HYCON 紘康科技

HY10P Family

User’s Guide

Mixed Signal Microcontroller

Table of Contents

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. Central Processing Unit (CPU)

2.1. CPU Core

To get higher efficiency, CPU Core (H08) is adopted Harvard architecture concept. It separates the program memory and data memory respectively. Furthermore, the address of program memory increases the convenience of program writing for user.

CPU Features include:

Separate design structures of program memory and data memory increase the instruction execution speed and improve the CPU efficiency.

Max addressing capabilities are 1MW for program memory and 4096KB for data memory respectively.

Max 46 operation instructions include block switching and stacking control of data memory.

One instruction can complete data moving of FSR register with max 16-bit and address table look-up instruction of 1MW program memory.

The operation of data memory includes the data moving of program counter (PC), status register (Status) and stack register (Stack).

The CPU core is H08B core of starter edition.

2.2. Memory

The structure of memory has two types. One is program memory, which is composed of OTP, and the other is data memory, which is composed of SRAM. On different model product, planned memory size is also different. Therefore, it must pay special attention on specification of the product when reading the operation manual.

Program Memory:

Main Program Memory (MPM)

Program Counter (PC)

Stack (STK)

Data Memory:

Special Register (SR)

General Purpose Register (GPR)

Related Register Abstract of Memory: (x: means it is composed of multiple registers.)

PC[10:0]	PCHSR[2:0],PCLATH[2:0],PCLATL[7:0]
TOS[10:0]	TOSH[2:0],TOSL[7:0]
FSR0[7:0]	FSR0L[7:0]
INDF0	INDF0[7:0]
POINC0	POINC0[7:0]
PODEC0	PODEC0[7:0]
PRINC0	PRINC0[7:0]
PLUSW0	PLUSW0[7:0]
STKCN	STKFL[0],STKOV[0],STKUN[0],STKPRT[2:0]
PSTATUS	SKERR[0]

2.2.1. Program Memory

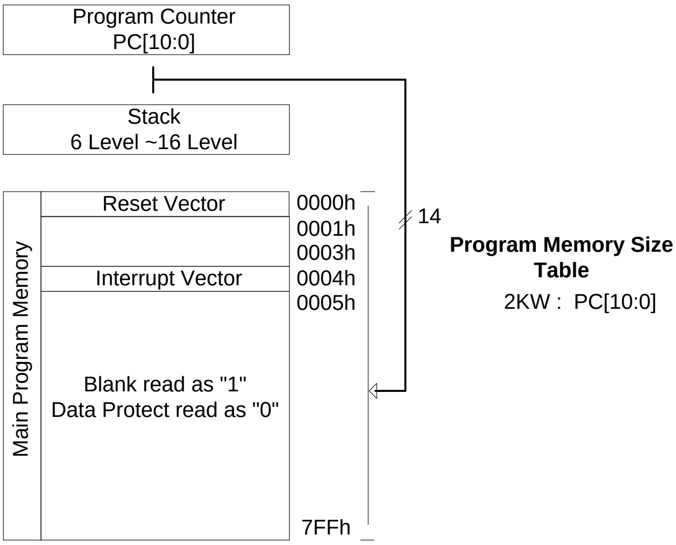


Fig 2-1 Program Memory Architecture

2.2.1.1. Main Program Memory (MPM)

Main Program Memory Architecture is as the following:

- Interrupt Vector Position
- Reset Vector Position

Addressing capability is from 0x00000h to 0x7FFh, with a total capacity of 2048 Word.

When the chip is not taken program writing, data type of all addresses is 1. After writing, the addresses will be 1 or 0 according to the written data type. Note: in program development, if the assemble option of simulation software (HYIDE) has burn protection function setting, the data type of chip is 0 at the addresses that burning can be read.

2.2.1.2. Program Counter (PC)

Program Counter (PC) is composed of shift register PCSR and buffer register PCLAT. See Fig 2-2.

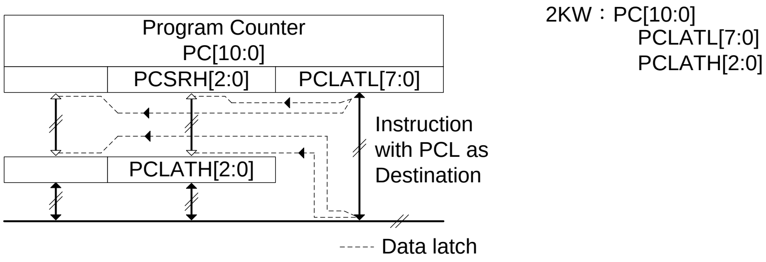


Fig 2-2 PC Architecture

Chip used in tool development of PC [10:0] has 11-bit data length. It is composed of two special registers, PCSRH [2:0] and PCLATL [7:0]. PCLATL [7:0] and PCLATH [2:0] can be read / written directly, while PCSRH [2:0] cannot be read / written directly and it must use buffer register PCLATH [2:0] as indirect reading /writing.

- Before reading PC [10:0], it must read PCLATL [7:0] first and then read PCLATH [2:0] before reading correct data. Reverse order will not read correct data.
- Before writing PC [10:0], it must write PCLATH [2:0] and then PCLATL [7:0] finally. Reverse order will not write correct data.

2.2.1.3. Stack (STK)

Stack (STK) is mainly composed of stack pointer control register STKCN, top stack register TOS0, stack layer register STKn[1], stack error flag SKERR(Stack Error) and stack error reset controller SKRST[0], as in Fig 2-3.

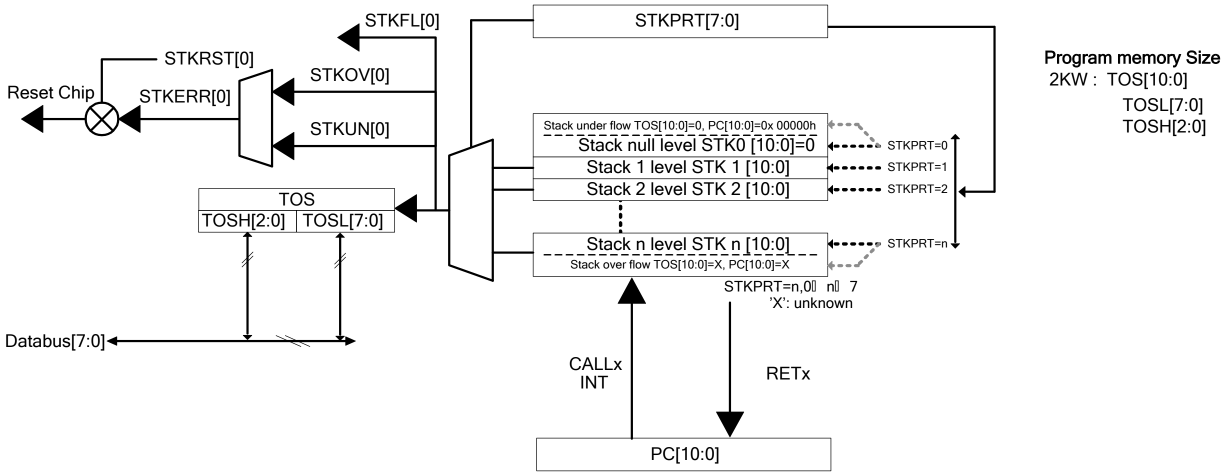


Fig 2-3 Stack Architecture

Top stack register TOS [10:0] has 11-bit data length. It is composed of two registers, TOSH [2:0] and TOSL [7:0]. When STKPRT [2:0] =<0>, TOS [10:0] =<0> is null. When the program executes CALL instruction or appears interrupt (INT) service, stack pointer STKPRT [2:0] makes plus one motion and writes PC address when the event appears into TOS [10:0] register of the time. When the program executes RETx instruction, stack pointer STKPRT [2:0] makes minus one motion. Before taking minus one motion, it will write TOS [10:0] data in PC [10:0] in advance. After writing is completed, STKPRT [2:0] makes minus one motion and changes the current TOS [10:0] value.

- It has no special rules for TOS [10:0] register reading and can be read directly.
- TOS[10:0] register writhing can use CALL instruction or interrupt (INT) to write PC[10:0] data, or uses POP instruction to discard current TOS[10:0] data and make STKPRT[2:0]minus one and load new TOS[10:0] data.

During stack operation process, it may appear Stack full STKFL [0], Stack overflow STKOV [0] or Stack underflow STKUN[0] and other events. Stack full is an indication flag before Stack overflow. At the moment, it can discard current TOS [10:0] and make STKPRT [2:0] minus one and write stack layer data with new pointer in TOS [10:0] via POP instruction execution. Note: when STKPRT [2:0]=<0>, POP instruction execution will not appear underflow condition. At the moment, STKPRT [2:0] is still <0>. Therefore, the user must judge whether it is empty stack by oneself.

[1] Stack layer register STKn: Each layer of stack has data register with the same length of top stack register TOS. When the stack pointer STKPRT is appointed, the content of data register is sent to TOS.

When the stack appears overflow and underflow, it may lead to unexpected execution results for program. When necessary, it can restart chip via setting. In program development process, it can set stack reset control bit SKRST [0][2] as <1> via software setting. When stack appears underflow or overflow, it may generate reset signal and shall restart chip after the SKERR [0] is set as <1>.

- Stack full: When STKFL [0] is set as <1>, PC[10:0] is not affected.
- Stack underflow : When STKUN[0] is set as <1>，PC[10:0] is moved to 0x00000hposition and stack pointer STKPRT points to 0 Level. If SKRST [0] is set as <1>, it may generate reset signal after stack underflow, and SKERR [0] is set as <1>. STKUN [0] is set as <0> after reset.
- Stack overflow: When STKOV[0] is set as <1>, PC[10:0] is not affected and STKPRT is still stopped at the final layer and it will press new value, i.e. it may reserve the last one pressing data after stack full. If SKRST[0] is set as <1>, it may generate reset signal after stack overflow, and SKERR [0] is set as <1>. STKOV[0] is set as <0> after reset.
- Error: When SKERR[0] is set as <1>, chip appears stack error. If SKRST[0] is set as <1>, STKUN[0] and STKOV[0] are set as <0> after reset.
- When it appears stack full, if it appears overflow condition as ignorance and continues executing POP instruction as ignorance and causes underflow condition, STKFL[0], STKOV[0] and STKOV[0] are set as <1> simultaneously. Therefore, it is suggested to make cleaning action for flags to avoid erroneous judgment of program when any one of above conditions appears.

If programming method has omitted the known overflow condition, it is suggested to use POP instruction to clear overflow flag and continue executing program after overflow appears. Otherwise, Interrupt or Call instruction generated stack writing motion after overflow will overlay current TOS[10:0] data.

[2] SKRST[0] is stack error generated reset signal control bit. It cannot be read /written directly and can only be set via software development at the program development stage, i.e. it must select whether it generates reset signal when stack error appears at the program development stage. If reset is selected, the bit is set as 1 after powering on the chip. Otherwise, it is set as <0>.

2.2.1.4. Register Instruction---Program Memory Controller

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
016h	TOSH	-	-	-	-	-	TOS[10]	TOS[9]	TOS[8]	…. xxxx	…. uuuu
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu
018h	SKCN	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. .000	u$$. .$$$
01Ah	PCLATH	-	-	-	-	-	PC[10]	PC[9]	PC[8]	…. .0000	…. 0000
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
02Ch	PSTATUS	POR	PD	TO	IDL	RST	SKERR	-	-	$000 $00.	uu$u u$u.

Table 2-1 Program Memory Control Register

TOSU/TOSH/TOSL: Stacked Top Stack Register

TOSH: TOS[10:8]

TOSL: TOS[7:0]

STKPTR: Stack Controller

STKFL: Stack Full Flag

1: Occurred

0: Not Occurred

STKUN: Stack Underflow Flag

1: Occurred

0: Not Occurred

STKOV: Stack Overflow Flag

1: Occurred

0: Not Occurred

STKPRT[2:0]: Stack Pointer Register

111: 7th layer

110: 6th layer

00000: 0 layer TOS[10:0]=0x0000h

PCLATU/PCLATH/PCLATL: Program Counter PC[10:0]

PCLATH: PC[10:8]

PCLATL: PC[7:0]

PSTATUS: Status Register

SKERR: Stack Error Generated Reset Flag

1: Occurred

0: Not Occurred

2.2.2. Data Memory (DM)

Data Memory (DM) is composed of Specially Register (SR) and General Purpose Register (GPR). Furthermore, it takes every 256byte as a block. 128byte Specially Register and 128 byte General Purpose Register is as in Fig 2-4.

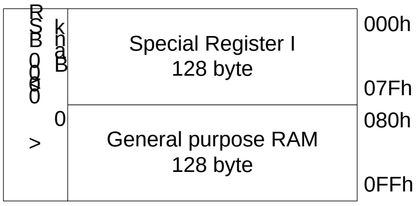


Fig 2-4 Data Memory (DM) Architecture

2.2.2.1. Memory and Instruction

H08 instruction set can be divided A and B two version, which have great difference in memory application, such as addressing capability, hardware multiplier, table look-up instruction, support functions and parameters definition. Here just illustrate definition of instruction memory parameters. See Instruction part of Instruction Set on detailed instruction parameter illustration.

Instructions with address computation function in instruction set have three parameters at most, i.e. “f”, “d” and “a”.

“f” refers to Data or Data Memory Address.

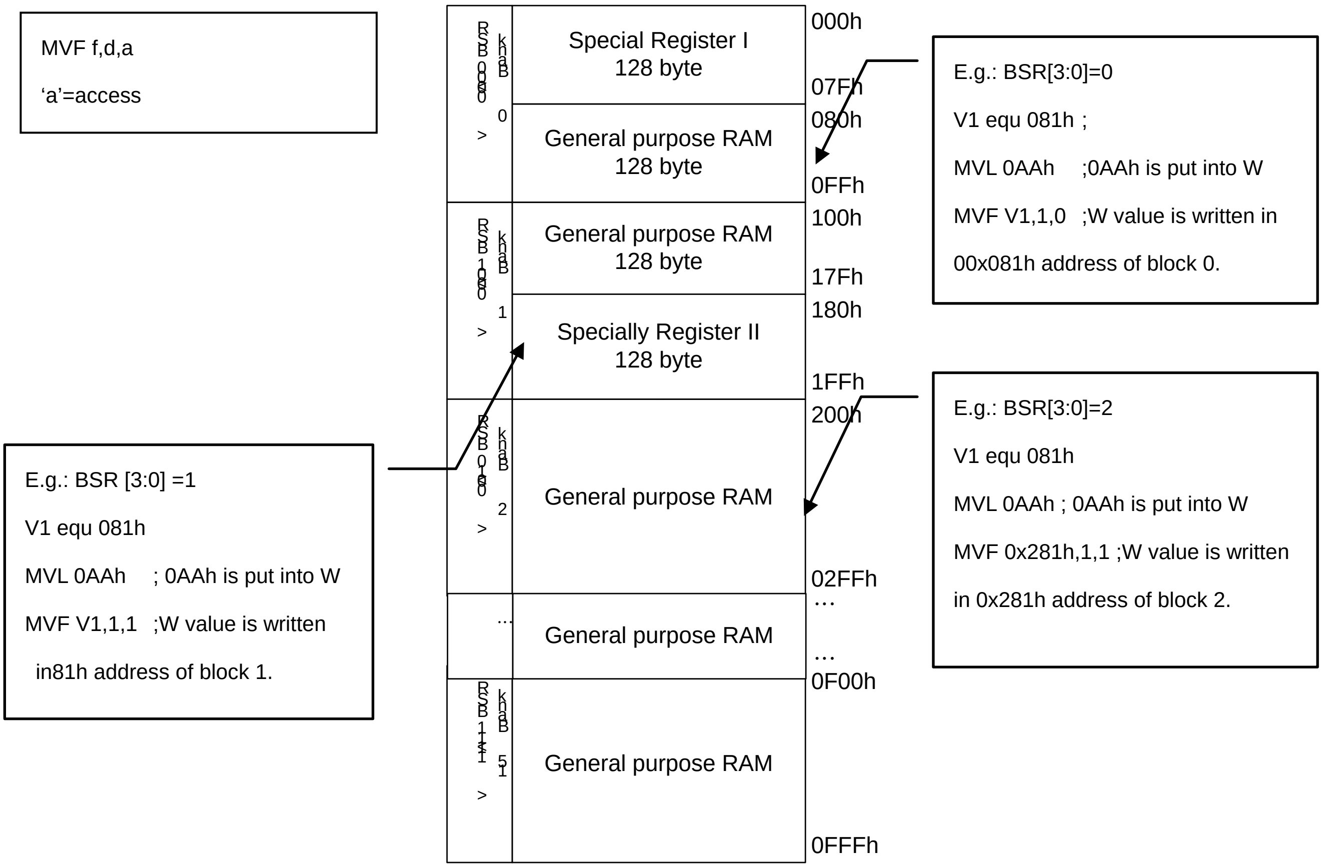
“d” refers to data storage place after computation. If d=0, it is stored in WREG register. If d=1, it is stored in Data Memory Register.

“a” is memory operation block appointing. If a=0, it is operated in block 0. If a=1, it is operated in BSR [3:0] appointed block.

2.2.2.2. Block Selection Control Register

Data memory is planned every 256byte as a block, i.e. 000h～0FFh is a block. If it is going to read / write data register after 0FFh address, it shall set block control register BSR [3:0] and instruction parameter "a" correctly. It is illustrated as the following:

- When a = 0, no matter BSR[3:0] is appointed at any block, the reading / writing of instruction to memory can only be at block 0.
- When a = 1, the reading / writing of H08A CPU Core instruction to data memory will be at BSR [3:0] appointed block. The reading / writing of H08B CPU Core instruction to data memory will be at block 0.



MVF f,d,a

'a'=access

E.g.: BSR[3:0]=0

V1 equ 081h ;

MVL 0AAh ;0AAh is put into W

MVF V1,1,0 ;W value is written in

00x081h address of block 0.

E.g.: BSR [3:0] =1

V1 equ 081h

MVL 0AAh ; 0AAh is put into W

MVF V1,1,1 ;W value is written

in81h address of block 1.

E.g.: BSR[3:0]=2

V1 equ 081h

MVL 0AAh ; 0AAh is put into W

MVF 0x281h,1,1 ;W value is written

in 0x281h address of block 2.

Example 2-1 Relation between Block Selector Example Program and Data Memory

2.2.2.3. Special Register (SR)

Special Register (SR) includes CPU Core peripheral function related registers. It mainly has control function register and data return register. If it takes reading for address which is not defined in data register or address used bit, read data is 0.

In SR, it also has several registers dedicated in instruction collocation. It just introduces two kinds of commonly used registers. One is working register WREG and the other is indirect addressing register FSR. The rest special registers, which are not introduced here, will be taken detailed illustration in each chapter.

2.2.2.3.1. Working Register (WREG)

Working register is shortened as W. It is the most frequently used register for instruction collocation, ranging from data movement, computation and judgment etc.

2.2.2.3.2. Indirect Addressing Register, FSR and INDF

FSR indirect addressing register is composed of index register FSR0[9:0] and index registers INDF0[7:0] and INDF1[7:0]. As the functions are similar, it just illustrates FSR0.

FSR0L [7:0] can write 16-bit data by using an instruction via a special instruction.

INDF0 [7:0] is index register and it can read the data of FSRL0 [7:0] pointed data memory address. Functions are described as the following:

- POINC0[7:0]: When read /write POINC0[7:0] register via instruction, the following events may appear:
  - Return the contents of current FSR0 [7:0] pointed address firstly.
  - The value of index register FSR0 [7:0] is added one and pointed to next address.
- PODEC0[7:0]: When read /write PODEC0[7:0] register via instruction, the following events may appear:
  - Return the contents of current FSR0 [7:0] pointed address firstly.
  - The value of index register FSR0 [7:0] is minus one and pointed to previous address.
- PRINC0[7:0]: When read /write POINC0[7:0] register via instruction, the following events may appear:
  - The value of index register FSR0 [7:0] is added one and pointed to next address firstly.
  - Return the contents of current FSR0 [7:0] pointed address then.
- PLUSW0 [7:0]: When read /write PLUSW0 [7:0] register via instruction, the following events may appear:
  - Add the value of index register FSR0 [7:0] to the content of working register W firstly.
  - Return contents of current FSR0 [7:0] pointed address. W content is values with sign bit, i.e. ±128d.

2.2.2.3.3. General Purpose Register (GPR)

General Purpose Register (GPR) takes data storage, computation, flag setting and other free planning area for the users.

2.2.2.4. Register Instruction---Data Memory Controller

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	uuuu uuuu
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	uuuu uuuu
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu
010H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu

Table 2-2 Data Memory Control Register

INDF0/POINC0/PODEC0/PRINC0/PLUSW0: Index register with different functions

INDF0[7:0]: .See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

POINC0[7:0]: See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

PODEC0[7:0]: See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

PRINC0[7:0]: See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

PLUSW0[7:0]: See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

FSR0: Indirectly Addresses Index Register

FSR0L[7:0]: See 2.2.2.3.2 Indirect Addressing Register, FSR and INDF Description in detail.

WREG: Indirectly Addresses Index Register

WREG[7:0]: See 2.2.2.3.1 Working Register, WREG Description in detail.

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"only write 1

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	uuuu uuuu
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	uuuu uuuu
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu
010H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu
016h	TOSH	-	-	-	-	-	TOS[10]	TOS[9]	TOS[8]	…. xxxx	…. uuuu
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu
018h	STKPTR	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. .000	u$$. .$$$
01Ah	PCLATH	-	-	-	-	-	PC[10]	PC[9]	PC[8]	…. .0000	…. 0000
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
023h	INTE1	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
024h	INTE2	-	-	-	-	I2CERIE	I2CIE	-	-	0000 0000	uuuu uuuu
026h	INTF1	-	ADIF	-	WDTIF	TB1IF	TMAIF	-	E0IF	.000 0000	.uuu uuuu
027h	INTF2	-	-	-	-	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu
02Bh	STATUS	-	-	-	C	-	-	-	Z	…x xxxx	...u uuuu
02Ch	PSTATUS	BOR	PD	TO	IDL	RST	SKERR	-	-	$000 $00.	uu$u u$u.
02Eh	BIECN	-	-	-	-	VPPHV	-	BIEWR	BIERD	1… $.00	1... $.uu
02Fh	BIEARH	ENBIE	-	-	-	-	11-bit look-up Table as BIEAH[2:0]			0... xxxx	u... uuuu
030h	BIEARL	BIE Address Register as BIEAL[5:0] or 11-bit look-up Table as BIEAL[7:0]								xxxx xxxx	uuuu uuuu
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]		-	-	ADRST	CSFON	0000 0000	uuuu u00u
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CPUS	0000 0000	uuuu uuuu
035h	OSCCN1	-	-	ADCS[2:0]			DTMB[1:0]		TMBS	0000 0000	uuuu uuu.
036h	OSCCN2	-	-	-	-	HAOM[1:0]		ENHAO	LPO	.000 0011	.uuu uu11
037h	WDTCN	-	-	-		ENWDT	DWDT[2:0]			0000 0000	uuuu $000
038h	TMACN	ENTMA	TMACL	TMAS	DTMA[2:0]			-	-	0000 00..	u0uu uu..
039h	TMAR	TMA counter Register								0000 0000	uuuu uuuu
041h	CSFCN0	SKRST	EN_RST_PIN	HAOTR[5:0]						0.10 0000	u.uu uuuu
043h	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
044h	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
045h	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
046h	ADCCN1	ENADC	ENHIGN	ENCHP	-	-	ADGN[2:0]			0000 0000	0000 0000
047h	ADCCN2	-	-	-	-	VREGN	DCSET[2:0]			…. 0000	…. 0000
048h	ADCCN3	OSR[3:0]				-	-	-	-	000. ..0.	000. ..0.

Table 2-3 Data Memory List

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
049h	AINET1	INH[2:0]			INL[2:0]			INIS	-	0000 000.	0000 000.
04Ah	AINET2	-	VRH[1:0]		INX[1:0]		VRL[1:0]		-	.000 000.	.000 000.
04Eh	TB1Flag	-	-	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu
04Fh	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	-	-	0000 0000	uuuu u0uu
050h	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu
051h	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu
052h	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu
053h	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
054h	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
055h	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
056h	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
057h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
058h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
061h	CFG	Rsv.					GCRst	ENI2CT	ENI2C	…. .000	…. .uuu
062h	ACT	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu
063h	STA	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu
064h	CRG	CRG[7:0]								0000 0000	uuuu uuuu
065h	TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu
066h	RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu
067h	TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu
068h	SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu
070h	PT1	-	-	-	-	-	-	-	PT10	xx.. ..xx	xx.. ..xx
071h	TRISC1	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
072h	PT1DA	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
073h	PT1PU	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
074h	PT1EG	-	-	FPWMA1	FPWMA0	-	-	E0EG[1:0]		…. 0000	…. uuuu
075h	PT2	-	-	-	-	-	-	PT21	PT20	…. ..xx	…. ..xx
076h	TRISC2	-	-	-	-	-	-	TC21	TC20	…. ..00	…. ..uu
077h	PT2DA	-	-	-	-	-	-	DA21	DA20	…. ..00	…. ..uu
078h	PT2PU	-	-	-	-	-	-	PU21	PU20	…. ..00	…. ..uu
079h	PT3	-	-	PT35	PT34	PT33	PT32	PT31	PT30	..xx xxxx	..xx xxxx
07Ah	TRISC3	-	-	TC35	TC34	TC33	TC32	TC31	TC30	..00 0000	..uu uuuu
07Bh	PT3DA	-	-	DA35	DA34	DA33	DA32	DA31	DA30	..00 0000	..uu uuuu
07Ch	PT3PU	-	-	PU35	PU34	PU33	PU32	PU31	PU30	..00 0000	..uu uuuu
080h ~ 0FFh	GPR0	General Purpose Register as 128Byte								uuuu uuuu	uuuu uuuu

Table 2-4 Data Memory List (Continued)

3. Oscillator, Clock Source and Power Consumption Management

HY10P Series have two clock sources, ie HAO and LPO, as in Table 3-1. It can distribute and manage CPU and peripheral working frequency feasibly through clock controller register. Furthermore, it can adjust power consumption of chip properly to reach the energy saving purpose.

Abstract of Clock Control Register:

OSCCN0	OSCS[1:0],DHS[1:0],DMS[2:0],CPUS[0]
OSCCN1	LCPS[1:0],ADCS[2:0],DTMB[1:0],TMBS[0]
OSCCN2	HAOM[1:0],ENHAO[0],LPO[0]

Sign	Frequency	Frequency Controller OSCCN2[7:0] Configuration			Instruction Execution Status	
		ENHAO[0]	HAOM[1]	HAOM[0]	SLP	IDLE
HAO	8MHz	1	1	1	Stop	Oscillation
	4MHz	1	0	1	Stop	Oscillation
	2MHz	1	0	0	Stop	Oscillation
LPO	14KHz	Oscillation is started after the chip is power on.			Stop	Oscillation

Table 3-1 Internal RC Oscillator Parameter, Frequency Controller Configuration and Instruction Status

3.1. Oscillator

3.1.1. HAO Oscillator

HAO is internal high speed RC oscillator. Typical output frequency is 2.0~8.0MHz.

When CPU of HY10P series products uses other oscillators as working clock source, it can shut off the HAO oscillator via ENHAO[0] setting.

3.1.2. LPO Oscillator

LPO is internal low speed RC oscillator. Typical output frequency is 14KHz. As the current consumption of LPO is about 0.7uA, it is mainly applied to low speed and power saving CPU working mode and Watch Dog Timer clock source.

After HY10P series of products execute Sleep instructions, LPO oscillator is shut off. LPO will be started oscillation automatically when the chip is awakened.

3.2. CPU and Peripheral Circuit Clock Source

3.2.1. Clock Source Configuration

Two groups of oscillators output (OSC_HAO、OSC_LPO) will be started /stopped, switched and pre-scaled frequency via pre-set working clock distributor, and then enter CPU and all peripheral circuits of chip, as in Fig 3-1 .

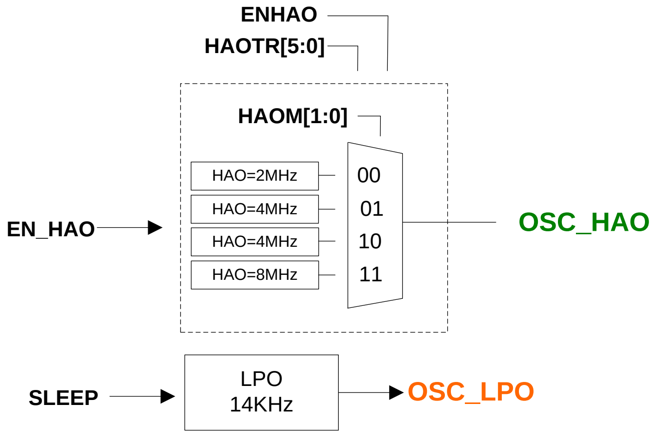


Fig 3-1 Pre-set Working Clock Distributor

3.2.2. CPU Clock Source

CPU has several working frequency for option. Via CPUS [0], optional working frequency is from HS_CK or DHS_CK.

Instruction working frequency adopts 1/4 CPU_CK design and frequency is divided to frequency source of INTR_CK.

- When operating ΣΔADC, it is suggested to divide current working frequency after using HS_CK for CPU to obtain better performance.
- When CPU_CK frequency and instruction execute cycle, it is as in Fig 3-2. Table 3-2 lists the relation between CPU working frequency and instruction cycle briefly.

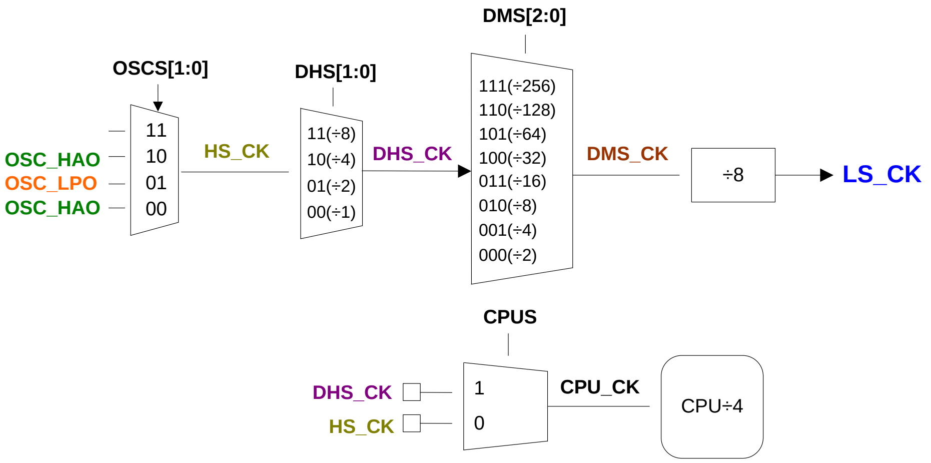


Fig 3-2 CPU and Peripheral Working Clock

Working Frequency CPU_CK	CPU	Instruction	
	Frequency	Frequency	Cycle
8MHZ	8MHZ	2MHz	0.5us
4MHz	4MHz	1MHz	1us
2MHz	2MHz	500kHz	2us
14KHz	14KHz	3.5KHz	285.7us

Table 3-2 CPU Working Frequency and Instruction Execution Cycle

3.2.3. CPU Peripheral Circuit Clock Source

Working clock of HY10P series peripheral circuits is configured by different configuration controller and frequency pre-scaler. The configuration will make detailed illustration in peripheral units, so peripheral working clock configuration diagram is just attached here, as in Fig 3-3.

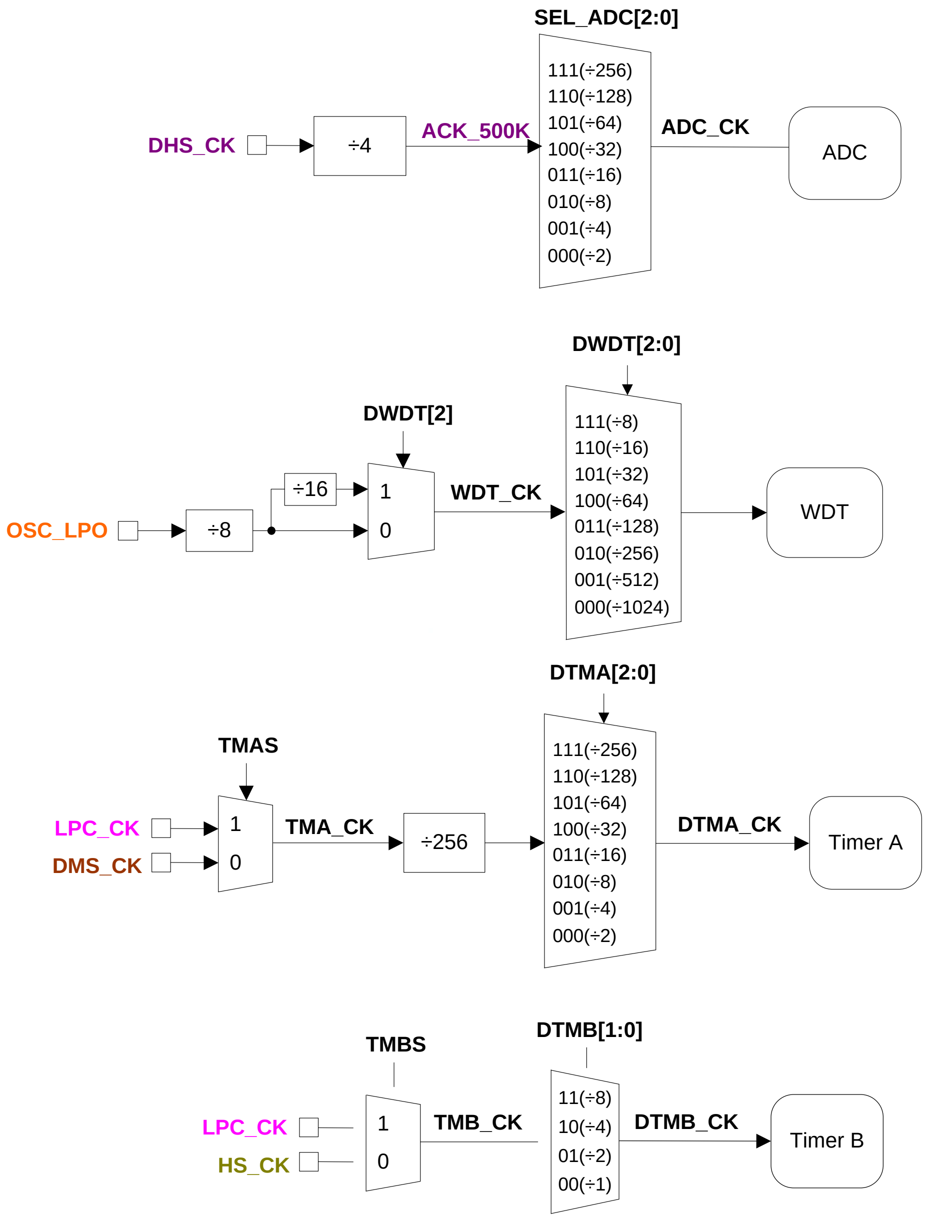


Fig 3-3 Peripheral Working Clock Configuration Diagram

3.3. Register Instruction-Working Clock Source Controller

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
033h	PWRCN								CSFON	0000 0000	uuuu u00u
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CPUS	0000 0000	uuuu uuuu
035h	OSCCN1	-	-	ADCS[2:0]			DTMB[1:0]		TMBS	0000 0000	uuuu uuu.
036h	OSCCN2	-	-	-	-	HAOM[1:0]		ENHAO	LPO	.000 0011	.uuu uu11
041h	CSFCN0	SKRST	EN_RST_PIN	HAOTR[5:0]						0.10 0000	u.uu uuuu

Table 3-3 Working Clock Source Control Register

OSCCN0 [7:0] Chip Working Frequency Control Register 0

Frequency Controller of OSCS [1:0] HS_CK

<11> Not Used

<10>OSC_HAO

<01>OSC_LPO

<00>OSC_HAO

Frequency Allocation Selector of DHS [1:0] DHS_CK

<11>HS_CK ÷ 8

<10>HS_CK ÷ 4

<01>HS_CK ÷ 2

<00>HS_CK ÷ 1

Frequency Allocation Selector of DMS [2:0] DMS_CK

<111>DHS_CK ÷ 256

<110>DHS_CK ÷ 128

<101>DHS_CK ÷ 64

<100>DHS_CK ÷ 32

<011>DHS_CK ÷ 16

<010>DHS_CK ÷ 8

<001>DHS_CK ÷ 4

<000>DHS_CK ÷ 2

Frequency Selector of CPUS [0] CPU_CK

<1>DHS_CK

<0>HS_CK

OSCCN1 [7:0] Chip Working Frequency Control Register 1

ADCS [2:0]: SD18 Peripheral Working Frequency Pre-eliminator

111: ADC_CK/256

110: ADC_CK/128

101: ADC_CK/64

100: ADC_CK/32

011: ADC_CK/16

010: ADC_CK/8

001: ADC_CK/4

000: ADC_CK/2

Frequency Allocation Selector of DTMB [1:0] DTMB_CK

<11>TMB_CK ÷ 8

<10>TMB_CK ÷ 4

<01>TMB_CK ÷ 2

<00>TMB_CK ÷ 1

Frequency Selector of TMBS [0] TMB_CK

<1>LPC_CK

<0>HS_CK

OSCCN2 [7:0] Chip Working Frequency Control Register 2

HAOM [1:0] Internal Oscillator HAO Oscillation Frequency Selector

<11>8MHz

<10> cannot be set

<01>4MHz

<00>2MHz

ENHAO: Internal HAO Start Control Bit

1: Start

0: Stop

LPO [0] Internal Oscillator LPO Status Flag

<1> Start

<0>Stop

※The bit is status bit. It can only be read but not written. After executing Sleep instruction, LPO oscillator will be stopped automatically. When the chip is awakened, LPO will be started automatically.

PWRCN [7:0] Linear Regulator and Analog Control Register

CSFON [0] CSF (Chip Special Function) Start writing controller

<1> Start CSF writing function. When the user is necessary to set block control register, it must set the CFSON[0] as <1> before writing in CSFCN0[7:0].

<0> Not start CSF function.

CSFCN0[7:0] Special Control Bit Register

HAOTR[5:0]HAO Frequency Center Adjustment Controller

<111111>Adjust -10%, minimum
<111110>Adjust -9.68%
<111101>Adjust -9.35%
<111100>Adjust -9.03%
<111011>Adjust -8.71%
<111010>Adjust -8.39%
<111001>Adjust -8.06%
<111000>Adjust -7.74%
<110111>Adjust -7.42%
<110110>Adjust -7.10%
<110101>Adjust -6.77%
<110100>Adjust -6.45%
<110011>Adjust -6.13%
<110010>Adjust -5.81%
<110001>Adjust -5.48%
<110000>Adjust -5.16%
<101111>Adjust -4.84%
<101110>Adjust -4.52%
<101101>Adjust -4.19%
<101100>Adjust -3.87%
<101011>Adjust -3.55%
<101010>Adjust -3.23%

<101001>Adjust -2.90%
<101000>Adjust -2.58%
<100111>Adjust -2.26%
<100110>Adjust -1.94%
<100101>Adjust -1.61%
<100100>Adjust -1.29%
<100011>Adjust -0.97%
<100010>Adjust -0.65%
<100001>Adjust -0.32%
<100000> Central point 0.00%
<011111>Adjust 0.31%
<011110>Adjust 0.63%
<011101>Adjust 0.94%
<011100>Adjust 1.25%
<011011>Adjust 1.56%
<011010>Adjust 1.88%
<011001>Adjust 2.19%
<011000>Adjust 2.50%
<010111>Adjust 2.81%
<010110>Adjust 3.13%
<010101>Adjust 3.44%
<010100>Adjust 3.75%

<010011>Adjust 4.06%
<010010>Adjust 4.38%
<010001>Adjust 4.69%
<010000>Adjust 5.00%
<001111>Adjust 5.31%
<001110>Adjust 5.63%
<001101>Adjust 5.94%
<001100>Adjust 6.25%
<001011>Adjust 6.56%
<001010>Adjust 6.88%
<001001>Adjust 7.19%
<001000>Adjust 7.50%
<000111>Adjust 7.81%
<000110>Adjust 8.13%
<000101>Adjust 8.44%
<000100>Adjust 8.75%
<000011>Adjust 9.06%
<000010>Adjust 9.38%
<000001>Adjust 9.69%
<000000>Adjust 10.00%, maximum

3.4. Power Consumption Management and Operation Status

HY10P series CPU provides three types of working modes to make user obtain best management on execution effectiveness and power saving. The three types of modes are operation mode, standby mode and sleep mode.

3.4.1. Operation Mode

Operation mode mainly refers to that CPU handles all appeared events according to clock source. At the moment, chip peripheral can be operated normally and power consumption handles max status under the same clock.

3.4.2. Standby Mode

Standby mode is entered via IDLE instruction. It mainly refers to that CPU stops operating waiting for wake when enter energy saving status, and sets IDLEB [0] flag bit of PSTATUS reset register as <1>. Under this mode, chip peripheral can be operated normally. When the peripheral appears interrupt event, it will awaken CPU[3]. Additionally, watch dog counter finally generated signal belongs to interrupt signal, but not reset signal.

Under standby mode, CPU is in pause mode and is stopped under the IDLE instruction. The internal oscillator is not affected and also is not closed. If the user wants to reach energy saving status, it depends on the application condition. At the moment, it shall switch off peripheral or oscillator and other resources. The chip must get to the standby mode status via external interrupt source or other peripheral resource interrupt signal.

Under standby mode, if it encounters interrupt resource and leaves standby mode, it needs 2 instruction cycles time to back to 04H position of interrupt vector. If the CPU frequency source is internal ideal 2MHZ, the other instruction cycle time is 2usec. Therefore, it needs 4usec program for awakening and to back to the position of interrupt vector. If CPU frequency source is internal ideal 14KHZ, the other instruction cycle time is 286usec. Therefore, it needs 536usec program for awakening and to back to the position of interrupt vector.

If under standby mode, CPU frequency source is internal 14KHZ, and internal 2MHZ oscillator has been closed, while 2MHZ oscillator is started after awakening, complete starting 2MHZ oscillator requires two 14KHZ instruction awakening time plus 128 2MHZ instruction oscillation time. It is equivalent to about 792usec. After it, internal 2MHZ oscillator can be completed oscillation normally.

[3] After CPU is suffered interrupt signal awakening, PC (Program Counter) will jump to interrupt vector position (0x004h). See *Reset and Interrupt Section* on detailed illustration on PSTATUS reset register and interrupt service vector.

4. RESET

HY10P series of reset circuits include the following 4 types of events to trigger reset signal. Reset block diagram is as Fig 4-1.

BOR	power interference reset
RST	external reset input pin
WDT	watch dog reset
SKERR	stack error reset (determined by the user)

Abstract of Operation Status Register:

PSTATUS BOR[0],PD[0],TO[0],IDL[0],SKERR[0]

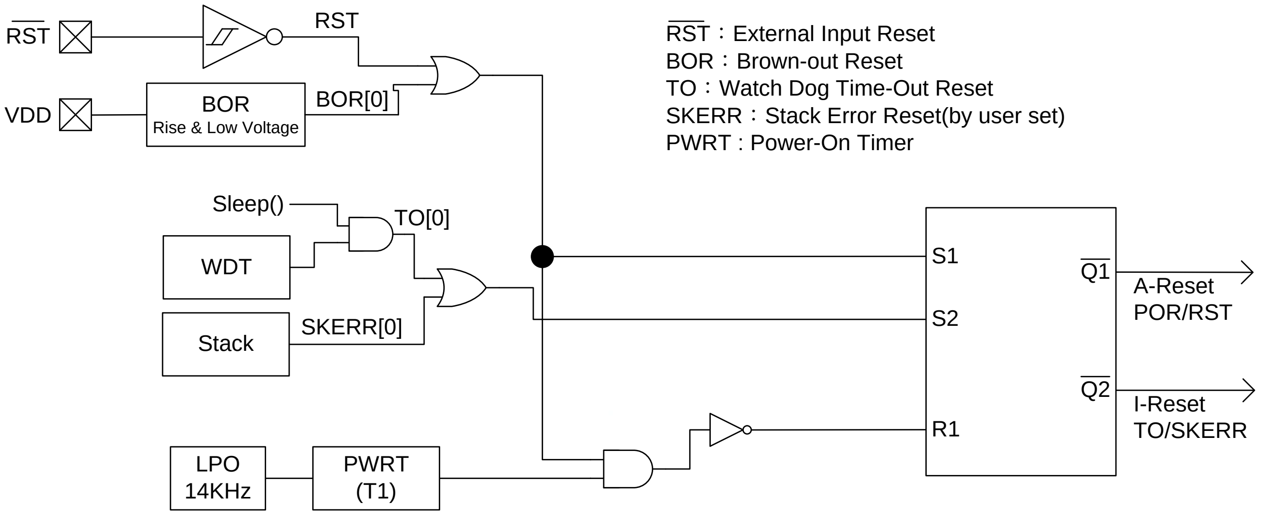


Fig 4-1 Reset Block Diagram

These reset events can be divided into software reset and hardware reset, as in Table 4-1. CPU reset program is started by 0x0000h.

Reset Type	Event	Sign	Description
Hardware Reset	BOR RST	A-RESET	CPU restart needs internal oscillator completes counter starting before entering normal operation status.
Software Reset	WDT SKERR	I-RESET	It only clears partial register. CPU is back to normal operation status rapidly.

Table 4-1 Reset Rank List

4.1. Reset Event Description

4.1.1. BOR Power Interference Reset

When CPU is suffered from external interference during powering on process or the power is suffered from external interference, CPU will enter into normal operation voltage from abnormal operation and low operation voltage. Therefore, if CPU cannot be in reset status when the operation voltage is too low, it may cause crash of CPU and make the operation of peripheral circuits abnormal. Therefore, it must rely on BOR circuit function. When it detects that operation voltage is suffered from interference and voltage level is lower than the designed value, it may generate reset signal and make chip enter restart status, until the operation voltage is recovered. Then it will relieve reset signal and make chip enter normal operation mode.

When BOR reset occurs, BOR[0] flag in PSTATUS[7:0] register is set as <1> to record the occurred event.

HY10P series of BOR circuits will generate about 0.6uA power consumption. It cannot be closed via program or other setting method.

4.1.2. RST External Input Reset

When the voltage level of external RST pin is lower than designed value[4], it may generate reset signal and make chip enter restart status, until the operation voltage is recovered. Then it will relieve reset signal and make chip enter normal operation mode.

4.1.3. WDT Watch Dog Counter Reset

WDT watch dog counter may generate reset signal and make chip enter rapid start status when the operation mode counter is ended. When WDT watch dog occurs reset, TO[0] flag in the PSTATUS[7:0] register will be set as <1> to record the occurred event.

Note: WDT watch dog finally generated signal has two types. It may generate reset signal when the chip is under operation mode. If the chip is under standby mode, it may generate interrupt event signal to awaken CPU. See *Watch DOG WDT Section* on detailed operation description.

4.1.4. SKERR Stack Error Reset

When the program occurs stack overflow or underflow, it may generate reset signal and make chip enter rapid start status. When SKERR stack error reset occurs, SKERR[0] flag in PSTATUS[7:0] register is set as <1> to record the occurred event. See *Memory Section* on detailed operation description.

[4] The pin has another two kinds of functions. One is when RST input voltage is pulled up to meet V_{IU} specification, the chip enters OTP program mode. The other is when RST input voltage meets V_{IL} specification, the chip enters current leakage test mode.

4.2. Status Register

Chip operation status is displayed in PSTATUS[7:0] reset register. Mutual relation is as in Table 4-2.

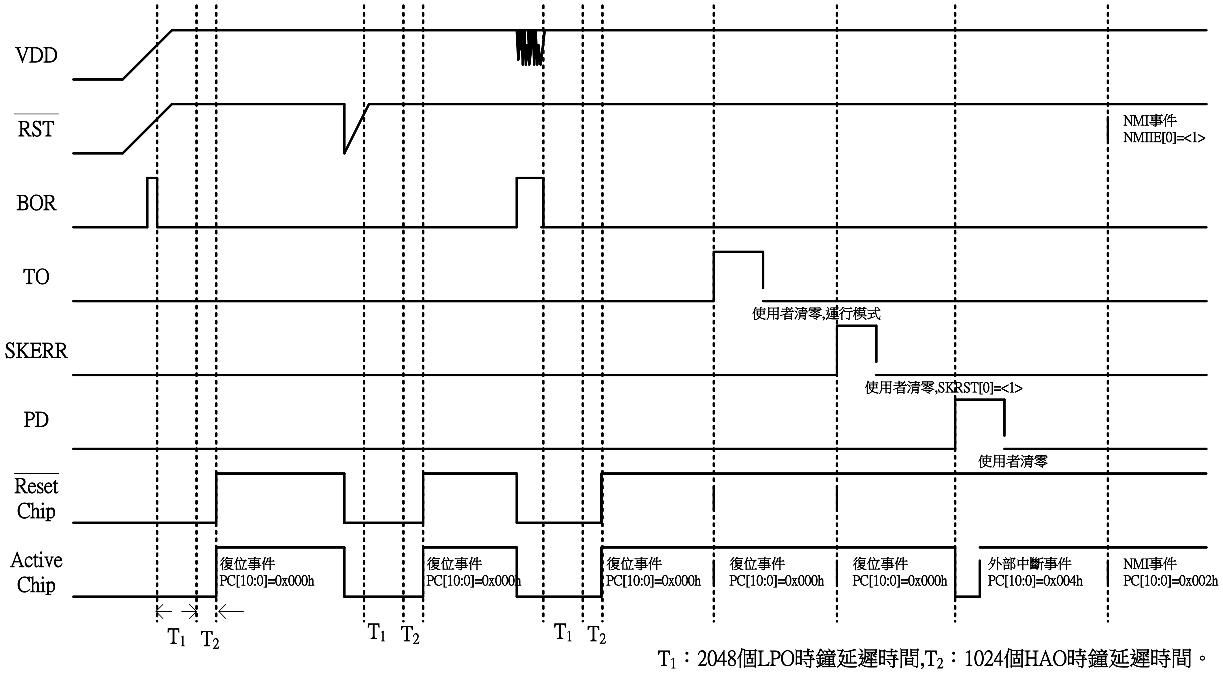
"0": Not Occured,"1": Occurred, "u": Not Changed,"-": Not Used

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

Table 4-2 Reset Status Flags Relation Table

4.2.1. Sequence Diagram of Reset Status

Sequence diagram from hardware reset signal occurrence to entering operation status is as in Fig 4-2. Time from different reset signal occurrence to entering operation status is as in Table 3-2(b).



T_1：2048個LPO時鐘延遲時間,T_2：1024個HAO時鐘延遲時間。

Fig 4-2 Sequence Diagram of Reset & Operation Mode and Status Flags

4.2.2. Register Instruction---Reset Status

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
02Ch	PSTATUS	BOR	PD	TO	IDL	RST	SKERR	-	-	$000 $00.	uu$u u$u.
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]		ENREFO	AD1RST	AD2RST	CSFON	0000 0000	uuuu u00u
041h	CSFCN0	SKRST	EN_RST_PIN							0.10 0000	u.uu uuuu

Table 4-3 Reset Register

PSTATUS: Status Register

BOR[0]； BOR reset event flag

<1>Power interference reset has been occurred. It shall use RST or instruction for clearing.

<0>Not occurred interference reset

PD[0]；SLEEP event flag

<1>Sleep event has been occurred. It shall use BOR, RST or instruction for clearing.

<0> Not occurred sleep event

TO[0]； WDT operation mode counting overflow flag

<1> WDT reset event has been occurred. It shall use BOR, RST or instruction for clearing.

<0> Not occurred WDT counting overflow event

IDL [0]；Standby IDLE event flag

<1> IDLE event has been occurred. It shall use BOR, RST or instruction for clearing.

<0> Not occurred IDLE event

RST[0]；External RST pin low potential event flag

<1> RST pin reset event has been occurred. It shall use BOR, or instruction for clearing.

<0> Not occurred RST pin reset event

SKERR[0]；Stack error reset flag

<1> Stack error. It shall use BOR, RST or instruction for clearing.

<0> No stack error

PWRCN[7:0] Linear Regulator and Analog Control Register

CSFON [0] CSF (Chip Special Function) Start writing controller

<1> Start CSF writing function. When the user is necessary to set block control register, it must set the CFSON[0] as <1> before writing in CSFCN0[7:0].

<0> Not start CSF function. CSFCN0[7:0] register cannot be read / written.

CSFCN0[7:0] Special Control Bit Register

SKRST[0] Stack error reset controller

<1> Start error reset chip

<0> Not start error reset chip

EN_RST_PIN[0] Reset chip pin set

<1>Not start hardware reset chip pin, set PT1.0 as general input.

<0> Start hardware reset chip pin. PT1.0 is set as RST pin.

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"only write 1

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	uuuu uuuu
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	uuuu uuuu
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu
010H	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu
016h	TOSH	-	-	-	-	-	TOS[10]	TOS[9]	TOS[8]	…. xxxx	…. uuuu
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu
018h	STKPTR	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. .000	u$$. .$$$
01Ah	PCLATH	-	-	-	-	-	PC[10]	PC[9]	PC[8]	…. .0000	…. 0000
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
023h	INTE1	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
024h	INTE2	-	-	-	-	I2CERIE	I2CIE	-	-	0000 0000	uuuu uuuu
026h	INTF1	-	ADIF	-	WDTIF	TB1IF	TMAIF	-	E0IF	.000 0000	.uuu uuuu
027h	INTF2	-	-	-	-	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu
02Bh	STATUS	-	-	-	C	-	-	-	Z	…x xxxx	...u uuuu
02Ch	PSTATUS	BOR	PD	TO	IDL	RST	SKERR	-	-	$000 $00.	uu$u u$u.
02Eh	BIECN	-	-	-	-	VPPHV	-	BIEWR	BIERD	1… $.00	1... $.uu
02Fh	BIEARH	ENBIE	-	-	-	-	11-bit look-up Table as BIEAH[2:0]			0... xxxx	u... uuuu
030h	BIEARL	BIE Address Register as BIEAL[5:0] or 11-bit look-up Table as BIEAL[7:0]								xxxx xxxx	uuuu uuuu
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]		-	-	ADRST	CSFON	0000 0000	uuuu u00u
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CPUS	0000 0000	uuuu uuuu
035h	OSCCN1	-	-	ADCS[2:0]			DTMB[1:0]		TMBS	0000 0000	uuuu uuu.
036h	OSCCN2	-	-	-	-	HAOM[1:0]		ENHAO	LPO	.000 0011	.uuu uu11
037h	WDTCN	-	-	-		ENWDT	DWDT[2:0]			0000 0000	uuuu $000
038h	TMACN	ENTMA	TMACL	TMAS	DTMA[2:0]			-	-	0000 00..	u0uu uu..
039h	TMAR	TMA counter Register								0000 0000	uuuu uuuu
041h	CSFCN0	SKRST	EN_RST_PIN	HAOTR[5:0]						0.10 0000	u.uu uuuu
043h	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
044h	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
045h	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
046h	ADCCN1	ENADC	ENHIGN	ENCHP	-	-	ADGN[2:0]			0000 0000	0000 0000
047h	ADCCN2	-	-	-	-	VREGN	DCSET[2:0]			…. 0000	…. 0000
048h	ADCCN3	OSR[3:0]				-	-	-	-	000. ..0.	000. ..0.

Table 4-4 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"only write 1

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
049h	AINET1	INH[2:0]			INL[2:0]			INIS	-	0000 000.	0000 000.
04Ah	AINET2	-	VRH[1:0]		INX[1:0]		VRL[1:0]		-	.000 000.	.000 000.
04Eh	TB1Flag	-	-	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu
04Fh	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	-	-	0000 0000	uuuu u0uu
050h	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu
051h	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu
052h	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu
053h	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
054h	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
055h	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
056h	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
057h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
058h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
061h	CFG	Rsv.					I2CRST	ENI2CT	ENI2C	…. .000	…. .uuu
062h	ACT	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu
063h	STA	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu
064h	CRG	CRG[7:0]								0000 0000	uuuu uuuu
065h	TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu
066h	RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu
067h	TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu
068h	SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu
070h	PT1	-	-	-	-	-	-	-	PT10	xx.. ..xx	xx.. ..xx
071h	TRISC1	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
072h	PT1DA	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
073h	PT1PU	-	-	-	-	-	-	-	-	0000 0000	uuuu uuuu
074h	PT1EG	-	-	FPWMA1	FPWMA0	-	-	E0EG[1:0]		…. 0000	…. uuuu
075h	PT2	-	-	-	-	-	-	PT21	PT20	…. ..xx	…. ..xx
076h	TRISC2	-	-	-	-	-	-	TC21	TC20	…. ..00	…. ..uu
077h	PT2DA	-	-	-	-	-	-	DA21	DA20	…. ..00	…. ..uu
078h	PT2PU	-	-	-	-	-	-	PU21	PU20	…. ..00	…. ..uu
079h	PT3	-	-	PT35	PT34	PT33	PT32	PT31	PT30	..xx xxxx	..xx xxxx
07Ah	TRISC3	-	-	TC35	TC34	TC33	TC32	TC31	TC30	..00 0000	..uu uuuu
07Bh	PT3DA	-	-	DA35	DA34	DA33	DA32	DA31	DA30	..00 0000	..uu uuuu
07Ch	PT3PU	-	-	PU35	PU34	PU33	PU32	PU31	PU30	..00 0000	..uu uuuu
080h ~ 0FFh	GPR0	General Purpose Register as 128Byte								uuuu uuuu	uuuu uuuu

Table 4-5 Data Memory Reset Status(Continued)

5. Interrupt

Interrupt is composed of interrupt start controller INTE and interrupt event flag INTF. When Interrupt service is established, if it appears interrupt event, program counter (PC) will jump to interrupt address 0x0004h in PM to execute interrupt service program.

Abstract of Interrupt Control Register:

INTE0	GIE[0],ADCIE[0],TMCIE[0],TMBIE[0],TMAIE[0],WDTIE[0],E1IE[0],E0IE[0]
INTE1	I2CERIE[0], I2CIE[0]
INTF0	ADCIF[0],TMCIF[0],TMBIF[0],TMAIF[0],WDTIF[0],E1IF[0],E0IF[0]
INTF1	I2CERIF[0],I2CIF[0]

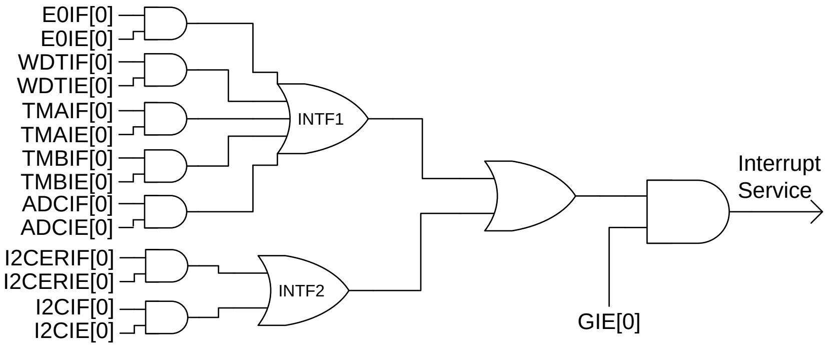


Fig5-1 Interrupt Vector Block Diagram

Interrupt service event governor has two layers totally. The higher layer is interrupt service event controller GIE [0] and the lower layer is interrupt event start control bit.

- To start interrupt event, it only needs to set the controller of corresponding interrupt event start controller INTEx [7:0] as <1>. On the contrary, it will close the interrupt event when it is set as <0>.
- To start interrupt service, it only needs to set the interrupt service controller GIE[0] of interrupt control register INTE0[7:0] as <1>. On the contrary, it will close the interrupt service when it is set as <0>.

When entering into interrupt service vector, GIE [0] will be set as <0> automatically. When it is going to return interrupt occurrence address after completing interrupt service program execution, it can execute interrupt return instruction RETI directly. At the moment, GIE [0] will be set as <1> automatically. Or it executes return instruction RET, and GIE [0] status maintains at 0 at the moment.

5.1. Register Instruction-Interrupt

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
023h	INTE0	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
024h	INTE1	-	-	-	-	I2CERIE	I2CIE	-	-	0000 0000	uuuu uuuu
026h	INTF0	-	ADIF	-	WDTIF	TB1IF	TMAIF	-	E0IF	.000 0000	.uuu uuuu
027h	INTF1	-	-	-	-	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu

Table 5-1 Interrupt Register

INTE0: Interrupt Start Control Register 0

GIE [0]: Interrupt Service Controller

1: Start

0: Close

ADCIE[0]: ADC Interrupt Event Start Controller

1: Start (Analog Digital Converter, SD18)

0: Close

TMBIE[0]: Timer-B Interrupt Event Start Controller

1: Start (Timing/ Timer B, TMB)

0: Close

TMAIE[0]: Timer-A Interrupt Event Start Controller

1: Start (Timing/ Timer A, TMA)

0: Close

WDTIE[0]: Watch Dog Interrupt Event Start Controller

1: Start (Watch Dog, WDT)

0: Close

E0IE[0]: Input Pin 0 Interrupt Event Start Controller

1: Start (External Input Pin, PT1.0)

0: Close

INTE1: Interrupt Start Control Register 1

I2CERIE[0] peripheral I2C error interrupt vector service controller

<1>Start I2C interrupt vector service

<0>Close I2C interrupt vector service

I2CIE[0] peripheral I2C interrupt vector service controller

<1>Start I2C interrupt vector service

<0>Close I2C interrupt vector service

INTF0: Interrupt Event Flag Register 0

ADCIF[0]: ADC Interrupt Event Flag

1: Occurred (Analog Digital Converter, SD18)

0: Not Occurred

TMBIF[0]: Timer-B Interrupt Event Flag

1: Occurred (Timing/Timer B,TMB)

0: Not Occurred

TMAIF[0]: Timer-A Interrupt Event Flag

1: Occurred (Timing/Timer A,TMA)

0: Not Occurred

WDTIF[0]: Watch Dog Interrupt Event Flag

1: Occurred (Watch Dog, WDT)

0: Not Occurred

E0IF[0]: Input Pin 0 Interrupt Event Flag

1: Occurred (External Input Pin, PT1.0)

0: Not Occurred

INTF1: Interrupt Event Flag Register 1

I2CERIF[0] peripheral I2C error interrupt event flag controller

<1> Occurred I2C interrupt event

<0> Not Occurred I2C interrupt event

I2CIF[0] peripheral I2C interrupt event flag controller

<1> Occurred I2C interrupt event

<0> Not Occurred I2C interrupt event

6. Input/ Output Port (I/O)

Every 8 pins of Input/ Output Port (I/O) are one port. It can be taken as digital input and output and analog signal measuring channel. Each port is controlled by a group of registers. It may have different I/O register composition for different products.

Abstract of I/O related register:

PT	PT1[0], PT2[1:0], PT3[5:0]
TRISC	TC2[1:0], TC3[5:0],
PTDA	DA2[1:0]
PTPU	PU2[1:0], PU3[5:0]
PT1EG	FPWMA[1:0], E0EG[1:0]

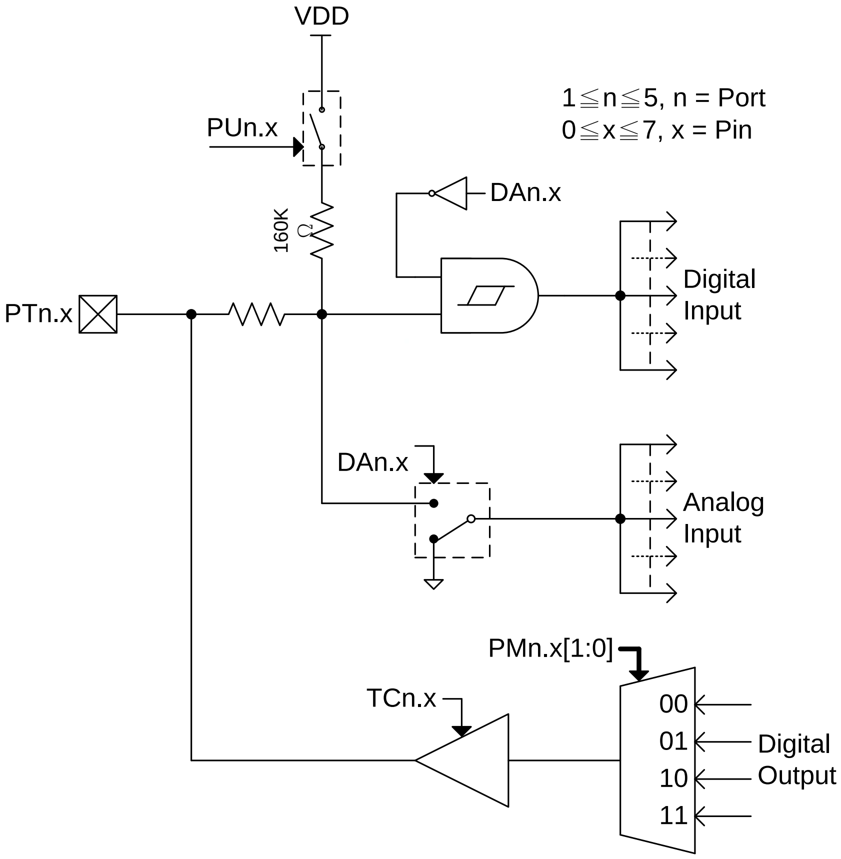


Fig 6-1 I/O Architecture Block Diagram

6.1. Introduction of PORT related register

PORT mainly provides digital or analog signal input and output pins.

6.1.1. PT Status Control Register

When I/O is set as input, it can read current I/O status at corresponding register position. I/O input is high potential when the reading is 1 at the moment, and low potential when is 0.

When I/O is set as output, it can control output status at corresponding register position. I/O output is high potential when it is set as <1> at the moment, and low potential when is<0>.

6.1.2. TRISC Input / Output Control Register

When I/O is selected as input or output, set <1>I/O as output status and <0> as input status. When I/O is set as input status, it must give definite input potential when the chip enters into sleep status. Don't make I/O in floating status to avoid causing power leakage phenomena of chip.

6.1.3. PTDA Digital or Analog Input Control Register

When I/O is set as analog or digital input status, set <1> is analog and <0> is digital input. It shall consider the setting status of other I/O related register when taking setting to avoid mutual interference of digital /analog signal.

6.1.4. PTPU Pull-up Resistor Control Register

When I/O is set as whether pull-up resistor function is started. Set <1> as I/O starting and set <0> as disconnection. Before the chip enters into sleep mode, if I/O is set as digital input status and external circuit connection way may cause floating phenomena of I/O, it can start pull-up resistor to avoid I.O floating and causing the chip entering into sleep mode and producing current leakage.

6.1.5. PTEG Interrupt Signal Creation Condition

When I/O external input potential belongs to certain changes, it may cause interrupt signal. Potential changes can be divided into rising edge (0→1) change, falling edge (1→0) change and potential transition (0→1or 1→0).

6.2. Input/ Output Port 1, I/O Port1

i": Input, "o": Output, "a": Analog , "c": cmos i/o, "x": No definition, "p": power ",

Pin Name	Design		Register Setting			Description
	Type	Buffer	CSFON[0]	EN_RST_PIN[0]	PT1EG[1:0]	
PT1.0	i	c	1	0	XX	Digital input pin
INT0	i	s	1	0	00~11	External interrupt source
VPP	p	p	X	X	XX	OTP burning voltage pin
RST	i	s	1	1	XX	Reset pin

Table 6-1 PORT1 Function

6.2.1. Register Instruction---PORT1

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
023h	INTE0	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
026h	INTF0								E0IF	.000 0000	.uuu uuuu
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]			AD1RST	AD2RST	CSFON	0000 0000	uuuu u00u
041h	CSFCN0		EN_RST_PIN							0.10 0000	u.uu uuuu
070h	PT1	-	-	-	-	-	-	-	PT10	xx.. ..xx	xx.. ..xx
074h	PT1EG	-				-		E0EG[1:0]		…. 0000	…. uuuu

Table 6-2 PORT1 Control Register

INTE0/INTF0: See Chapter *Interrupt* .

PT1: PORT1 Status Control Register

PT1.0: External pin control bit

1: high potential

0: low potential

PT1EG[1:0]: Pin interrupt way control register

E0EG[1:0] pin PT1.0 interrupt way controller

<11>potential transition (0→1 or 1→0）, i.e. Producing interrupt event; interrupt event shall be occurred after reading PT1.0[0].

<10> potential transition (0→1 or 1→0）,i.e. Producing interrupt event; interrupt event will be occurred so long as potential transition.

<01>rising edge (0→1）

<00>falling edge (1→0）

PT1.0 default is reset mode(when BOR occurs)，built-in 180K ohm resistors pull high.

If PT1.0 changes setting, it must set ADDR 41h CSFUN bit<6>=1 after setting ADDR 33h PWRCN bit<0>=1. Change PT1.0 setting as input mode. Then, set ADDR 33h PWRCN bit<0>=0 to avoid modifying false writing to the setting.

PWRCN[7:0]: Linear Regulator and Analog Control Register

CSFON [0] CSF (Chip Special Function) Start writing controller

<1> Start CSF writing function. When the user is necessary to set block control register, it must set the CFSON[0] as <1> before writing in CSFCN0[7:0].

<0> Not start CSF function. CSFCN0 register cannot be read and written.

CSFCN0[7:0]: Special Control Bit Register

EN_RST_PIN[0] Hardware reset pin setting

<1>Not start hardware reset chip pin, set PT1.0 as general input.

<0> Start hardware reset chip pin. PT1.0 is set as RST pin.

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

i": Input, "o": Output, "a": Analog, "c": cmos i/o, "x": No definition, "p": power

Pin Name	Design		Register Setting			Description
	Type	Buffer	TC[0]	DA[0]	FPWMA[0]	
PT2.0	i/o	c	x	0	0	Digital input /output pin
AI6	a	a	x	1	0	Analog input pin
PWMA0	o	c	1	0	1	PWM output pin
PT2.1	i/o	c	X	0	0	Digital input /output pin
AI7	a	a	x	1	0	Analog input pin
PWMA1	o	c	1	0	1	PWM output pin

Table 6-3 PORT2 Function

6.3.1. Register Instruction---PORT2

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
074h	PT1EG	-		FPWMA1	FPWMA0	-				…. 0000	…. uuuu
075h	PT2	-	-	-	-	-	-	PT21	PT20	…. ..xx	…. ..xx
076h	TRISC2	-	-	-	-	-	-	TC21	TC20	…. ..00	…. ..uu
077h	PT2DA	-	-	-	-	-	-	DA21	DA20	…. ..00	…. ..uu
078h	PT2PU	-	-	-	-	-	-	PU21	PU20	…. ..00	…. ..uu

Table 6-4 PORT2 Control Register

PT2: PORT2 Status Control Register

PT2.x: External pin control bit (0≦x≦1)

1: high potential

0: low potential

TRISC2: I/O Control Register

TC2.x: External pin input /output control bit (0≦x≦1)

1: output

0: input

PT2DA: Digital or analog input control register

DA2.x: External pin input analog or digital signal control bit (0≦x≦1)

1: analog

0: digital

PT2PU: Pull-up resistor control register

PU2.x: External pin pull-up resistor control bit (0≦x≦1)

1: Start

0: Close

PT1EG[1:0] Pin interrupt way control register

PWMA1[0]: PWMA1 pin output controller

<1>output

<0>not output

PWMA0[0]: PWMA0 pin output controller

<1>output

<0>not output

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

"i": Input, "o": Output, "a": Analog, "c": cmos i/o, "s": Smith trigger, "x": No definition, "p": power

Pin Name	Design		Register Setting		Description
	Type	Buffer	TC[0]	DA[0]	
PT3.0	i/o	c	x	0	Digital I/O pin
AI0	a	a	0	1	Analog input pin
PT3.1	i/o	c	x	0	Digital I/O pin
AI1	a	a	0	1	Analog input pin
PSCK	i	s	0	0	OTP reading/writing interface SCK pin
PT3.2	i/o	c	x	0	Digital I/O pin
AI2	a	a	0	1	Analog input pin
PSDI	i	s	0	0	OTP reading/writing interface SDI pin
PT3.3	i/o	c	x	0	Digital I/O pin
AI3	a	a	0	1	Analog input pin
PSDO	o	c	1	0	OTP reading/writing interface SDO pin
PT3.4	i/o	c	x	0	Digital I/O pin
AI4	a	a	0	1	Analog input pin
SCL	i/o	s	x	0	I2C communication interface pin
PT3.5	i/o	c	x	0	Digital I/O pin
AI5	a	a	0	1	Analog input pin
SDA	i/o	s	x	0	I2C communication interface pin

Table 6-5 PORT3 Function

6.4.1. Register Instruction---PORT3

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
Address	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**ARST**	**IRST**
079h	PT3	-	-	PT35	PT34	PT33	PT32	PT31	PT30	..xx xxxx	..xx xxxx
07Ah	TRISC3	-	-	TC35	TC34	TC33	TC32	TC31	TC30	..00 0000	..uu uuuu
07Bh	PT3DA	-	-	DA35	DA34	DA33	DA32	DA31	DA30	..00 0000	..uu uuuu
07Ch	PT3PU	-	-	PU35	PU34	PU33	PU32	PU31	PU30	..00 0000	..uu uuuu

Table 6-6 PORT3 Control Register

PT3: PORT3Status Control Register

PT3.x: External Pin Control Bit (0≦x≦5）

1: high potential.

0: low potential

TRISC3: I/O Control Register

TC3.x: External Pin I/O Control Bit (0≦x≦5）

1: Output

0: Input

PT3DA: Digital or analog input control register

DA3.x: External Pin input analog or digital signal control bit (0≦x≦5）

1: analog

0: digital

PT3PU: Pull-up resistor control register

PU3.x: External pin pull-up resistor control bit (0≦x≦5）

1: Start

0: Close

7. Watch Dog Timer

Namely, watch dog WDT is the guarder of chip. It is mainly used to genrate awakening event.

Operation Mode

- Overflow of watch dog counter produces reset signal to restart the chip.
- Software can be used to zero timer.

Sleep Mode

- Watch dog WDT is closed, and cannot be used.

Standby Mode

- Overflow of watch dog counter produces interrupt event to awaken the chip.

Abstract of WDT Related Register:

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

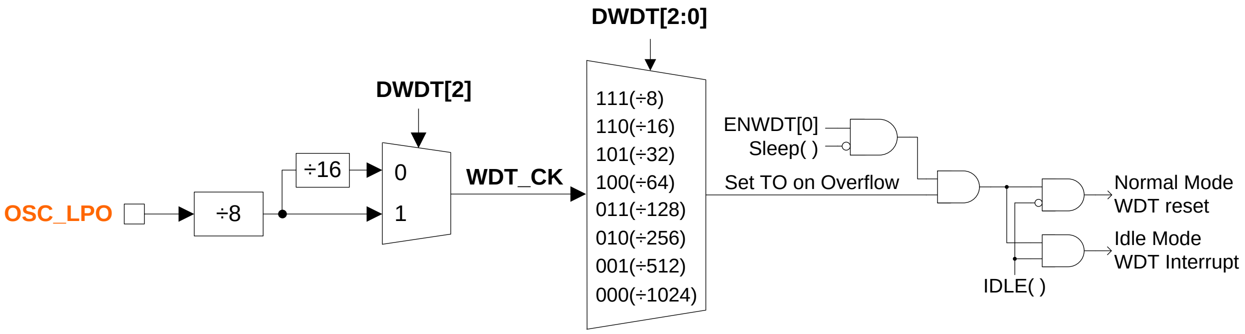


Fig 7-1 Watch Dog Block Diagram

7.1. WDT Manual

7.1.1. WDT Initialization Setting

WDT timing controller DWDT[2:0] can determine the working frequency WDT_CK of WDT counter and overflow. After overflow of timer, it can produce WDT reset signal TO or interrupt event WDTIF[5].

7.1.2. WDT Interrupt Event Service

WDT interrupt event can only be operated at standby mode of the chip. When WDTIE[0] and GIE[0] are set as <1>, after overflow of WDT counter, it may produce interrupt event. It may

[5] WDT uses internal clock source LPO. Therefore, it can be operated at Normal Mode and Idle Mode of the chip. Under normal mode, it can use software to clear the timer and make it not reset the chip as the ending of counting. However, under idle mode, it cannot clear WDT timer via any method.

set WDTIF[0] as<1>. Furthermore, PC jumps to interrupt vector position <0>x0004h. On the contrary, when WDTIE[0] and GIE[0] are set as <0>, it will not produce any interrupt.

7.1.3. WDT Starting

WDT must be started when the chip is under operation mode, i.e. set WDT start controller ENWDT[0] as <1> to start WDT. After starting, it cannot use software to set ENWDT[0] as <0>. But when WDT is under standby mode, if awakened interrupt event is produced when WDT counting is ended, the hardware will set ENWDT[0] as<0> automatically.

After DWDT[2:0] setting, when WDT reset or interrupt occurs, DWDT will be cleared to 000b and it needs software for resetting.

7.2. Register Instruction-WDT

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1											
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition											
Address	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**ARST**	**IRST**
023h	INTE0	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
026h	INTF0	-			WDTIF					.000 0000	.uuu uuuu
02Ch	PSTATUS	POR	PD	TO	IDL	RST	SKERR	-	-	$000 $00.	uu$u u$u.
037h	WDTCN					ENWDT	DWDT[2:0]			0000 0000	uuuu $000

Table 7-1 WDT Control Register

INTE0/INTF0：See Chapter Interrupt

PSTATUS：See Chapter RESET

TMACN: Timing / Timer A Control Register

ENWDT: WDT Start Controller

1: Start

0: Close ;(unable set software<0>)

DWDT[2]；Watch Dog WDT_CK Working Frequency Selector

<1>LPO ÷8

<0>LPO ÷128, (LPO ÷ 8 ÷ 16)

DWDT[2:0]: Counting Overflow Frequency

<111>WDT_CK ÷ 8, (LPO ÷ 8 ÷ 8)

<110>WDT_CK ÷ 16, (LPO ÷ 8 ÷ 16)

<101>WDT_CK ÷ 32, (LPO ÷ 8 ÷ 32)

<100>WDT_CK ÷ 64, (LPO ÷ 8 ÷ 64)

<011>WDT_CK ÷ 128, (LPO ÷ 8 ÷ 16 ÷ 128)

<010>WDT_CK ÷ 256, (LPO ÷ 8 ÷ 16 ÷ 256)

<001>WDT_CK ÷ 512, (LPO ÷ 8 ÷ 16 ÷ 512)

<000>WDT_CK ÷ 1024, (LPO ÷ 8 ÷ 16 ÷ 1024)

8. Timer-A

Timer A is 8-bit design architecture. TMA can be operated under operation mode and standby mode.

Increasing Type Timer

4-segment overflow value selection

Overflow produced interrupt event

Readable Timer Value

Abstract of TMA Register:

TMACN	ENTMA[0],TMACK[0],TMAS[1:0]
TMAR	TMAR[7:0]
INTE0	TMAIE[0]
INTF0	TMAIF[0]

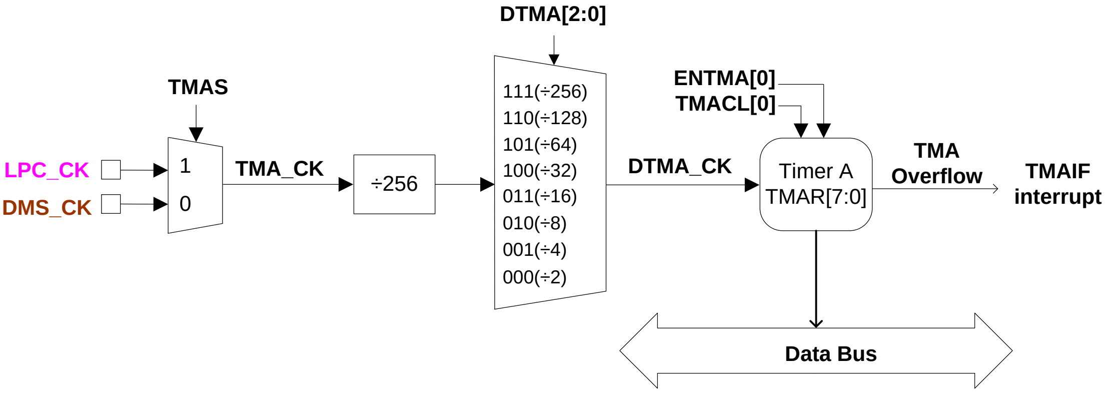


Fig 8-1 Timer A Block Diagram

Operation Instructions:

- Set TMAS[0] selected TMA_CK frequency. Reduce the frequency via frequency pre-eliminator 256. Then input DTMA frequency eliminator.
- Set ENTMA[0] as <1> to start TMA; on the contrary, set as<0> to close and clear TMAR[7:0].
- When DTMA[2:0] timing condition is established, it may produce interrupt event and make TMAR[7:0] progressively increase 1.
- TMA interrupt event TMAIF[0] must have interrupt service when TMAIE[0] is set as <1> and GIE[0] is set as <1>.
- Reading TMAR[7:0] will not make TMA timer zeroed.
- After the user sets TMACL[0] as <1> and clear all timers of TMA, TMACL[0]will be set as <0> automatically.
- TMAR[7:0] can read TMA progressively increased value of the timer, and can write motion to clear the timing value of TMAR[7:0].

8.1. Register Instruction-TMA

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
023h	INTE0	GIE		-			TMAIE	-		0000 0000	0uuu uuuu
026h	INTF0	-		-			TMAIF	-		.000 0000	.uuu uuuu
02Ch	PSTATUS			TO				-	-	$000 $00.	uu$u u$u.
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu
035h	OSCCN1	LCPS[1:0]								0000 0000	uuuu uuu.
036h	OSCCN2	ENRTC	-	XTS[1:0]		HAOM[1:0]		ENHAO	LPO	.000 0011	.uuu uu11
038h	TMACN	ENTMA	TMACL	TMAS	DTMA[2:0]			-	-	0000 00..	u0uu uu..
039h	TMAR	TMA counter Register								0000 0000	uuuu uuuu

Table 8-1 TMA Control Register

INTE0/INTF0: See Chapter *INTERRUPT*

TMACN: Timer A Control Register

ENTMA: Timer-A Start Controller

1: Start

0: Close; Zeroing of Timer

TMACL[0]: TMA Zeroing of Timer

<1>TMA Counting.

<0>TMA Zeroing of Timer.

TMAS[0] : TMA Working Frequency Selector

<1>LPC_CK

<0>DMS_CK

DTMA[2:0] Start and Close Controller

<111>TMA_CK ÷ 256

<110>TMA_CK ÷ 128

<101>TMA_CK ÷ 64

<100>TMA_CK ÷ 32

<011>TMA_CK ÷ 16

<010>TMA_CK ÷ 8

<001>TMA_CK ÷ 4

<000>TMA_CK ÷ 2

TMAR: Increment Counter of TMA can be read but not be written.

9. 16-bit Timer B (TMB)

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

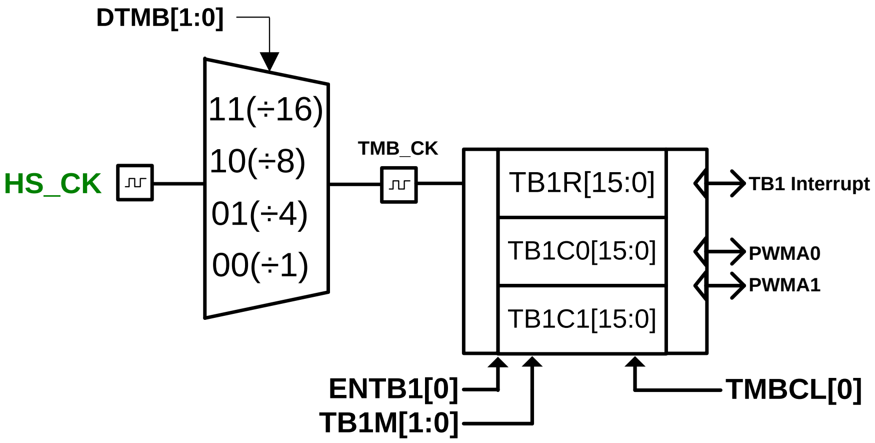


Fig 9-1 Timer B Block Diagram

- Timer Registers of TMB are
  - Increment / decrement timer TB1R [15:0] (this is a hardware counter and cannot be read or written)
  - Overflow event condition controller TB1C0[15:0]
  - PWMA condition controller TB1C1[15:0]
  - Enable controller ENTMB[0]
  - Mode controller TB1M[1:0]
  - Zeroing controller TB1CL[0]
  - Operation frequency pre-eliminator DTMB[1:0]
- 4 types of counting modes of TMB
  - 16-bit counting
  - 16 bit pulse generator mode
  - Dual 8-bit PWM mode
  - 8+8bit PWM mode
- System power consumption operation of TMB

  Operation Mode

  Idle Mode

- TB1R[15:0] Zeroing and Re-counting Condition
  - Read TMB related register, and it will not make TB1R[15:0] zero and re-count.
  - Writing TB1C0[15:0] and TB1C1[15:0] will not make TB1R[15:0] zero and re-count.
  - Writing TB1CN0 control register will not make TB1R[15:0] zero and re-count.
  - When TB1R[15:0] progressive counting is up to more than TB1C0[15:0], it will make TB1R[15:0] zero and re-count.

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

9.1. 4 Types Counting Modes of TMB

9.1.1. 16-bit counting

Set the counting mode selector TB1M[1:0] as <00> to make TMB operate under 16-bit counting mode. Under this mode, it has the following features:

◆ When TB1R[15:0] progressive counting is equal to TB1C0[15:0], it generator overflow event TB1IF[0] and zero and re-count TB1R[15:0].

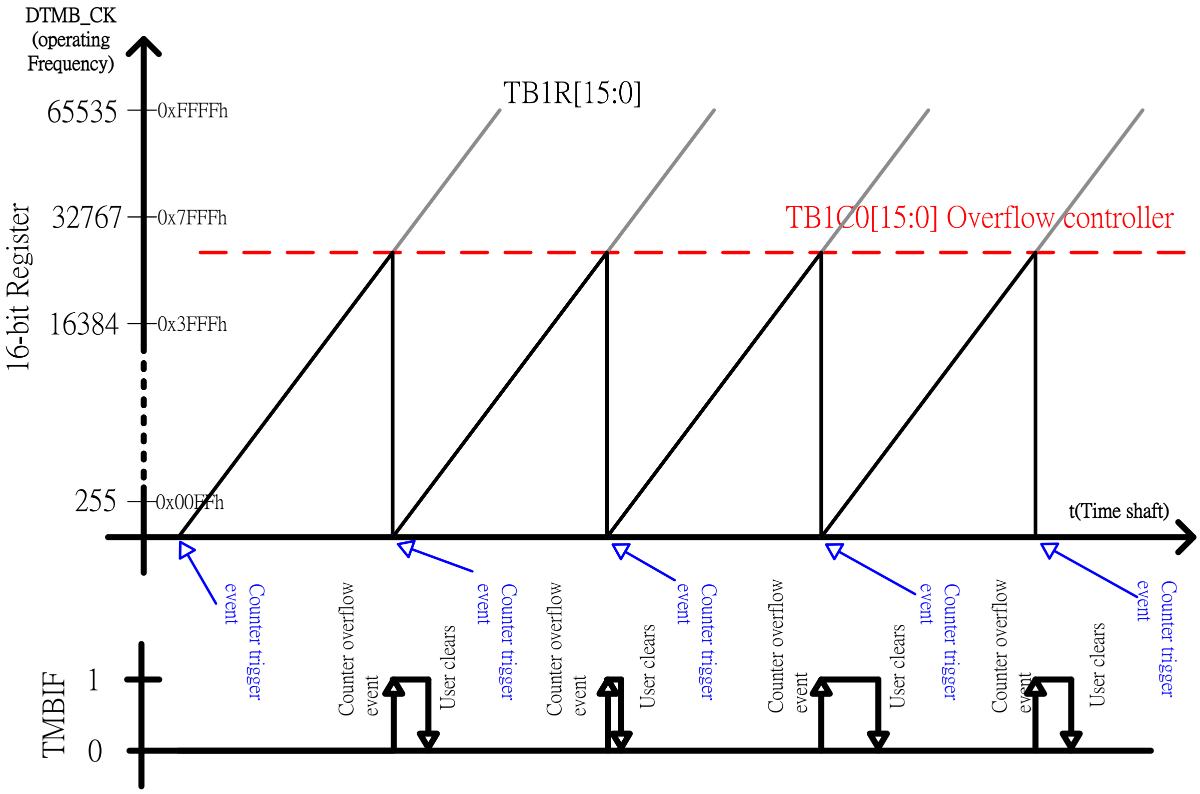


Figure 9-2 16-bit Timer Waveform and Using Schematic Diagram

◆ Operation Instructions of 16-bit Counting Mode

■ Initial Configuration

- TMB source operating frequency is fixed at HAO Set DTMB[1:0] to determine TMB operation frequency.
- TB1M[1:0 is set as<00>. Plan TMB1 as 16-bit timer.
- Write data to TB1C0[15:0]. (TB1C0H*256+TB1C0L)

■ Trigger counting signal as Always Enable status, i.e. cycle count.

■ Set ENTMB[0] as <1> to start timer.

- When TB1R[15:0] counting value is equal to TB1C0[15:0], it will produce overflow event and make TB1IF[0] set as <1>, and it is zeroed and re-taken increment counting. At the moment, it will produce interrupt event service when GIE[0] 、TB1IE[0] is set as <1>.

        - During counting process, the user can set counting zeroing controller TB1CL[0] as <1> to recount, and TB1CL[0] will be set as <0> automatically.
    - Set ENTMB[0] as <0> to close the timer.
- Operation Instructions of 16-bit PWM Mode
    - Initialization Configuration (PWM Frequency and Duty Cycle Setting)
        - TMB source operating frequency is fixed at HAO Set DTMB[1:0] to determine TMB operation frequency.
        - TB1M[1:0] is set as <00>. Plan TMB1 as 16-bit timer.
        - Trigger counting signal as Always Enable, i.e. cycle count.
        - Write data to TB1C0[15:0] (TB1C0H*256+TB1C0L)to determine the frequency of PWM.
        - Write data to TB1C1[15:0] (TB1C1H*256+TB1C1L) to determine the Duty Cycle of PWM.
        - Set ENTMB[0] as <1> to start timer.
    - Generator PWM0 Waveform
        - When TB1R[15:0] counting value is equal to TB1C1[15:0], it makes the status of PWM0 be 0 ➔1.
        - When TB1R[15:0] recounting value is equal to TB1C0[15:0], it makes the status of PWM0 be 1➔0. Furthermore, it generator overflow event and makes TMBIF[0] set as <1> and zero and retake increment counting. At the moment, it will produce interrupt event service when GIE[0] 、TMBIE[0] is set as <1>.
    - PWM Output Control
        - Set the pin output PWM waveform status as output status, and set ENPWM0[0] as <1>to start output as PWM function. Confirm whether the pin related setting is correct.
    - When ENTMB[0] is set as <0>, it will close timer and PWM output.
    - Calculation equation of PWM1 frequency Duty Cycle is:

$$\text{PWM0 Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0}[15:0]+1}$$

$$\text{PWM0 Duty Cycle} = \frac{(\text{TB1C0}[15:0]+1)-\text{TB1C1}[15:0]}{\text{TB1C0}[15:0]+1}$$

9.1.2. 16 bit pulse generator mode

The counting mode selector TB1M[1: 0] settings <01> so TMB operates in 16-bit mode pulse generator, which generates a pulse wave number is (TB1C1H * 256 + TB1C1L).

After the pulse generator, TMB will automatically shut down. To the end of the pulse generator issues an interrupt, set (TB1C0H * 256 + TB1C0L) settings (TB1C1H * 256 + TB1C1L) the same.

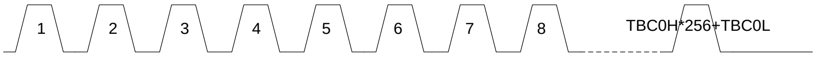


Figure 9-316 bit pulse generator Waveform and Using Schematic Diagram

9.1.3. Dual 8-bit PWM mode

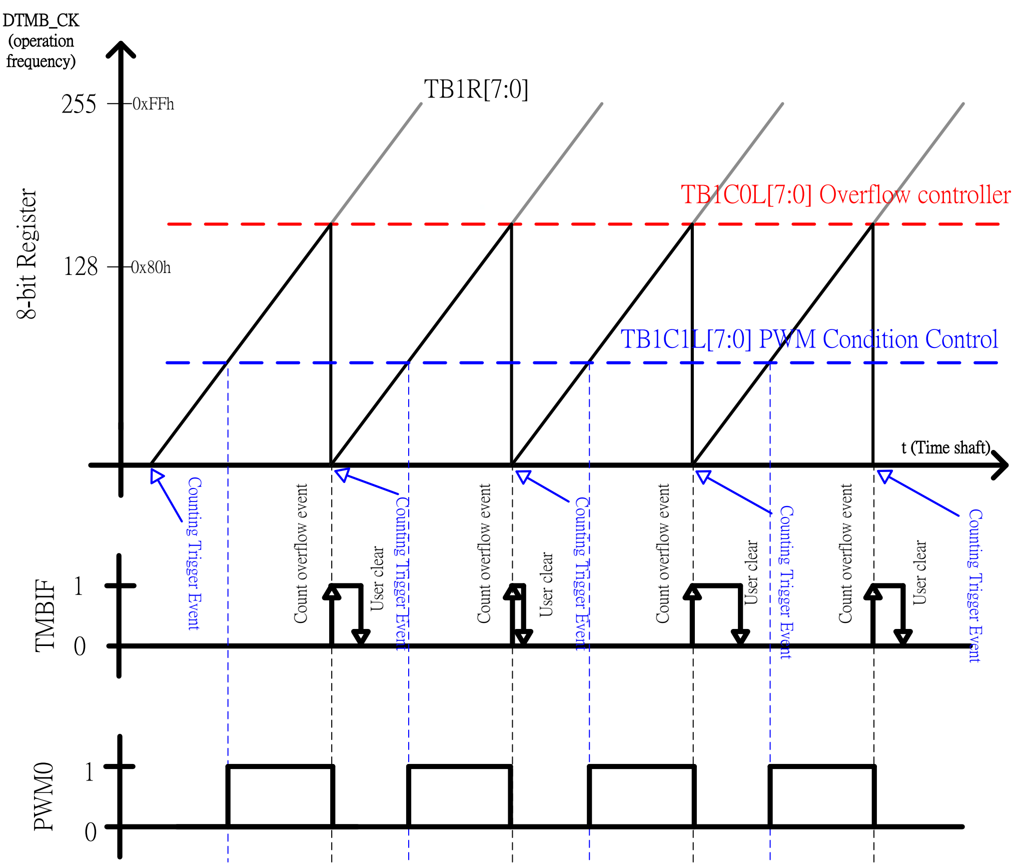


Figure 9-4 PWM1 Waveform and Using Schematic Diagram

- PWM0 Output Operation Instructions
  - Initialization Configuration (PWM Frequency and Duty Cycle Setting)
    - TMB source operating frequency is fixed at HAO Set DTMB[1:0] to determine TMB operation frequency.
    - TB1M[1:0] is set as <10>. Plan TMB1 as Dual 8-bit count.

- Trigger counting signal as Always Enable, i.e. cycle count.
- Write data to TB1C0L[7:0] to determine the frequency of PWM.
- Write data to TB1C1L[7:0] to determine the Duty Cycle of PWM.
- Set ENTMB [0] as <1> to start timer.

■ Generator PWM0 Waveform

- When TB1R[7:0] counting value is equal to TB1C1L[7:0], it makes the status of PWM0 be 0->1.
- When TB1R[7:0] recounting value is equal to TB1C0L[7:0], it makes the status of PWM0 be 1->0. Furthermore, it generator overflow event and makes TMBIF[0] set as <1> and zero and retake increment counting. At the moment, it will produce interrupt event service when GIE[0]、TMBIE[0] is set as <1>.

■ PWM Output Control

- Set the pin output PWM waveform status as output status, and set ENPWM0[0] as <1>to start output as PWM function. Confirm whether the pin related setting is correct.

■ When ENTMB[0] is set as <0>, it will close timer and PWM output.

■ Calculation equation of PWM0 frequency Duty Cycle is:

$$\text{PWM0 Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0L}[7:0]+1}$$

$$\text{PWM0 Duty Cycle} = \frac{(\text{TB1C0L}[7:0]+1)-\text{TB1C1L}[7:0]}{\text{TB1C0L}[7:0]+1}$$

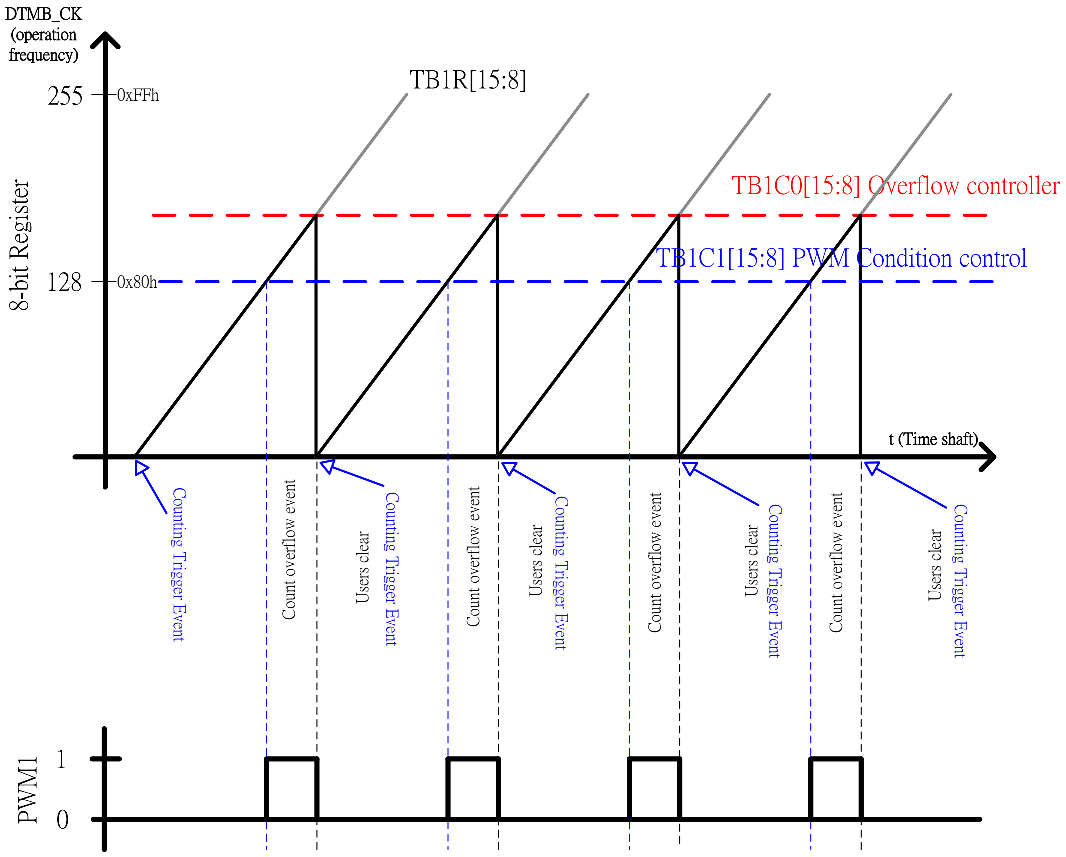


Figure 9-5 PWM1 Waveform and Using Schematic Diagram

- PWM1 Output Operation Instructions
  - Initialization Configuration (PWM Frequency and Duty Cycle Setting)
    - TMB source operating frequency is fixed at HAO Set DTMB[1:0] to determine TMB operation frequency.
    - TB1M[1:0] is set as <10>. Plan TMB1 as Dual 8-bit count.
    - Trigger counting signal as Always Enable, i.e. cycle count.
    - Write data to TB1C0H[7:0] to determine the frequency of PWM.
    - Write data to TB1C1H[7:0] to determine the Duty Cycle of PWM.
    - Set ENTMB[0] as <1> to start timer.
  - Generator PWM1 Waveform
    - When TB1R[15:8] counting value is equal to TB1C1H[15:8], it makes the status of PWM0 be 0➔1.
    - When TB1R[15:8] recounting value is equal to TB1C0H[15:8], it makes the status of PWM0 be 1➔0. Furthermore, it generator overflow event and makes TMBIF[0] set as <1> and zero and retake increment counting. At the moment, it will produce interrupt event service when GIE[0] 、TMBIE[0] is set as <1>.
  - PWM Output Control
    - Set the pin output PWM waveform status as output status, and set ENPWM1[0] as <1>to

start output as PWM function. Confirm whether the pin related setting is correct.

- When ENTMB[0] is set as <0>, it will close timer and PWM output.
- Calculation equation of PWM0 frequency Duty Cycle is:

$$\text{PWM1 Frequency} = \frac{\text{DTMB\_CK}}{\text{TB1C0H}[15:8]+1}$$

$$\text{PWM1 Duty Cycle} = \frac{(\text{TB1C0H}[15:8]+1) - \text{TB1C1H}[15:8]}{\text{TB1C0H}[15:8]+1}$$

9.1.4. 8+8-bit PWM

When TMB is set as 8+8-bit mode and PWM output waveform is selected as PWM5, it can produce 8+8bit PWM output.

8+8-bit PWM is composed of TB1R[7:0], TB1C0L[15:8], TB1C1L[7:0] and TB1C1H[15:8] control registers and internal digital circuit. TB1C0[7:0] is PWM frequency controller, TB1C1L[7:0] is PWM duty cycle controller, and TB1C1H[15:8] is 8+8-bit PWM duty cycle trimmer.

Setting and instruction of 8+8-bit PWM duty cycle spinne TB1C1H[15:8] are as in the following table.

Setting / Weighted Quantity	TB1C1H[15:8]							
	80h	40h	20h	10h	08h	04h	02h	01h
Fine tuning of PWM duty cycle	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
Description	2 times of TMB overflow，one half N+1and one half N	4 times of TMB overflow，one half N+1and one half N	8 times of TMB overflow，one half N+1and one half N	16 times of TMB overflow，one half N+1and one half N	32 times of TMB overflow，one half N+1and one half N	64 times of TMB overflow，one half N+1and one half N	128 times of TMB overflow，one half N+1and one half N	256 times of TMB overflow，one half N+1and one half N

Table 9-1 Duty Cycle Trimmer Setting Table

- Description of duty cycle trimmer TB1C1H[15:8], in which N is the width of duty cycle, (*Note: N = TB1C1[7:0]*）
  - Basic Type

- TB1C1H[15:8] is set as 80h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 2 output periods as a group--- one is N+1 and the other is N.
- TB1C1H[15:8] is set as 40h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 4 output periods as a group--- 2 continuous output is N+1 and the other 2 is N.
- TB1C1H[15:8] is set as 20h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 8 output periods as a group--- 4 continuous output is N+1 and the other 4 is N.
- TB1C1H[15:8] is set as 10h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 16 output periods as a group--- 8 continuous output is N+1 and the other 8 is N.
- TB1C1H[15:8] is set as 08h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 32 output periods as a group---16 continuous output is N+1 and the other 16 is N.
- TB1C1H[15:8] is set as 04h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it produces waveform taking 64 output periods as a group--- 32 continuous output is N+1 and the other 32 is N.
- TB1C1H[15:8] is set as 02h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 128 output periods as a group--- 64 continuous output is N+1 and the other 64 is N.
- TB1C1H[15:8] is set as 01h, and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 256 output periods as a group--- 128 continuous output is N+1 and the other 128 is N.

■ Logic Computation OR Superimposition Type

(only takes1/2+1/4,1/2+1/8,~,1/2+1/4+1/8+1/16+1/32+1/64+1/128,1/2+1/4+1/8+1/16+1/32+ 1/64+1/256 as description and demonstration)

- TB1C1H[15:8] is set as C0h(1/2+1/4), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 4 output periods as a group--- 1 output is N+1 and the other 3 is N.
- TB1C1H[15:8] is set as A0h(1/2+1/8), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 8 output periods as a group--- 2 output is N+1 and the other 6 is N.
- TB1C1H[15:8] is set as 90h(1/2+1/16), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 16 output periods as a group--- 4 output is N+1 and the other 12 is N.

- TB1C1H[15:8] is set as 88h(1/2+1/32), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 32 output periods as a group--- 8 output is N+1 and the other 24 is N.
- TB1C1H[15:8] is set as 84h(1/2+1/64), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 64 output periods as a group--- 16 output is N+1 and the other 40 is N.
- TB1C1H[15:8] is set as 90h(1/2+1/128), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 128 output periods as a group--- 32 output is N+1 and the other 96 is N.
- TB1C1H[15:8] is set as 90h(1/2+1/256), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 256 output periods as a group---64 output is N+1 and the other 192 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 8 output periods as a group--- 1 output is N+1 and the other 7 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 8 output periods as a group--- 1 output is N+1 and the other 7 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8+1/16), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 16 output periods as a group--- 1 output is N+1 and the other 15 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8+1/16+1/32), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 32 output periods as a group--- 1 output is N+1 and the other 31 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8+1/16+1/32+1/64), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 64 output periods as a group--- 1 output is N+1 and the other 63 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8+1/16+1/32+1/64+1/128), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 128 output periods as a group--- 1 output is N+1 and the other 127 is N.
- TB1C1H[15:8] is set as C0h(1/2+1/4+1/8+1/16+1/32+1/64+1/128+1/256), and it makes waveform of PWM duty cycle occur N+1 and N output, i.e. it generator waveform taking 256 output periods as a group--- 1 output is N+1 and the other 255 is N.

◆ Following Table 9 -2 and Figure 9-6 sections list 8+8-bit PWM waveform changing of TB1C1H[15:8] under different setting for user's reference.

8+8bit PWM			TBN Overflow Times																		
Type	TBC2 [7:0]	Logic computation	0	1	2	3	4	5	6	7	8	9	10	~	127	128	~	252	253	254	255
Basic Waveform	80h	1/2	N	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N+1	N	~	N	N+1	N	N+1
	40h	1/4	N	N	N+1	N	N	N	N+1	N	N	N	N+1	~	N	N	~	N	N	N+1	N
	20h	1/8	N	N	N	N	N+1	N	N	N	N	N	N	~	N	N	~	N+1	N	N	N
	10h	1/16	N	N	N	N	N	N	N	N	N+1	N	N	~	N	N	~	N	N	N	N
	08h	1/32	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	04h	1/64	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	02h	1/128	N	N	N	N	N	N	N	N	N	N	N	~	N	N	~	N	N	N	N
	01h	1/256	N	N	N	N	N	N	N	N	N	N	N	~	N	N+1	~	N	N	N	N
Logic Computation Superimposition Type	C0h	3/4	N	N+1	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N+1	~	N+1	N	~	N	N+1	N+1	N+1
	A0h	5/8	N	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	N	~	N+1	N	~	N+1	N+1	N	N+1
	E0h	7/8	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	N+1	N	~	N+1	N	~	N+1	N+1	N+1	N+1
	F0h	15/16	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	~	N+1	N	~	N+1	N+1	N+1	N+1
	A1h	161/256	N	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	N	~	N+1	N+1	~	N+1	N+1	N	N+1
	F1h	241/256	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	~	N+1	N+1	~	N+1	N+1	N+1	N+1
	FFh	255/256	N	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	~	N+1	N+1	N+1	N+1

Table 9 -2 8+8-bit PWM Output Waveform and Using Schematic Table

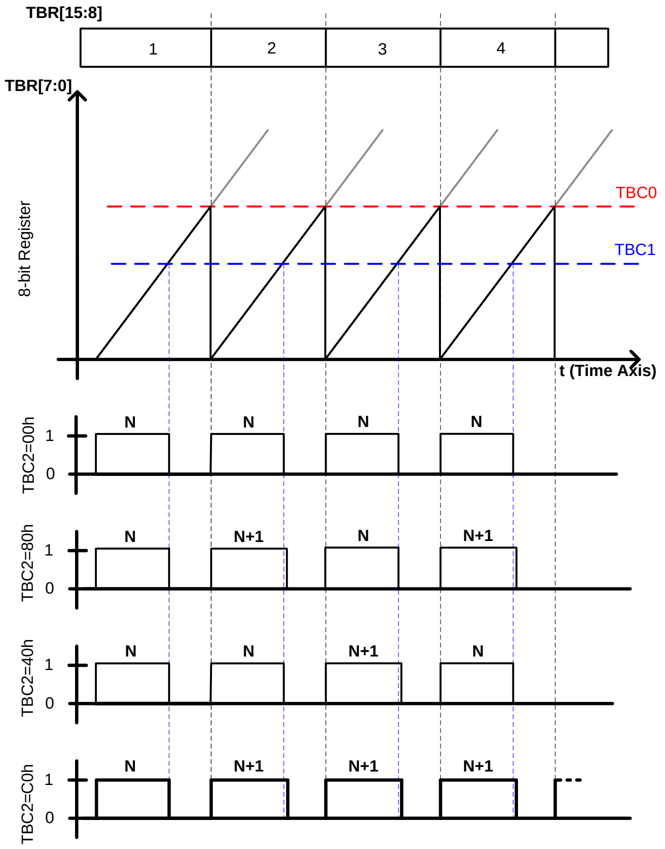


Figure 9-6 8+8-bit PWM Output Waveform Schematic Diagram

- 8+8-bit PWM Output Operation Instructions
  - Initialization Configuration (PWM Frequency and Duty Cycle Setting)
    - TMB source operating frequency is fixed at HAO Set DTMB[1:0] to determine TMB operation frequency.
    - TB1M[1:0] is set as <11>. Plan TMB1 as 8+8-bit count.
    - Trigger counting signal as Always Enable, i.e. cycle count.
    - Write data to TB1C0L[7:0] to determine the frequency of PWM.
    - Write data to TB1C1H[15:8] to determine the Duty Cycle of PWM.
    - Set ENTMB[0] as <1> to start timer.
  - Generator 8+8-BIT PWM Waveform
    - When TB1R[7:0] counting value is equal to TB1C0L[7:0], it makes the status of 8+8-Bit PWM be 0➔1.
    - When TB1R[7:0] recounting value is equal to TB1C1L[7:0], it makes the status of 8+8-Bit PWM be 1➔0.

✓ Furthermore, it produces overflow event and makes TB1IF[0] set as <1> and zero and retake increment counting. At the moment, it will produce interrupt event service when GIE[0]、TB1IE[0] is set as <1>.

✓ At the moment, all TB1C1H[7:0] set data is adjust 8+8-Bit PWM output as N+1 and N. As in Table 8-1, N＝TB1C1L[7:0].

■ PWM Output Control

● Set the pin output PWM waveform status as output status, and set ENPWM0[0] as <1>to start output as PWM function. Confirm whether the pin related setting is correct.

■ When ENTMB[0] is set as <0>, it will close timer and PWM output.

■ Calculation equation of 8+8-Bit PWM frequency Duty Cycle is:

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

$$\text{PWM Duty Cycle} = \frac{(\text{TB1C1L}[7:0]+1)+\dfrac{\text{TB1C1H}[15:8]}{256}}{\text{TB1C0L}[7:0]+1}$$

9.2. TMB1 Control Register List and Instructions

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
Address	**File Name**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**	**A-RESET**	**R/W**
58H	TB1CN0	ENTMB	TB1M[1:0]		DTMB[1:0]		-	-	TMBCL	0000 0000	*,*,*,* *,*,*,*
59H	TB1C0L	TimerB1 counter Condition Register0 [7:0]								xxxx xxxx	*,*,*,* *,*,*,*
5AH	TB1C0H	TimerB1 counter Condition Register0 [15:8]								xxxx xxxx	*,*,*,* *,*,*,*
5BH	TB1C1L	TimerB1 counter Condition Register1 [7:0]								xxxx xxxx	*,*,*,* *,*,*,*
5CH	TB1C1H	TimerB1 counter Condition Register1 [15:8]								xxxx xxxx	*,*,*,* *,*,*,*

Table 9-3 TMB1/2/3 Related Register

INTE0/INTF0 : Please refer to *Interrupt* Chapter

TB1CN0 : Timer-B Control Register

Bit	Name	Description
Bit7	ENTMB	Timer-B enable controller <1> Enable count <0> Disable Count ; Zero counters
Bit6~5	TB1M[1:0]	TMB Operation Mode Selector <00>16bit counter mode. TMB_CLK/(TBC0H*256+TBC0L) Periodic interruption occurs <01> 16bit pulse generator mode. Pulse generation quantity (TB1C1H*256+TB1C1L) <10> Dual 8-bit PWM mode. PWMO0 Duty isTB1C1L/TB1C0L PWMO1 Duty is TB1C1H/TB1C0H <11> 8+8bit PWM mode. Output Duty is TB1C1L/TB1C0L+TB1C1H/256
Bit4~3	DTMB	Timer-B operating frequency prescaler <00>TMB clock=HS(default) <01>TMB clock=HS/4 <10>TMB clock=HS/8 <11>TMB clock=HS/16
Bit0	TMBCL	TMB clear of counter <1> TMB clear of counter, When writing “1” clear TMBR and Pre-counter, automatically returns to “0” <0> TMB Counting

TB1C0H : TMB1 Count condition register TB1C0[15:8]

TB1C0L : TMB1 Count condition register TB1C0 [7:0]

TB1C1H : TMB1 Count condition register TB1C1[15:8]

TB1C1L : TMB1 Count condition register TB1C1 [7:0]

10. Power System

Power System (PWR) has a linear regulator power VDDA and analog circuit ground power ACM. The provided chip analog peripheral circuit is used appropriately to drive external circuit.

VDDA Linear Regulator Power

- 3-segment voltage adjustment design
- 4 types of operation modes
- External bias voltage design
- Low temperature drift coefficient

ACM Internal Analog Circuit Ground Power

- Output Voltage 1.2V
- Low temperature drift coefficient

Abstract of PWR Register:

PWRCN ENLDO[1:0],VDDAX[1:0]

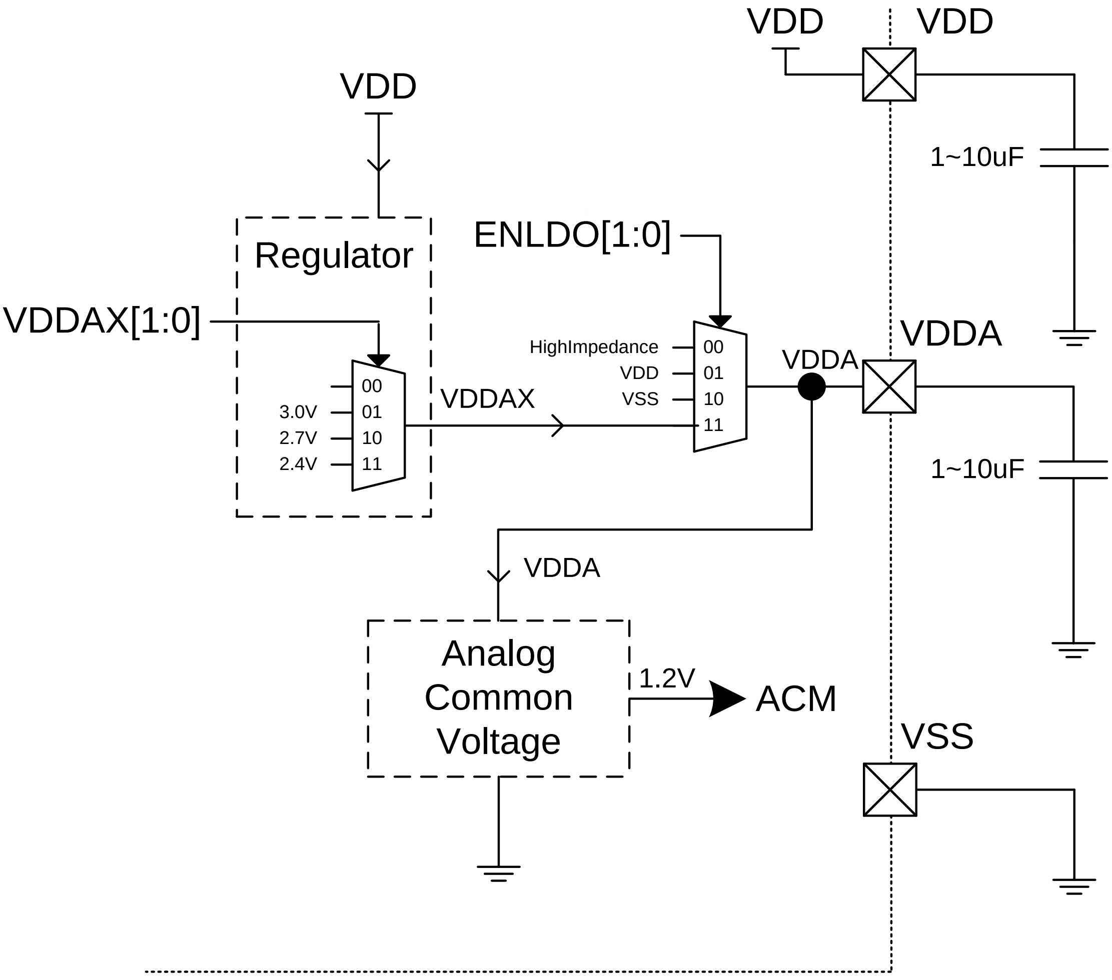


Fig 10-1 Power System Block Diagram

10.1. VDDA Manual

10.1.1. VDDA Initialization Setting:

Voltage stabilization selector VDDAX[1:0] can set VDDA pin output voltage as 3.0V, 2.7V and 2.4V. As VDDA is linear voltage stabilization power, it must mote whether the VDD working voltage value is lower than the set value of VDDA output voltage when using, so as to avoid causing unexpected circuit malfunction.

10.1.2. VDDA Used External Bias:

VDDA can adopt external input voltage design. When the user provided power source by oneself, it must be input by VDDA pin impressed voltage method. When this method is adopted, it must close VDDA, i.e. ENLDO[1:0] is set as 00. It must note this method may affect the efficiency of analog circuit. Therefore, it shall be careful.

10.1.3. VDDA Starting

When ENLDO[1:0] is set as <11>, it will start VDDA regulator. When starting VDDA regulator, it shall avoid SD18 in starting status. Furthermore, it can only start SD18 when VDDA voltage is stable. When it is external connection with 1uF(10uF) voltage stabilization capacitor, it needs 500uS(5mS) stabilization time.

10.2. ACM Manual

10.2.1. ACM Initialization Setting:

When internal analog circuit ground power ACM is used, it must start VDDA first. ACM internal produced output voltage is fixed as 1.2V.

10.3. Register Instruction-PWR

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1											
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition											
Address	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**ARST**	**IRST**
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]			AD1RST	AD2RST	CSFON	0000 0000	uuuu u00u

Table 10-4 PWR Register

PWRCN: Power System Control Register

ENLDO [1:0]: Internal linear regulator start/ close controller

<11> Output VDDA voltage

<10>Close LDO, and output is pulled to VSS.

<01> Output VDD voltage

<00> Close LDO, and VDDA pin has high input impedance mode.

VDDAX[1:0]: VDDA Voltage Stabilization Selector

11: 2.4V

10: 2.7V

01: 3.0V

00: Not Used

11. Analog Digital Converter SD18,ΣΔADC

SD18 is Over Sampling Sigma Delta Analog-to-Digital Converter with high resolution. It has 24-bit output, and includes 4 parts, i.e. multi-functional input multiplexer, Input Buffer and pre- Programmable Gain Amplifier (PGA,),ΣΔAD, Sigma Delta Modulator and Comb Filter.

Multi-functional Input Multiplexer:

- Switch and select several groups of different channels. Single chip can take several kinds of measuring.
- Input channel can make inversion and short circuit, and eliminate ADC wandering of zero point.
- Built-in temperature sensing circuit output voltage

ΣΔ Sigma Delta Modulator:

- Adjust the input voltage magnification factor. Optional magnification factors are 1/4, 1/2, 1, 2, 4, 8 and 16 times
- Optional reference voltage magnification is 1 or 1/2.
- 3-bit DC input bias setting
- Adjustable sampling frequency of modulator: 31.5kHz~250kHz

Comb Filter:

- Adjustable OSR(Over Sampling Ratio)= 32~32768，ADC output speed is about 7.81kHz~8Hz(sampling frequency=250kHz)
- Produce interrupt event

Abstract of SD18 register:

ADCR[23:0]	ADCRH[7:0], ADCRM[7:0], ADCRL[7:0],
ADCCN1	ENADC[0],ENHIGN[0],ENCHP[0],ADGN[2:0]
ADCCN2	VREGN[0],DCSET[2:0]
ADCCN3	OSR[3:0]
AINET1	INH[2:0],INL[2:0],INIS[0]
AINET2	VRH[1:0],INX[1:0],VRL[1:0]

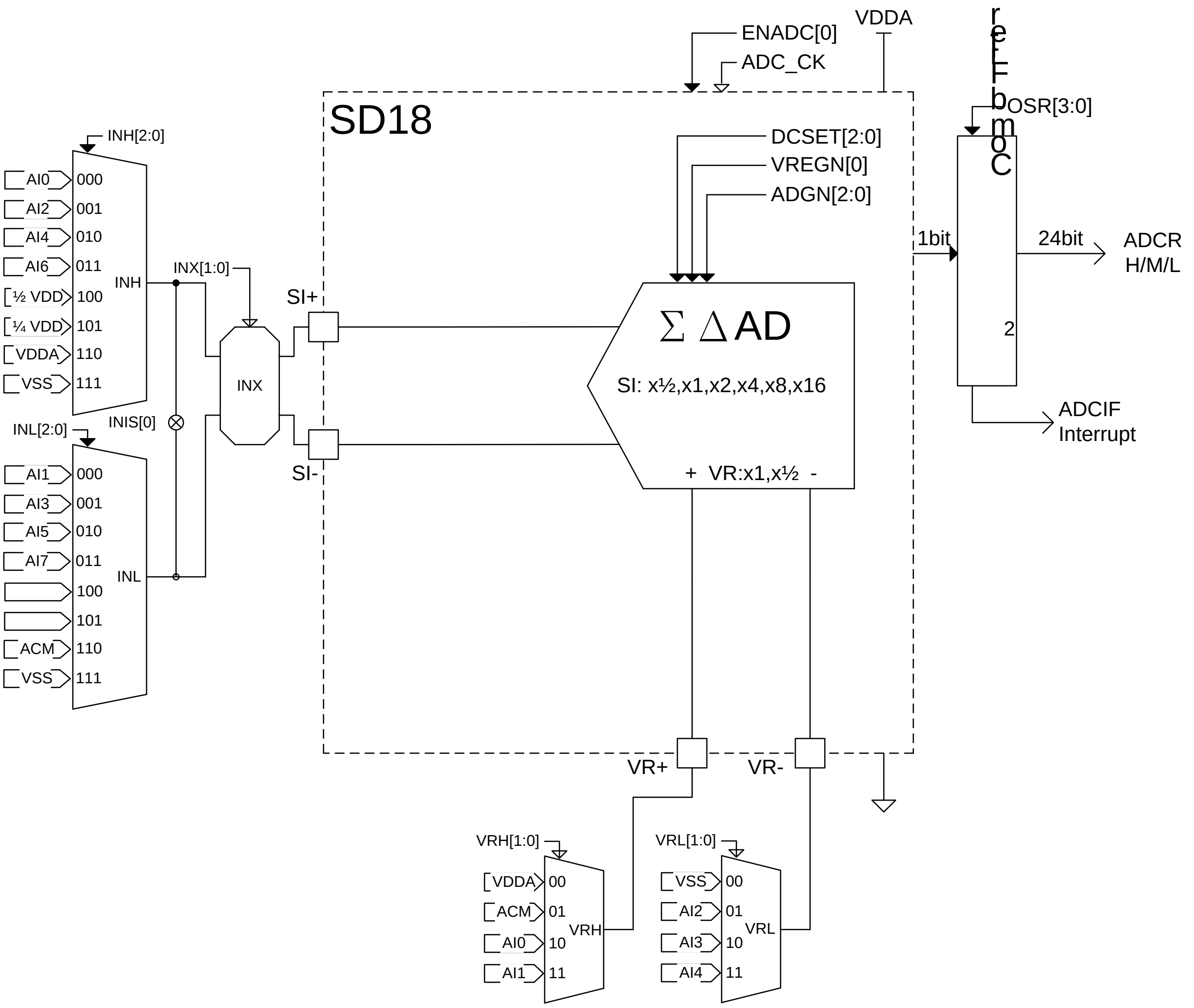


Fig 11-1 SD18 Block Diagram

11.1. SD18 Manual

11.1.1. SD18 Initialization Setting

11.1.1.1. Work Frequency Collocation Mode

SD18 sampling frequency is set by sampling frequency selector ADCCK [0], and SD18 work frequency is provided by DHS_CK. The max sampling frequency is no more than 300kHz and min sampling frequency is no less than 25kHz. Quicker sampling frequency can obtain better resolution under the same output speed, but the input impedance will also be reduced. When DHS_CK frequency surpasses max allowable value, it must take frequency adjustment via over-sampling frequency pre-eliminator ADCS [2:0], as in Table 11-1.

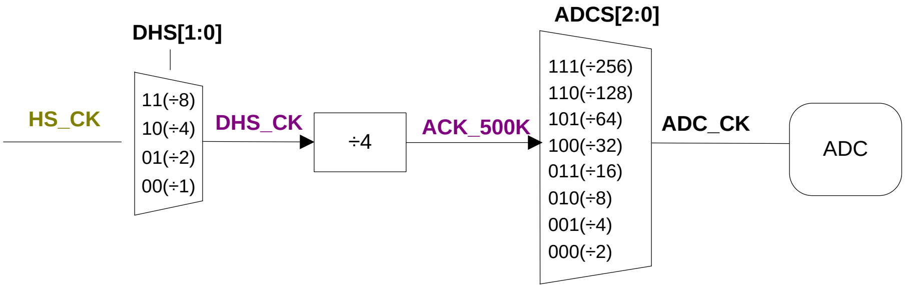


Fig 11-2 SD18 Work Frequency Block Diagram

"-": SD18 non-operational work frequency(Unit: KHz)

	Setting	ADCS[2:0]							
HS_CK	DHS_CK	000	001	010	011	100	101	110	111
8000	HS_CK/1	-	-	250	125	62.5	31.2	-	-
8000	HS_CK/2	-	250	125	62.5	31.2	-	-	-
4000	HS_CK/1	-	250	125	62.5	31.2	-	-	-
4000	HS_CK/2	250	125	62.5	31.2	-	-	-	-
2000	HS_CK/1	250	125	62.5	31.2	-	-	-	-

Table 11-1 SD18 Work Frequency Setting Table

11.1.1.2. Multi-functional Input Multiplexer Collocation Mode

Multi-functional Input Multiplexer may produce two groups of Differential Output signals, i.e. signal for testing **SI+**,**SI-** and reference voltage **VR+**、**VR-**.

- **SI±** input signal selector INH[2:0], INL[2:0] and **SI±** input signal converter INX[1:0] can send the input signals to **SI+** or **SI-** end via the following paths, such as Fig 11-3 and Table 11-2 (a)；
  - AI0~AI7pins via INH and INL channel
  - LNOP output signal OPO
  - Reference power source ACM
- **VR±** voltage signal selectors VRH[1:0] and VRL[1:0] can determine SD18 reference voltage is sent to **VR+** or **VR-** end via the following paths, as in Table 11-2 (b) .
  - AI0~AI4 pins via VRH and VRL channel
  - Reference power source ACM
  - Work power source VSS
- When **SI±** input signal short circuit INIS[0] is set as <1>, it can make INH and INL channels short circuits. On the contrary, when it is set as <0>, INH and INL channels are not short circuits.

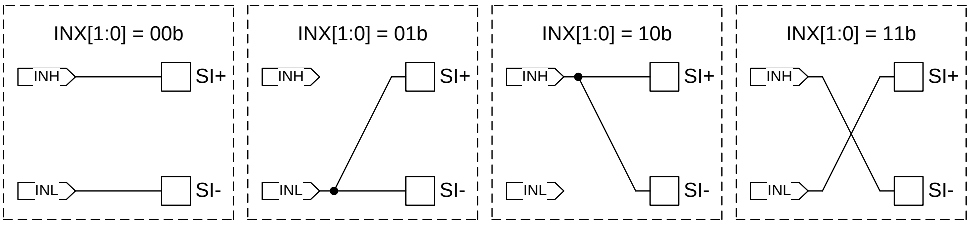


Fig 11-3 Four Types of Combinations of INX Input Signal Converter

Setting / Signal for Testing	INH[2:0],INL[2:0]							
	000	001	010	011	100	101	110	111
SI+	AI0	AI2	AI4	AI6	VDD/2	VDD/4	VDDA	VSS
SI-	AI1	AI3	AI5	AI7			ACM	VSS

Table 11-2 (a)SI± Input Selector

Setting / Input	VRH[1:0],VRL[1:0]			
	00	01	10	11
VR+	VDDA	ACM	AI0	AI1
VR-	VSS	AI2	AI3	AI4

Table 11-2 (b) VR± Input Selector

11.1.1.3. ΣΔ Modulator Configuration Mode

SD18 is adopted second order ΣΔ modulator. The signal for testing and reference voltage can be taken magnification and bias adjustment via the following settings.

- When ΔVR± magnification adjuster VREGN[0] is set as <1>, it will take 1/2 magnification adjustment for reference voltage signal, and also change the ratio of input signals ΔSI± = (SI+ － SI-) and ΔVR± = (VR+ － VR-). When it is set as <0>, it takes 1 time of adjustment.
- Via the setting of magnification adjuster ADGN[2:0], input signal can be up to max 16 times of signal magnification, as in Table 11-3 (a).
- When ENCHP[0] is set as <0>, it can make the input signal reduce frequency caused frequency noise via chopper. On the contrary, when ENCHP[0] is set as <1>, input signal will come round the chopper.
- Input signal SI± can adjust the input signal zero position to increase measurement range via DC input bias adjuster DCSET[2:0]. Bias method is adopted magnification value of weighted reference signal VR±, as in Table 11-3 (b).
- When taking signal measuring, it shall note the matching problem of external input signal impedance and ADC. See *11.2 Analog Channel Input Features*

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

Table 11-3 (a)ADGN[2:0] Magnification Collocation List

Setting / 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 11-3 (b) SI± Input Signal Weighted Reference Voltage Magnification List

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

Formula 11 -1

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

Formula 11 -2

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

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

11.1.1.4. Comb Filter Setting Mode

ΣΔ Modulator output1-bit data is sent to second order Comb Filter, then it is covered into 24-bit value via Comb Filter and stored in ADCR[23:0] register. The update speed of ADCR[23:0] data is the output speed of SD18, and the calculation mode is the ratio of SD18 sampling frequency to SD 18 output speed. SD 18 output speed frequency is also called OSR (Over Sampling Ratio).

Therefore, the SD18 output speed is ADC_CK÷OSR, and OSR value can produce different SD18 conversion frequency via OSR[2:0] settings, as in Table 11-3 (c).

Setting / ADC_CK	OSR[2:0]										
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
250K	7812	3906	1953	976	488	244	122	61	30	15	7
125K	3906	1953	976	488	244	122	61	30	15	7	3

Table 11-3 (c) Simple List of SD18 Over-sampling Rate Frequency Collocation

ADCR [23:0] is composed of ADCRH[7:0], ADCRM[7:0] and ADCRL[7:0] respectively, and is used to store Comb Filter output 24-bit data. Comb Filter data format is as in Fig 11-4.

+FSR/-FSR: Max measuring range of positive phase and negative phase

	Equivalent Signal for Test	ADCR[23:0]	
		Hex	Binary
Bipolar Output Two Complement Format	ΔVR_I	7FFFFF[*1]	0111-1111- 1111-1111-1111-1111
	$\Delta VR\_I \times \frac{1}{2^{22}}$	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 11-4 Relationship Table between ADCR[23:0] and Input Signal

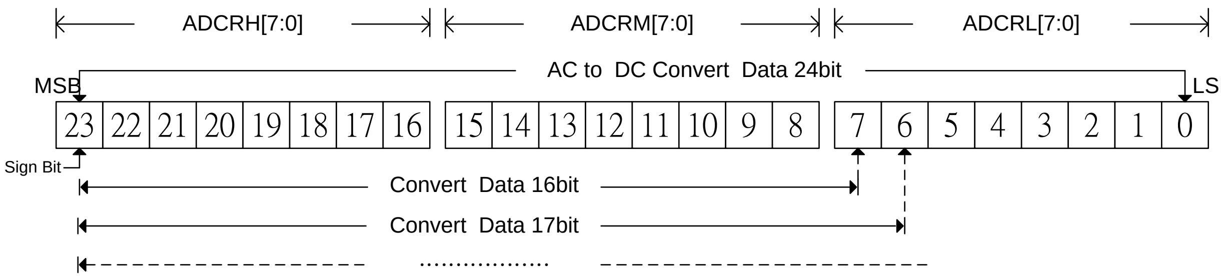


Fig 11-4 ADCR [23:0] Resolution Schematic Diagram

Note: * 1. ADCR ΔVR_I full scale is 3FFFFFh. When overflow is 7FFFFFh, not 400000h i.e. 3FFFFF +1 -> 7FFFFF

11.1.2. Interrupt Service Setting

When comb filter value is completed converting and stored in ADCR[23:0] register, it will produce interrupt signal and ADCIF[0] is set as <1>. At the moment, it shall set ADCIE[0] and GIE[0] as <1> if interrupt event service is needed.

11.1.3. SD18 Starting

It can start SD18 to take analog digital conversion when ENADC[0] is set as <1>. On the contrary, when ENADC[0] is set as <0>, SD18 will be closed. The power of SD18 is VDDA, and ACM is used as the reference point of internal common-mode voltage. Therefore, it must start VDDA and ACM before starting SD18.

The work voltage of SD18 is provided by VDDA, and the Aix input pin voltage shall be no more than VDDA voltage. When VDDA power is closed (which is not taken internal start or external input), it may cause power leakage of the network and cause large chip wear and tear and current consumption indirectly if SD18 input signal network **SI±** and reference voltage network **VR±** exist voltage. Therefore, when VDDA power is closed, SD18 input signal network or reference voltage network must be selected properly. The network switch is adjusted internal ACM or VSS to avoid external voltage causing power leakage of network.

11.2. Analog Channel Input Features

SD18 is taken analog signal handling by using switchable capacitor circuit. When input buffer is not used, to ensure getting correct value of sampling capacitor voltage, max output impedance of input signal must be restricted. Furthermore, it shall have mutual dependence relationship with SD18 sampling frequency and signal magnification selection.

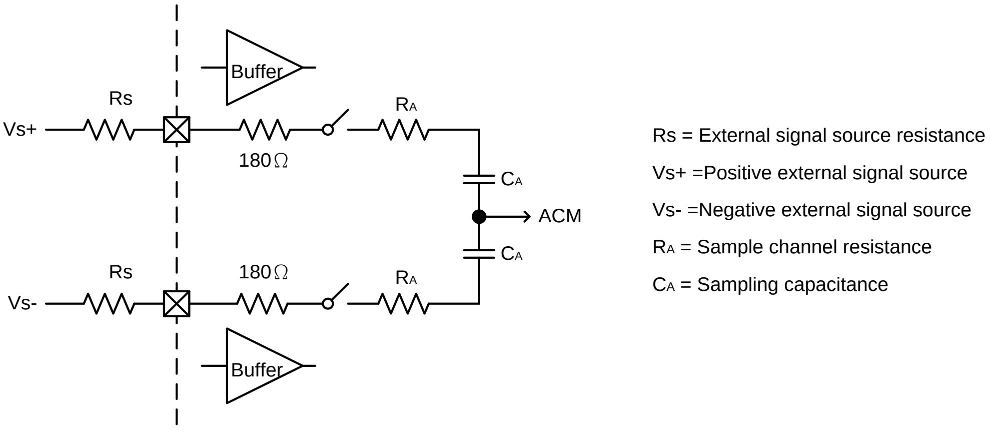

Buffer
Rs
RA
Vs+
180Ω
CA
ACM
CA
Rs
180Ω
RA
Vs-
Buffer
Rs = External signal source resistance
Vs+ =Positive external signal source
Vs- =Negative external signal source
RA = Sample channel resistance
CA = Sampling capacitance

Fig 11-5 Aix Input Capacitor and Impedance Module

From Fig 11-5, when input signal is input directly without via buffer, it must consider the effects of input signal internal resistance Rs and sampling frequency ADC_CK & parasitic resistance R_A and capacitor C_A of SD18. Related formulas are as the following:

Formula 11 -3

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

t_S: SD18 shortest sampling time

ENOB: Expected to obtain SD18 Effective Number of Bits

Gain: (PGA Gain)×(ΣΔAD Gain)

Formula 11 -4

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

Fs: SD18 shortest sampling time

As the composition of SD18 includes PGA and ΣΔAD and the two parts have respective R_A and C_A value on design, the calculation of shortest sampling time t_S is considered and measured according to the directly matched part of input signal .

If PGA is used and amplification factor of ΣΔAD is set as 4 directly, R_A=10KΩ and C_A=2pF when t_s is calculated.

If PGA is used as pre- amplification and amplification factor is 2, while the amplification factor of ΣΔAD is still set as 4 to make the integral amplification factor be up to 8 times, but the calculation of t_S only depends on input signal and directly matched amplifier, corresponding relationship of R_A

and C_A in formulas of R_A=1.25KΩ and C_A=16pF to all Gain parts of SD18 is as in Table 11-5(a)/(b)/(c) .

ΣΔAD 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 11-5(a) Relationship Table of SD18 Gain and R_A& C_A

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

Table 11-5(b) Relationship Table of PGA Gain and R_A & C_A

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

Table 11-5(c) Relationship Table of VR Gain and R_A& C_A

SD18 is mainly applied to measure low-frequency signals. However, actually, signals for testing may include many high-frequency noise signal. According to the signal sampling principle, high-frequency noise signal higher than sampling frequency may produce zero drift and low frequency noise signal after sampling, and cause measurement error further. Therefore, we recommend adding 10nF~100nF filter capacitor at the chip differential signal for test and reference voltage ends to strengthen the accuracy of measurement.

11.3. Register Instruction-SD18

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
023h	INTE0	GIE	ADIE	-	WDTIE	TB1IE	TMAIE	-	E0IE	0000 0000	0uuu uuuu
026h	INTF0	-	ADIF							.000 0000	.uuu uuuu
033h	PWRCN	ENLDO[1:0]		VDDAX[1:0]			-	ADRST	CSFON	0000 0000	uuuu u00u
034h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu
035h	OSCCN1	LCPS[1:0]		ADCS[2:0]			DTMB[1:0]		TMBS	0000 0000	uuuu uuu.
036h	OSCCN2	ENRTC	-	XTS[1:0]		HAOM[1:0]		ENHAO	LPO	.000 0011	.uuu uu11
043h	ADCRH	ADC conversion memory HighByte								xxxx xxxx	uuuu uuuu
044h	ADCRM	ADC conversion memory Middle Byte								xxxx xxxx	uuuu uuuu
045h	ADCRL	ADC conversion memory Low Byte								xxxx xxxx	uuuu uuuu
046h	ADCCN1	ENADC	ENHIGN	ENCHP	-	-	ADGN[2:0]			0000 0000	0000 0000
047h	ADCCN2	-	-	-	-	VREGN	DCSET[2:0]			…. 0000	…. 0000
048h	ADCCN3	OSR[3:0]				-	-	-	-	000. ..0.	000. ..0.
049h	AINET1	INH[2:0]			INL[2:0]			INIS	-	0000 000.	0000 000.
04Ah	AINET2	-	VRH[1:0]		INX[1:0]		VRL[1:0]		-	.000 000.	.000 000.
075h	PT2	-	-	-	-	-	-	PT21	PT20	…. ..xx	…. ..xx
076h	TRISC2	-	-	-	-	-	-	TC21	TC20	…. ..00	…. ..uu
077h	PT2DA	-	-	-	-	-	-	DA21	DA20	…. ..00	…. ..uu
078h	PT2PU	-	-	-	-	-	-	PU21	PU20	…. ..00	…. ..uu
079h	PT3	-	-	PT35	PT34	PT33	PT32	PT31	PT30	..xx xxxx	..xx xxxx
07Ah	TRISC3	-	-	TC35	TC34	TC33	TC32	TC31	TC30	..00 0000	..uu uuuu
07Bh	PT3DA	-	-	DA35	DA34	DA33	DA32	DA31	DA30	..00 0000	..uu uuuu

Table 11-6 SD18 Register

INTE0/INTF0：See Chapter ***Interrupt***

PWRCN：See Chapter **Power System**

ADRST[0]: ΣΔADC and Comb Filter Reset Controller

<1> Reset; writing motion occurs.

<0>Not reset

OSCCN0/ OSCCN1/ OSCCN2: See *Chapter* ***Oscillator, Clock Source and Power Consumption Management***

ADC0RH/M/L: Out Register of SD18

ADCCN1: SD18 Control Register 1

ENADC: SD18 Start Controller

1: Start

0: Close

ENHIGN: Retain original test using

1: Setting Disable. When it is set as 1, it may cause resolution reduction of SD18.

0: Definition Setting

ENCHP: SD18 Internal chopper

1: Setting Disable. When it is set as 1, it may cause resolution reduction of SD18.

0: Definition Setting

ADGN[2:0]: AD Magnification Adjuster

111: Not using

110: x16

101: x8

100: x4

011: x2

010: x1

001: x1/2

000: Not using

ADCCN2: SD18 Control Register 2

VREGN: VR± Magnification Adjuster

1: x1/2

0: x1

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[3:0]: SD18 Over-sampling rate frequency eliminator

1010: 32768

1001: 16384

1000: 8192

0111: 4096

0110: 2048

0101: 1024

0100: 512

0011: 256

0010: 128

0001: 64

0000: 32

AINET1: AI Network Control Register 1

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

111: VSS

110: VDDA

101: VDD/4

100: VDD/2

011: AI6

010: AI4

001: AI2

000: AI0

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

111: VSS

110: ACM

101: Not using

100: Not using

011: AI7

010: AI5

001: AI3

000: AI1

INIS: SI± input signal short circuit controller

1: Short circuit

0: No short circuit

AINET2: AI Network Control Register 2

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

11: AI1

10: AI0

01: ACM

00: VDDA

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: AI4

10: AI3

01: AI2

00: VSS

12. BIE and 16-bit Hardware Data Recorder

Built-in EPROM (shortened as BIE) is composed of BIEAR[11:0] index register, BIEDR[15:0] value register and BIECN[7:0] control register. Related controllers and flags are BIEARL[7:0]/BIEARH[3:0] address controller, ENBIE[0] start controller, BIEWR[0] writing controller, BIERD[0] reading controller and VPPHV[0] voltage condition flags.

- BIE Features:
  - Use BIE function to store product serial number, security code and data and materials after program computation.
  - Provide additional 64words (equivalent to 128 bytes) for PM. Storage address range is 00H~3FH. Take data processing.
  - Have reading, writing and 16-bit table look-up function.
  - External hardware only needs connect VPP 6V burn voltage with VPP pins externally.
  - VPPHV[0] is flag register that reflects VPP pin voltage status immediately.
  - The reading of BIE is unnecessary to consider pin externally connected voltage. It is only necessary to consider effective voltage is equivalent to VDD.
  - Burn time of each byte of data is about 150ms.

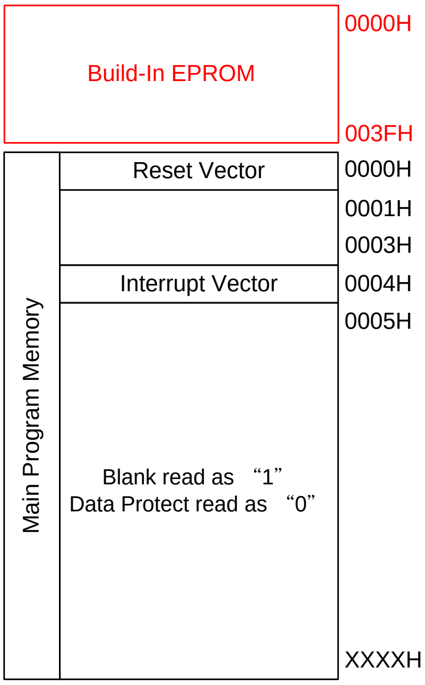


Fig 12-1 Built-In EPROM Architecture

Abstract of BIE Register:

BIECN	VPPHV[0], BIEWR[0], BIERD[0]
BIEARH	ENBIE[0],BIE_BIEAH[2:0]
BIEARL	BIEAL[7:0]
BIEDRH	BIEDH[15:8]
BIEDRL	BIEDL[7:0]

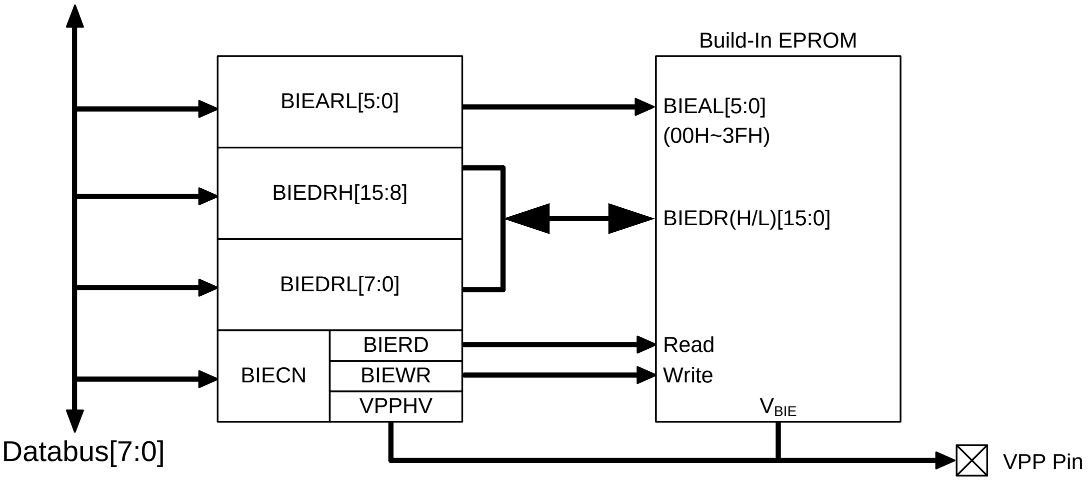


Fig 12-2 BIE Block Diagram

12.1. BIE Manual

12.1.1. BIE Reading

- When the user reads BIE:
    - Set **BIE**ARL as readable **BIE** Address (Max 3FH，**BIE**ARL[7]=0、**BIE**ARL[6]=0)
    - BSF **BIE**CN, **BIE**RD,F
        - The instruction is invalid when **BIE** Address surpasses 3FH.
        - BIE READ motion has no relation with VPP potential, but it cannot be 0V.
    - Judge whether BIECN [BIERD] is cleared to be 0 automatically after **BIE** reading is completed.
    - Read BIEDRH, BIEDRL for the BIE Data

Notes: Before reading **BIE**, close ADC function (ADCCN1 [ENADC] =0b) first and capacity of resisting disturbance of ADC can be enhanced. After the **BIE** reading is completed, restart ADC function to take signal measurement.

12.1.2. BIE Writing

- Before the user writes **BIE**, read VPPHV first to confirm whether VPP voltage is correct.
- When the user writes BIE:
  - Set **BIE**ARL as writable **BIE** address (max 3FH，**BIE**ARL[7]=0, **BIE**ARL[6]=0)
  - Set **BIE**DRH, **BIE**DRL as writable **BIE** data
  - BSF **BIE**CN, **BIE**WR,F
    - The instruction is invalid when **BIE** address surpasses 3FH.
    - The instruction is invalid when VPP is not 6V.
  - Judge whether BIECN [BIEWR] is cleared to be 0 automatically after **BIE** writing is completed.
  - When current consumption is increased obviously, it is suggested taking **BIE** writing motion at ADC testing mode.

Notes: Before writing **BIE**, close ADC function (ADCCN1 [ENADC] =0b) first and capacity of resisting disturbance of ADC can be enhanced. After the **BIE** writing is completed, restart ADC function to take signal measurement.

12.1.3. Notes

- Before reading and writing BIE, please set CPU frequency source as HAO first. Otherwise, the motion may be abnormal.
- After BIE READ/WRITE motion is completed, BIERD/BIEWR is cleared to be 0 automatically, and BIEARL is increased automatically (max 3FH).
- It is suggested using BSF instruction to set BIERD of BIEWR. The instruction is invalid if the both are set as 1 simultaneously.
- When VPP is in high potential, after CPU is reset, PT1.5 maintains 65ms unknown potential (high potential or low potential) status output.
- Power on sequences: 1.Powering on VDD first, 2. Powering on VPP next.
- Before reading **BIE** or writing **BIE**, close ADC function (ADCCN1 [ENADC] =0b) first and capacity of resisting disturbance of ADC can be enhanced. After the **BIE** reading or **BIE** writing is completed, restart ADC function to take signal measurement.

12.2. Hardware Data Recorder

When BIE peripheral does not start burn function, it can be used as 16-bit hardware data recorder, but it can only read program memory and cannot take data writing.

It is unnecessary to consider pin externally connected power voltage for hardware data recorder reading. The following are operation instructions:

- Set the address of the table to be read as BIEARH, BIEARL(Up to 7FFH)
- After BIECN[BIERD] is set as <1>, check BIECN [BIERD] is automatically cleared to 0, determine whether the completion of the look-up table.
- Then reads BIEDRH, BIEDRL registers, look-up table is the Data.

```
16BITS_READ:
      MVL   HIGH Table          ;Set table look-up address
      MVF   BIEARH,F,ACCE
      MVL   LOW Table
      MVF   BIEARL,F,ACCE
      BSF BIECN, BIERD,F
WAITRDBIE:
      BTSZ   BIECN, BIERD, 1          ; check BIECN [BIERD] is automatically cleared to 0,
                                      ; determine whether the completion of the look-up table.
      JMP    WAITRDBIE
      MVF    BIEDRL, W, 1
      MVF    BUF0, F, 1               ; move BIEDRL data to BUF0

      MVF    BIEDRH, W, 1
      MVF    BUF1, F, 1               ; move BIEDRH data to BUF1
```

Example 12-1 16Bits Table Look-up Software Setting Example Program

12.3. Register Instruction-BIE

"-"no use,"*"read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1											
"$"for event status,"."unimplemented bit,"x"unknown,"u"unchanged,"d"depends on condition											
Address	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**ARST**	**IRST**
02Eh	BIECN	-	-	-	-	VPPHV	-	BIEWR	BIERD	1… $.00	1... $.uu
02Fh	BIEARH	ENBIE	-	-	-	-	11-bit look-up Table as BIEAH[2:0]			0... xxxx	u... uuuu
030h	BIEARL	BIE Address Register as BIEAL[5:0] or 11-bit look-up Table as BIEA[7:0]								xxxx xxxx	uuuu uuuu
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu

Table 12-7 BIE Register

BIECN: BIE Control Register

VPPHV: Check VPP

0: VPP No external connection with burn power 6V

1: VPP external connection with burn power 6V

BIEWR: Write BIE control bit

0: Non- writable

1: Writable

BIERD: Read BIE control bit

0: Non-readable

1: Readable

BIEARH: BIE Address Definition

ENBIE: Mode Selection

0: OTP READ MODE

1: BIE MODE

BIEAH[2:0]: OTP address

BIEARL: BIE Low Byte Address Definition

BIEAL[5:0]: OTP address

BIEDH: BIE High Byte Data Definition

BIEDL: BIE Low Byte Data Definition

13. Communication interface (CI)

Main type of communication interface (shortened as CI) is I2C serial communication.

13.1. I2C Inter-Integrated Circuit Serial interface

I2C communication interface includes two types of operation modes, i.e. mater mode and slave mode. Master mode can combine Transmission Controller (Tx Controller) to transmit 12C packet format signal to 12C Bus according to the system requirements, and use Clock Generator to determine the required transmission ratio. Slave Controller can receive signal on 12C Bus. It receives master communication requirements on Bus, and combines Transmission Controller to post back the master required data. Additionally, Slave controller built-in receiving circuit is also a channel for Master Controller to receive posted back data.

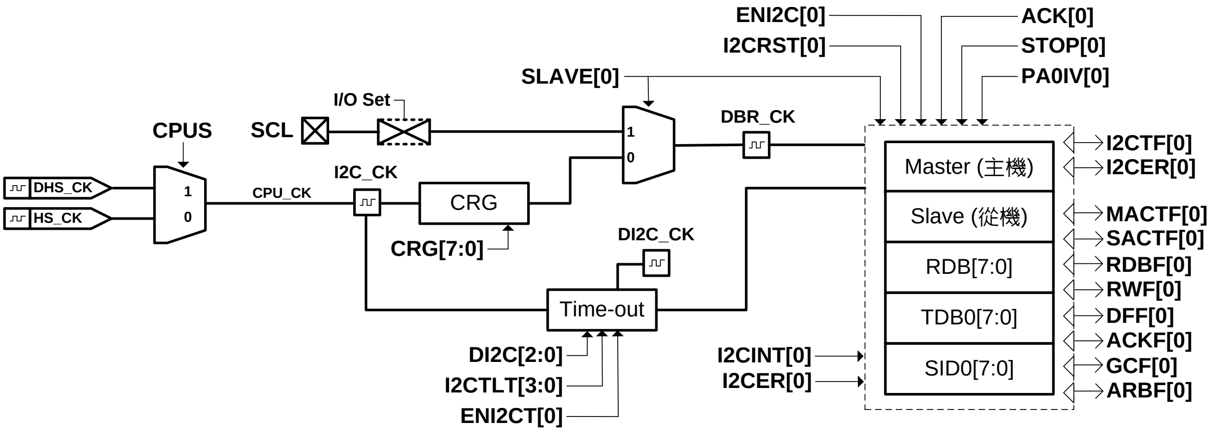


Fig 13-1 I2C System Architecture

- Functional Features of I2C Serial Interface:
  - Standard I2C serial interface includes 2 pins---serial data and (SDA) and serial clock.
  - The pin is Open Drain output structure. It needs external pull-up resistor, to ensure high potential output.
  - Standard I2C serial interface can be configured as Master, Slave or Master /Slave mode.
  - Programmable clock is allowable to adjust I2C transmission speed.
  - Data transmission between the master and slave is two-way.
  - I2C allows rather large work voltage range.
  - The reference design of I2C uses a 7-bit address space, but reserves 16-bit address. Therefore, it can communicate with 112 nodes in a group of bus-bar.

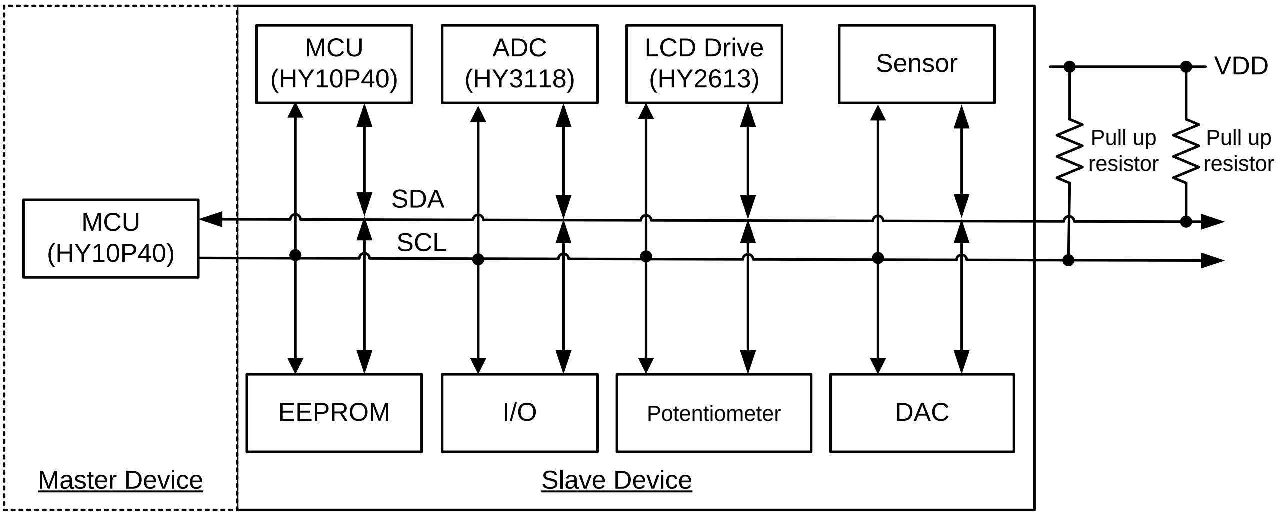


Fig 13-2 I2C Communication Wiring Schematic Diagram

- I2CSerial Interface Signal:
  - START: When master SCL is in high potential, it issues SDA and transfers from high potential to low potential. Then it starts transmit data.
  - DATA or ADDRESS signal: I2C serial interface protocol requires data on SDA can only be changed when SCL is in low potential.
  - Acknowledge Signal: When device (slave) for data receiving receives the 8th bit, it sends low potential to device (slave) for data sending, and indicates data has been received.
  - STOP Signal: When Master SCL is in high potential, it issues SDA and transfers from low potential to high potential to. Then data transition is ended.

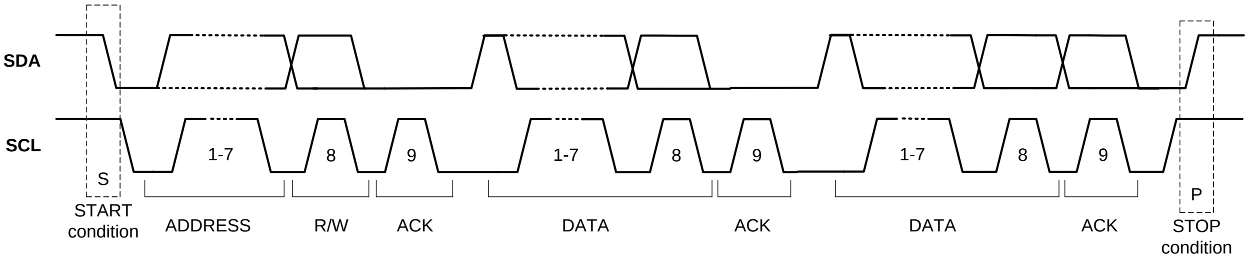


Fig 13-3 I2C Bus-bar Timing Diagram

13.2. Data Transmission Ratio Calculation

12C internal register CRG[7:0] can control data transmission speed under master mode. The value produces SCL pin signal of master via internal counter. Therefore, data transmission ratio can be calculated by using the following formula according to the frequency of 12C clock source I2C_CK.

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

13.3. Time-Out Function

Time-out control is to avoid 12C controller locks the 12C communication bus-bar deadly and provides sufficient time for 12C controller handling requirements during the operation process of 12C. Therefore, 12C will pull SCL to low after every acknowledge bit and make Master cannot transmit next clock signal, i.e. Clock Stretching. However, when MCU is too busy to acknowledge 12C controller or any other causes make MCU cannot acknowledge 12C controller, SCL of 12C communication bus-bar will be locked at Low deadly.

To avoid the above conditions, Time-out controller can determine the Time-out conditions when SCL is in Low status via work frequency eliminator DI2C[2:0] and time condition controller I2CTLT[3:0] according to the user's requirements. Condition handling has the following status:

When it is detected the time of SCL pulled by the local host satisfies the conditions, 12C controller will release SCL compulsively and issue interrupt event to CPU.

When SCL is not up to Time-out time and is released as High, the internal timer of Time-out controller will be reset, and take recounting when SCL is pulled to Low again next time.

13.4. I2C Serial Interface Communication Flowchart

- I2CSerial Interface Terms
  - (SPIA): It means issued orders of Action Control Register (ACT).

    S is Start instruction.

    P is Stop instruction.

    I is interrupt flag.

    A is Acknowledge instruction.
  - SPIA: It means Action Control Register reading value. It can be applied to judge whether the interrupt flag or other instruction is operated completely.
  - STA: It means Status register (STA) reading value. It is used to indicate current 12C circuit operation status.
  - The following flowchart takes Fig 13-4 indicated "Grey Bottom Round Block" , "White Bottom Round Block" and "Square Block" to indicate 12C interface status respectively:

    Grey Bottom Round Block : It means interrupt flag has been set as 12C status.

    White Bottom Round Block : It means interrupt flag has not been set. It needs the MCU read the12C status actively.

    Square Block : It means it needs the MCU issue instruction to 12C.

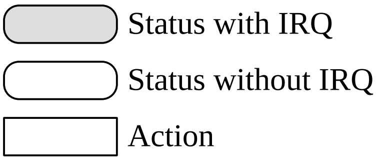


Fig 13-4 Flowchart Symbols

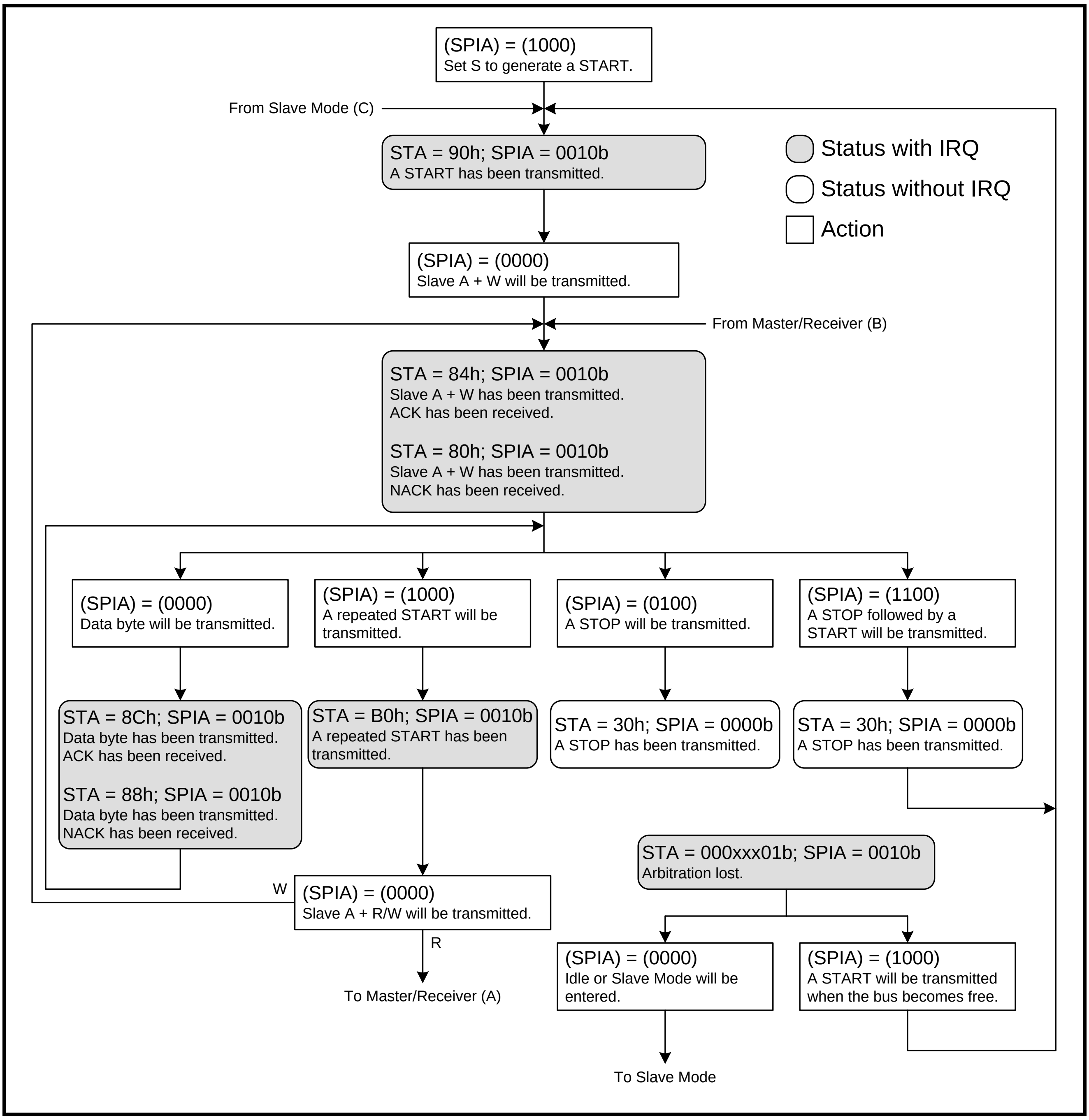

(SPIA) = (1000)
Set S to generate a START.
From Slave Mode (C)
STA = 90h; SPIA = 0010b
A START has been transmitted.
Status with IRQ
Status without IRQ
Action
(SPIA) = (0000)
Slave A + W will be transmitted.
From Master/Receiver (B)
STA = 84h; SPIA = 0010b
Slave A + W has been transmitted.
ACK has been received.
STA = 80h; SPIA = 0010b
Slave A + W has been transmitted.
NACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
(SPIA) = (1000)
A repeated START will be transmitted.
(SPIA) = (0100)
A STOP will be transmitted.
(SPIA) = (1100)
A STOP followed by a START will be transmitted.
STA = 8Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
STA = 88h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = B0h; SPIA = 0010b
A repeated START has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
STA = 000xxx01b; SPIA = 0010b
Arbitration lost.
W
(SPIA) = (0000)
Slave A + R/W will be transmitted.
R
To Master/Receiver (A)
(SPIA) = (0000)
Idle or Slave Mode will be entered.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
To Slave Mode

Fig 13-5 Master Transmitter Mode

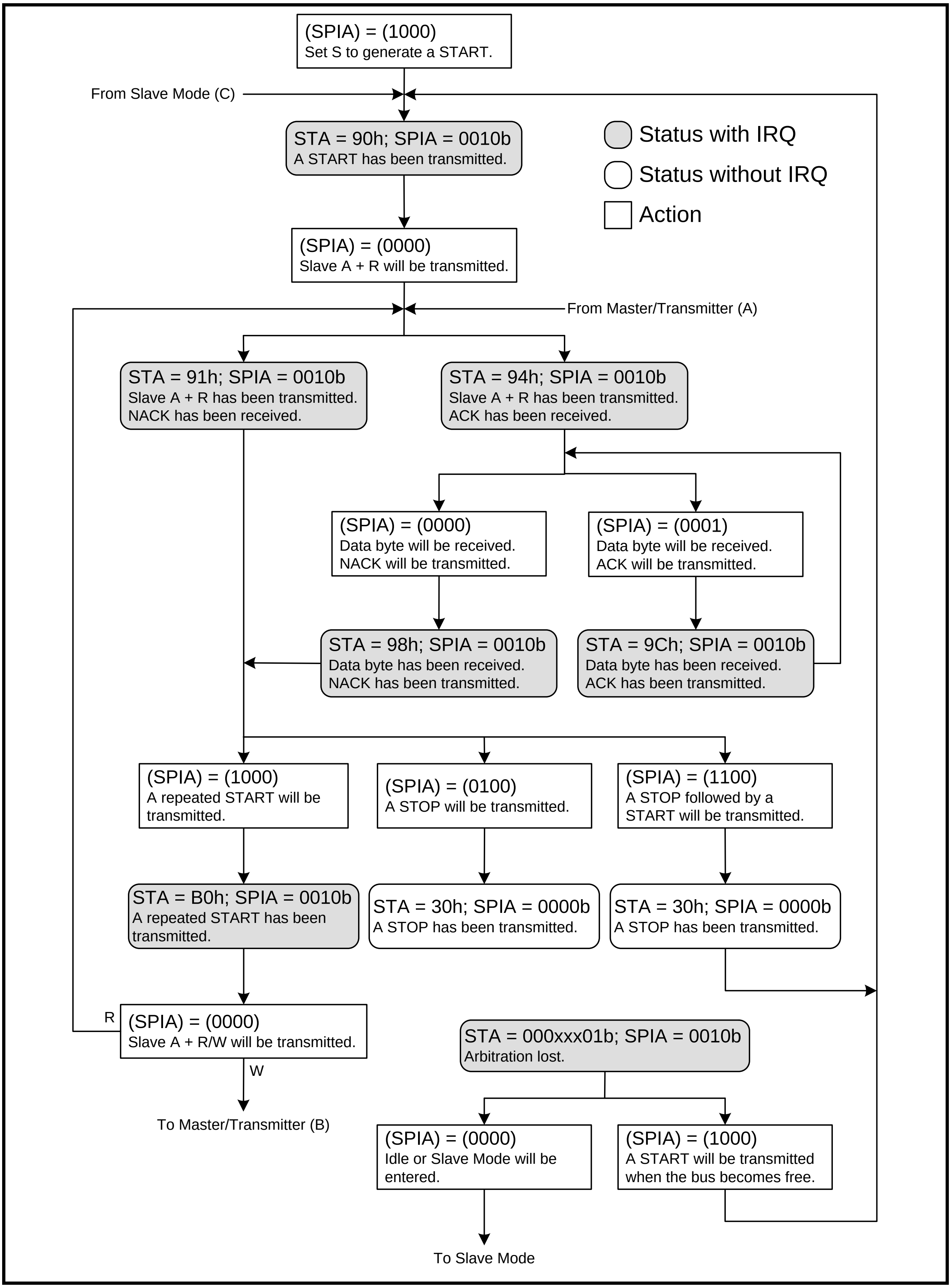

(SPIA) = (1000)
Set S to generate a START.
From Slave Mode (C)
STA = 90h; SPIA = 0010b
A START has been transmitted.
Status with IRQ
Status without IRQ
Action
(SPIA) = (0000)
Slave A + R will be transmitted.
From Master/Transmitter (A)
STA = 91h; SPIA = 0010b
Slave A + R has been transmitted.
NACK has been received.
STA = 94h; SPIA = 0010b
Slave A + R has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 98h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 9Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
(SPIA) = (1000)
A repeated START will be transmitted.
(SPIA) = (0100)
A STOP will be transmitted.
(SPIA) = (1100)
A STOP followed by a START will be transmitted.
STA = B0h; SPIA = 0010b
A repeated START has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
STA = 30h; SPIA = 0000b
A STOP has been transmitted.
R
(SPIA) = (0000)
Slave A + R/W will be transmitted.
W
To Master/Transmitter (B)
STA = 000xxx01b; SPIA = 0010b
Arbitration lost.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
To Slave Mode

Fig 13-6 Master Receiver Mode

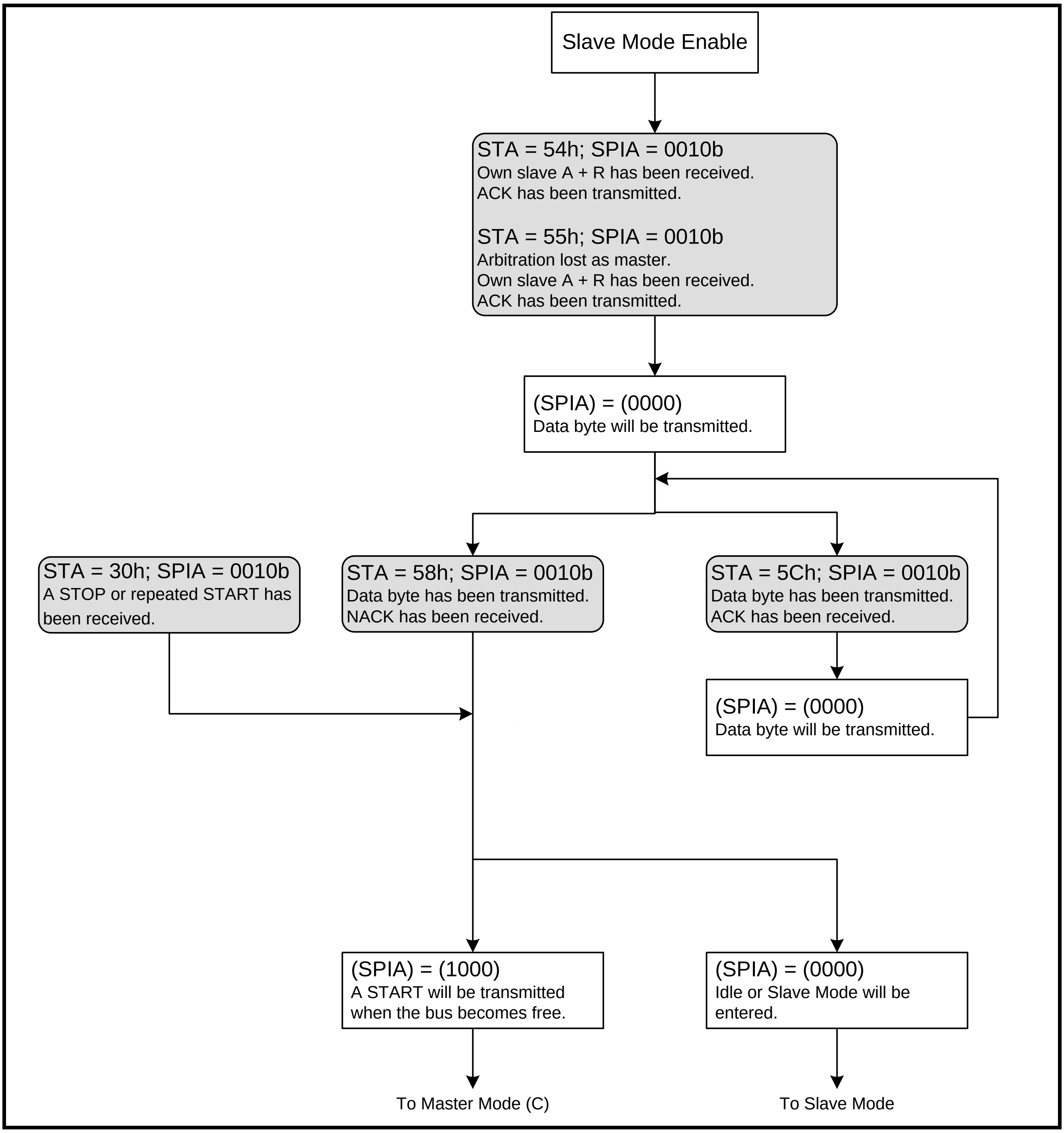

Slave Mode Enable
STA = 54h; SPIA = 0010b
Own slave A + R has been received.
ACK has been transmitted.
STA = 55h; SPIA = 0010b
Arbitration lost as master.
Own slave A + R has been received.
ACK has been transmitted.
(SPIA) = (0000)
Data byte will be transmitted.
STA = 30h; SPIA = 0010b
A STOP or repeated START has been received.
STA = 58h; SPIA = 0010b
Data byte has been transmitted.
NACK has been received.
STA = 5Ch; SPIA = 0010b
Data byte has been transmitted.
ACK has been received.
(SPIA) = (0000)
Data byte will be transmitted.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

Fig 13 −7 Slave Transmitter Mode

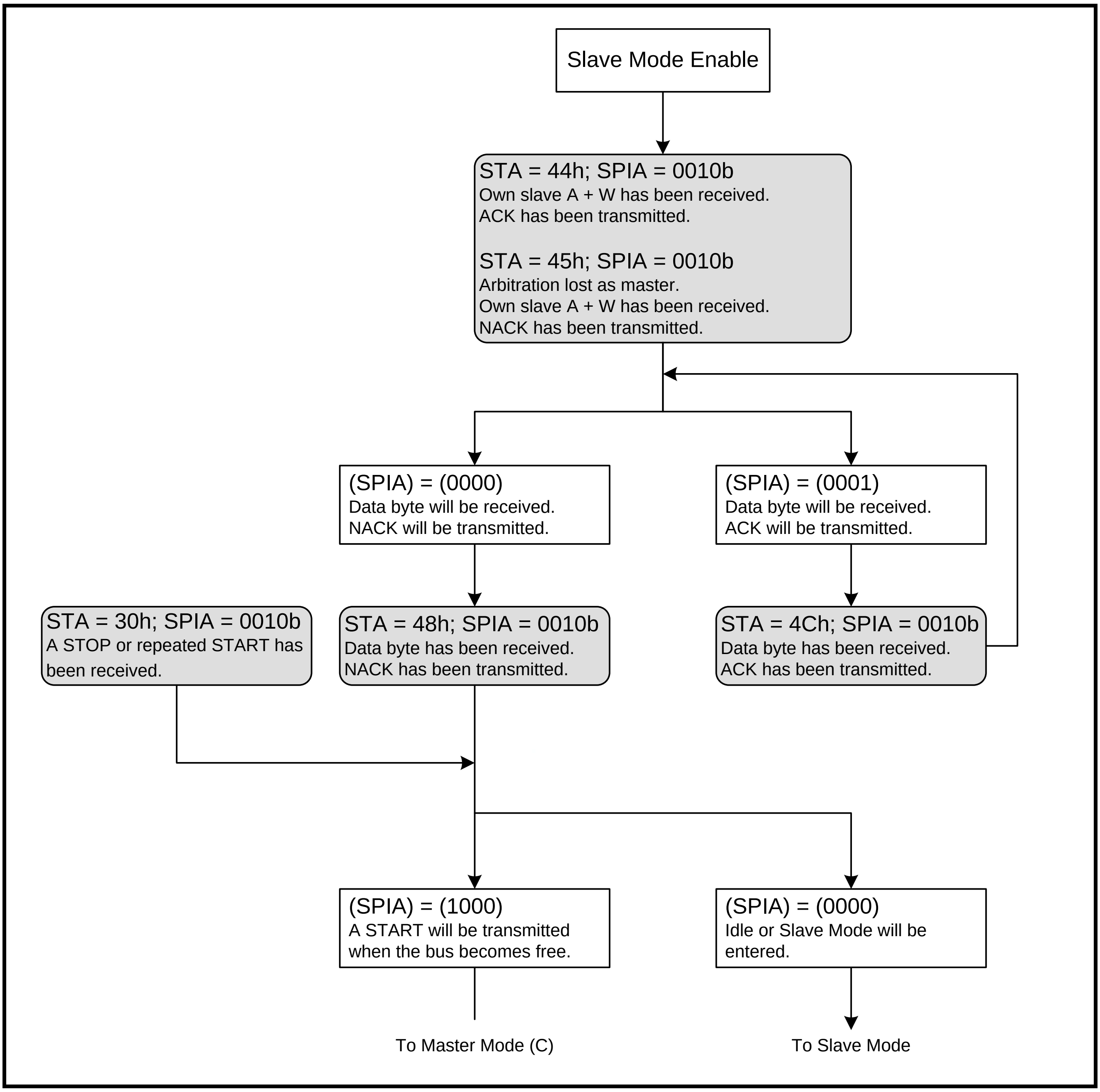

Slave Mode Enable
STA = 44h; SPIA = 0010b
Own slave A + W has been received.
ACK has been transmitted.
STA = 45h; SPIA = 0010b
Arbitration lost as master.
Own slave A + W has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 30h; SPIA = 0010b
A STOP or repeated START has been received.
STA = 48h; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Ch; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

Fig 13-8 Slave Receiver Mode

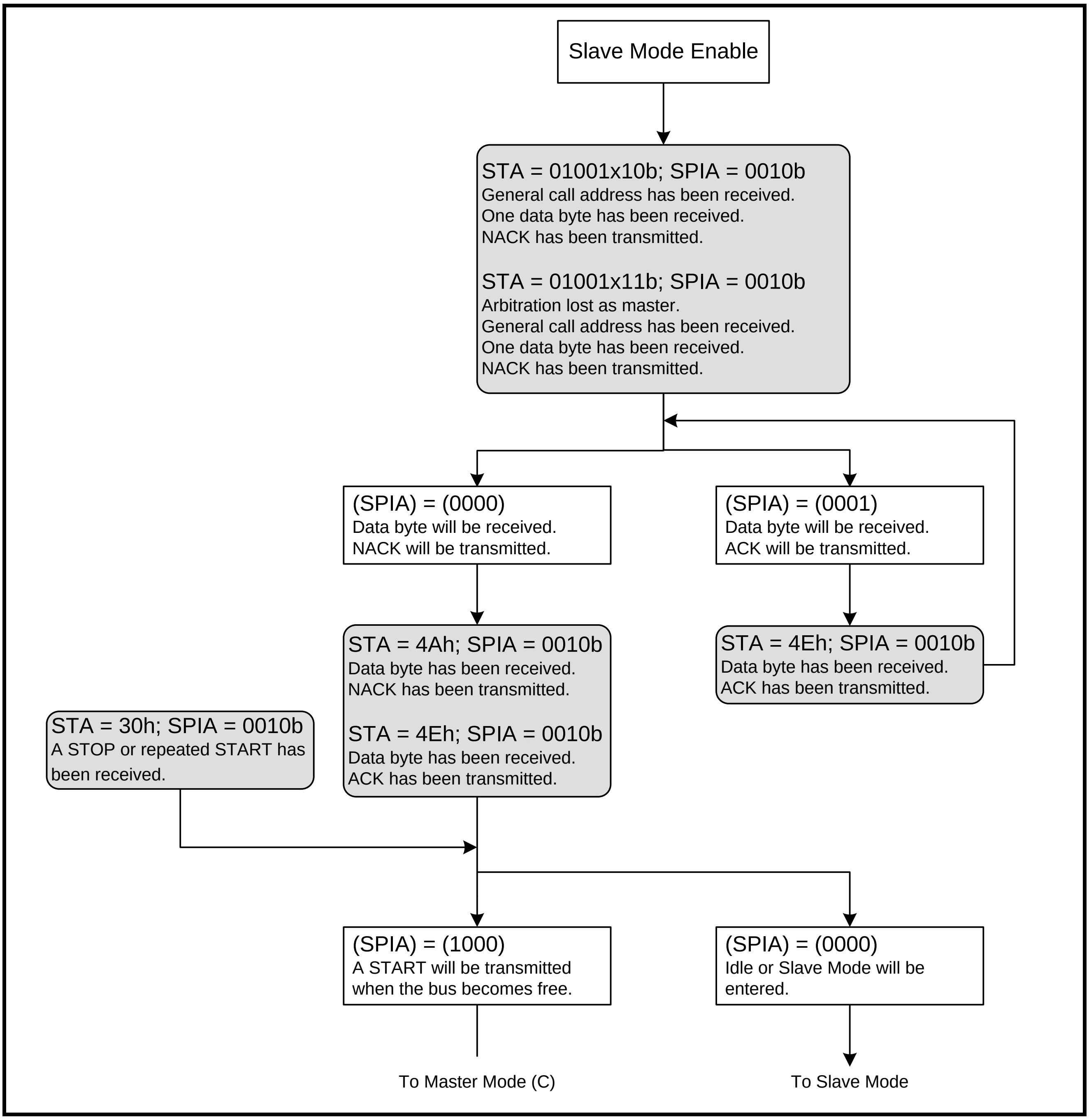

Slave Mode Enable
STA = 01001x10b; SPIA = 0010b
General call address has been received.
One data byte has been received.
NACK has been transmitted.
STA = 01001x11b; SPIA = 0010b
Arbitration lost as master.
General call address has been received.
One data byte has been received.
NACK has been transmitted.
(SPIA) = (0000)
Data byte will be received.
NACK will be transmitted.
(SPIA) = (0001)
Data byte will be received.
ACK will be transmitted.
STA = 4Ah; SPIA = 0010b
Data byte has been received.
NACK has been transmitted.
STA = 4Eh; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 4Eh; SPIA = 0010b
Data byte has been received.
ACK has been transmitted.
STA = 30h; SPIA = 0010b
A STOP or repeated START has been received.
(SPIA) = (1000)
A START will be transmitted when the bus becomes free.
(SPIA) = (0000)
Idle or Slave Mode will be entered.
To Master Mode (C)
To Slave Mode

Fig 13-9 General Call Mode

13.5. I2C Register Instructions

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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST
024h	INTE2	-	-	-	-	I2CERIE	I2CIE	-	-	0000 0000	uuuu uuuu
027h	INTF2	-	-	-	-	I2CERIF	I2CIF	-	-	0000 0000	uuuu uuuu
061h	CFG	Rsv.					I2CRST	ENI2CT	ENI2C	…. .000	…. .uuu
062h	ACT	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu
063h	STA	MACTF	SACTF	RDBF	RWF	DFF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu
064h	CRG	CRG[7:0]								0000 0000	uuuu uuuu
065h	TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu
066h	RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu
067h	TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu
068h	SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu

Fig 13-1 I2C Register

INTE1/INTF1: See Chapter Interrupt.

CFG: I2C Setting Register

GCRst: I2C General call reset enable control

1: Start

0: Close

Note: When the I2C Slave mode is turned on with the GCRst function at the same time, if the I2C Controller receives the General call ID 00h and the first data is "06h", the General Call Reset condition holds, The interrupt signal (Interrupt) that was originally sent to the local processor will be replaced by a reset signal, Providing an external host can reset functions of the machine via a chip I2C Bus.

ENI2CT: I2C Start Time-out Monitoring Function Bit

1: I2C Start Time-out Monitoring Function

0: Close

ENI2C: I2C Start Function Control Bit

1: Start I2C CI

0: Close

※ Notes: When ENI2C is closed, it will close internal Clock of 12C, except Configuration Register can take writing motion. The rest registers cannot write data.

ACT: Action Register

SLAVE: Slave Start Control

1: Start

0: Close

I2CER: Error Interrupt Flag

1: Occurred error interrupt

0: Normal, 0 writing will eliminate error interrupt flag to enable 12C take next status execution.

START: Start Order Bit

1: Produce Start Signal at I2C Bus

0: Normal

STOP: Stop Order Bit

1: Produce Start Signal at I2C Bus

0: Normal

I2CINT: Interrupt Flag

1: Occurred 12C interrupt

0: Normal, 0 writing will eliminate interrupt flag to enable 12C take next status execution.

ACK: ACK(Acknowledge) Bit

1: ACK Acknowledged

0: Not Acknowledged ACK or Acknowledged NACK

STA: I2C Status Register

MACTF: Master Mode Active Flag

1: Started

0: Not Started

SACTF: Slave Mode Active Flag

1: Started

0: Not Started

RDBF: Received Stop/Repeat-Start Flag

1: Received Stop/Repeat-Start Flag has been sent or received,

0: Normal

RWF: Read/Write State Flag)

1: Read order has been sent or received,

0: Write order has been sent or received,

DFF: Data Field Flag

1: I2C Data has been sent or received,

0: Normal

ACKF: Acknowledge Flag (ACK Flag)

1: ACK has been sent or received,

0: ACK has not been sent or received,

GCF: General Call Flag

1: Currently General Call Operation

0: Normal

ARBF: Arbitration Lost Flag

1: Arbitration Lost

0: Normal

CRG: I2C Clock Control Register

CRG [7:0]: I2C Bus Data Baud Rate Control

Data transmission on I2C Bus is determined by clock signal on SCL pins, and the clock rate of SCL pin can be calculated by clock source frequency CPU_CK and CRG according to the following formula:

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

TOC: I2C Time-out Control Register

I2CTF: Time-out Flag

1: I2C Bus Clock Stretching Time-out

0: Normal

DI2C[2:0]: Time-out Clock Pre-scale

0: CLKPS = CPU_CK / 1

1: CLKPS = CPU_CK / 2

2: CLKPS = CPU_CK / 4

3: CLKPS = CPU_CK / 8

4: CLKPS = CPU_CK / 16

5: CLKPS = CPU_CK / 32

6: CLKPS = CPU_CK / 64

7: CLKPS = CPU_CK / 128

I2CTLT[3:0]: Time-out Limit；The occurrence of Time-out is triggered after CLKPS is counted I2CTLT + 1 times.

0: 1x CLKPS Cycle

1: 2x CLKPS Cycle

2: 3x CLKPS Cycle

3: 4x CLKPS Cycle

…

15: 16x CLKPS Cycle

RDB: Data Receiving Register

RDB [7:1]: The content is receiving address (A7~A1) or data (D7~D1).

RDB [0]: The content is receiving read/write order or data (D0).

TDB0: Data Transition Register

TDB0 [7:1]: The content is receiving address (A7~A1) or data (D7~D1).

TDB [0]: The content is receiving read/write order or data (D0).

※ Notes : During communication process, when the machine belongs to non Address or Data transmission status, it must set the register as FFh, as Bit 7 of TDB0 is 0 and it may lock the SDA Bus at Low deadly.

SID0: Slave Mode ID Code Setting Register

SID[7:1]: Slave Mode ID Code (A7~A1)

SIDV[0]: Effective Control of Slave Mode ID Code

0: Invalid Slave Mode ID Code

1: Valid Slave Mode ID Code

14. REVISION RECORD

Major differences are stated thereinafter:

Date	Version	Page	Revision Summary
2013/08	V03	All	First edition
2017/08/30	V06	73 ALL 52~64	1. Added ADCR ΔVR_I Full Scale Note 2. Synchronize Chinese version 3. Modify the description of the Chapter 9 TMB
2017/11/16	V07	81~85	Modify the description of the BIE and 16-bit hardware look-up table