter will transfer the data to 24-bit value and save the value in register, ADCR[23:0]. ADCR[23:0] data update rate equals to that of SD18 output rate. The calculated equation is SD18 sampling frequency and SD18 output rate frequency ratio. It can be called as OSR (Over Samplings Ratio) as well.

Thus, SD18 output rate is ADC_CK÷OSR. However, OSR value can be set by OSR[2:0] in order to generate different SD18 output conversion frequency, as Table 19-5(c).

Configuration / ADC_CK	OSR[2:0]							
	000	001	010	011	100	101	110	111
250K	976	488	244	122	61	30	15	7
125K	488	244	122	61	30	15	7	3

Table 19-5 (c) SD18 Over-Sampling Rate Configuration

ADCR[23:0] is constituted by ADCRH[7:0] , ADCRM[7:0] and ADCRL[7:0] where 24-bit Comb Filter outputted data is stored. Comb Filter data format is presented in Figure 19-4.

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

	Equivalent unmeasured signal	ADCR[23:0]	
		Hexadecimal	Binary
The output format is two complement	ΔVR_I	400000	0100-0000 0000-0000 0000-0000
	$\Delta VR\_I \times \frac{1}{2^{22}}$	000001	0000-0000-0000-0000-0000-0001
	0	000000	0000-0000 0000-0000 0000-0000
	$-\Delta VR\_I \times \frac{1}{2^{22}}$	FFFFFF	1111-1111-1111-1111-1111-1111
	$-\Delta VR\_I$	C00000	1100-0000 0000-0000 0000-0000

Table 19-6 ADCR[23:0] and Input Signal Correlation

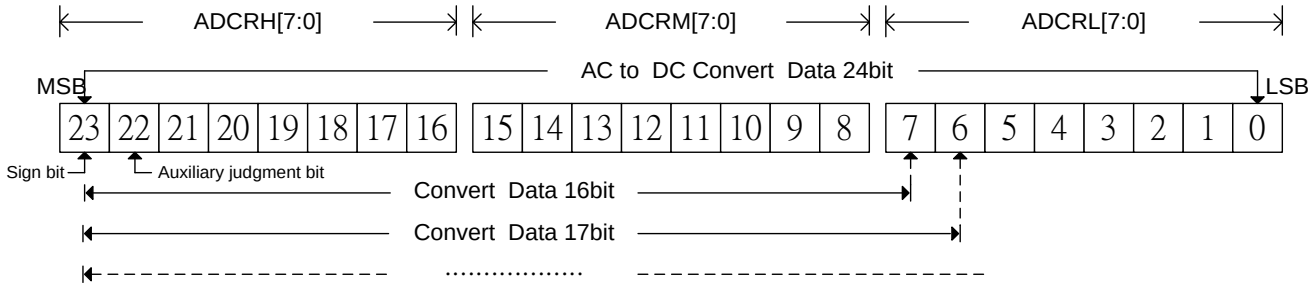


Figure 19-37 ADCR[23:0] Resolution Chart

19.1.2. Interrupt Service Configuration

After comb filter converted the value, it will be stored in register, ADCxR[23:0]. At this time, interrupt event signal will be generated and ADCIF[0] will be placed as <1>. If you need to interrupt the event service at this time need ADCIE [0] and GIE [0] set <1>.

19.1.3. SD18 Initiation

Configuring ENADC[0] as <1> enables SD18 to carry out analog to digital conversion. On the other hand, when ENADC[0] is set as <0>, SD18 is disabled. VDDA supplies SD18 power and uses ACM as the internal common mode reference voltage point. Thus, VDDA and ACM must be initiated first before starting SD18.

SD18 operating voltage source comes from VDDA and the voltage of AIx input pin cannot exceed VDDA voltage (please refers to Table 19-4 Input Signal CMR). When VDDA power turns off (neither from internal initiate nor external input), voltage exists in SD18 input signal network SI± and voltage reference network VR± may bring about power leakage and may indirectly damage the chip and enlarge power consumption. Thus, before turning off VDDA power, SD18 input signal network or voltage reference network must be properly selected. To avoid power leakage caused by external voltage, the network switch must be adjusted to internal ACM or VSS.

19.2. Analog Channel Input Characteristics

SD18 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 SD18 sampling frequency and signal amplification.

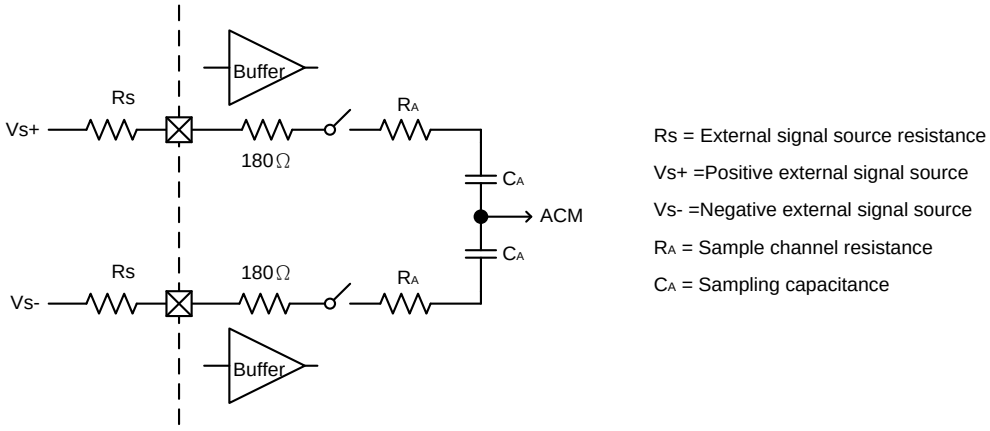


Figure 19-38 AIx Input Capacitors and Impedance Module

As Figure 19-5 illustrated, if input signal does not pass through buffer, further consideration of input signal impedance, Rs and SD18 sampling frequency, ADC_CK and

parasitical resistor RA, capacitor CA effect must also been taken into account. Related calculation is given by：

Equation 19-8

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

t_s：SD18 shortest sampling time,
ENOB：Expected SD18 effective bit
Gain：(PGA Gain)×(ΣΔAD Gain)

Equation 19-9

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

Fs：SD18 shortest sampling frequency

SD18 is composed by PGA and ΣΔAD; these two parts have separate R_A and C_A value in design. The shortest sampling time, t_s are calculated by direct and input signal matching consideration.

- If PGA is not implemented, signal is directly configured as 4 amplifications by ΣΔAD. t_s in calculation, R_A=10KΩ and C_A=2pF.
- If PGA is used, the amplification is 2. However, ΣΔAD amplification is 4, the entire amplification is 8. In calculation, the ts only depends on input signals and matching amplifier, therefore, R_A=1.25KΩ and C_A=16pF

In equation, R_A and C_A corresponding SD18 gain relationship is shown in Table 19-7 (a)/ (b)/ (c).

Σ ΔAD Gain	C_A	R_A
x1/4	0.125pF	10KΩ
x1/2	0.25pF	10KΩ
x1	0.5pF	10KΩ
x2	1pF	10KΩ
x4	2pF	10KΩ
x8	4pF	5KΩ
x16	8pF	2.5KΩ

Table 19-7 (a) SD18 Gain and R_A and C_A Relation

PGA Gain	C_A	R_A
x2	16pF	1.25KΩ
x4	32pF	0.625KΩ
x8	64pF	0.3KΩ

Table 19-7 (b) PGA Gain and R_A and C_A Relation

VR Gain	C_A	R_A
x1/2	0.25pF	10KΩ
x1	0.5pF	10KΩ

Table 19-7 (C) VR Gain and R_A and C_A Relation

■

SD18 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.

19.3. 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: $V_{TPS@0°K}$ =0V.
- Through measurement method, ADC bias ($V_{ADC-OFFSET}$) and BJT asymmetry ($I_{S1} \neq I_{S2}$) can be offset automatically.
- Single point temperature calibration.

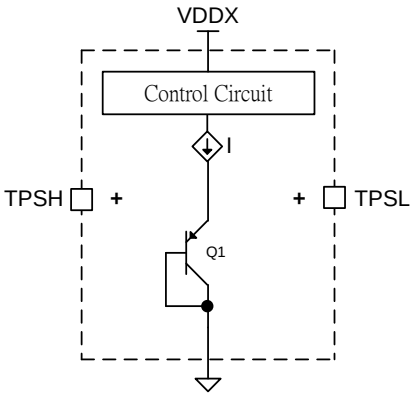


Figure 19-39 Block Diagram of Absolute Temperature Sensor Application

19.3.1. TPS Initial Configuration and Calculation

TPS function can be automatically initiated when SD18 is active. Configuring input signal selector, INH[2:0] and INL[2:0] as INH=[111]、INL=[110] to measure voltage signal, VTPS0. Configuring INH=[110]、INL=[111] to measure voltage signal, VTPS1.

Under the same temperature TA(°C), after SD18 measured the value of VTPS0 and VTPS1, add the value together and get the mean, TPS corresponding voltage value, VTPS@TA can be acquired.

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

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

.

$$= \frac{V_{TPS@T_A}}{289 + T_A}$$

Equation 19-10 TPS Gain

19.3.2. TPS Example Description

Presumed the calibration temperature is $T_A = 25℃ = 298.15°K$

Configuration 1: $V_{TPS0@25℃} = 52.515mV$ is measured....(1)

Configuration 2: $V_{TPS1@25℃} = -53.626mV$.is measured...(2)

((1) − (2)) ÷ 2 gets $\Delta V_{TPS@25℃} = 53.0705mV$

∵$V_{TPS@0°K} = 0.0mV$

∴curve slope $G_{TPS} = (V_{TPS@298.15°K} - V_{TPS@0°K}) \div (298.15 - (0)) \fallingdotseq 178uV/°K$

Temperature value on any point of the curve: $T_A = V_{TPS@T_A} \div G_{TPS}$

19.4. Register Description-SD18

"-"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000
26H	INTF1		ADCIF							.000 0000	.000 0000
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA	DA5.7	DA5.6	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	1111 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

Table 19-8 SD18 Register

INTE1/INTF2：Please refer to Interrupt Chapter

PWRCN：Please refer to Power System Chapter

MCKCN1：Please refer to Oscillator, Clock Sources and Power Managed Modes Chapter

ADC0RH/M/L：SD18 Output Register

ADCCN1：SD18 Control Register 1

ENADC：SD18 start controller

1：Start

0：Shutoff

ENHIGN：Reserved for manufacturer test

1：Forbidden, set 1 will lead to unstable SD18 output value

0：Define configuration

ENCHP：SD18 internal chopper

1：Forbidden, set 1 will lead to lower SD18 resolution

0：Define configuration

PGAGN[1:0]：PGA amplification adjuster

11：x8；By pass input buffer

10：x4；By pass input buffer

01：x2；By pass input buffer

00：off；By pass PGA, using input buffer must choose this option

"-"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	ETIE	EOIE	0000 0000	0000 0000
26H	INTF1		ADCIF							.000 0000	.000 0000
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA	DA5.7	DA5.6	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	1111 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

ADGN[2:0]：AD amplification adjuster

111：Unused

110：x16

101：x8

100：x4

011：x2

010：x1

001：x1/2

000：x1/4

ADCCN2：SD18 Control Register 2

INBUF：**SI**± input buffer

1：Start

0：Shutoff

VRBUF：**VR**± input buffer

1：Enable

0：Disable

VREGN：**VR**± amplification adjuster

1：x1/2

0：x1

"-"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1
"."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
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	ETIE	EDIE	0000 0000	0000 0000
26H	INTF1		ADCIF							.000 0000	.000 0000
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA	DA5.7	DA5.6	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	1111 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

DCSET[2:0]：**SI**± bias adjuster

111：-3/4 **VR**±

110：-1/2 **VR**±

101：-1/4 **VR**±

100：No bias

011：+3/4 **VR**±

010：+1/2 **VR**±

001：+1/4 **VR**±

000：No bias

ADCCN3：SD18 Control Register 3

OSR[2:0]：SD18 over-sampling rate divider

111：32768

110：16384

101：8192

100：4096

011：2048

010：1024

001：512

000：256

"-"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMEIE	TMAIE	WDTIE	ETIE	EOIE	0000 0000	0000 0000
26H	INTF1		ADCIF							.000 0000	.000 0000
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA	DA5.7	DA5.6	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	1111 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

AINET1：AI Network Control Register 1

INH[2:0]：**SI±"+"** input signal selector

111：TPSH0

110：TPSH1

101：ACM

100：VDD/2

011：AI0

010：AI4

001：AI6

000：AI2

INL[2:0]：**SI±"-"** input signal selector

111：TPSL1

110：TPSL0

101：VSS

100：OPO

011：AI1

010：AI5

001：AI7

000：AI3

INIS：**SI±**input signal short selector

1：Short

0：Not short

OPIS：LNOP "+", "-"edge input signal short control bit

1：Short

0：Not short

"-"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1											
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMEIE	TMAIE	WDTIE	ETIE	EOIE	0000 0000	0000 0000
26H	INTF1		ADCIF							.000 0000	.000 0000
30H	PWRCN	ENVDDA	VDDAX[1:0]		ENACM					0000	0000 ....
31H	MCKCN1	ADCS[2:0]			ADCCK	XTHSP	XTSP	ENXT	ENHAO	0000 0001	0000 0001
32H	MCKCN2			LSCK	HSCK	HSS[1:0]		CPUCK[1:0]		..00 0000	..00 0000
33H	MCKCN3	LCDS[2:0]				PERCK	BZS[2:0]			000. 0000	000. 0000
39H	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
3AH	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
3BH	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
3CH	ADCCN1	ENADC	ENHIGN	ENCHP	PGAGN[1:0]		ADGN[2:0]			0000 0000	0000 0000
3DH	ADCCN2			INBUF	VRBUF	VREGN	DCSET[2:0]			..00 0000	..00 0000
3EH	ADCCN3	OSR[2:0]								000.	000. ....
3FH	AINET1	INH[2:0]			INL[2:0]			INIS	OPIS	0000 0000	0000 0000
40H	AINET2		VRH[1:0]		INX[1:0]		VRL[1:0]			.000 000.	.000 000.
6AH	PT4	PT4.7	PT4.6	PT4.5	PT4.4	PT4.3	PT4.2	PT4.1	PT4.0	xxxx xxxx	uuuu uuuu
6BH	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	DA4.3	DA4.2	DA4.1	DA4.0	1111 1111	1111 1111
6CH	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	PU4.3	PU4.2	PU4.1	PU4.0	0000 0000	0000 0000
192H	PT5	PT5.7	PT5.6	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	uuuu uuuu
193H	PT5DA	DA5.7	DA5.6	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	1111 1111
194H	PT5PU	PU5.7	PU5.6	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	0000 0000

AINET2：AI Network Control Register2

VRH[1:0]：**VR±"+"** voltage signal selector

11：ACM

10：AI2

01：AI8

00：AI6

INX[1:0]：**SI±** input signal converter

11：INH→ADL, INL→ADH

10：INH floating, INH→ADH & ADH

01：INL→ADH & ADL, INH floating

00：INH→ADH, INL→ADL

VRL[1:0]：**VR±"-"** voltage signal selector

11：VSS

10：AI3

01：AI9

00：AI5

20. 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
- 4 steps adjustable driving voltage state
- Supporting 4 LCD waveform types operation
  - Static operation
  - 2-mux, 1/3 bias (2-mux, 1/3 bias)
  - 3-mux, 1/3 bias (3-mux, 1/3 bias)
  - 4-mux, 1/3 bias (4-mux, 1/3 bias)
- Selectable input clock source and output frequency
- Equips with Blinking capability

LCD Registers：

LCDCN1	ENLCD[0],LCDPR[0],VLCDX[1:0],LCDBF[0],LCDBI[1:0],
LCDCN2	LCDBL[0],LCDMX[1:0]
LCD[159:0]	LCD0[7:0]~LCD19[7:0]

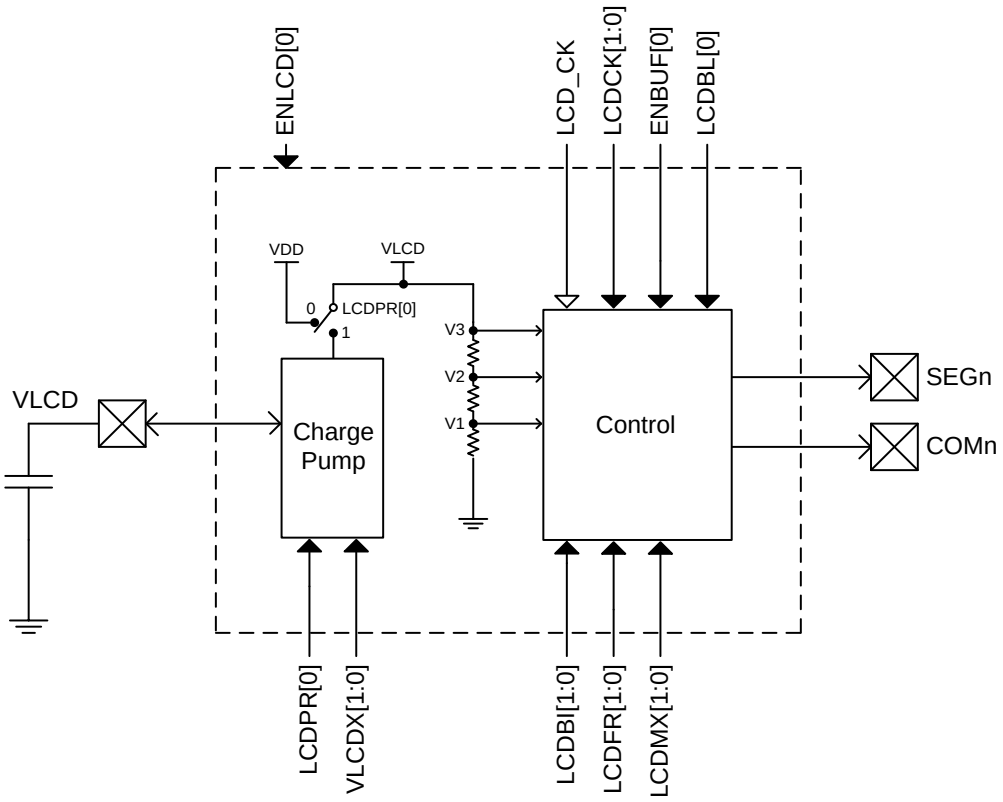


Figure 20-40 LCD Block Diagram

In LCD design, when 2-mux or 3-mux, COM0 and COM1 can multiplexed SEG0 and SEG1. Registers used by SEG0/1 can be multiplexed with SEG2/3. As Table 20-1 indicates.

"-" : Unused

Digit Register								
Name		Digit Pin	Name		Digit Pin	Name		Digit Pin
LCD0	[28]	SEG0	LCD6	3:0	SEG14	LCD14	3:0	SEG30
	[28]	SEG1		7:4	SEG15		7:4	SEG31
	3:0	SEG2	LCD7	3:0	SEG16	LCD15	3:0	SEG32
	7:4	SEG3		7:4	SEG17		7:4	SEG33
LCD1	-	-	LCD8	3:0	SEG18	LCD16	3:0	SEG34
	[28]	SEG1		7:4	SEG19		7:4	SEG35
	3:0	SEG4	LCD9	3:0	SEG20	LCD17	3:0	SEG36
	7:4	SEG5		7:4	SEG21		7:4	SEG37
LCD2	3:0	SEG6	LCD10	3:0	SEG22	LCD18	3:0	SEG38
	7:4	SEG7		7:4	SEG23		7:4	SEG39
LCD3	3:0	SEG8	LCD11	3:0	SEG24	LCD19	3:0	SEG40
	7:4	SEG9		7:4	SEG25		7:4	SEG41
LCD4	3:0	SEG10	LCD12	3:0	SEG26			
	7:4	SEG11		7:4	SEG27			
LCD5	3:0	SEG12	LCD13	3:0	SEG28			
	7:4	SEG13		7:4	SEG29			

Table 20-43 (a) LCDn Digit Register and Digit Pin Comparison

"-" : unused

Register	Waveform	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LCD0	4-MUX	SEG3.3	SEG3.2	SEG3.1	SEG3.0	SEG2.3	SEG2.2	SEG2.1	SEG2.0
	3-MUX	**SEG1.1**	SEG3.2	SEG3.1	SEG3.0	**SEG1.0**	SEG2.2	SEG2.1	SEG2.0
	2-MUX	**SEG1.1**	**SEG0.1**	SEG3.1	SEG3.0	**SEG1.0**	**SEG0.0**	SEG2.1	SEG2.0
LCD1	4-MUX	SEG5.3	SEG5.2	SEG5.1	SEG5.0	SEG4.3	SEG4.2	SEG4.1	SEG4.0
	3-MUX	-	SEG5.2	SEG5.1	SEG5.0	**SEG1.2**	SEG4.2	SEG4.1	SEG4.0
	2-MUX	-	-	SEG5.1	SEG5.0	**-**	-	SEG4.1	SEG4.0

Table 20-1 (b) LCD0 and LCD1 Register Multiplexed Digit Pin Comparison

[28] Multiplexed register : When LCD operating waveform is configured as 2-MUX or 3-MUX, digit control bit will change as shown in Table 20-1(b).

20.1. LCD Manual

20.1.1. LCD Initial Configuration

Operating Frequency and Output Frame Frequency Configuration

Operating frequency is offered by PERA_CK, after prescaler, LCDS[2:0] offers appropriate operating frequency to LCD to output frame frequency. Output waveform controller, LCDMX[1:0] can configure LCD operating waveform, frame frequency and operating waveform must be configured correctly in accordance with external LCD monitor specification, otherwise LCD monitor will display shadow or abnormal digit display. LCD operating frequency and frame frequency is presented in Figure 20-2 。

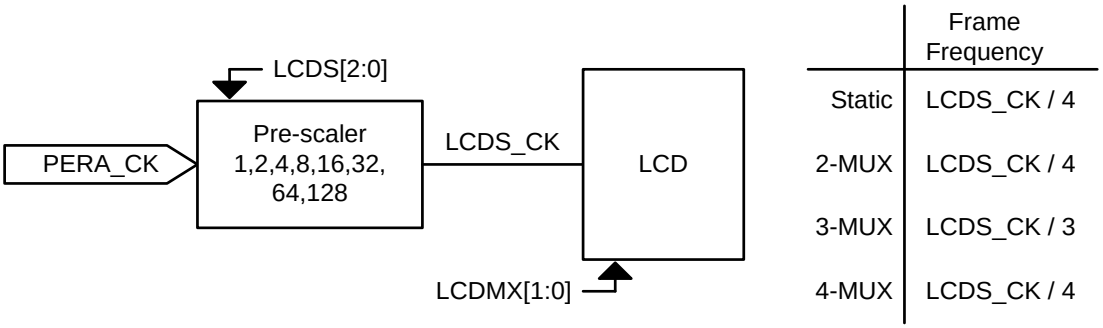


Figure 20-41 LCD Operating Frequency and Frame Frequency

PERA_CK=976Hz		LCDS[2:0]							
		0	1	10	11	100	101	110	111
LCDS_CK		976	488	244	122	61	31	15	8
Frame Frequency	Static, LCDMX[1:0]=00	244	122	61	31	15	8	4	2
	2-MUX, LCDMX[1:0]=01	244	122	61	31	15	8	4	2
	3-MUX, LCDMX[1:0]=10	325	163	81	41	20	10	5	3
	4-MUX, LCDMX[1:0]=11	244	122	61	31	15	8	4	2

Unit: Hz

Table 20-44 LCD Operating Frequency Configuration

Charge Pump Circuit and LCD Operating Voltage Configuration

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

- VLCD is input externally, LCDPR[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 VLCDX[1:0] will not influence LCD operating voltage.
  - When driving over-sized or over-loaded LCD monitor, LCD output buffer, LCDBF[0] can be set up as <1> and can initiate buffer to increase LCD driving ability. On the contrary, if LCDBF[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, LCDPR[0] as <1> and configure charge pump voltage state controller, VLCDX[1:0] , VLCD will be produced to supply LCD power that prevents it against IC operating voltage changes.
  - VLCDX[1:0] can be configured as 4 different kind of functional operating voltage if the charge pump is being initiated first. Charge pump circuit may influence analog-to-digital convertor, SD18 performance in high resolution conversion.
  - Using internal charge pump circuit to produce VLCD, LCD buffer will be started automatically by internal hardware circuit.

Bias and Blinking Configuration

Waveform bias controller, LCDBI[1:0] can configure every LCD frame is consisted by how many bias. There are two options, configuring <00> LCD waveform in static operation and configuring <10> LCD waveform operation in 1/3 bias.

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 LCD1[7:0]~ LCD19[7:0].

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]. Whether the control bit is valid or invalid, it depends on waveform output controller, LCDMX[1:0] configuration. For example, when output waveform is 4-mux, SEGn[3:0] 4-bit is valid. Besides, if output waveform is 2-mux, only the lowest SEGn[3:0] 2-bit is valid, as illustrated in Table 20-1. Because of the valid and invalid characteristic, LCD0 and LCD1 digit register has SEG0[3:0] and SEG1[3:0] multiplexed design.

20.1.2. LCD Initiation

Configuring ENLCD[0] as <1> can start LCD driver. On the other hand, when ENLCD[0] is configured as <0>, LCD driver will be closed.

```
MVL     01100000B
MVF     LCDCN2,1,0      ;1/4 duty, LCD digit display
MVL     11011100B
MVF     LCDCN1,1,0      ;1/3 bias, LCD start. Initiating LCD charge pump power
                        VLCD=3V
CALL    DELAY           ;LCD charge pump power regulated time (at VLCD
                        CAP-4.7uF)
                        ;VDD=2.2V, VLCD=3V, Stable time ~ 85msec
                        ;VDD=3.6V, VLCD=3V, Stable time ~ 15msec
```

Example 20-17 LCD Example Program

20.2. LCD Output Waveform

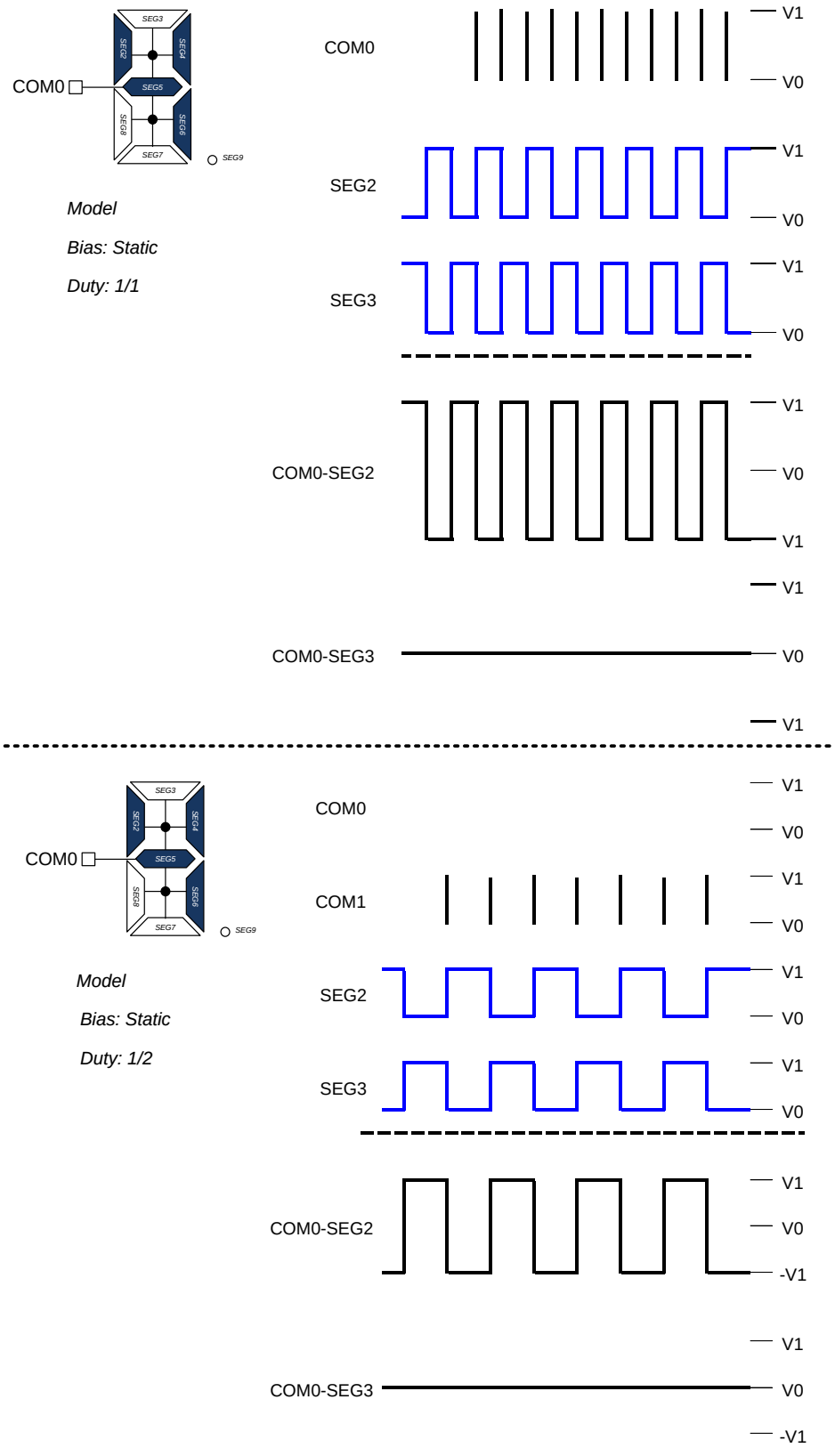


Figure 20-42(a) Output Waveform-Static Operation

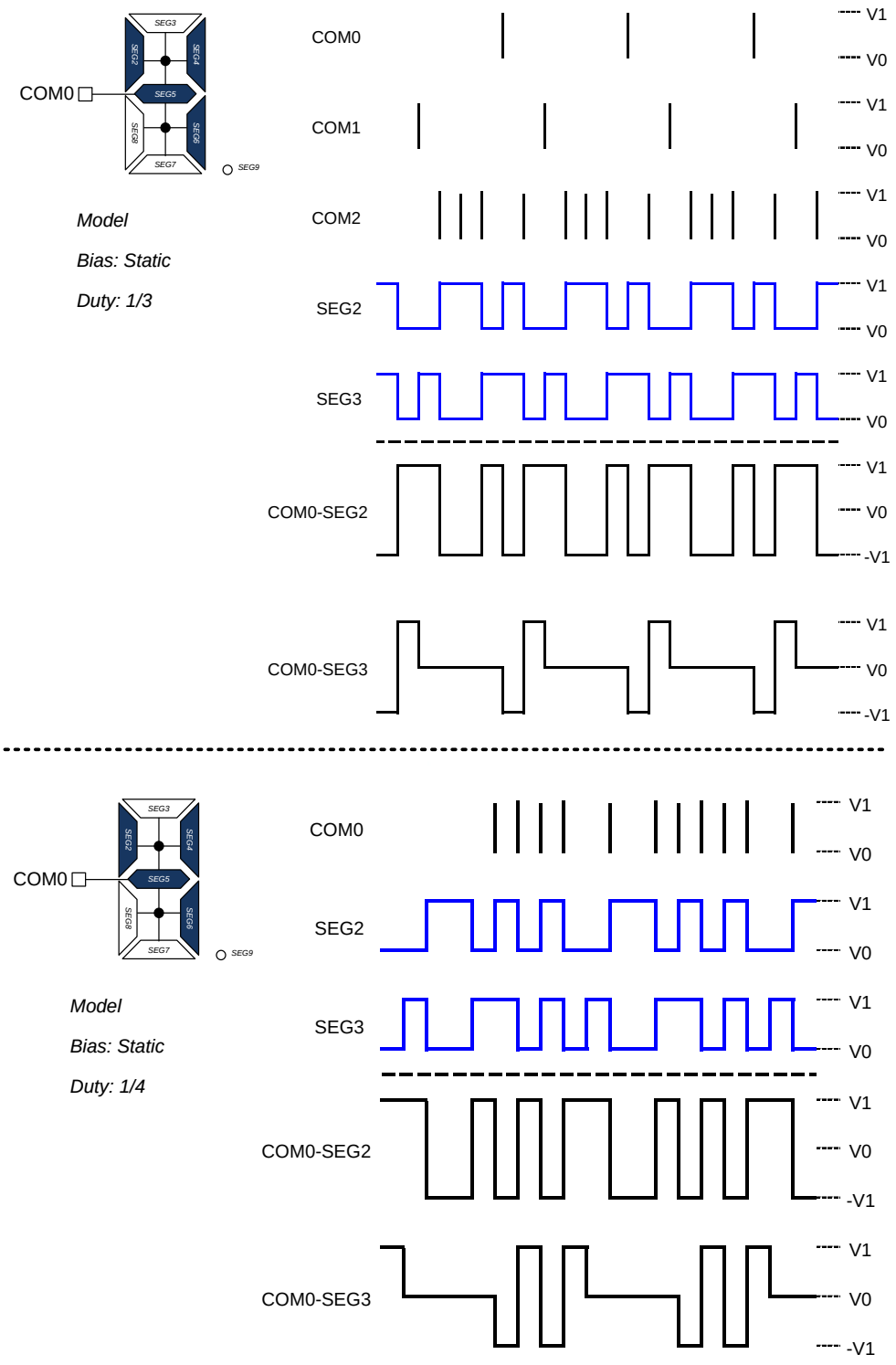


Figure 20-3(b) Output Waveform-Static Operation (continued)

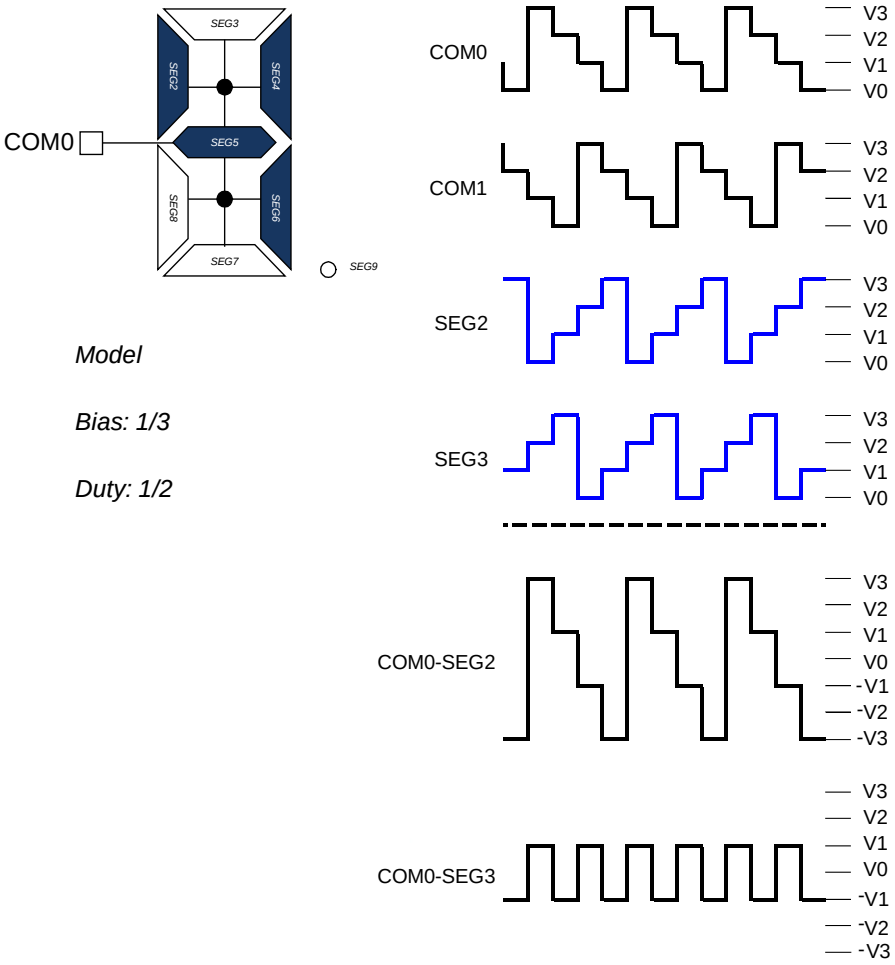


Figure 20-43 Output Waveform -2-Mux

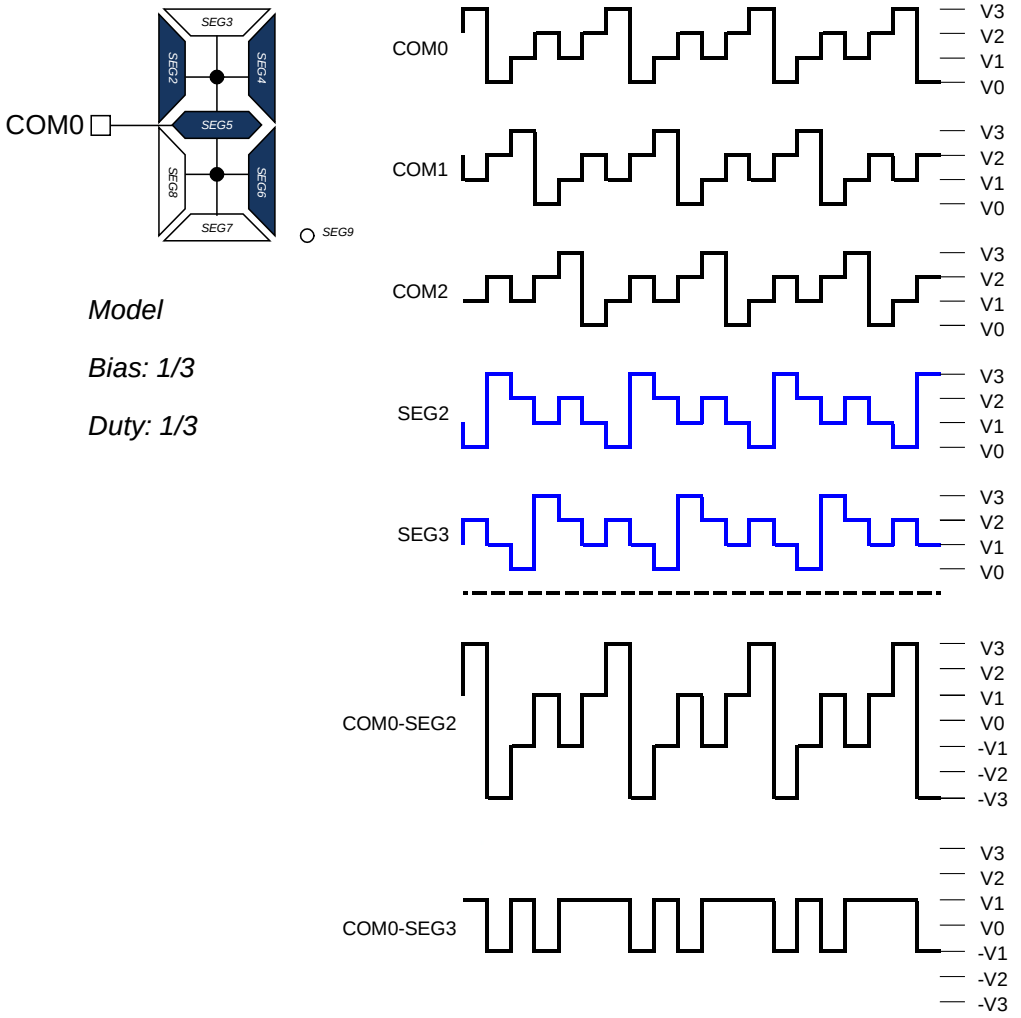


Figure 20-44 Output Waveform-3-Mux

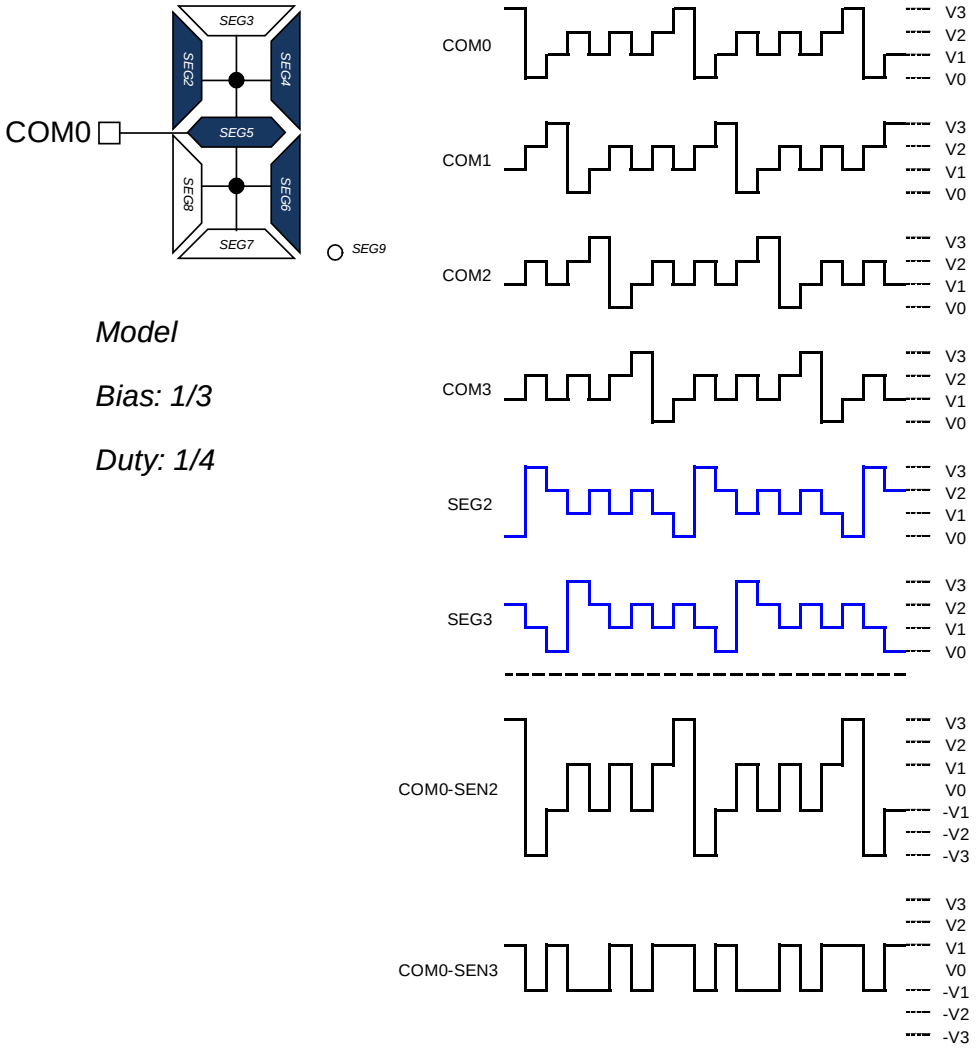


Figure 20-45 Output Waveform-4-Mux

20.3. Register Description-LCD

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
33H	MCKCN3	LCDS[2:0]								000. 0000	000. 0000	*,*,*,-,*,*,*
52H	LCDCN1	ENLCD	LCDPR	VLCDX[1:0]		LCDBF	LCDBI[1:0]			0000 000.	0000 000.	*,*,*,*,*,*,-,-
53H	LCDCN2	LCDBL	LCDMX[1:0]							000.	000.	*,*,*,-,-,-,-,-
54H	LCD0	Segment SEG2@[3:0] and SEG3@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
55H	LCD1	Segment SEG4@[3:0] and SEG5@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
56H	LCD2	Segment SEG6@[3:0] and SEG7@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
57H	LCD3	Segment SEG8@[3:0] and SEG9@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
58H	LCD4	Segment SEG10@[3:0] and SEG11@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
59H	LCD5	Segment SEG12@[3:0] and SEG13@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5AH	LCD6	Segment SEG14@[3:0] and SEG15@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5BH	LCD7	Segment SEG16@[3:0] and SEG17@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5CH	LCD8	Segment SEG18@[3:0] and SEG19@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5DH	LCD9	Segment SEG20@[3:0] and SEG21@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
180H	LCD10	Segment SEG22@[3:0] and SEG23@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
181H	LCD11	Segment SEG24@[3:0] and SEG25@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
182H	LCD12	Segment SEG26@[3:0] and SEG27@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
183H	LCD13	Segment SEG28@[3:0] and SEG29@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
184H	LCD14	Segment SEG30@[3:0] and SEG31@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
185H	LCD15	Segment SEG32@[3:0] and SEG33@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
186H	LCD16	Segment SEG34@[3:0] and SEG35@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
187H	LCD17	Segment SEG36@[3:0] and SEG37@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
188H	LCD18	Segment SEG38@[3:0] and SEG39@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
189H	LCD19	Segment SEG40@[3:0] and SEG41@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 20-45 LCD Register

MCKCN3：Please refer to Oscillator, Clock Sources and Power Managed Modes Chapter

LCDCN1：LCD Control Register 1

ENLCD：LCD start controller

1：Start

0：Shutoff

LCDPR：LCD charge pump circuit controller

1：Start；Voltage source, VLCD is generated from internal IC

0：Shutoff；Voltage source, VLCD is inputted from external pin.

VLCDX[1:0]：Charge pump voltage state select controller

11：VLCD = 2.55V。

10：VLCD = 2.8V。

01：VLCD = 3.05V。

00：VLCD = 3.3V。

LCDBF：LCD output buffer

1：Start

0：Shutoff

LCDBI[1:0]：LCD waveform bias controller

11：Unused

10：1/3 bias

01：Reserved

00：Static operation

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
33H	MCKCN3	LCDS[2:0]								000. 0000	000. 0000	*,*,*-*,*,*
52H	LCDCN1	ENLCD	LCDPR	VLCDX[1:0]		LCDBF	LCDBI[1:0]			0000 000.	0000 000.	*,*,*,*,*,*,-
53H	LCDCN2	LCDBL	LCDMX[1:0]							000.	000.	*,*,*- -,-,-,-
54H	LCD0	Segment SEG2@[3:0] and SEG3@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
55H	LCD1	Segment SEG4@[3:0] and SEG5@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
56H	LCD2	Segment SEG6@[3:0] and SEG7@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
57H	LCD3	Segment SEG8@[3:0] and SEG9@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
58H	LCD4	Segment SEG10@[3:0] and SEG11@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
59H	LCD5	Segment SEG12@[3:0] and SEG13@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5AH	LCD6	Segment SEG14@[3:0] and SEG15@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5BH	LCD7	Segment SEG16@[3:0] and SEG17@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5CH	LCD8	Segment SEG18@[3:0] and SEG19@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
5DH	LCD9	Segment SEG20@[3:0] and SEG21@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
180H	LCD10	Segment SEG22@[3:0] and SEG23@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
181H	LCD11	Segment SEG24@[3:0] and SEG25@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
182H	LCD12	Segment SEG26@[3:0] and SEG27@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
183H	LCD13	Segment SEG28@[3:0] and SEG29@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
184H	LCD14	Segment SEG30@[3:0] and SEG31@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
185H	LCD15	Segment SEG32@[3:0] and SEG33@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
186H	LCD16	Segment SEG34@[3:0] and SEG35@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
187H	LCD17	Segment SEG36@[3:0] and SEG37@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
188H	LCD18	Segment SEG38@[3:0] and SEG39@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*
189H	LCD19	Segment SEG40@[3:0] and SEG41@[7:4] data register of LCD								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

LCDCN2：LCD Control Register 2, LCD Control Register

LCDBL：LCD digit blink controller

1：LCD digit off

0：LCD digit display

LCDMX[1:0]：LCD Waveform output controller

00：Static status (COM0)

01：1/2 duty (COM0, COM1)；At this time, COM3=SEG1、COM2=SEG0

10：1/3 duty (COM0, COM1, COM2)；At this time, COM3=SEG1

11：1/4 duty (COM0, COM1, COM2, COM3)

LCD0~LCD19：LCD Digit Data Register

21. Serial Peripheral Interface

Serial Peripheral Interface, SPI equips with the following features：

- SPI module allows synchronously transmit and receive 8 bit data
- Can be utilized as serial interface of other devices. These devices mainly are EEPROM, shift register…etc.
- Can be used in master mode and slave mode.
- Master mode pin configuration is as listed below. The corresponding input/output pin must be configured while using.
  - Serial data output, SDO(PT1.5)
  - Serial data input, SDI(PT1.2)
  - Serial clock source, SCK(PT1.6)
- When in slave mode, slave synchronous pin, SCE (PT1.1) can be applied to control.

SPI Registers：

SSPCON1	SSPEN[0],CKP[0],CKE[0],SMP[0],SSPM[1:0]
SSPSTA	SSPBUY[0],SSPOV[0],BF[0]
SSPBUF	SSPBUF[7:0]

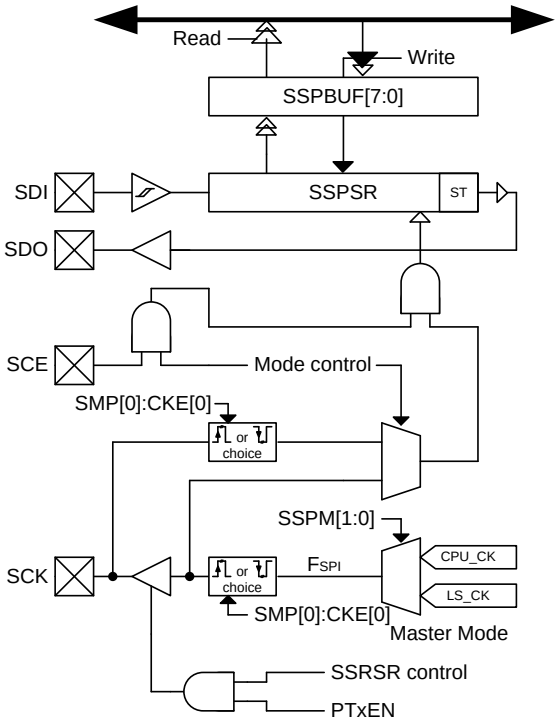


Figure 20-46 SPI Block Diagram

21.1. SPI Manual

Not only the corresponding pin must be configured as input/output pin while using, the pin will be different in diverse modes.

- Master Mode：Using SCK (clock output), SDI (data input), SDO (data output).
- Slave Mode：Using SCK (clock input), SDI (data input), SDO (data output), SCE (enable synchronous data receiving)

SPI control bit can be configured through register, SSPCON1：

- SPI is activated by configuring SSPEN bit.
- Configuring CKP bit, SCK potential status is determined after transmission.
- Configuring CKE bit, transmission data is determined to be in SCK rising edge or falling edge.
- Configuring SMP bit, input value sampling time can be determined (sampling in the middle or at the end of clock).
- Configuring SSPM[1:0] byte, to determine whether master mode SCK frequency source or passive mode SCE pin is enabled.

Through configuring register, SSPBUF, system can decide whether to transmit or receive data. Register, SSPSTA will reflect transmit and receive status:

- SPI transmit and receive register is composed by register, SSPBUF and SSPSR.
- SSPBUF will keep the last written value in SSPSR until the next received data is ready. When 8 bit receiving data is accomplished, the data will be move to register, SSPBUF. And BF bit of register, SSPSTA and SSPIF bit of register, INTF2 will be configured as 1.
- Double register allows to Read SSPBUF and receive the next data in register, SSPSR at the same time. When data received, BF bit of SSPSTA register must be determined first. If the value is 1, it means that SSPBUF has received data but has not been read yet. Users must read the SSPBUF data first. After reading, BF bit will be automatically cleared as 0 by hardware. If users do not read out SSPBUF data when BF is being configured as 1, when next data is received, SSPOV bit of register, SSPSTA will be configure as 1. And the data will be missing.
- When data transmission, SSPBUY bit of register, SSPSTA will be configured as 1. Any written action into register, SSPBUF will be ignored. After data transmission finished, SSPBUY bit will automatically be cleared as 0.
- When SPI is in master mode, the received value of register, SSPBUF can be ignored. It is only needed to write in the specified transmitting data.

21.2. SPI Master Mode

When SPI is designated in master mode, data transmission can be carried out at any time. After data is written into register, SSPBUF, the data will be outputted by SDO pin along with SCK clock source.

At this time, if the received data module is the same as SPI, the SDO pin configuration of SPI slave module, can be configured as input pin, no ineffective data will be transmitted to master module. If the received module must transmit data synchronously to master module, SDO pin configuration of slave module can be designated as output pin. Then, SDI pin of master module will receive move in data continuously. After data received, it will be write into register, SSPBUF. The corresponding BF bit of register, SSPSTA and SSPIF bit of register, NTF2 will be configured as 1.

SSPM[1:0] byte of register, SSPCON1 can configure master module frequency source, and CKE, CKP configuration can determine clock source polarity.

Master mode configuration description:

- Drive pin setup: SCK (clock output), SDI (data input), SDO (data output).
- Be sure to correctly set I/O input/output function and SCK, SDO output module function.
- Allocate registers, SSPCON1 to reach SPI function control bit.
- Set CKP bit, determining potential status of SCK after transmission.
- Set CKE bit, determining data transmission to SCK's arising/falling edge.
- Set SMP bit, determining input data sampling time (in middle or end of sampling time)
- Set SSPM[1:0] bit, determining master mode SCK frequency source
- Set SSPEN bit to start SPI communication module.
    - Allocate register, SSPBUF to decide transmission data, register SSPSTA reflects transmission status
- SPI transmission register is constituted by SSPBUF and SSPSR register.
- After data was written into SSPBUF, hardware will move SSPBUF data to register, SSPSR and the data of SSPSR will be sent through SCK clock source.
- After sent out the data, SSPSR synchronically receive input data from SDI pin. After transmission finished, hardware will move data of SSPSR to SSPBUF. BF flag helps to determine if the reception is completed.
- SSPBUF will keep last data received from SSPSR until the next received data is ready.
- After 8 bit data transmission completed, this data will be moved to SSPBUF and SSPIF bit of INTF2 register will be set as 1 when SCK finished transmission 8 bit data. And after BF bit of register, SSPSTA was moved by hardware to SSPBUF, it would be set as 1.

- When data transmission, SSPBUY bit of register, SSPSTA is set as 1. Any write in action to register, SSPBUF will be ignored. SSPBUY bit will be auto-0 after completion of data transmission.
- When SPI is in master mode, data received at register, SSPBUF can be neglected. Users only need to write in transmission data. If the data received by SSPBUF is usable, users should move data out by themselves and write in transmission data later.

Relative configuration is illustrated below:

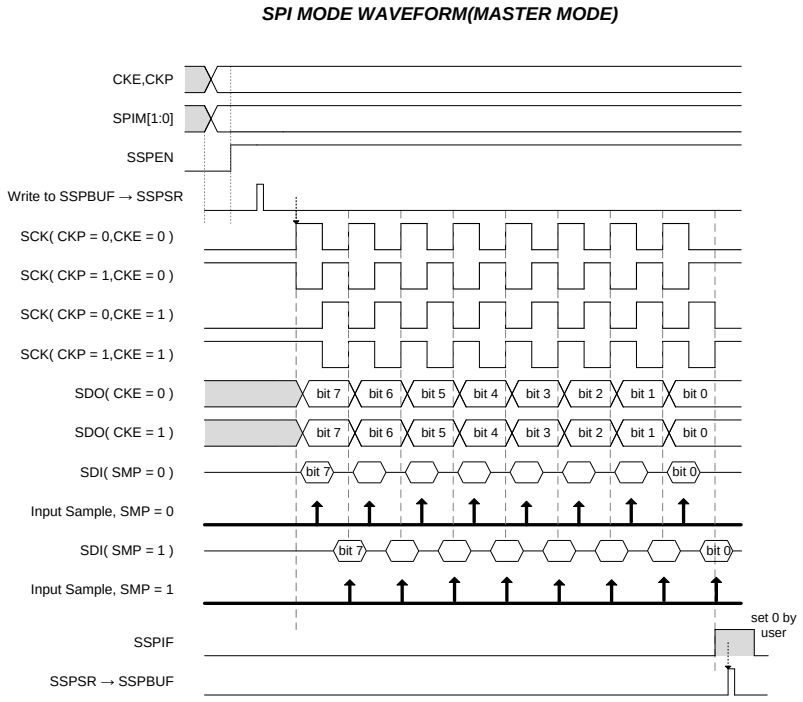


Figure 21-47 SPI Master Module Sequence Waveform

21.3. SPI Slave Mode

Under slave mode, SCK clock source is offered externally and SCK pin must be in idle status. SCK pin is configured as input pin. In configuring polarity of master module clock source, CKE and CKP bit configuration is used for determine polarity of slave module.

Slave mode configuration description:

- Drive pin setup：SCK (clock output), SDI (data input), SDO (data output), SCE (enable synchronous data reception)
  - Be sure to correctly set I/O input/output function and SDO output module function.
- Allocate registers, SSPCON1 to reach SPI function control bit
  - Set CKP bit, determining potential status of SCK after transmission.
  - Set CKE bit, determining data transmission to SCK's rising/falling edge.

- Set SMP bit, determining input data sampling time (in middle or end of sampling time)
- Set SSPM[1:0] bit, determining whether slave mode SCE pin is started.
- Set SSPEN bit to start SPI communication module.

◆ Allocate register, SSPBUF to decide receiving or transmitting synchronous data, register SSPSTA reflects reception status

- SPI reception register is constituted by SSPBUF and SSPSR register.
- Before receiving data, users should write synchronized transmission data to SSPBUF and wait the clock source of master end arrived.
- Even synchronized transmission data is not necessary, users need to finish writing SSPBUF action and it is recommended to write in data of 0FFh.
- Before write in synchronized transmission data and clock source of master end arrived, it is required to delay for 5 instruction cycle time for hardware to move SSPBUF data to right register, SSPSR.
- When master end SCK clock source inputted, except capturing data from SDI input pin, the slave module may synchronically output SSPSR data via SDO pin to the master end.
- SSPBUF will keep the last received data of SSPSR until the next received data is ready.
- After 8 bit data reception completed, this data will be moved to SSPBUF and SSPIF bit of INTF2 register will be set as 1 when SCK finished transmission 8 bit data. And after BF bit of register, SSPSTA was moved to SSPBUF by hardware, it would be set as 1.
- Dual register allows synchronously reading the received data (Read SSPBUF) and received the next data in register, SSPSR. When receiving data, users should judge whether BF bit of register, SSPSTA is set as 1. If it is 1, it represents SSPBUF has received complete data but has not been read by users. Users should read out SSPBUF data first, and then BF bit will be erased by hardware as 0. If users did not readout SSPBUF data when BF is configured as 1, SSPOV bit will be set as 1 while receiving data and the data will be lost, never be written into register, SSPBUF.

Relative configuration is given in below：

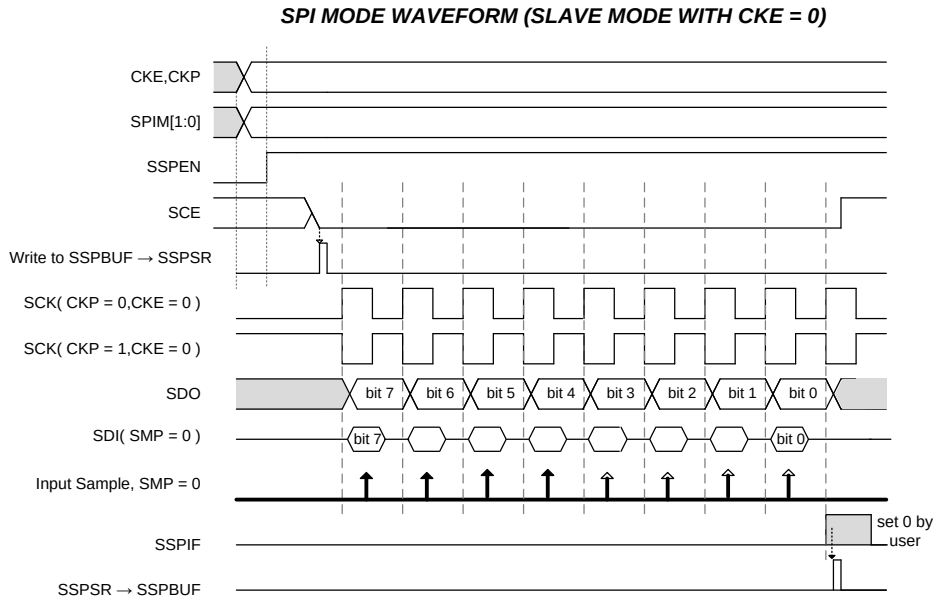


Figure 21-48 SPI Slave Module Sequence Waveform (CKE=0)

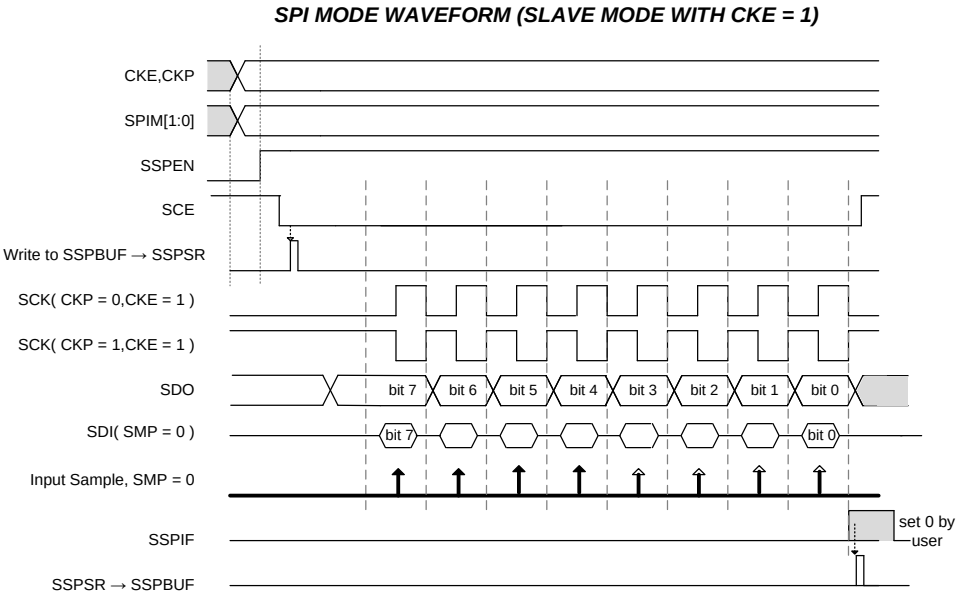


Figure 21-49 SPI Slave Module Sequence Waveform (CKE=1)

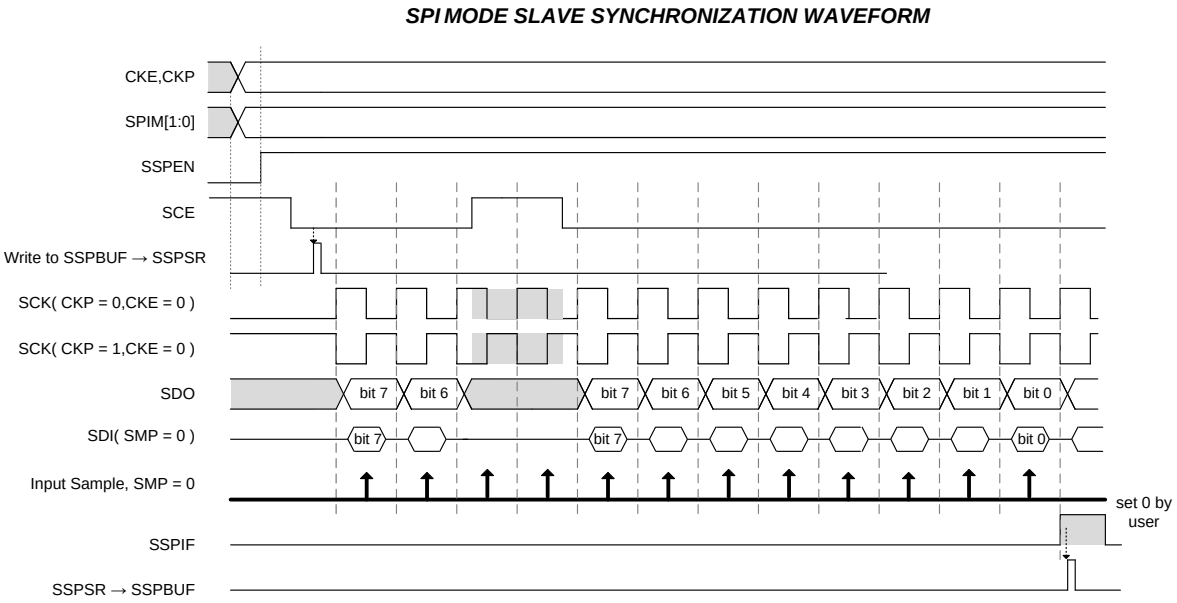


Figure 21-50 SPI Slave Module Synchronous Sequence Waveform

Under Sleep mode (SLP MODE), if SSPIE bit of register, INTE2 interrupt is enabled, the IC will be awakened after receiving 8 bit data.

Another SCE pin in slave module, the configuration of SCE pin allows slave synchrony mode. SSPM[1:0] byte of register SSPCON1 can configure the pin.

When SCE pin is in low state, data can be received or transmitted regularly, SDO pin can operate normally as well. When SCE is in high state, SDO output pin will be pended not drive.

21.4. SPI Master Module Transmission Way

The figure hereinafter shows 2 sets of SPI module master/slave connected way of HYCON processor:

- Master module will transmit register, SSPBUF data through shift register, SSPSR and output data through SDO. While data transmission, it can synchronously receive data that transmitted from slave module to shift register, SSPSR. After finished, the data will be written into register, SSPBUF.
- Slave module will save the received data in shift register, SSPSR. After the receiving is done, the data will then be written into register, SSPBUF.

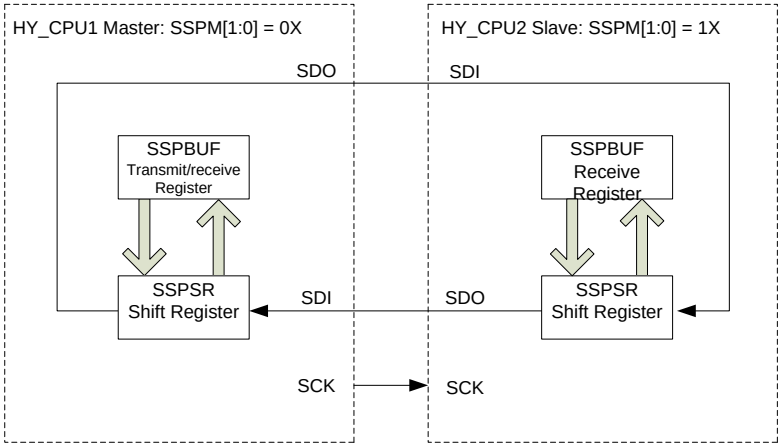


Figure 21-51 2 Processor Sets, SPI Master/Slave Mode Connected Way

```
CLRF    PT1DA,0        ；Set PT16(SCK), PT15(SDO) as digital output pin
                       ；Set PT12(SDI) as digital input pin
CLRF    PT1PU,0
MVL     060H
MVF     TRISC1,1,0
MVL     00010100b
MVF     PT1M2,1,0      ；Set SCK, SDO function
BSF     INTE2,2,0      ；Set SSPIE interrupt service
MVL     080H           ；Start SPI function, set frequency source as CPU_CK
MVF     SSPCON1,1,0    ；Set as Master module, allocating CKP, CKE setup waveform of frequency source
MVL     055H           ；Write in data to master module
MVF     SSPBUF,1,0
….

SPI Interrupt：        ；SPI interrupt event service program
BCF     INTF2,SSPIF,0  ；Clear SPI interrupt event flag
BTSS    SSPSTA,BF,0
RJ      SPI Interrupt  ；Determine whether the reception of slave module has been finished synchronously
MVFF    SSPBUF,BUF0    ；Move the received data from slave module to register, BUF0
…

RETI                   ；Return from interrupt service
```

Example 21-18 SPI Master Mode Example Program

```
CLRF    PT1DA,0         ；Set PT15(SDO) as digital output pin
                        ；Set PT16(SCK), PT12(SDI) digital input pin
CLRF    PT1PU,0
MVL     020H            ；PT1.5（SDO),PT.6(SCK),PT1.2(SDI),PT1.1(SCE)
MVF     TRISC1,1,0
MVL     00000100b       ；Set SDO function
MVF     PT1M2,1,0
BSF     INTE2,SSPIE,0   ；Enable SPI interrupt
MVL     11000011b       ；Start SPI, falling edge data transmission
MVF     SSPCON1,1,0     ；Set as slave mode and start SCE pin control function
MVF     SSPBUF,0,0      ；Read out data of SPI buffer zone
CLRF    SSPSTA,0
MVL     05Ah            ；Write in data to SSPBUF, preparing to transmit synchronously
MVF     SSPBUF,1,0
NOP                     ；Other program operation
.............
SPI_interrupt:
BCF     INTF2,SSPIF,0
BTSS    SSPSTA,BF,0
RJ      SPI_Interrupt   ；Determine whether data is received completely
MVFF    SSPBUF,BUF0     ；Save data sent by master end
BTSZ    SSPSTA, BF,0
RJ      SPI_InterruptA  ；Determine whether another data is inputted while receiving
BTSZ    SSPSTA, SSPOV,0
RJ      SPI_InterruptB  ；Determine whether data overflow
BCF     SSPSTA, SSPOV,0         ；If conflict occurred, it must be cleared
MVL     069h            ；Write in data to SSPBUF，
                        ；Ready to receive the next data sent by the master end and sent
                        out the date synchronically
                        ；If synchronous transmission is no need, please fill in 0FFh
MVF     SSPBUF,1,0      ；Data written in SSPBUF is the same as the sent out data
SPI_InterruptA:
…
SPI_InterruptB:
…
RETI
```

Example 21-2 SPI Slave Mode Example Program

21.5. Register Description-SPI

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	E1IE	E0IE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2						SSPIE			00.. 0000	00.. 0000	*,*,-,- *,*,*,*
27H	INTF2	TXIF	RCIF			CPOIF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,- *,*,*,*
5EH	SSPCON1	SSPEN	CKP	CKE	SMP			SSPM<1:0>		0000 ..00	uuuu ..uu	*,*,*,* -,-,*,*
60H	SSPSTA	SSPBUY	SSPOV						BF	00.. ...0	00.. ...0	r,r,-,- -,-,-,r
61H	SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

Table 21-46 SPI Register

SSPCON1: SPI Control Register

SSPEN：SPI enable bit

1：Start SPI and configure SCK, SDO, SDI SCE pin sets for serial interface utilization

0：Shutoff SPI and configure SCK, SDO, SDI SCE pin sets for I/O utilization

Notice：After starting SPI, input/output pin must be configured appropriately.

CKP：SPI clock source polarity select bit

1：Configure high state clock source as idle status

0：Configure low state clock source as idle status

CKE：SPI clock source select bit

1：Transmit when clock changes from effective status to idle status

0：Transmit when clock changes from idle status to effective status

Notice：Clock source polarity is determined by CKP bit (SSPCON1<6>)

SMP：SPI sample bit

SPI Master Mode:

1：Sampling input data at the end of transmission time

0：Sampling input data in the middle of transmission time

SPI Slave Mode:

Users must clear SMP BIT as 0

SSPM[1:0]：SPI mode select byte

00：SPI Master Mode, clock source = LS_CK

01：SPI Master Mode, clock source = CPU_CK

10：SPI Slave Mode, clock source = SCK pin, close SCE pin control function, SCE pin set is for I/O usage

11：SPI Slave Mode, clock source = SCK pin, start SCE pin control function

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."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**
23H	INTE1	GIE	ADCIE	TMCIE	TMBIE	TMAIE	WDTIE	EXIE	EOIE	0000 0000	0000 0000	*,*,*,*,*,*,*,*
24H	INTE2						SSPIE			00.. 0000	00.. 0000	*,*,-,- *,*,*,*
27H	INTF2	TXIF	RCIF			CP0IF	SSPIF	CCP1IF	CCP0IF	00.. 0000	00.. 0000	*,*,-,- *,*,*,*
5EH	SSPCON1	SSPEN	CKP	CKE	SMP			SSPM<1:0>		0000 ..00	uuuu ..uu	*,*,*,* -,-,*,*
60H	SSPSTA	SSPBUY	SSPOV						BF	00.. ...0	00.. ...0	r,r,-,- -,-,-,r
61H	SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu	*,*,*,*,*,*,*,*

SSPSTA：SPI Status Register

SSPBUY：SPI write in collision detect bit (only for data transmission use)

1：The data is still in transmitting status

0：No collision

SSPOV：SPI receive overflow flag

SPI Slave Mode:

1：SSPBUF still holding the previous data and a new byte is received. Once SSPSR overflowed, register, SSPSR data is lost. SSPOV only occur in slave mode. The user must read register, SSPBUF data, even if only transmitting data, to avoid setting SSPOV as 1 (must be cleared by instruction).

0：No overflow

Notice: In master mode, SSPOV bit will not be configured as 1. Every transmitted (reception) data must be written to the register, SSPBUF.

BF：Buffer full status bit (only for data reception)

1：Reception accomplished, SSPBUF is full

0：Reception not accomplished, SSPBUF is empty

SSPBUF：SPI Receive Buffer Register or Transmit Register

22. Enhanced Universal Asynchronous Receiver Transmitter

Enhanced Universal Asynchronous Receiver Transmitter, EUART peripheral is usually called serial communications interface or SCI. The EUART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROM, etc.

The Enhanced EUART implements additional features, including Frame error detection and auto address identification. Frame error detection can determine whether a frame is effective or not through frame stop bit. Auto address identification function can compare address frame content with single chip address; serial interrupt can only be generated when matching. These 2 functions are implemented through hardware circuit and software respectively.

EUART can be configured in the following modes：

- Asynchronous (full-duplex) with：
  - Auto-baud rate generator
- Frame Error Mode
  - Frame error detection[29]
  - Overrun error detection[30]
  - Hardware same bit check code(Parity check code)
- Data Transmission and reception
  - Asynchronous transmission(8 bit or 9 bit)
  - Asynchronous receive(8 bit or 9 bit)
- Auto-wake-up on character reception

EUART Registers：

URCON	ENSP[0],ENTX[0],TX9[0],TX9D[0],PARITY[1:0]
URSTA	RC9D [0],PERR[0],FERR[0],OERR[0],RCIDL[0],TRMT[0],ABDOVF[0]
BAUDCON	ENCR[0],RC9[0],ENADD[0],ENABD[0]
BRGR[15:0]	BRGRH[7:0], BRGRL[7:0]
TXREG	TXREG[7:0]
RCREG	RCREG[7:0]

[29] Frame error detection (FERR)：UART does not receive the initial bit that usually aroused from the noise on signal line. UART cannot ontain correct data from shift register.

[30] Overflow error detection (OERR)：The latest data has covered the previous data.

22.1. EUART Manual

22.1.1. To Set up an Asynchronous Data Transmission

- Configure TRISC1 of register, TX as output pin. Configure TX pin out mode of PT1M2 register.
- Configure TXIE bit of INTE2 register and GIE bit of INTE1 register to determine whether to allow transmit enabled interrupt (TXIF bit of INTF2 register is default as High, related enabled interrupt must be configured after confirmation).
- Configure BRGRH, BRGRL register to define appropriate baud rate.
- Configure ENSP bit of URCON register to start EUART serial I/O module.
- Configure TX9 bit of URCON register to decide whether the 9th bit data transmit function is enabled (If the function is enabled, the data must be written into TX9D bit. The 9th bit can be address or data).
- Configure ENTX bit of URCON register to start data transmission function.
- Write the data to TXREG register; determine to transmit data (Start transmitting after writing the data).

22.1.2. To Set up an Asynchronous Data Reception

- Configure TRISC1 register, set RC as input pin.
- Configure RCIE bit of INTE2 register and GIE bit of INTE1 register to determine whether to allow enabled interrupt reception.
- Configure BRGRH, BRGRL register, determine the appropriate baud rate.
- Configure ENSP bit of URCON register to start EUART serial I/O module.
- Configure RC9 bit of BAUDCON register to determine whether to enable the 9^{th} bit data reception function.
- Configure ENCR bit of BAUDCON register to start data reception function.
- Read RC9D bit of URSTA register to obtain the 9^{th} bit value and determine if error occurred during reception processes.
- Read RCREG registers to obtain the received 8 bit data.
- Read whether the FERR bit of URSTA register is configured. Make sure if the read data is wrong. FERR can be cleared through ENCR bit.

22.1.3. To Set up an Asynchronous Data Reception (9 Bit, RS-485 Mode)

- Configure TRISC1 register. Set RC as input pin.
- Configure BRGRH, BRGRL register, to determine appropriate baud rate.
- Configure ENSP bit of URCON register to start EUART serial I/O module.
- Configure RC9 bit of BAUDCON register and determine whether to start the 9^{th} bit data reception function.
- Configure ENADD bit of BAUDCON register enabled address detect function.
- Configure ENCR bit of BAUDCON register to start data reception function.

- Configure RCIE bit of INTE2 register and GIE bit of INTE1 register to determine whether to allow enabled interrupt reception. When data reception accomplished, RCIF bit will be configured.
- Read RC9D bit of URSTA register to obtain the 9th value of the received data. Determine whether error occurred during reception processes.
- Read RCREG registers to obtain 8 bit reception data.
- Read whether FERR bit of URSTA register is configured. Make sure the creativeness of the read data. FERR bit can be cleared through ENCR bit.
- Configure ENADD bit of BAUDCON register to shut off address detect, making the next data as reception.

22.2. Baud Rate Generator, BRG

BRG is a dedicated 13 bit generator that supports asynchronous mode of the EUART. BRGR[15:0] register that controls the period of a free-running timer. Table 22-1 is the equation of baud rate computation that only applies in Master Mode.

Given the desired baud rate and the operating clock source is OSC_HAO, the nearest integer value for BRGR[15:0] register can be calculated using the equation in Table 22-1. From this, the error of baud rate is determined.

BRG/EUART MODE	Baud Rate Equation
13 bit/ asynchrony	OSC_HAO÷[4 (n + 1)]
OSC_HAO= operating frequency； n = BRGRH:BRGRL register correct value	

Table 22-47 Baud Rate Equation

Operating under asynchronous mode, the operating frequency is OSC_HAO(2MHz). And the desired baud rate is 9600bps. What is BRGR[15:0]=< ? > BRGRH[7:0]:BRGRL[7:0]=< ? >

The known equation：desired baud rate＝OSC_HAO÷(4 (<BRGR[15:0]>+1))：

Therefore BRGR[15:0]＝((OSC_HAO÷ desired baud rate)÷4)–1

＝((2000000÷9600)÷4)–1

＝51.08

≒51 means BRGRH[7:0]=<00>, BRGRL[7:0]=<33>：Note: 33 is hexadecimal

Actually, BRG calculated result is：actual baud rate＝2000000÷4(51+1))＝9615.38

Certain error exists, the computed way is：

Error rate＝(actual baud rate– desired baud rate)/ desired baud rate

＝(9615-9600)/9600

＝0.16%

Example 22-19 Calculation of Baud Rate Error

22.2.1. Operation in Power Managed Modes

The IC clock is used to generate desired baud rate. When one the power managed modes is entered, the new clock source may be operating at different frequency. This may require adjustment to the value of BRGR[15:0] register.

22.2.2. RC Sampling

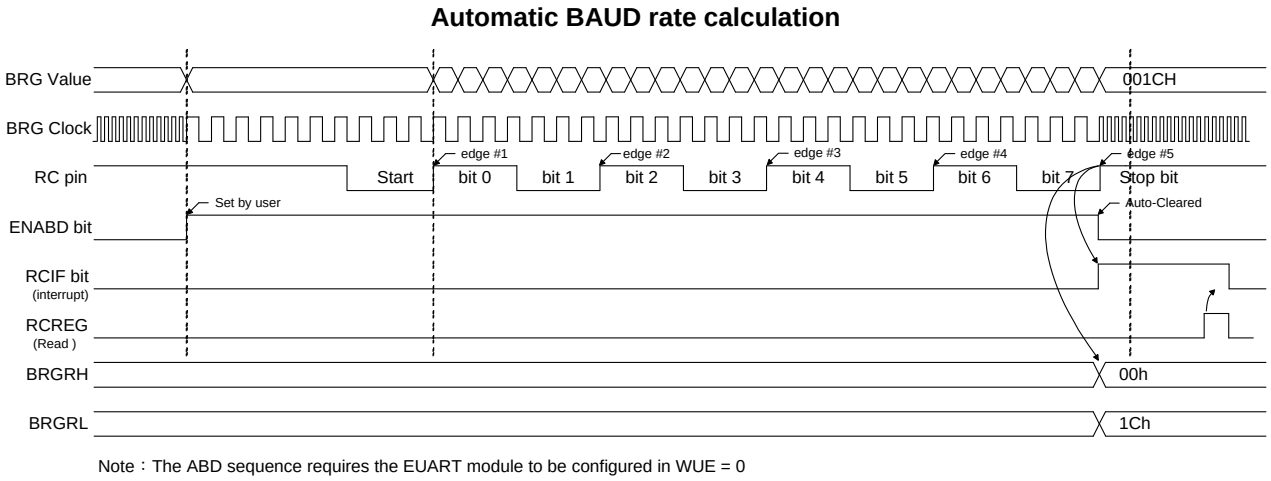
Sampling circuit will conduct sampling in the center point of baud rate period to determine if a high or a low level is presented at the RC pin.

22.2.3. Auto Baud Rate

EUART module supports auto detection and calibration of baud rate. Auto baud rate is only active in awakening start controller; WUE[0] is cleared. To start this function, ENABD[0] must be set as 1.

Auto baud rate detection begins whenever a start bit is received (the reception value must be 055H). After auto detection and calibration finished, the calculated result will be written to register, BRGRH[7:0] and BRGRL[7:0]. Relative sequences are detailed in Figure 22-1.

When BRGR[15:0] overflowed, the content generates overflow from 01FFFH to 00000H, auto baud rate flag will be placed 1. Users can clear ABDOVF[0] by instruction or through clearing the ENABD[0] configuration. After ABDOVF[0] is set up as 1, ENABD[0] status will remain as 1. Relative sequences are detailed in Figure 22-2.



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

Figure 22-52 Auto Baud Rate Calculation Waveform

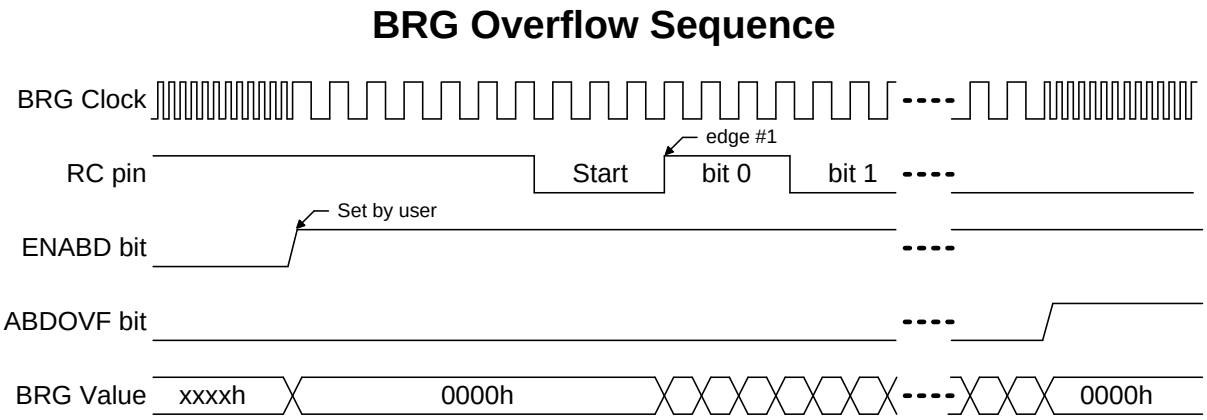


Figure 22-53 Auto Baud Rate Overflow (ABDOVF) Waveform

22.3. Hardware Parity Check

EUART supports hardware odd/even calibration function; the calibrated bit is saved in the 9th data bit. Parity check (ENADD[0]=0) is carried out in accordance with registers configuration, relative configurations are shown in Table 22-2.

Transmit/Receive 8/9 bit data		PARITY	Status
TX9	RC9		
0	0	0	Transmit/Receive data no parity check message
0	0	1	Transmit/Receive data no parity check message
0	1	0	Data received has parity l check function, even parity
0	1	1	Data received has parity check function, odd parity
1	0	0	Data transmitted has parity check function, even parity
1	0	1	Data transmitted has parity check function, odd parity
1	1	0	
1	1	1	
Note：When iRC9[0] is configured as 1, parity check function is started. When even/odd parity error, PERR[0] is configured as 1. If RC9[0] and ENADD[0] is configured as a in the same time, do not care error value of PERR[0] bit.			

Table 22-48 Parity Check Status

22.4. EUART Asynchronous Mode

In this mode, the EUART uses standard "Non-Return-to-Zero, NRZ" format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bit. An on-chip dedicated 13-bit Baud Rate Generator can be applied to derive standard baud rate frequencies from the oscillator.

Moreover, the EUART transmits and receives the last LSB. The transmitter and receiver are functionally independent but use the same data format and baud rate. Parity is supported by hardware and can be stored as the 9th data bit.

22.4.1. EUART Asynchronous Transmitter (Interrupt is generated from UART TXIF/RCIF flag, 0->1)

Figure 22-3 is the sequence of UART transmitter. The core of the transmitter is Transmit Shift Register, TSR; users cannot read/write TSR.

TSR obtains data from the Read/Write Transmit Buffer register, TXREG[7:0]. The TXREG[7:0] register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. Once the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

As soon as the TXREG register transfer the data to the TSR register, the TXREG register is empty and TXIF flag is set from 1 to 0 (when ENTX bit of URCON register is configured, TXIF bit will be configured as 1). TXIF will not be cleared immediately when new data is loaded. Instead, it is cleared in the second instruction period following the load instruction. TXIF will be configured as 1 again at the end of one instruction cycle. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE. TXIE will be set regardless of the state of TXIE; when interrupt occurs, TXIF will be configured from 1 to 0 and cannot be cleared in software. TXIF will be configured as 1 again at the end of one instruction cycle. If TSR register data has not been transmitted from the previous load and data has been written into TXREG register. TXIF is cleared in the second instruction period following the load instruction. TXIF will be configured as 1 again when Stop bit occurred.

Hence, after new data is loaded into TXREG register, TXIF will indicate the status of TXREG. Another bit, TRMT, presents TSR register status. TRMT is only readable, it will be set up as 1 when TSR register is empty (no loaded action). TRMT bit has no connection with any interrupt logic. Hence, users can only check to the status of TSR to determine whether it is empty. Asynchronous data transmission sequence can be referred to Figure 22-4 and Figure 22-5.

- UART actions are irrelevant to CPU instruction cycle except read and write action.
- TXIF and RCIF is for interrupt purpose, they are irrelevant to other events.
- When using CPU to monitor peripheral components, be cautioned about the corresponding operating speed.

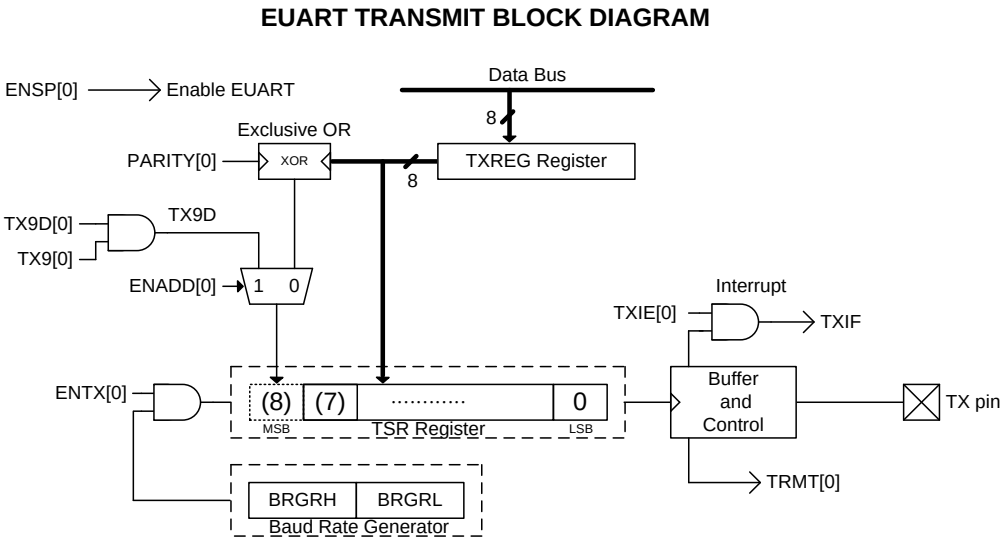


Figure 22-3 EUART Transmission Block Diagram

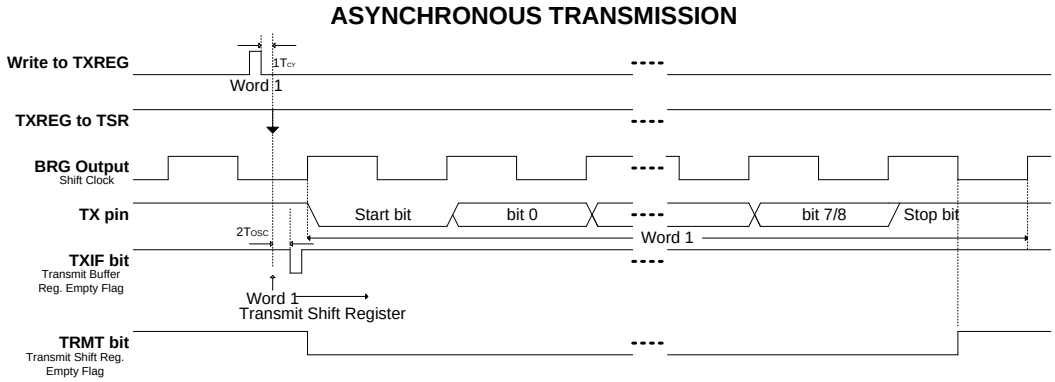


Figure 22-4 Asynchronous Transmission Sequence

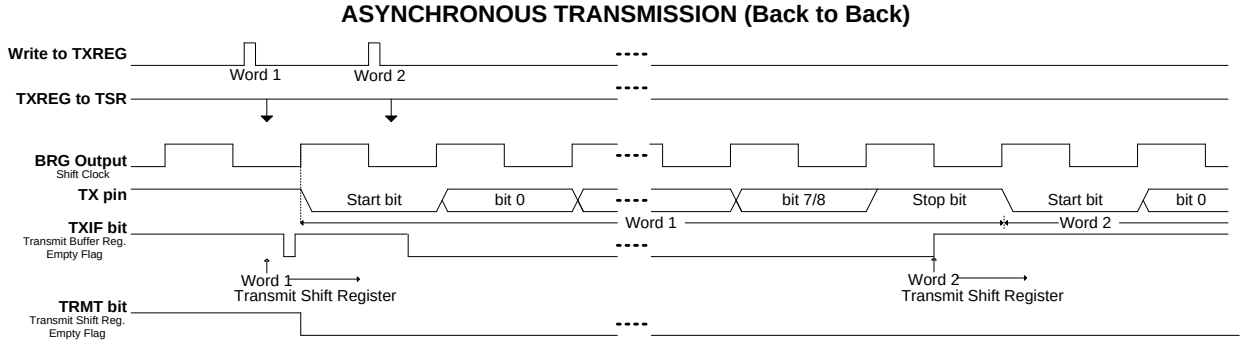


Figure 22-5 Asynchronous Transmission Sequence (Back to Back)

22.4.2. EUART Asynchronous Receiver

Figure 22-6 and 22-7 indicates receiver block diagram while Figure 22-8 illustrates asynchronous receive sequence. The data is received on the RC pin drives the data recovery circuit. Data recovery circuit is actually a high-speed shifter that operating at 13-bit auto baud rate, whereas the main receive serial shifter operates at baud rate or at OSC_RC2M. This mode is typically be used in RS-232 systems.

If RC pin does not receive complete byte (Start bit, 8(9) bit data, Close bit), FERR bit will be set as 1 and it can be cleared by ENCR bit.

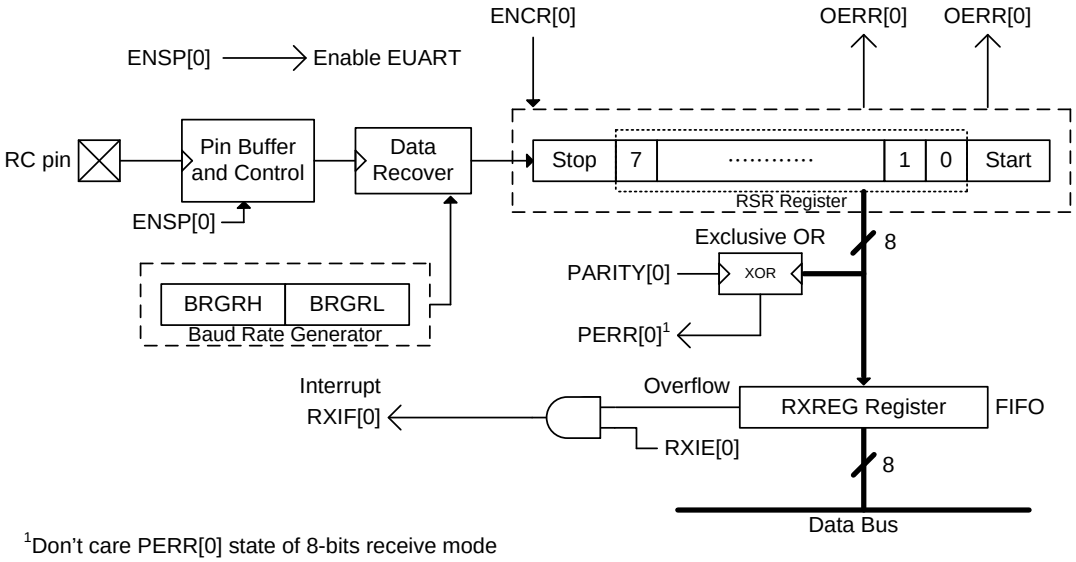
When RC pin has received two complete byte data (have not read the data of RCREG register), OERR bit will be configured as 1. OERR bit can be cleared by ENCR bit.

When complete data reception accomplished, RCIF bit of INTF2 register will be set up. Instructions cannot clear RCIF bit once it has been configured. RCIF status can be cleared by reading RCREG register.

RCIDL bit of URSTA register will reflect if it is in reception status. By determining the status, users can know whether the data reception has been completed.

When receiving data, hardware will conducts the received 8 bit data exclusive or. If RC9 is set as 1, the received RC9D data (9 bit) will be calculated by exclusive or. After operation, the result will be calculated again by exclusive or with PARITY bit and it will be displayed in PERR bit. If the received data is correct, PERR is configured as 0. Conversely, if the data received is incorrect, PERR will be set as 1. PERR bit cannot be cleared in software. PERR will be set as 0 whenever the next data is being correctly received.

EUART 8-BITs RECEIVE BLOCK DIAGRAM



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

Figure 22-6 EUART 8-Bit Receive Block Diagram

EUART 9-BITs RECEIVE BLOCK DIAGRAM

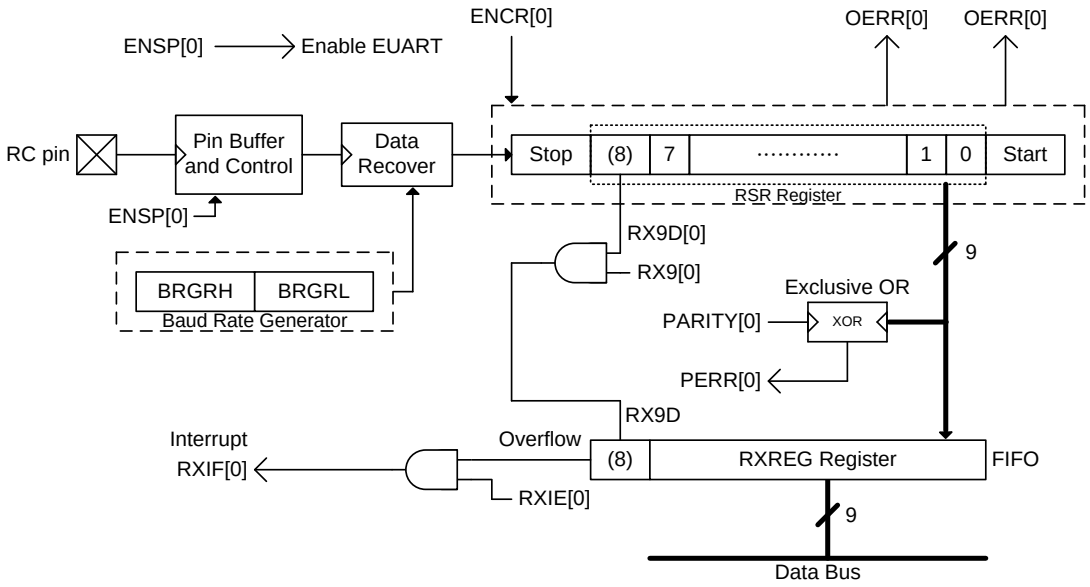


Figure 22-7 EUART 9-Bit Receive Block Diagram

ASYNCHRONOUS RECEPTION

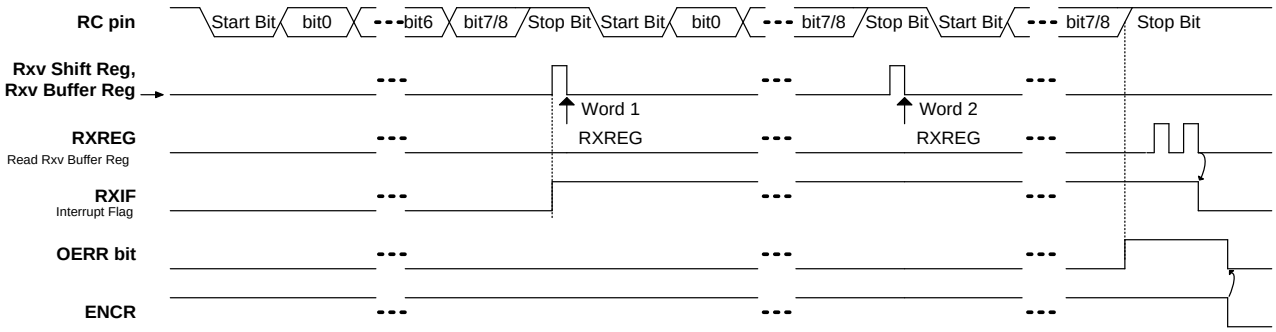


Figure 22-8 Asynchronous Receive Sequence

22.4.3. 9 Bit Address Detect Mode

This mode is typically be used in RS-485 systems. Users can operate asynchronous reception with reference to EUART Manual section. ENADD bit of BAUDCON register can be used to configure address detect or data detect.

22.4.4. Auto-Wake-Up on Character Reception

Under Sleep mode, all clocks to the EUART are suspended. As a result, the Baud Rate Generator is inactive and a correct byte reception cannot be conducted. The auto-wake-up function allows the controller to be awakened up when the activity on the RC line while the EUART is operating in Asynchronous mode. The auto-wake-up function is enabled by configuring the WUE bit of URCON register. After initiation, the typical receive sequence on

RC is disabled and the EUART remains in an Idle state, monitoring for a wakeup event（it is not related with CPU Run mode）.

A wake up event consists of a high state to low state transition on the RC line. Followed by a wake up event, the module generates an RCIF interrupt. The interrupt is generated synchronously to the Q clock in normal operating mode (Figure 22-9). If the IC is in Sleep or Idle mode, it is asynchronously (Figure 22-10).The interrupt is cleared by reading RCREG register.

After wake up event, when low state to high state transition occurs on the RC line, WUE bit is automatically be cleared. At this time, EUART module returns to normal Run mode from Idle mode. Users can know from the signals that the event is ended.

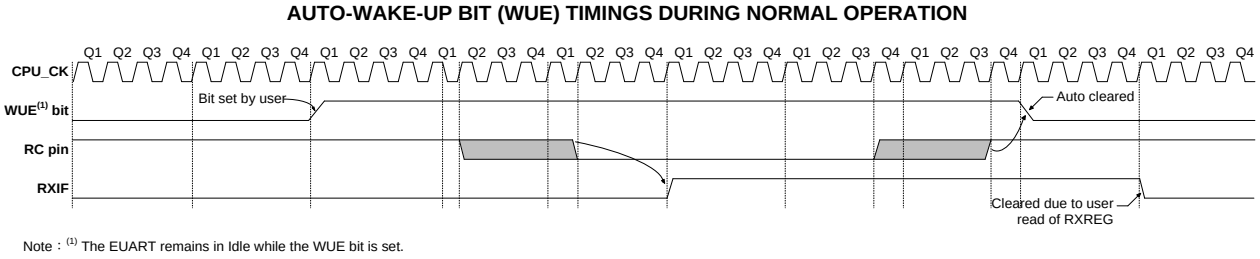


Figure 22-9 Auto-Wake-Up Sequence in Normal Mode

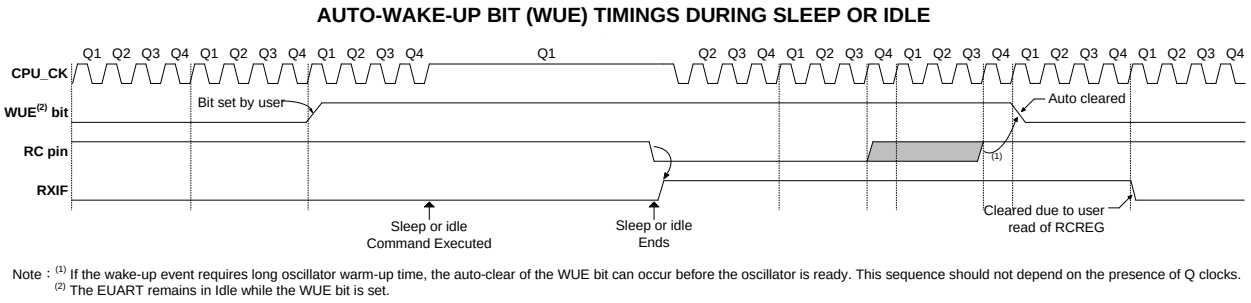


Figure 22-10 Auto-Wake-Up Sequence in Sleep or Idle Mode

22.4.4.1. Notice of Using Auto-Wake-Up Function

Due to the fact that auto-wake-up functions by sensing rising edge transitions on RC, information with any state changes before the Stop bit may output a false character and result in data or frame errors. Thus, the initial character in the transmission must be all "0" bit. This can be 00h (8 bit) for standard RS-232 ICs.

Oscillator start-up time must be considered as well, particularly in applying oscillators with longer start-up delay. The auto-wake-up character must be of sufficient length and of sufficient length of time interval to allow enough time for the selected oscillator to start and offer appropriate initialization of the EUART.

22.4.4.2. Notice of Using WUE Bit

Using WUE and RCIF event timing to determine the validity of received data may bring about some confusion. As noted, setting the WUE as 1 may place the EUART to an standby mode. The wake up event generates a receive interrupt and RCIF is placed 1. The

WUE bit is cleared after a rising edge is seen on RC. The interrupt condition is cleared by reading the RCREG register.

Under normal condition, the data of RCREG after wake up is ineffective and should be discarded. The fact that WUE bit has been cleared (or is still set as 1) and RCIF flag is set should not be used as an indicator of the integrity of the data in RCREG. Users should consider deploying a firmware method to verify received data integrity. In order to assure no effective data is lost, check the RCIDL bit to verify that a receive operation is not in progress. If a receive operation is not executed, the WUE can be placed 1, forcing the IC entering the Sleep mode.

```
BSF     INTE2,6,0       ; Configure RCIE receive interrupt service
CLRF    PT1DA,0         ; Configure PT14 (TX) as digital output pin
                        ; Configure PT13 (RX) as digital input pin
MVL     010H
MVF     TRISC1,1,0
MVL     001H
MVF     PT1M2,1,0
MVL     000H            ; Set Baud rate as 9600Hz
MVF     BRGRH,1,0
MVL     033H
MVF     BRGRL,1,0
MVL     0F0H            ; Start EUART function, initiate data transmission and
                          the 9th bit output
MVF     URCON,1,0       ; Configure TX9D=1. Set parity check code as 0.
MVL     00CH            ; Enable data reception, start the 9th bit reception, the 9th
                          bit is data
MVF     BAUDCON,1,0
MVL     055H            ; Write EUART transmission data
MVF     TXREG,1,0
….
RC Interrupt：          ; Receive interrupt event service program
BCF     INTF2,RCIF,0    ; Clear receive interrupt event flag
BTSZ    URSTA,5,0       ; Determine if PERR bit is 0, to assure data integrity
JMP     FAIL_LOOP       ; Data error determine loop
MVFF    RCREG,BUF0      ; Move the received data to BUF0 register
MVFF    URSTA,BUF1      ; Move the 9th received data to BUF1 register
….
RETI                    ; Return from interrupt
FAIL_LOOP:              ; Data receive error  loop
….
```

Figure 22-11 EUART Example Program

23. Built-in EPROM

Built-in EPROM (BIE), using OTP features to save product serial number, security code and data that generated after program operation...etc., is applied to HY11P3x/HY11P4x/HY11P5x series. Collocating with programming instruction, the external hardware is only needed to connect to 6V VBIE in VPP/RST pin or to use low voltage programming control circuit and this function can be utilized to save address ranges from 00H to 3FH, totally 64 words (equals to 128 bytes).

Build-in EPROM (BIE) can separately support H08A and H08B, two different CPU cores. H08A CPU with Build-in EPROM (BIE) embedded IC includes HY11P33/HY11P35/HY11P36/HY11P41/HY11P42/HY11P54…etc. H08B CPU with Built-in EPROM (BIE) embedded IC includes HY11P32/HY11P52/HY11P52B. Some of the ICs, ex. HY11P52/HY11P52B/HY11P54 support low voltage programming control Build-in EPROM (BIE) function.

When using external VBIE power (6V) to program BIE block, one word can be written to BIE block through instructions; when using HY11P52 to start low voltage programming control circuit, BIE block programming can be implemented without connected to external VBIE power but vice program (WR1WORDBIEDATA) must be called to carry out programming. Only 1 word will be programmed in every calling of programming vice program and it will take about 500msec to complete. When using HY11P52B IC to start low voltage programming control circuit, BIE block can be programmed without connecting to external VBIE power but it is required to call programming vice program (LVWRBIE) to implement programming. Every time calling the programming vice program to carry out programming action, only one word data can be programmed and it will take about 150msec to finish.

Notice: HY11P52/HY11P54 programming spent time is calculated by CPUCK=2MHZ (CPUCLK=00b). Users need to switch frequency source as CPUCLK=00b when calling programming vice program (WR1WORDBIEDATA, WR5WORDBIEDATA). If programming vice program was called under configuration of CPUCLK=01b =2Mhz, then the programming time will take about 1000msec. Below description aims at giving example program for different CPU cores and low voltage programming control Build-in EPROM (BIE).

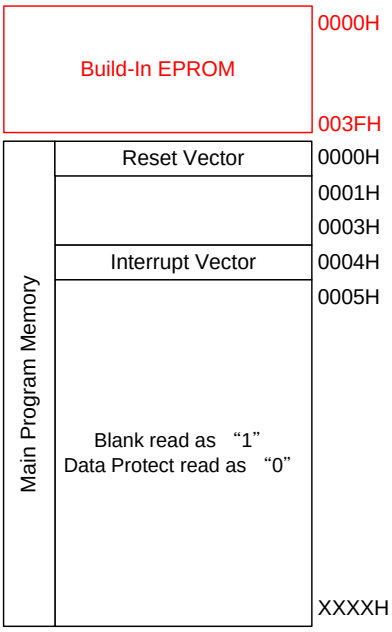


Figure 23-54 Built-In EPROM Framework

BIE Register：

BIECTRL VPP_HIGH[0], **BIE**WR[0], **BIE**RD[0]

BIEPTRL **BIE**_ADDR[5:0]

BIEDH **BIE**_DATA[15:8]

BIEDL **BIE**_DATA[7:0]

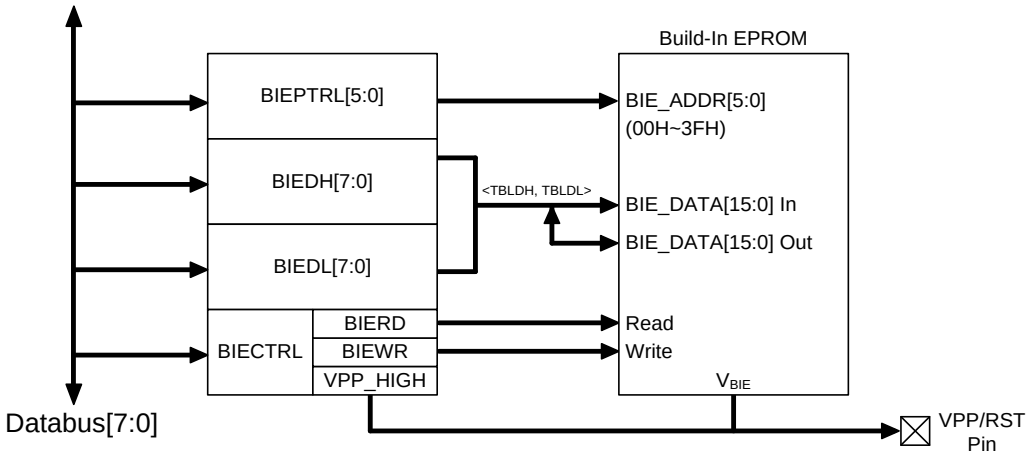


Figure 23-2 BIE Block Diagram

23.1. BIE Manual

23.1.1. Read EPROM

- When the user reads the EPROM：
  - Setup BIEPTRL as BIE_ADDR (utmost to 3FH, BIEPTRL[7]=0、BIEPTRL[6]=0)
  - Read BIEDH,BIEDL as BIE_DATA
  - BSF BIECTRL,BIERD,F

- If BIE_ADDR exceeds 3FH, then this instruction is invalid
- EPROM READ is irrelevant to VBIE potential, but cannot be 0V.

- Users must judge whether BIECTRL[BIERD] is cleared as 0 after EPROM read finished

Notice: ADC function (ADCCN1[ENADC]=0b) must be turned off before reading EPROM. This will help to enhance ADC anti-interference ability. Restart ADC function after EPROM read is completed in order to measure signal.

23.1.2. Write in EPROM

- Before writing EPROM, users are suggested to read VPP_HIGH to confirm VBIE voltage.
- When writing EPROM, users are suggested：
  - To configure BIEPTRL as BIE _ADDR (utmost to 3FH, BIE PTRL[7]=0, BIE PTRL[6]=0)
  - To set up BIE DH, BIE DL as BIE _DATA (unused Data Bit can set 1)
  - BSF BIE CTRL, BIE WR,F
    - If BIE _ADDR exceeds 3FH, this instruction is invalid
    - If VPP is not 6V, this instruction is invalid
  - Wait and judge whether BIECTRL[BIEWR] is cleared as 0 after EPROM write is finished
  - Current consumption will increase distinctively. It is recommend not writing EPROM between ADC test modes.

Notice: Turned off ADC function before write in EPROM (ADCCN1[ENADC]=0b),it helps to improve ADC anti-interference ability. Wait until EPROM write in finished, restarts ADC function to carry out signal measurement.

23.1.3. Read EPROM (HY11P52/HY11P52B /HY11P54 only)

- Users read EPROM：
  - Configure BIEPTRH [7]=1b, starting BIE block read function；
  - Configure **BIE**PTRL as reading **BIE**_ADDR (utmost 3FH, **BIE**PTRL[7]=0, **BIE**PTRL[6]=0)
  - Read **BIE**DH, **BIE**DL as **BIE**_DATA
  - BSF **BIE**CTRL, **BIE**RD,F
    - If **BIE**_ADDR exceeds 3FH, this instruction is invalid
    - EPROM READ is irrelevant to VBIE potential, and cannot be 0V
  - Judge whether BIECTRL[BIERD] is cleared as 0 after EPROM read out completed

Notice: Turned off ADC function (ADCCN1 [ENADC] =0b) before reading EPROM, it will help to enhance ADC anti-interference ability. Restarting ADC function to implement signal measurement until EPROM read is accomplished.

23.1.4. Write EPROM (HY11P52/HY11P52B/HY11P54 only)

- Users write EPROM：
  - Configure BIEPTRH [7]=1b, starting write in function of BIE block；
  - Configure **BIE**PTRL as writing **BIE**_ADDR (utmost 3FH, **BIE**PTRL[7]=0, **BIE**PTRL[6]=0)
  - Configure **BIE**DH, **BIE**DL as write-in **BIE**_DATA(Data Bit not in use as be set as 1)
  - BSF **BIE**CTRL, **BIE**WR,F
    - If **BIE**_ADDR exceeds 3FH, then this instruction is invalid
    - If VPP is not 6V, then this instruction is invalid
  - After EPROM write-in finished, judge whether BIECTRL[BIEWR] is cleared as 0
  - Current consumption will increase a lot, it is recommended not to write in EPROM under ADC test mode

Notice: Turned off ADC function (ADCCN1 [ENADC] =0b) before writing EPROM, this will increase ADC anti-interference ability. Until EPROM write in finished, restarting ADC function to enable signal measurement.

23.1.5. Write in EPROM (HY11P52 only, starting low voltage programming control circuit)

- Users write in EPROM：
  - Configure BIEPTRH [7]=1b, starting write in function of BIE block；
  - Configure **BIE**PTRL as write in **BIE**_ADDR (utmost 3FH, **BIE**PTRL[7]=0, **BIE**PTRL[6]=0)
  - Configure **BIE**DH, **BIE**DL as write in **BIE**_DATA(Data Bit not in use can be set as 1)
  - CALL WR1WORDBIEDATA
    - If **BIE**_ADDR exceeded 3FH, this instruction is invalid
    - Users must make sure VDD≧3.05V for successful calibration
    - Return WREG=0, programming of this word is completed
    - Return WREG=1; representing VDD power is insufficient or the low voltage control circuit is invalid
    - This vice program only judges write in, it does not check the correctness of programming value

- Must attach WR2.obj file

- Current consumption will increase a lot, it is recommended not to write in EPROM under ADC test mode.

Notice: Turned off ADC function (ADCCN1 [ENADC] =0b) before writing EPROM, this will increase ADC anti-interference ability. Until EPROM write in finished, restarting ADC function to enable signal measurement. When low voltage programming control circuit is activated, VBIE voltage cannot be connected.

23.1.6. Write in EPROM (HY11P54 only, starting low voltage programming control circuit)

- Users write in EPROM:
  - Configuring BIEPTRH [7]=1b to start BIE block write in function ;
  - Configuring **BIE**PTRL as write in **BIE**_ADDR (utmost 3FH, **BIE**PTRL[7]=0, **BIE**PTRL[6]=0)
  - Configuring **BIE**DH, **BIE**DL as write in **BIE**_DATA (Data bit not in use can be set as 1)
  - CALL WR5WORDBIEDATA
    - **If BIE**_ADDR exceeded 3FH, then this instruction is invalid
    - Users must make sure that VDD≧3.05V for success calibration.
    - Return WREG=0, finished the programming of word data
    - Return WREG=1; represents insufficient VDD voltage or invalid low voltage control circuit.
    - This vice program only judge write in function, it does not check the correctness of programming value.
    - WR5.obj file must be attached.
  - Current consumption will increase a lot, it is suggested not to execute ERPOM write in action during ADC test modes.

Notice: ADC function must be turned off (ADCCN1[ENADC]=0b) before write in EPROM. It will increase ADC inti-interference ability. Start ADC function to execute signal measurement after EPROM write in finished. When low voltage programming control circuit is activated, please do not connect to external VBIE voltage.

23.1.7. Write in EPROM (HY11P52B only, starting low voltage programming control circuit)

- Users write in EPROM:
  - Configuring BIEPTRH [7]=1b to start BIE block write in function ;
  - Configuring **BIE**PTRL as write in **BIE**_ADDR (utmost 3FH, **BIE**PTRL[7]=0, **BIE**PTRL[6]=0)

- Configuring **BIE**DH, **BIE**DL as write in **BIE**_DATA (Data bit not in use can be set as 1)
- CALL LVWRBIE
  - LCD display will temporarily close while the function was executed. Display will recover when finished.
  - **If BIE**_ADDR exceeded 3FH, then this instruction is invalid
  - Users must make sure that VDD≧2.75V for success calibration.
  - Return WREG=0, finished the programming of word data
  - Return WREG=1; represents insufficient VDD voltage or invalid low voltage control circuit.
  - This vice program only judge write in function, it does not check the correctness of programming value.
  - WR3.obj file must be attached.
- Current consumption will increase a lot, it is suggested not to execute ERPOM write in action during ADC test modes.

Notice: ADC function must be turned off (ADCCN1[ENADC]=0b) before write in EPROM. It will increase ADC inti-interference ability. Start ADC function to execute signal measurement after EPROM write in finished. When low voltage programming control circuit is activated, please do not connect to external VBIE voltage.

23.1.8. Notice

- Please configure CPU frequency source as HAO first before READ/WRITE BIE, otherwise it may lead to abnormal action.
- After EPROM READ/WRITE finished, BIERD/BIEWR will be cleared as 0. BIEPTRL auto ascends (utmost to 3FH).
- It is commended to use BSF instruction to set BIERD or BIEWR. If instruction configures BIERD and BIEWR as 1 in the same time, this instruction is invalid.
- When VPP is in high potential after CPU reset, PT1.5 remains outputting 65ms unknown potential (high/low) status.
- Power on sequence: Power on VDD first, then power on VBIE.
- ADC function (ADCCN1[ENADC]=0b) must be turned off before reading/writing EPROM. This will help to enhance ADC anti-interference ability. Restarting ADC signal measurement function after reading/writing EPROM is completed.
- When using HY11P52 to start low voltage programming control circuit, the related restrictions of its programming vice program, WR1WORDBIEDATA are described as follows：
  - Influenced registers:
    Special Register: WREG, LVDCN, FSR0L, INDF0, INTE1[GIE].

Data Register: 0F0h~0F4h

- Reserved stack layers:

  At least 2 stack layers must be reserved to call this vice program

- Option Function:

  Vice program, WR1WORDBIEDATA, will start LVD, low voltage detection circuit of the IC automatically

  LVD function will be closed after leaving this vice program.

PS: Users must pay attention when using this programming vice program that VDD must larger or equal to 3.05V for successful programming!

- Code Size: 61 lines of instruction space
- Function return value:

  Return WREG=1, representing insufficient VDD voltage or low voltage control circuit is invalid.

  Return WREG=0, representing the completion of programming word data to BIE block.

  The correctness of programmed value was not checked, users need to reassure about this.

- Other notice:
  - Low voltage programming control circuit function of HY11P52 only suited to be operated in temperature ranges from 0℃~40℃ and under 3.05V ≦VDD≦3.4V condition. If the IC uses external VBIE power (6V) to program BIE block, it can program data through signal instruction (BSF BIECTRL, BIEWR, 0) and the programming vice program (WR1WORDBIEDATA) cannot be used under this mode as to avoid inaccurate programming data.
  - After low voltage programming control circuit is activated, the voltage source must be configured as 3.05V≤VDD≤3.4V to enable normal programming.
  - Starting low voltage programming control circuit, GIE interrupt control source will be turned off compulsively.
  - It is recommended to start BIE block programming by low voltage programming control circuit after all calibration value was completely measured.
  - Users must be cautioned about the influenced registers after calling programming vice program.
  - Development kit (ICE) only supports HY11S14-DK02/DK03, it does not support HY11S14-DK01.

- Development kit (ICE) cannot emulate low voltage program and 16bits look-up-table function.
- When users utilize development kit (ICE) to emulate HY11P52 low voltage programming function, an external VBIE power (6V) must be connected to operate the program normally (if the emulation is not done on ICE, the VBIE (6V) can be disconnected.

- When using HY11P54 to start low voltage programming control circuit, the related restrictions of its programming vice program, WR5WORDBIEDATA are described as follows：
  - Influenced registers:
    Special Register: WREG, LVDCN, FSR0L, INDF0, INTE1[GIE].
    Data Register: 0F0h~0F5h
  - Reserved stack layers:
    At least 2 stack layers must be reserved to call this vice program
  - Option Function:
    Vice program, WR5WORDBIEDATA, will start LVD, low voltage detection circuit of the IC automatically
    LVD function will be closed after leaving this vice program.

PS: Users must pay attention when using this programming vice program that VDD must larger or equal to 3.05V for successful programming!

- Code Size: 81 lines of instruction space
- Function return value:
  Return WREG=1, representing insufficient VDD voltage or low voltage control circuit is invalid.
  Return WREG=0, representing the completion of programming word data to BIE block.
  The correctness of programmed value was not checked, users need to reassure about this.
- Other notice:
  - Low voltage programming control circuit function of HY11P54 only suited to be operated in temperature ranges from 0℃~40℃ and under 3.05V ≦VDD≦3.4V condition. If the IC uses external VBIE power (6V) to program BIE block, it can program data through signal instruction (BSF BIECTRL, BIEWR, 0) and the programming vice program (WR5WORDBIEDATA) cannot be used under this mode as to avoid inaccurate programming data.

- After low voltage programming control circuit is activated, the voltage source must be configured as 3.05V≤VDD≤3.4V to enable normal programming.
- Starting low voltage programming control circuit, GIE interrupt control source will be turned off compulsively.
- It is recommended to start BIE block programming by low voltage programming control circuit after all calibration value was completely measured.
- Users must be cautioned about the influenced registers after calling programming vice program.
- Development kit (ICE) only supports HY11S14-DK 05,
- When users utilize development kit (ICE) to emulate HY11P54 low voltage programming function, an external VBIE power (6V) must be connected to operate the program normally.

- When using HY11P52B to start low voltage programming control circuit, its programming vice program, LVWRBIE must be controlled as follows:
  - Influenced registers:

    Special Register: WREG, LVDCN, FSR0L, INDF0, INTE1[GIE]. ADCCN1[ENADC], LCDCN2[LCDBL].

    Data Register: 0F0h~0F5h
  - Save stack layer:

    At least 3 stack layers must be reserved to call this vice program.
  - Option Function:

    GIE interrupt will be closed when entering to vice program. Users must restart GIE function after finished. LVD low voltage detect circuit will automatically be activated in vice program and will be turned off when exit vice program, users must setup in accordance with their needs. Vice program will temporarily turn off LCD display function (LCDCN2[LCDBL]=1b) and it will be turned on after the vice program was completely executed (LCDCN2[LCDBL]=0b).

Note: Users must make sure that VDD≧2.75V when using this programming vice program to ensure successful programming!

- Code Size: 83-line instruction space
- Function return value:

  Return WREG=1, represents insufficient VDD voltage or invalid low voltage control circuit.

Return WREG=0, represents the completion of word data programming in BIE block. The correctness of programming value was not checked, users must examine it by themselves.

- Other notice:
  - HY11P52B uses low voltage programming control circuit function that is suitable to temperature range from 0℃~40℃ and 2.75V≦VDD≦3.6V condition. If the IC connects to external VBIE power (6V) to program BIE block, only single instruction can be given to program data (BSF BIECTRL,BIEWR,0) and under this mode, programming vice program (LVWRBIE) cannot be used to execute programming action to avoid error.
  - Voltage source must be configured 2.75V≤VDD≤3.6V as to perform programming when low voltage programming control circuit is activated.
  - GIE interrupt control source and LVD function will be turned off compulsively when low voltage programming control circuit is activated.
  - It is recommended to start low voltage programming control circuit to implement BIE block programming when all calibration value was measured.
  - Turn off ADC function before connecting to external VBIE power (6V) to grogram BIE block to enhance anti-interference ability.
  - Users must caution about the influenced registers after calling vice programming program.
  - Only support HY11S14-DK02 emulator, HY11S14-DK01 is not supported;
  - Emulator cannot simulate low voltage programming and 16bits look-up-table function;
  - When users use emulator to simulate HY11P52B low voltage programming function, VBIE power (6V) must be connected to the emulator to perform successfully.

```
(1) Read EPROM
BCF       ADCCN1,ENADC,ACCE      ;Turn off ADC function before read EPROM
MVL       00000000B
MVF       BIEPTRL, F, ACCE       ; Define EPROM address as  00H
BSF       BIECTRL, BIERD, ACCE   ; Give instruction to read EPROM and store the data in register, BIEDH,
                                 ; BIEDL, After EPROM read/write finished, BIECTRL[BIERD] will be 0
                                 ; BIEPTRL(BIE_ADDR) auto-ascends, utmost to  3FH
WAITRDBIE:
 BTSZ     BIECTRL, BIERD, ACCE  ; wait until determination of EPROM read completed,
                                 ;BIECTRL[BIERD] is clear as 0
JMP       WAITRDBIE
MVF       BIEDL, W, 0
MVF       BUF0, F, 0             ; Move BIEDL data to BUF0
MVF       BIEDH, W, 0
MVF       BUF1, F, 0             ; Move BIEDH data to BUF1
;BSF  ADCCN1,ENADC,ACCE          ;Start ADC function depends on users' need
(2)Write EPROM
BCF       ADCCN1,ENADC,ACCE      ;Turn off ADC function before write in EPROM
VPPCHK:
BTSS      BIECTRL,VPP_HIGH,0     ;Check if external VPP=6V voltage exists. If yes, continue  programming
JMP       VPPCHK
MVL       00000000B
MVF       BIEPTRL, F, ACCE       ; Define EPROM address as 00H
MVL       12H                    ; Define write-in data [BIEDH, BIEDL]=[12H,34H]
MVF       BIEDH, F, ACCE
MVL       34H
MVF       BIEDL, F, ACCE
BSF       BIECTRL, BIEWR, ACCE   ;(1)If BIECTRL[VPP_HIGH]=0 or  BIEPTRL＞3FH
                                 ;this instruction is invalid
                                 ;(2) If  BIECTRL[VPP_HIGH]=1 and  BIEPTRL≦3FH
                                 ;this instruction is valid
                                 ; After EPROM read/write finished, BIECTRL[BIEWR] is 0
                                 ; BIEPTRL auto-ascends, upmost to  3FH
WAITWRBIE:
 BTSZ     BIECTRL, BIEWR, ACCE  ;Wait until determination of EPROM write in completed,
                                 ;BIECTRL[BIEWR] is automatically clear as 0
JMP       WAITWRBIE
;BSF       ADCCN1,ENADC,ACCE     ; Start ADC function depends on users' need
```

Figure 23-3 H08B BIE Example Program (Applicable to HY11P32)

```
(2) Read EPROM
BCF     ADCCN1,ENADC,ACCE       ;Turn off ADC function before read EPROM
LBSR    01H                     ; Declared BANK location
MVL     00000000B
MVF     BIEPTRL, F, BANK        ; Define  EPROM location as 00H
BSF     BIECTRL, BIERD,BANK     ; Give instruction to read  EPROM and store the data in  BIEDH, BIEDL register
                                ; After EPROM reading finished, BIECTRL[BIERD] will be 0
                                ; BIEPTRL (BIE_ADDR) auto-ascends,  utmost to 3FH
WAITRDBIE:
BTSZ    BIECTRL, BIERD, ACCE    ; Wait until the determination of EPROM reading finished, BIECTRL [BIERD] is auto 0
JMP     WAITRDBIE
MVF     BIEDL, W, 0
MVF     BUF0, F, 0              ; Move  BIEDL data to BUF0
MVF     BIEDH, W, 0
MVF     BUF1, F, 0              ; Move BIEDH data to BUF1
;BSF    ADCCN1,ENADC,ACCE       ;Starts ADC function depends on users' need
(2) Write in EPROM
BCF     ADCCN1,ENADC,ACCE       ;Turn off ADC function before write in EPROM
VPPCHK:
BTSS    BIECTRL,VPP_HIGH,0      ;Check if external VPP =6V voltage exists. If yes, continue programming
JMP     VPPCHK
MVL     00000000B
MVF     BIEPTRL, F, BANK        ; Define  EPROM location as  00H
MVL     12H                     ; Define write in data  [BIEDH, BIEDL]=[12H,34H]
MVF     BIEDH, F, BANK
MVL     34H
MVF     BIEDL, F, BANK
BSF     BIECTRL, BIEWR, BANK    ;(1) If BIECTRL[VPP_HIGH]=0 or BIEPTRL＞3FH ; Then this instruction is invalid
                                ;(2) If  BIECTRL[VPP_HIGH]=1 and BIEPTRL≦3FH; Then this instruction is valid
                                ; After EPROM writing finished, BIECTRL[BIEWR] will be auto-0
                                ; BIEPTRL auto-ascends, utmost to  3FH
LBSR    00H                     ; Return to BANK location
WAITWRBIE:
BTSZ    BIECTRL, BIEWR, ACCE    ; Wait until the determination of  EPROM reading was completed, BIECTRL[BIEWR] is auto 0
JMP     WAITWRBIE
;BSF    ADCCN1,ENADC,ACCE; Start ADC function depends on users' need
```

Figure 23-4 Example Program of H08A BIE(Applicable to HY11P33/HY11P35/HY11P36/HY11P41/HY11P42)

```
(3)Read EPROM (for HY11P52 /HY11P52B only)
BCF     ADCCN1,ENADC,ACCE   ;ADC function must be turned off before read  EPROM
MVL     10000000B
MVF     BIEPTRH, 1, 0       ;BIEPTRH [7]=1b starts  BIE block  read function
MVL     00000000B
MVF     BIEPTRL, F, ACCE    ;Define  EPROM address as  00H
BSF     BIECTRL, BIERD, ACCE ;Give instruction to read EPROM and store the data to register,
                             ;BIEDH, BIEDL
                             ;After reading EPROM completed, BIECTRL[BIERD] is auto 0
                             ;BIEPTRL(BIE_ADDR) auto ascend, upmost to  3FH
WAITRDBIE:
BTSZ    BIECTRL, BIERD, ACCE ;Wait until the determination of  EPROM reading was
                             ; completed, BIECTRL[BIERD] is auto 0
JMP     WAITRDBIE
MVF     BIEDL, W, 0
MVF     BUF0, F, 0          ; Move  BIEDL data to BUF0
MVF     BIEDH, W, 0
MVF     BUF1, F, 0          ; Move  BIEDH data to BUF1
;BSF    ADCCN1,ENADC,ACCE   ;Start ADC function depends on users' need
```

Figure 23-5 Example Program of reading BIE Control (Applicable to HY11P52 /HY11P52B)

```
(4)Read EPROM (for Hy11P54)
BCF     ADCCN1,ENADC,ACCE   ; ADC function must be turned off before write in EPROM
LBSR    1
MVL     10000000B
MVF     BIEPTRHA, 1, 1      ;BIEPTRH [7]=1b starts BIE block programming function
MVL     00000000B
MVF     BIEPTRLA, F, 1      ; Define EPROM address as 00H
BSF     BIECTRLA, BIERD, 1  ; give order to read EPROM, and save the data in register
                            ;BIEDHA,BIEDLA.
                            ;After finish reading EPROM，BIECTRLA[BIERD] automatically
                            ;erased to 0
                            ;BIEPTRLA(BIE_ADDR) automatically increase，up most is 3FH
WAITRDBIE:
BTSZ    BIECTRLA, BIERD, 1  ;After wait for reading EPROM，BIECTRLA[BIERD]
                            ;automatically be erased to 0
JMP     WAITRDBIE
MVF     BIEDLA, W, 1
MVF     BUF0, F, 0          ; to transfer BIEDL data to BUF0
MVF     BIEDHA, W, 1
MVF     BUF1, F, 0          ; to transfer BIEDH data to BUF1
LBSR    0
;BSF    ADCCN1,ENADC,ACCE   ;Start ADC function depends on users' need
```

Figure 23-6 Example Program of reading BIE Control (Applicable to HY11P54)

(5) Write in EPROM (for HY1P52B only)

```
BCF     ADCCN1,ENADC,ACCE     ;ADC function must be turned off before write in  EPROM
MVL     10000000B
MVF     BIEPTRH, 1, 0         ;BIEPTRH [7]=1b starts  BIE block programming function
MVL     00000000B
MVF     BIEPTRL, F, ACCE      ;Define  EPROM address as  00H
MVL     12H                   ;Define write in data  [BIEDH, BIEDL]=[12H,34H]
MVF     BIEDH, F, ACCE
MVL     34H
MVF     BIEDL, F, ACCE
MVF     FSR0L,0,0
MVF     FSR_BUF,1,0           ;Protect FSR0 register value
        ;…                    ;If there's a need to protect operation register, please protect WREG register
BCF     MCKCN2,1,0            ;Restore CPUCLK=00b=2Mhz
BCF     MCKCN2,0,0
CALL    LVWRBIE               ; LCD display will be off when this function was executed, display will recover when finished
                              ;Users must assureVDD≧2.75V to achieve success calibration
                              ;BIEPTRL＞3FH or this instruction is invalid (BIEPTRH=0x80)
                              ;BIEPTRL≦3FH, this instruction is valid (BIEPTRH=0x80)
                              ;After EPROM finished write in, BIEPTRL auto ascends 1 (upmost to 3FH)
                              ;Return WREG=0, data programming of this byte is finished
                              ;Return WREG=1, represents insufficient VDD voltage Or low   voltage control circuit is invalid
                              ; This subroutine is only for write, it does not check the correctness of the programming value
MVF     FSR_BUF,0,0           ;Restore  FSR0 register value
MVF     FSR0L, 1,0
;BSF    ADCCN1,ENADC,ACCE     ;Starts ADC function depends on users' needs
 BSF    INTE1,GIE             ;Programming vice program already turned off interrupt source, GIE
                              ;After programming complete, users need to turn on GIE
TFSZ    WREG,0                ;WREG determines whether the programming was completed,
JMP     FAIL                  ;WREG=0b, WRITE BIE FAIL
;…                            ;Other executive program
FAIL:
IDLE
NOP
INCLUDE WR3.obj               ;WR3.obj file must be placed in the very last of the program
END
```

Figure 23-7 Example Program of H08B BIE Low Voltage Programming Control (Applicable to HY11P52B)

(6)Write in EPROM (for HY11P52 only)

```
BCF     ADCCN1,ENADC,ACCE   ; ADC function must be turned off before write in  EPROM
MVL     10000000B
MVF     BIEPTRH, 1, 0       ;BIEPTRH [7]=1b starts  BIE block programming function
MVL     00000000B
MVF     BIEPTRL, F, ACCE    ;Define  EPROM address as  00H
MVL     12H                 ;Define write in data [BIEDH, BIEDL]=[12H,34H]
MVF     BIEDH, F, ACCE
MVL     34H
MVF     BIEDL, F, ACCE
MVF     FSR0L,0,0
MVF     FSR_BUF,1,0         ;Protect  FSR0 register value
;…                          ;Protect WREG register if needed
BCF     MCKCN2,1,0          ;Return to CPUCLK=00b=2Mhz
BCF     MCKCN2,0,0
CALL    WR1WORDBIEDATA      ;Start  LVD detect function,
                            ;Users make sure that  VDD≧3.05V for successful calibration
                            ;BIEPTRL＞3FH, invalid instruction (BIEPTRH=0x80)
                            ;BIEPTRL≦3FH, valid instruction (BIEPTRH=0x80)
                            ;EPROM write in finished, BIEPTRL auto adds 1, utmost  3FH
                            ;Return WREG=0, this word programming is finished
                            ;Return WREG=1, insufficient  VDD or invalid low voltage control circuit
                            ;Vice program only judges write in not the correctness of programming value
MVF     FSR_BUF,0,0         ; Restore to  FSR0 register value
MVF     FSR0L, 1,0
;BSF    ADCCN1,ENADC,ACCE   ;Start ADC function depends on users' need
BSF     INTE1,GIE           ;GIE interrupt is closed by vice programming program
                            ;Users must turn on GIE after completion of programming
TFSZ    WREG,0              ;WREG judge program, whether the programming was completed
JMP     FAIL                ;WREG=0b, WRITE BIE FAIL
;…                          ;Other executed program
FAIL:
IDLE
NOP
INCLUDE WR2.obj             ;WR2.obj file must be placed at the end of program
```

Figure 23-8 Example Program of H08B BIE Low Voltage Programming Control (Applicable to HY11P52)

23.2. Register Description-BIE

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"omly write 1												
"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition												
H08B Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**i-RESET**	**R/W**
1CH	BIECTRL					VPP_HIGH		BIEWR	BIERD	1000 d000	1000 d000	-,-,-,- *,r0,*,*
1DH	BIEPTRH	SBMSEL				BIE_ADDR[10:8]				..00 0000	..00 0000	-,-,*,*,*,*,*
1EH	BIEPTRL	0	0	BIE_ADDR[5:0]						0000 0000	0000 0000	*,*,*,*,*,*,*
1FH	BIEDH	BIE_DATA[15:8]								0000 0000	0000 0000	*,*,*,*,*,*,*
20H	BIEDL	BIE_DATA[7:0]								0000 0000	0000 0000	*,*,*,*,*,*,*
H08A Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**i-RESET**	**R/W**
195H	BIECTRLA					VPP_HIGH		BIEWR	BIERD	1000 d000	1000 d000	-,-,-,- *,r0,*,*
196H	BIEPTRHA	SBMSEL				BIE_ADDR[10:8]				0000 0000	0000 0000	w0,w0,W0,W0 W0,W0,W0,W0
197H	BIEPTRLA	0	0	BIE_ADDR[5:0]						0000 0000	0000 0000	w 0,w 0,*,*,*,*,*
198H	BIEDHA	BIE_DATA[15:8]								xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*
199H	BIEDLA	BIE_DATA[7:0]								xxxx xxxx	xxxx xxxx	*,*,*,*,*,*,*

BIECTRL/ BIECTRLA：BIE Register controller

VPP_HIGH：Check VPP

0：VPP not connect with external writer power supply 6V

1：VPP connect with external writer power supply 6V

BIEWR：Write in EPROM control bits.

0：Not allow to write

1：Allow to write

BIERD：Read EPROM control bits.

0：Cannot read

1：Can read

BIEPTRH/ BIEPTRHA：EPROM Address Definition

BIESEL：Mode Selection (this bit is only applicable to HY11P52/HY11P52B/HY11P54; the rest products must write 0 compulsively)

0：OTP READ MODE

1：BIE MODE

BIE_ADD[10:8]：OTP address, when BIESEL =0b, HY11P52 selects OTP READ MODE

BIEPTRL/ BIEPTRHA：EPROM Address Definition

BIE_ADDR [7]：0 must be compulsively written into (For HY11P52/HY11P52B/HY11P54, selects BIE MODE)

BIE_ADDR [6]：0 must be compulsively written into (For HY11P52//HY11P52B, selects BIE MODE)

BIE_ADDR[5:0]：EPROM address, only 00H~3FH, total 64 words

BIE_ADDR[7:0]：OTP address, when BIESEL =0b, HY11P52/HY11P52B selects OTP READ MODE

BIEDH/ BIEDHA：EPROM High Byte Data Definition

BIEDL/ BIEDHA：EPROM Low Byte data definition

24. Revisions

The following describes the major changes made to the document, excluding the punctuation and font changes.

Version	Page	Revision Summary
V10	16	Revise BSR content and example
	19	Revise BSRCN content
	26	Add in oscillator example program
	29	Revise content
	50	Revise Figure 7-1 I/O frame block diagram
	53	Revise I/O port 1 content
	55	Revise Table 7-3 content description
	57~58	Revise Table 7-6 content and add in Table 7-7 content
	95	Add in duty cycle configuration content
	96	Revise Equation 14-4
	104	Add in ACM description
	114	Add in voltage generator II description
	121	Revise Table 18-1 SD18 operating freq. configuration
	123	Add in 18.1.3 SD18 content
	127	Delete 18.3 SD18 noise characteristics
	128~129	Table 18-7 (b) PGA Gain and R_A and C_A Relation revision
	130	18.3.1 TPS Initial Configuration and Calculation revision
	131	Add in TPS Example Description
	139	Update Figure 19-2 & Table 19-2
	149	Figure 23-55 SPI Block Diagram revision
	151	Add in Figure 20-2 content
	152~153	Revise Figure 20-3 to 20-5
	154~155	Revise Example 20-1 and add in Example 20-2
	163	Delete content
	170	Figure 23-3 BIE Example Program revision
	172	Add Chapter 23 revision record
V12	10	Revise the remark of Maximum Address Ability
	15	Revise Figure 23-56 Data Memory Frame
	16~17	Revise Example 23-20 Relation of Segment Selector Example Program And Data Memory
	20~21	Revise Table 23-49 Data Memory List
	23	Add Oscillator oscillation time description
	27	Revise Table 23-50 CPU Operating Frequency and Instruction Execution Cycle

	30	Change notice description
	32~35	Add IDLE Mode and Sleep Mode description
	36	Revise symbol description
	109	Revise register name
	117	Revise OPM[1:0] ADC_CLK name description
	137	Revise Table 19-1 (b) LCD0 and LCD1 Register Multiplexed Digit Pin Comparison
	150~156	Add SPI Master/Slave Mode description & Revise Example 23-1 & 20-2
	172~173	Revise example program of H08B and H08A BIE Example Program
V13	71	Add description to example program of WDT interrupt event
	156	Revise Figure 20-6
V14	59	Revise Table 7-7, CCP1 Description
	170~180	Update Chapter 22, Built-in EPROM Description
V15	170~183	Add inHY11P52B built-in EPROM related description
V16	All	Add in HY11P54 LNO2 and built-in EPROM related description