CMOS single-chip 8-bit MCU

with 12-bit A/D Converter, Analog Comparator and OP-AMP

ABOV SEMICONDUCTOR

Main features

- **8-bit Microcontroller with High Speed 8051 CPU**
- **Basic MCU Function**
  - 8Kbytes Flash Code Memory
  - 512bytes SRAM Data Memory
- **Built-in Analog Function**
  - Power-On Reset
  - Brown Out Detect Reset
  - Internal 16MHz RC Oscillator
  - Internal WDT RC Oscillator
- **Peripheral features**
  - 12-bit Analog to Digital Converter
  - Communication Interface: UART, I2C
  - 16-bit Timer/Counter/PWM
  - Buzzer
- **PPG Control Function**
  - 5 Analog Comparator
  - 2 OP-AMP
- **I/O and packages**
  - Up to 18 programmable I/O lines with 20 SOP
  - 20 SOP, 16 SOPN
- **Operating conditions**
  - 2.7V to 5.5V wide voltage range
  - -40°C to 85°C temperature range
- **Application**
  - PPG MCU for IH Cooker

MC97F6108A

User's manual

V 1.9

Revised 12 Dec, 2017

Revision history

Version	Date	Revision list
1.0	2014.10.22	Upload Initial version.
1.1	2015.09.01	1. Add VDD rise rate. 2. Remove LVDRF flag in RSFR register. 3. Remove IRCOFF in SCCR register. 4. Remove LVROFF in BODR register 5. Change the timing for detecting the Comparator2 output when using auto-period mode. 6. Remove POL (PPGCR1) 7. Change the offset spec of OP-Amp & Analog Comparator.
1.2	2015.09.22	1. Fix incorrect phrases.
1.3	2015.11.19	1. Change development tools. - No longer support PGMPlusUSB, PGMPlusLC ADAM Single Writer, ADAM Gang8.
1.4	2015.11.26	1. Change the default settings for the BOD to enable.
1.5	2016.01.07	1. Change the initial value of RSFR register.
1.6	2016.02.24	1. Change the initial value of BODR register.
1.7	2016.03.14	1. Correct FLASH Endurance Spec : 10,000 times
1.8	2016.06.29	1. Delete 16/20 PDIP package information 2. Correct Internal RC-OSC error rate in Table 7-2
1.9	2017.12.12	Add Device nomenclature.

Version 1.9

Published by FAE team

Additional information of this manual may be served by ABOV Semiconductor offices in Korea or distributors.

ABOV Semiconductor reserves the right to make changes to any information here in at any time without notice.

The information, diagrams and other data in this manual are correct and reliable;
however, ABOV Semiconductor is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

1 Overview

1.1 Description

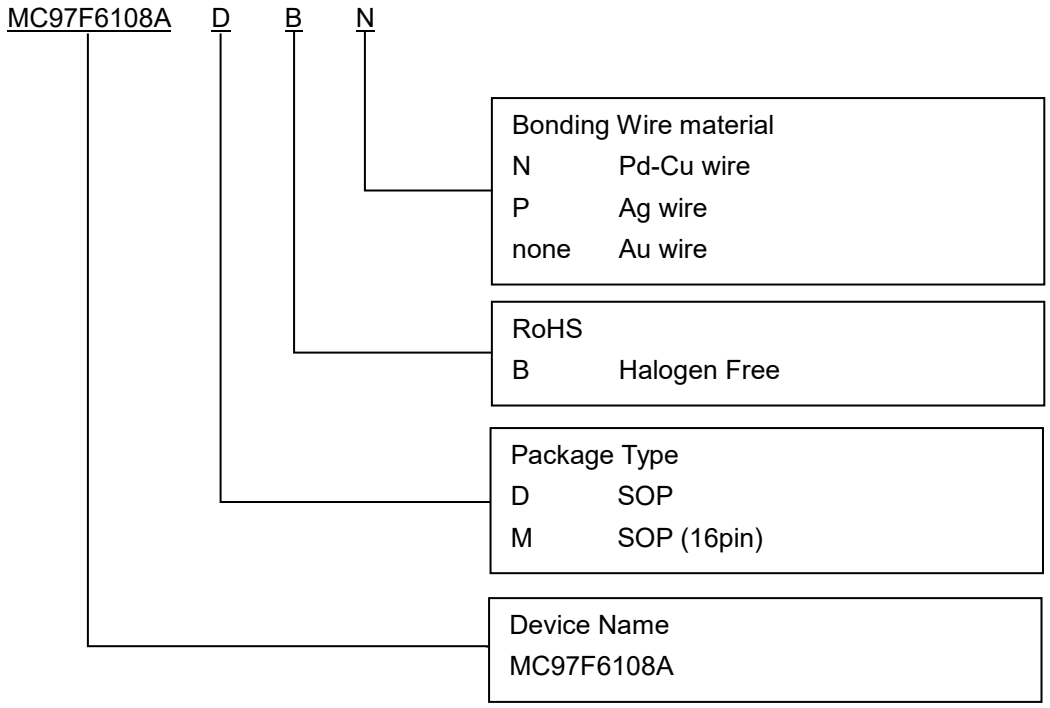
The MC97F6108A is an advanced CMOS 8-bit microcontroller with 8Kbytes of FLASH. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 8Kbytes of FLASH, 256bytes of IRAM, 256bytes of XRAM, general purpose I/O, 16-bit timer/counter, PPG (Programmable Pulse Generator), buzzer, Basic Interval Timer, watchdog timer, 12-bit ADC, USART, I2C, on-chip POR (Power On Reset), BOD (Brown Out Detect) and LVR(Low Voltage Reset), Internal RC-oscillator, Internal WDT-oscillator and analog comparators, on-chip OP-AMP. The MC97F6108A also supports power saving modes to reduce power consumption.

1.2 MC97F6108A ORDERING INFORMATION

Device Name	FLASH	XRAM	IRAM	ADC	I/O PORT	Package
MC97F6108AD	8 Kbytes	256 bytes	256 bytes	8 inputs	18	20 SOP
MC97F6108AM				7 inputs	14	16 SOP

Table 1-1. Ordering Information of MC97F6108A

1.3 Device nomenclature



1.4 Features

- **CPU**
  - 8-bit CISC core (M8051 compatible)
  - 2 clocks per cycle
- **ROM (FLASH) capacity**
  - 8Kbytes Flash Memory
  - Self-Writing
  - In-system programming(ISP)
  - Endurance : 10,000times
  - Retention : 10years
- **256bytes IRAM**
- **256bytes XRAM**
- **GPIO (General purpose I/O)**
  - Normal I/O : 18 ports (20 pin)
    P0[7:0], P1[7:0], P2 [1:0]
  - Normal I/O : 14 ports (16 pin)
    P0[7:1], P1[7:3], P2[1:0]
- **BIT (Basic Interval Timer)**
  - 8-bit × 1-ch
- **WDT (Watch Dog Timer)**
  - 8-bit × 1-ch
- **Timer/Counter/PWM output**
  - 16-bit × 4-ch(T0/T1/ T2/T3)
- **UART**
  - 8-bit x 1-ch
- **I2C**
  - 8-bit x 1-ch
- **12-bit A/D converter**
  - 8 input channels
- **Analog Comparator**
  - 5 Comparators
    1 comparator for Sync.
    1 comparator for zero-cross
    3 comparators for protection
    (Surge, IGBT over voltage, over current)
    Internal reference voltage for comparator
- **OP-AMP**
  - 2 OP-Amp 2stage amplifier
- **PPG**
  - 1-ch output (PPG step < 0.1us)
- **Buzzer**
  - 1-ch
- **POR (Power On Reset)**
  - Reset release level (1.4V @1ms VDD rising)
- **LVR (Low Voltage Reset)**
  - 1-level 1.8V
- **BOD (Brown Out Detect)**
  - 6 levels(2.2V(default) / 2.5V / 2.7V / 3.2V / 3.7V / 4.2V)
- **PCI (Pin Change Interrupt)**
  - P1[7:0]
- **Interrupt sources**
  - External Interrupt (EINT0, EINT1, EINT2, PCI)
  - Comparator output(5)
  - I2C(1)
  - USART(2)
  - Timer(4)
  - PPG(1)
  - ADC(1)
  - WDT(1)
  - BIT (1)
  - BOD (1)
- **IRC (Internal RC oscillator)**
  - 16MHz ±3.0% (TA = -40℃ ~ +85℃)
- **WDTOSC (Internal WDT RC oscillator)**
  - 8kHz
- **Power down mode**
  - STOP1, STOP2, IDLE mode
- **Operating voltage and frequency**
  - 2.7V ~ 5.5V (at 16MHz internal RC)
- **Minimum instruction execution time**
  - 125ns (at 16MHz, NOP Instruction)
- **Operating temperature: – 40℃ ~+ 85℃**
- **Package type**
  - 20 SOP, 16 SOPN

1.5 Development Tools

1.5.1 Compiler

We do not provide the compiler. Please contact third parties.

The MC97F6108A's CPU is Mentor Graphic 8051. Anyway, device ROM size is smaller than 64Kbytes. Developer can use all kinds of third party's standard 8051 compiler.

1.5.2 OCD2 emulator and debugger

The OCD2 emulator supports ABOV Semiconductor's 8051 series MCU emulation.

The OCD2 interface uses two wires interfacing between PC and MCU which is attached to user's system. The OCD2 can read or change the value of MCU internal memory and I/O peripherals. And also the OCD2 controls MCU internal debugging logic, it means OCD2 controls emulation, step run, monitoring, RAM break etc.

The OCD2 Debugger program works on Microsoft-Windows NT, 2000, XP, Vista (32-bit) operating system.

If you want to see more details, please refer OCD2 debugger manual. You can download debugger S/W and manual from our web-site.

There are two types of OCD2 mode connection.

Connection 1:

- P02 (MC97F6108A DSCL pin)
- P01 (MC97F6108A DSDA pin)

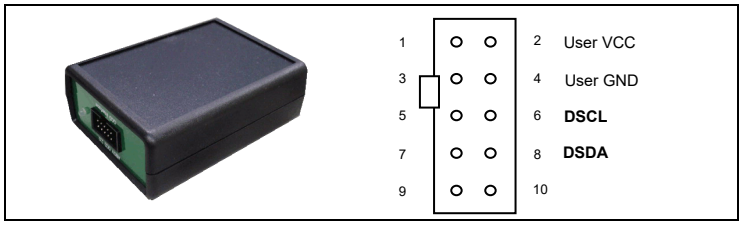


Figure 1.1 On Chip Debugger 2 and Pin description (OCD2 mode)

Connection 2:

- P11 (MC97F6108A DSCL1 pin)
- P12 (MC97F6108A DSDA1 pin)

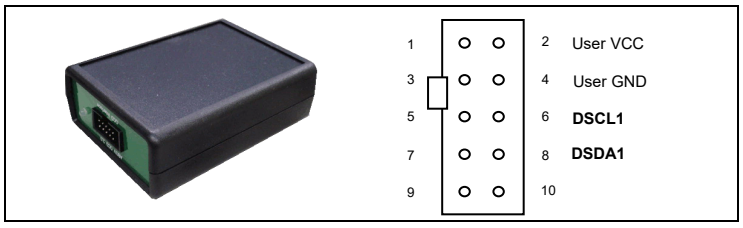


Figure 1.2 On Chip Debugger 2 and Pin description (OCD2 mode 1)

1.5.3 Programmer

E-PGM +

- Support ABOV / ADAM devices
- 2~5 times faster than S-PGM+
- Main controller : 32 bit MCU @ 72MHz
- Buffer memory : 1 MByte

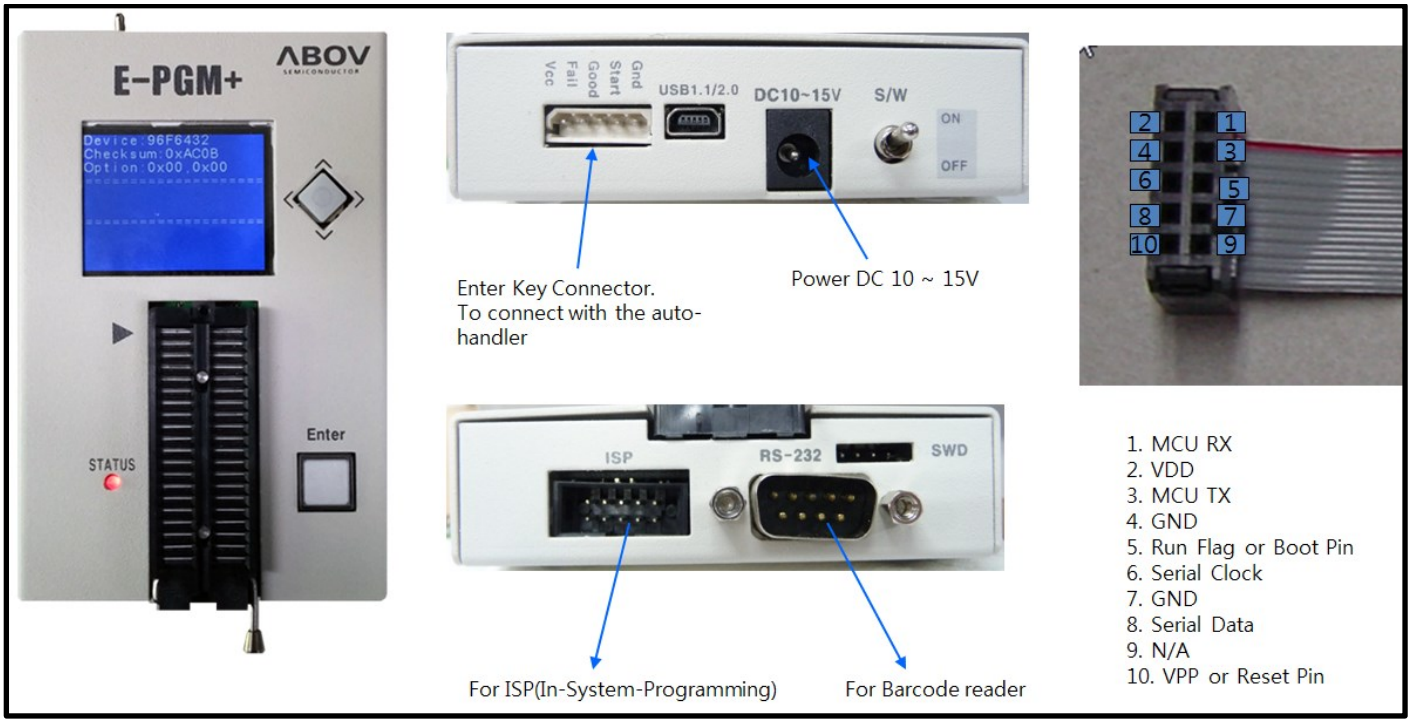


Figure 1.3 PGMplus USB

PGMPlusLC 2

Description

PGMPlusLC2 is for ISP (In System Programming). It is used to write into the MCU Which is already mounted on target board using 10pin cable.

Features

- PGMplusLC2 is low cost writing Tool.
- USB interface is supported.
- Not need USB driver installation.
- Connect the external power adaptor (5v@2A).
- Fast 32-bit Cortex-M3 MCU is used.
- Supported high voltage Max 18V.
- PGMplusLC2 is based on PC environment.
- PGMplusLC2 is faster than PGMplusLC.
- Transmission speed is 64Kbyte/s

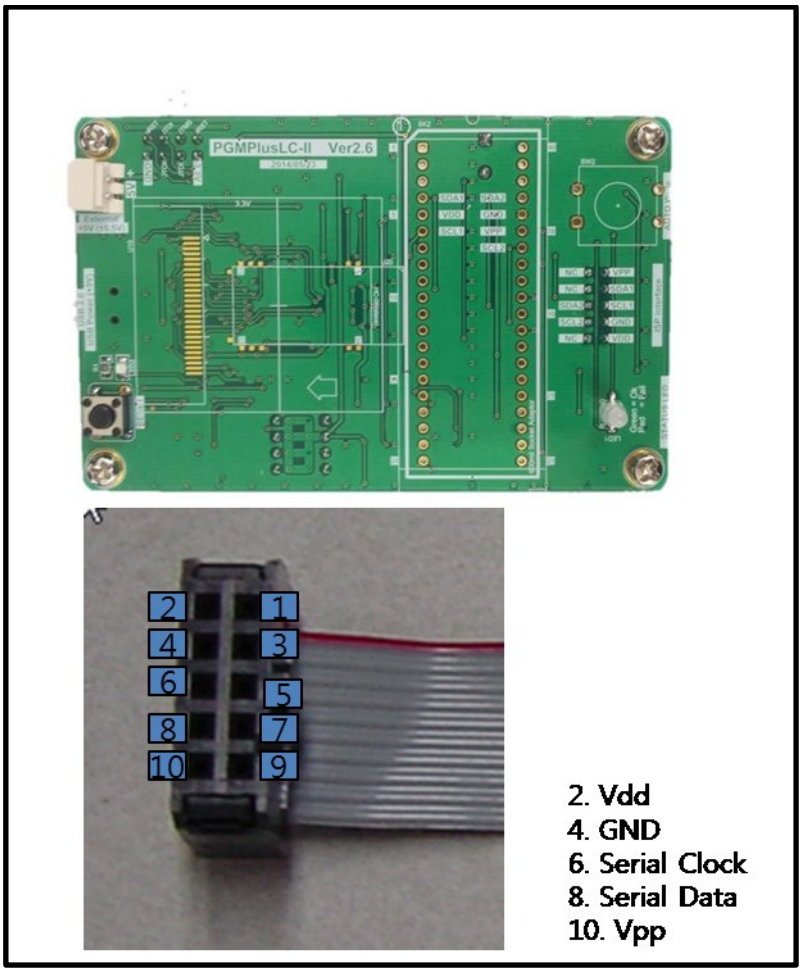


Figure 1.4 PGMplusLC Writer

E-PGM+ Gang4/6

- Product name : **E-PGM+ GANG 4**
- Dimension(x , y, h) : 33.5 x 22.5 x35mm
- Weight : 2.0kg
- Input Voltage : DC Adaptor 15V/2A
- Power Consumption :
- Operating Temp : -10 ~ 40℃
- Storage Temp : -30 ~ 80℃
- Water Proof : No

- Product name : **E-PGM+ GANG 6**
- Dimension(x , y, h) : 148.2 x 22.5 x35mm
- Weight : 2.8kg
- Input Voltage : DC Adaptor 15V/2A
- Power Consumption :
- Operating Temp : -10 ~ 40℃
- Storage Temp : -30 ~ 80℃
- Water Proof : No

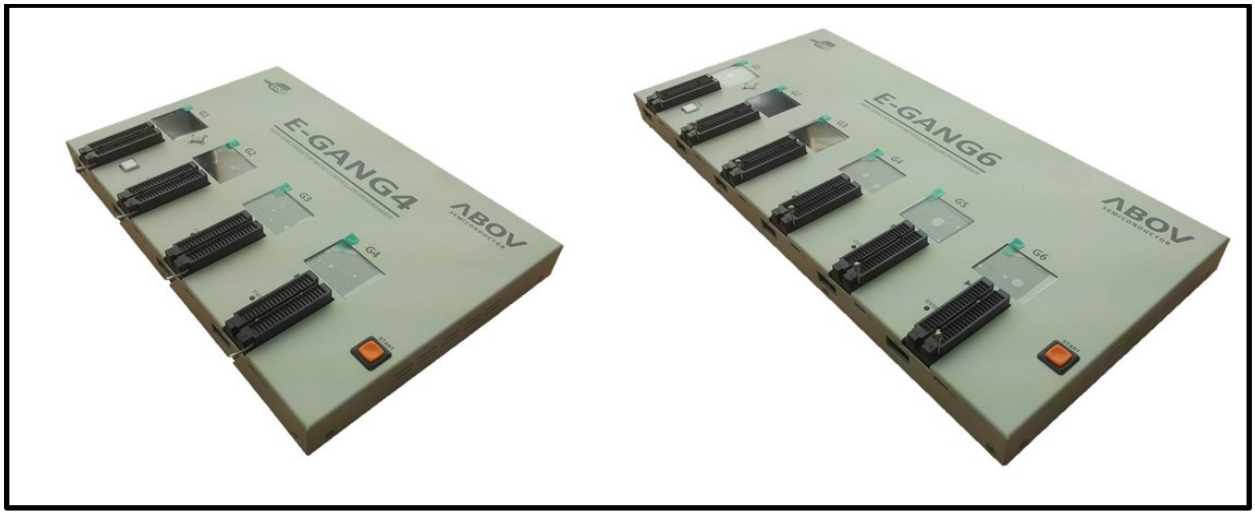


Figure 1.5 Gang Programmer

2 Block diagram

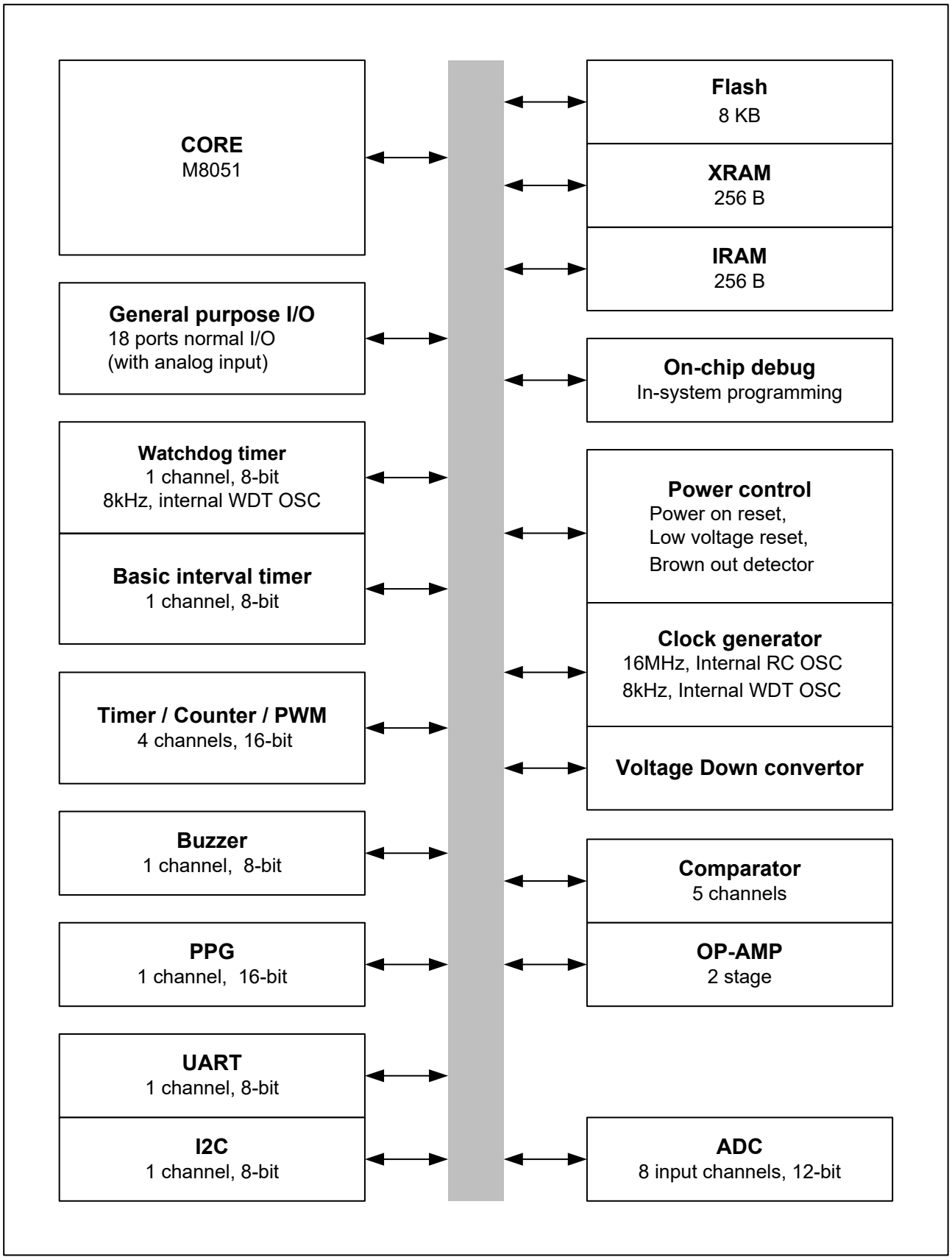


Figure 2.1 Block diagrom of MC97F6108A

3 MC97F6108A pin assignment

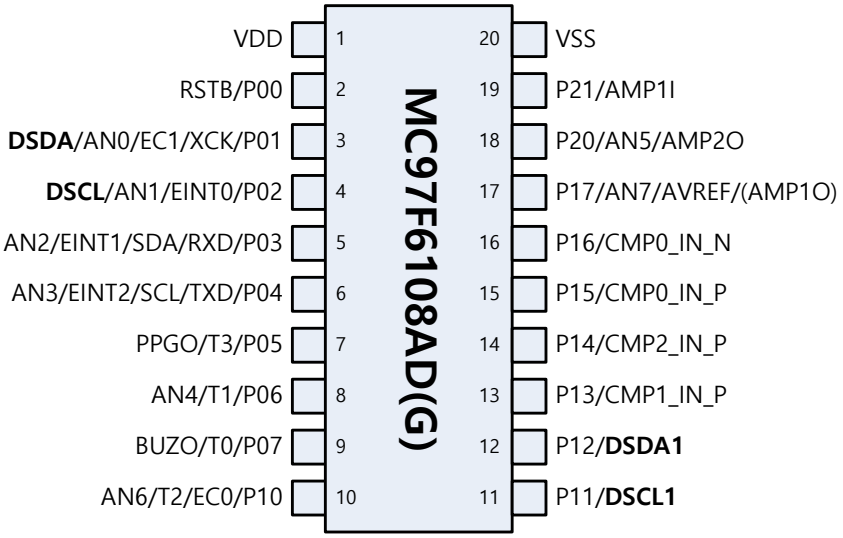


Figure 3.1 Pin assignment (20 SOP)

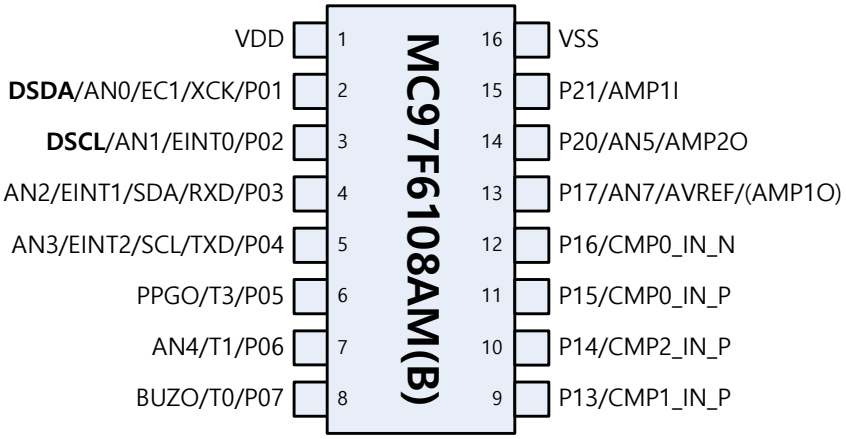


Figure 3.2 Pin assignment (16 SOPN)

Monitor function: P01/CMPXO, P04/CLKO, P07/TPPGO

USART function: P01/XCK, P04/TXD, P03/RXD

SPI function: P02/SS, P04/MOSI, P03/MISO, P01/SCK (when USART is used as spi mode)

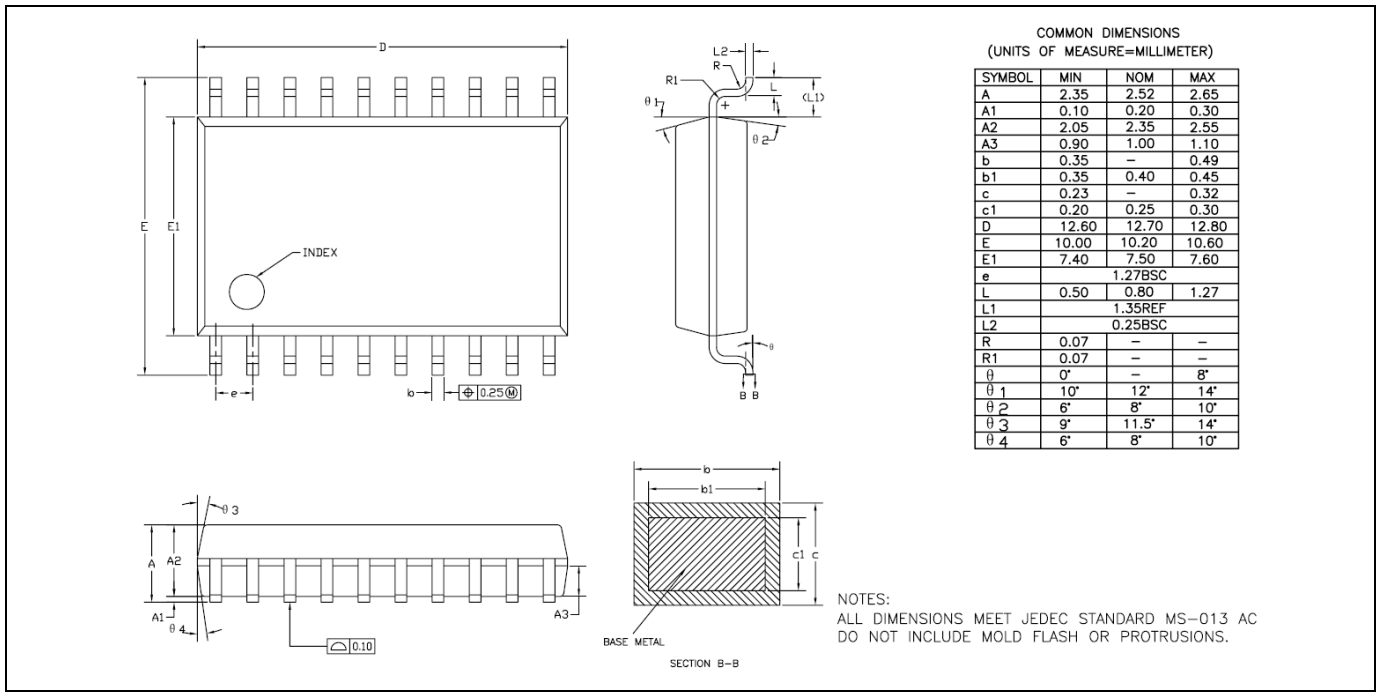
I2C function: P03/SDA, P04/SCL

Note)

1. When using 16-pin products, it is recommended that configure internal pull-up to the floating pin in order to prevent current consumption.
2. Second functions of DSDA and DSCL port are not supported in OCD mode. (EC1, XCK, EINT0)

4 Package Diagram

4.1 20 SOP

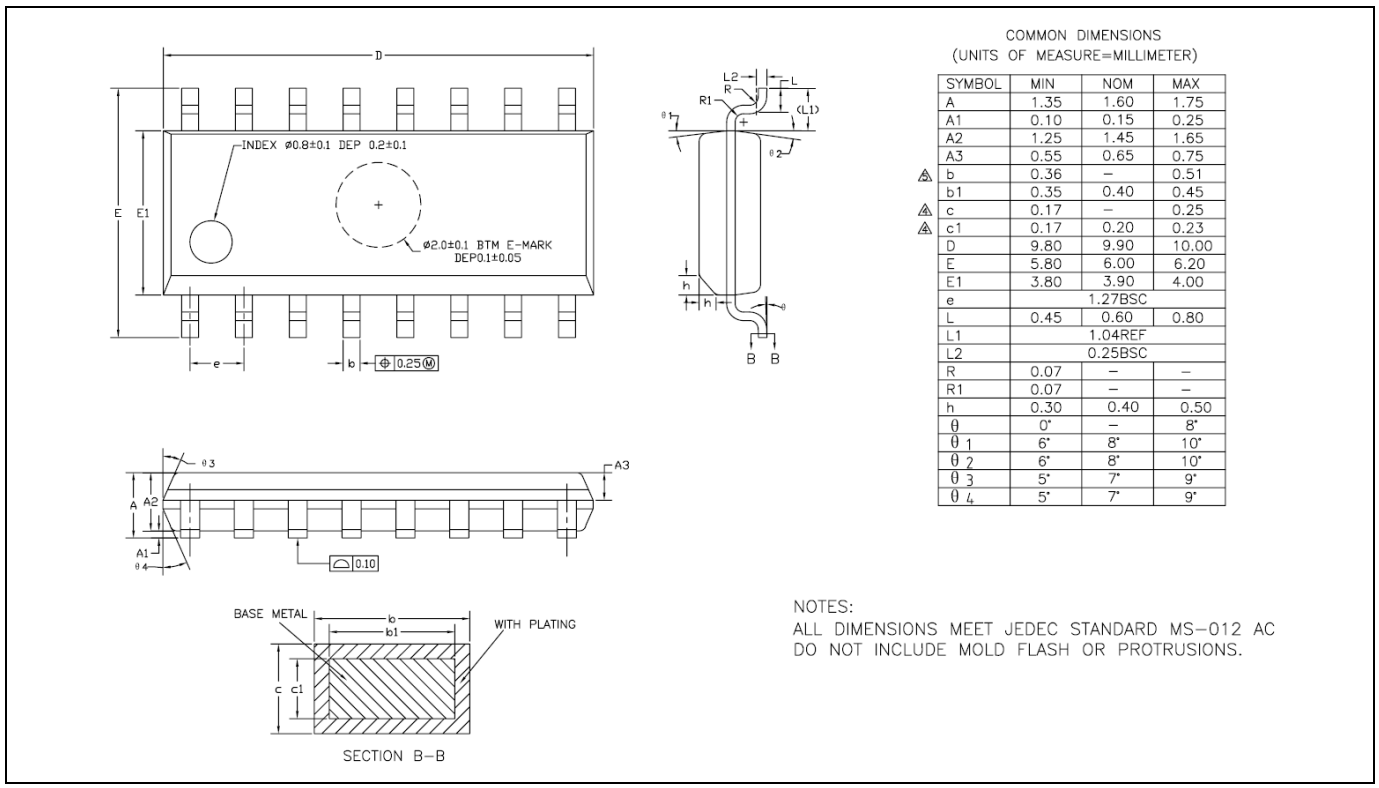


COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	2.35	2.52	2.65
A1	0.10	0.20	0.30
A2	2.05	2.35	2.55
A3	0.90	1.00	1.10
b	0.35	–	0.49
b1	0.35	0.40	0.45
c	0.23	–	0.32
c1	0.20	0.25	0.30
D	12.60	12.70	12.80
E	10.00	10.20	10.60
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.50	0.80	1.27
L1	1.35REF		
L2	0.25BSC		
R	0.07	–	–
R1	0.07	–	–
θ	0°	–	8°
θ1	10°	12°	14°
θ2	6°	8°	10°
θ3	9°	11.5°	14°
θ4	6°	8°	10°

Figure 4.1 20-pin SOP package

4.2 16 SOPN



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	1.35	1.60	1.75
A1	0.10	0.15	0.25
A2	1.25	1.45	1.65
A3	0.55	0.65	0.75
⚠4 b	0.36	–	0.51
b1	0.35	0.40	0.45
⚠4 c	0.17	–	0.25
⚠4 c1	0.17	0.20	0.23
D	9.80	9.90	10.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
L	0.45	0.60	0.80
L1	1.04REF		
L2	0.25BSC		
R	0.07	–	–
R1	0.07	–	–
h	0.30	0.40	0.50
θ	0°	–	8°
θ1	6°	8°	10°
θ2	6°	8°	10°
θ3	5°	7°	9°
θ4	5°	7°	9°

Figure 4.2 16-pin SOPN package

5 Pin Description

PIN Name	I/O	Function	@RESET	Shared with
P00	I/O	Port P0 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	RSTB
P01				AN0/EC1/XCK/CMPXO/DSDA
P02				AN1/EINT0/DSCL
P03				AN2/EINT1/SDA/RXD
P04				AN3/EINT2/SCL/TXD/CLKO
P05				PPGO/T3
P06				AN4/T1
P07				BUZO/T0/TPPGO
P10	I/O	Port P1 8-Bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	AN6/T2/EC0
P11				DSCL1
P12				DSDA1
P13				CMP1_IN_P
P14				CMP2_IN_P
P15				CMP0_IN_P
P16				CMP0_IN_N
P17				AN7/AVREF/AMP1O
P20	I/O	Port P2 2bit I/O Port Can be set in input or output mode in 1-bit units Internal pull-up register can be used via software when this port is used as input port Open Drain enable register can be used via software when this port is used as output port	Input	AN5/AMP2O
P21				AMP1I

Table 5-1. Pin Description

6 Port Structures

6.1 General Purpose I/O Port

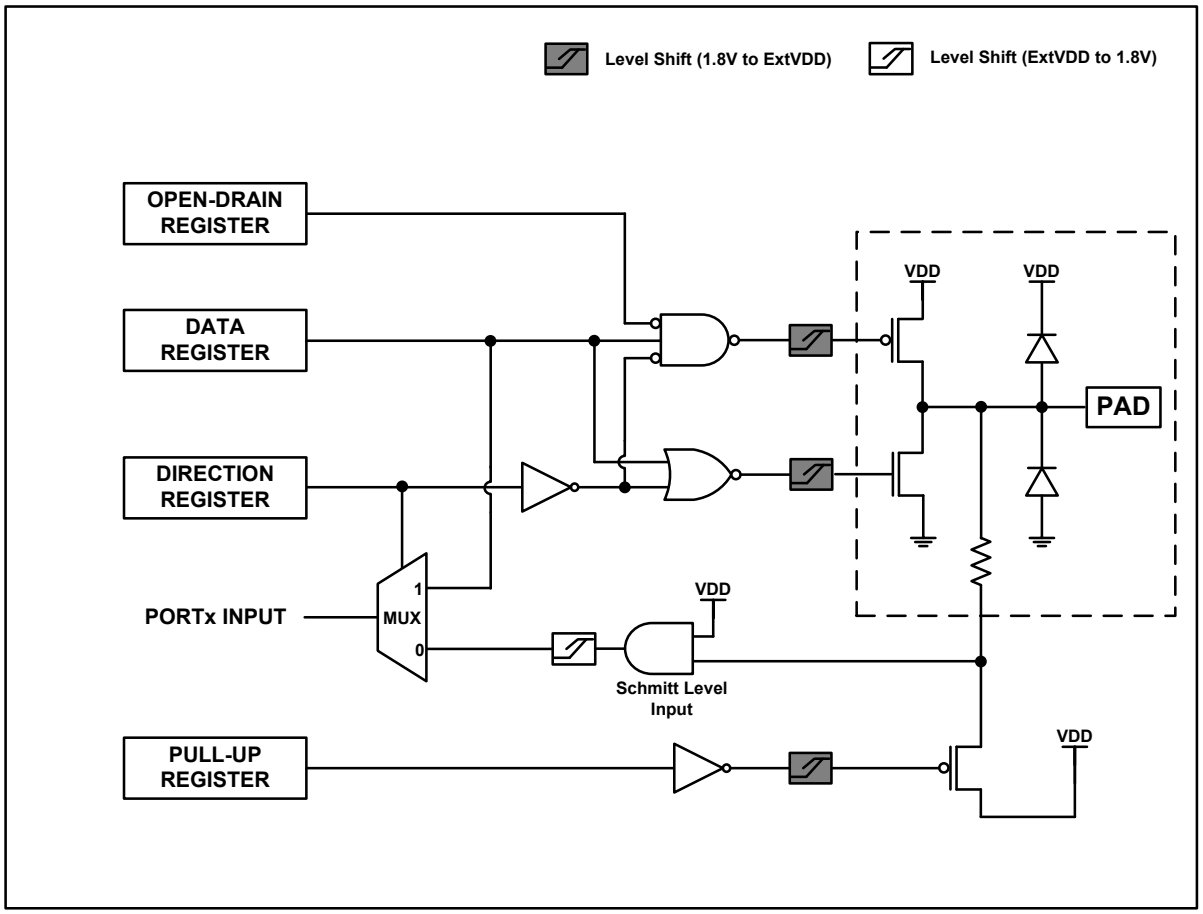


Figure 6.1 General Purpose I/O Port

6.2 Second Function I/O Port

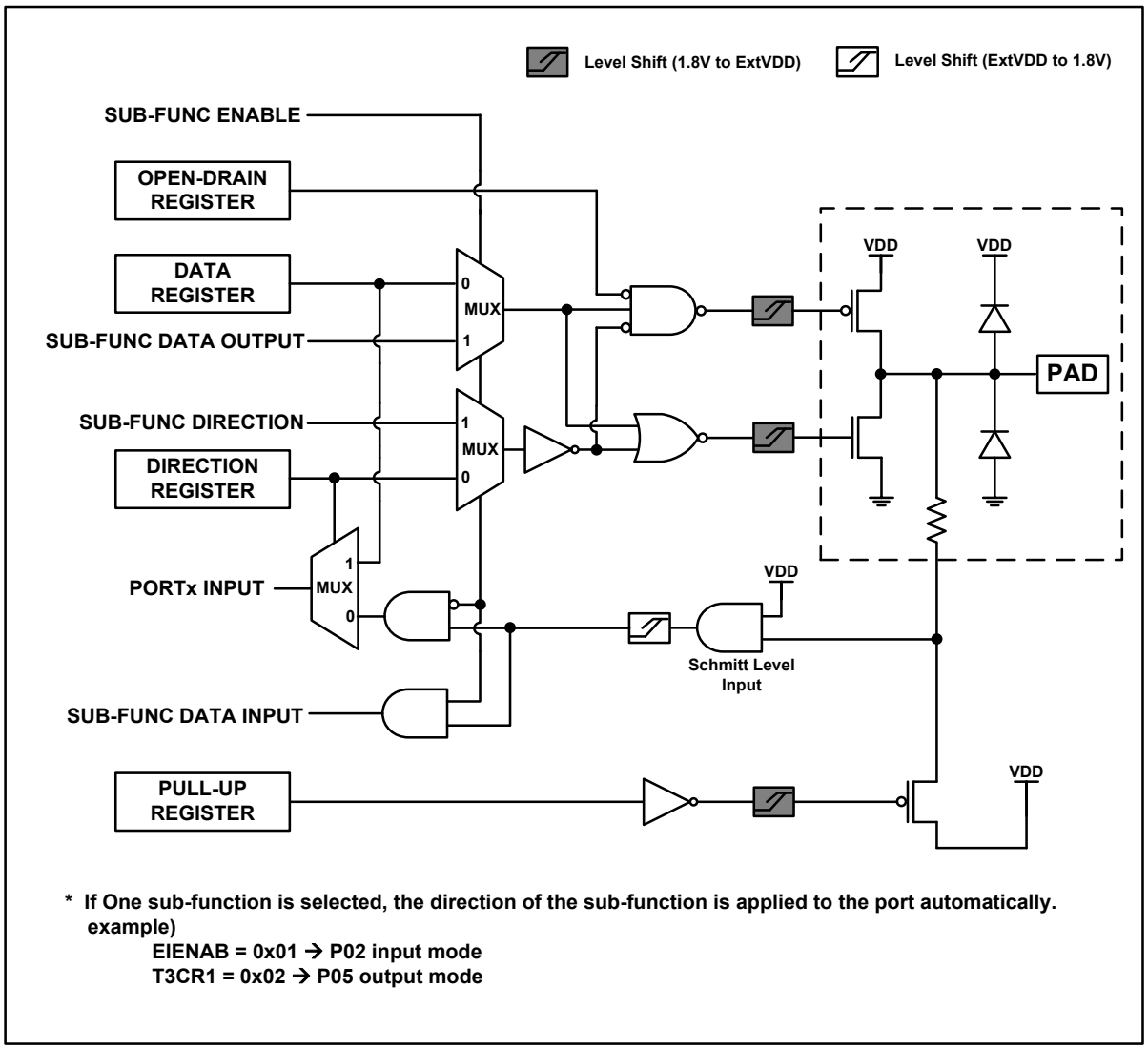


Figure 6.2 Second Function I/O Port

6.3 Analog Input I/O Port

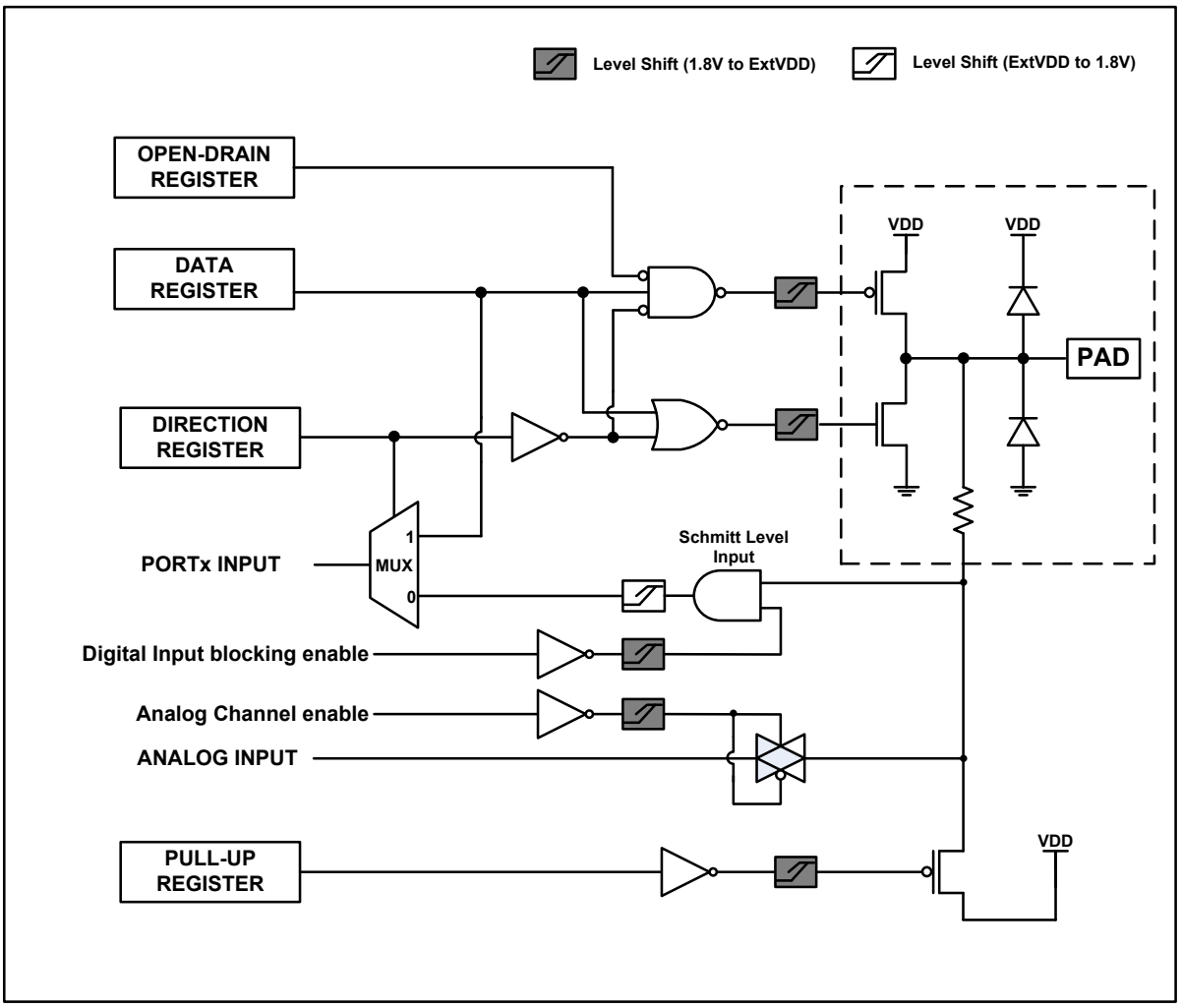


Figure 6.3 Analog Input I/O Port

7 Electrical Characteristics

7.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply Voltage	VDD	-0.3~+6.5	V
	VSS	-0.3~+0.3	V
Normal Voltage Pin	V_I	-0.3~VDD+0.3	V
	V_O	-0.3~VDD+0.3	V
	I_{OH}	-25	mA
	$\sum I_{OH}$	-160	mA
	I_{OL}	45	mA
	$\sum I_{OL}$	320	mA
Total Power Dissipation	P_T	600	mW
Storage Temperature	T_{STG}	-65~+150	℃

Table 7-1. Absolute Maximum Ratings

Note) Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Supply Voltage	VDD	f_{IRC}=2, 4, 8, 16MHz	2.7	-	5.5	V
Operating Temperature	T_{OPR}	-	-40	-	85	°C
		Internal RC-OSC	15.52	16	16.48	MHz
		Internal WDT-OSC	4	8	12	kHz

Table 7-2. Recommended Operation Conditions

7.3 Internal RC Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	-	+85	°C
Frequency		25°C	15.84	16	16.16	MHz
		-20°C ~ +70°C	15.68	16	16.32	MHz
		-40°C ~ +85°C	15.52	16	16.48	MHz
Stabilization Time		-	-	1	-	ms
Operating Current	I_{DD}	-	-	400	-	uA

Table 7-3. Internal RC Oscillator Characteristics

7.4 Internal WDT Oscillator Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	-	+85	°C
Frequency		-	4	8	12	kHz
Stabilization Time		-	-	1	-	ms
Operating Current	I_{DD}	-	-	5	-	uA

Table 7-4. Ring-Oscillator Characteristics

7.5 Voltage Dropout Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	2.7	-	5.5	V
Operating Temperature		-	-40	-	+85	°C
Regulation Voltage		-	1.62	1.8	1.98	V
Current Drivability		-	-	20	-	mA
Operating Current	I_{DD1}	RUN/IDLE	-	0.5	-	mA
	T_{TRAN}	STOP to RUN	-	200	-	ns
		STOP1	-	0.5	-	uA
		STOP2	-	0.5	-	uA

Table 7-5. Voltage Dropout Converter Characteristics

Note) STOP1: WDTRC running- STOP2: WDTRC disable

7.6 A/D CONVERTER CHARACTERISTICS

(T_A = - 40°C to + 85°C, V_{DD} = 2.7 – 5.5V, Vss=0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Resolution	–	–		–-	12	–	bit
Integral Non-Linearity	INL	AVREF= 2.7V – 5.5V fx= 8MHz		–	–	±4	LSB
Differential Non-Linearity	DNL			–	–	±1	
Zero Offset Error	ZOE			–	–	±7	
Overall Accuracy	OA			–	–	±3	
Conversion Time	t_{CON}	12-bit resolution, 8MHz		–	60	–	Cycle
Analog Input Voltage	V_{AN}	–		VSS	–	AVREF	V
Analog Reference Voltage	AVREF	–		2.7[NOTE]	–	VDD	
Analog Input Leakage Current	I_{AN}	AVREF=5.12V		–	–	2	uA
ADC Operating Current	I_{ADC}	Enable	VDD=5.12V	–	1	2	mA
		Disable		–	–	0.1	uA

Table 7-6. A/D Converter Characteristics

Note) 1. Zero offset error is the difference between 000000000000 and the converted output for zero input voltage (VSS);
2. Full scale error is the difference between 111111111111 and the converted output for full-scale input voltage (VDD).
3. Under 2.7V, the ADC resolution decrease.

7.7 Low Voltage Reset Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
LVR Voltage	V_{LVR}	-	-	1.8	-	V
VDD voltage rising time	t_R	-	-	50	-	mV
VDD voltage off time	t_{OFF}	-	-	-	50	uA
Hysteresis voltage of LVR	△V					
Current consumption	I_{LVR}					

Table 7-7. Low Voltage Reset Characteristics

7.8 Brown Out Detector Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Detection Level	BOD6	-	3.8	4.2	4.6	V
	BOD5	-	3.3	3.7	4.1	V
	BOD4	-	2.8	3.2	3.6	V
	BOD3	-	2.3	2.7	3.1	V
	BOD2		2.1	2.5	2.9	V
	BOD1		1.8	2.2	2.6	V
Hysteresis		-	-	50	-	mV
Operating Current	I_{DD}	-	-	-	50	uA

Table 7-8. Brown Out Detector Characteristics

7.9 DC Characteristics

(VDD =5.5V, VSS =0V, TA=-40~+85℃)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input Low Voltage	V_{IL}	P0, P1, P2, P3	0	-	0.2VDD	V
Input High Voltage	V_{IH}	P0, P1, P2, P3	0.8VDD	-	VDD	V
Output Low Voltage	V_{OL}	ALL I/O (IOL=40mA)	-	-	1	V
Output High Voltage	V_{OH5}	ALL I/O (IOH=-20mA) @ VDD=5V	VDD-1.5	-	-	V
	V_{OH3}	ALL I/O (IOH=-10mA) @ VDD=3.3V	VDD-1.5	-	-	V
Input High Leakage Current	I_{IH}	ALL PAD	-1	-	1	uA
Input Low Leakage Current	I_{IL}	ALL PAD	-1	-	1	uA
Pull-Up Resister	R_{PU}	ALL PAD	25	50	75	kΩ
Power Supply Current	I_{DD}	Run Mode, f_{IRC}=8MHz @5V	-	3	5	mA
	I_{IDLE}	IDLE Mode, f_{IRC}=8MHz @5V	-	2	5	mA
	I_{STOP1}	STOP1 Mode, WDTRC Enable @5V	-	10	-	uA
	I_{STOP2}	STOP2 Mode, WDTRC Disable @5V	-	5	-	uA

Table 7-9. DC Characteristics

NOTE)

1. STOP1: WDT only running, STOP2: All function disable.

7.10 Power-On Reset Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
RESET Release Level		-	1.2	1.4	1.6	V
Operating Current	I_{DD}	-	-	0.1	-	uA
VDD Rise Rate	V_{RR}	-	0.05	-	5	V/ms

Table 7-10. Power-On Reset Characteristics

7.11 AC Characteristics

(VDD=5.0V±10%, VSS=0V, TA=-40~+85℃)

Parameter	Symbol	PIN	MIN	TYP	MAX	Unit
Operating Frequency	fMCP	-	2	-	16	MHz
System Clock Cycle Time	tSYS	-	500	-	62.5	ns
Oscillation Stabilization Time (16MHz)	tMST1	-	-	-	8	ms
External Interrupt Input Width	tIW	INT0~INTx	2	-	-	tSYS
External Interrupt Transition Time	tFI,tRI	INT0~INTx			1	us
nRESET Input Pulse “L” Width	tRST	nRESET	8	-	-	us
External Counter Input “H” or “L” Pulse Width	tECW	EC0~ECx	2	-	-	tSYS
Event Counter Transition Time	tREC,tFEC	EC0~ECx	-	-	20	ns

Table 7-11. AC Characteristics

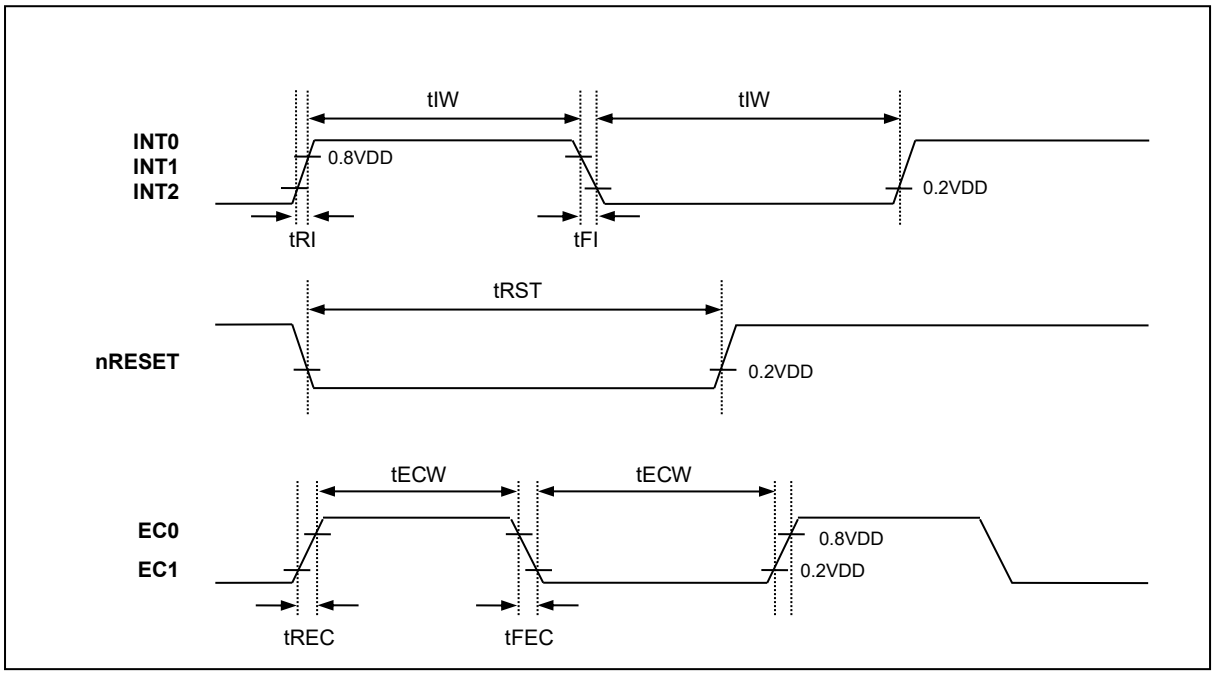


Figure 7.1 AC Timing

7.12 Analog comparator characteristics

Parameter	Condition	Min	Typ	Max	Unit
Operating Voltage (VDD)		2.4		5.5	V
I_{DD}		150		200	uA
Power Down Current				0.1	uA
Input Offset Voltage	Without calibration	-15		+15	mV
	With calibration Comparator 0	-4		+4	mV
	With calibration Comparator 1~4	-10		+10	mV
Input common Mode Voltage Range		VSS		VDD-0.2V	V
Response Time	Hysteresis disable and Input step with 5mV overdrive	0.5		1	us
Hysteresis	Comparator hysteresis enable	20	40	60	mV

Table 7-12. analog comparator dc characteristics

7.13 Operational amplifier characteristics

Parameter	Condition	Min	Typ	Max	Unit
Operating Voltage (VDD)		2.8		5.5	V
Power Down Current				0.1	uA
Input Offset Voltage	Without calibration	-15		+15	mV
	With calibration by 4x5 gain	-4		+4	mV
Common Mode Voltage Range		VSS		VDD-0.2V	V
Power Supply Rejection Ratio		60	90		dB
Common Mode Rejection Ratio	Vcm=0~VDD-0.1V	60	90		dB
Slew Rate +, Slew Rate -	No load	4	5		V/us
Gain Band Width	CL=100pF		3		MHz
Programmable DC Gain	AMP1 : x2.5, x3, x3.53, x4, x4.62, x5, x6, x8				
	AMP2 : x1, x1.5, x2, x2.5, x3, x4, x5, x6				

Table 7-13. operational amplifier characteristics

7.14 Typical Characteristics

These graphs and tables provided in this section are for design guidance only and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

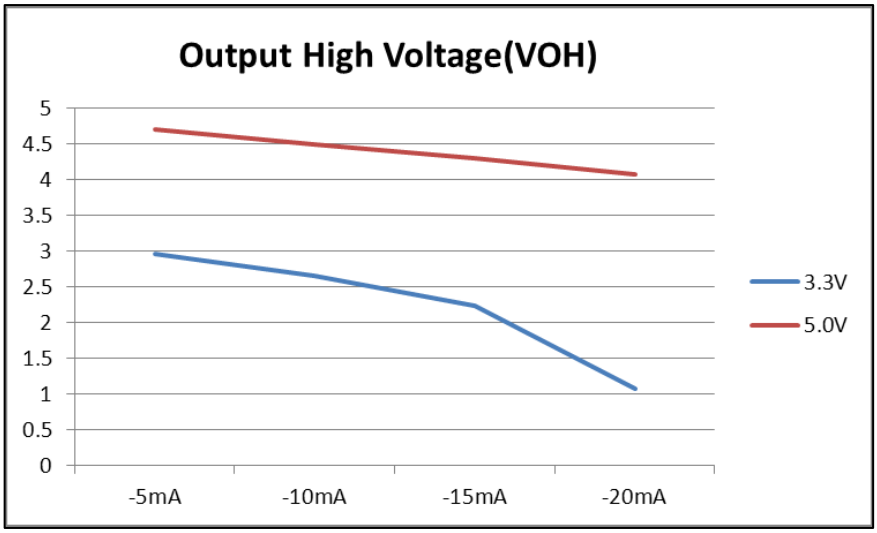


Figure 7.2 Output High Voltage (VOH)

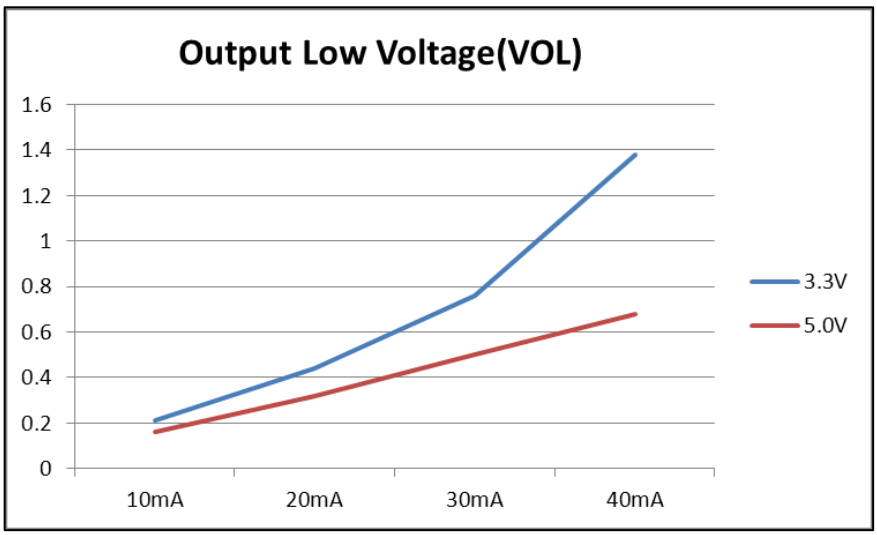


Figure 7.3 Output Low Voltage (VOL)

8 Memory

The MC97F6108A addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by 8-bit CPU. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

Program memory can only be read, not written to providing up to 8Kbytes of Program memory on-chip. Data memory can be read and written to up to 256bytes internal memory (IRAM) including the stack area and 256bytes of external memory (XRAM).

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64Kbytes for one bank of memory space, but this device has 8Kbytes program memory space.

Figure 1.1 shows a map of program memory. After reset, the CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine.

External interrupt 0, for example, is assigned to location 0003H. If external interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8bytes unit. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

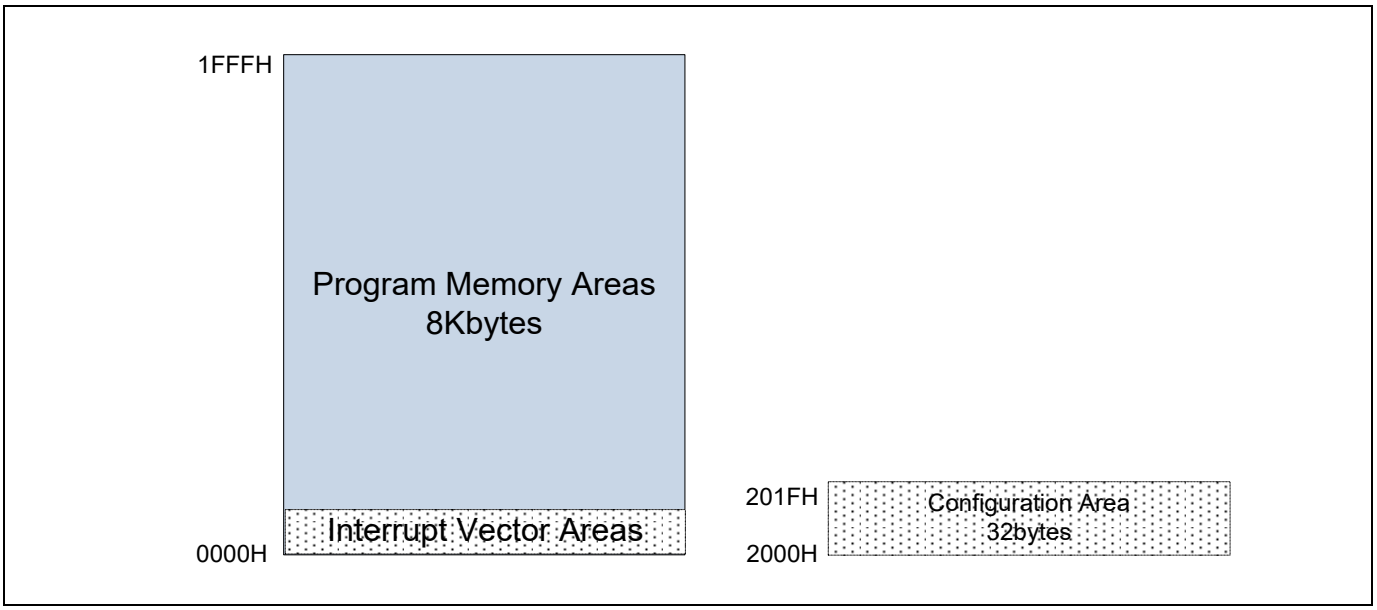


Figure 1.1 Program Memory

- User Function Mode: 8Kbytes Program Memory Area included Interrupt Vector Region
- Non-volatile and reprogramming memory: FLASH memory based on EEPROM cell

8.2 Data Memory

Figure 1.2 shows the external and internal Data memory space available.

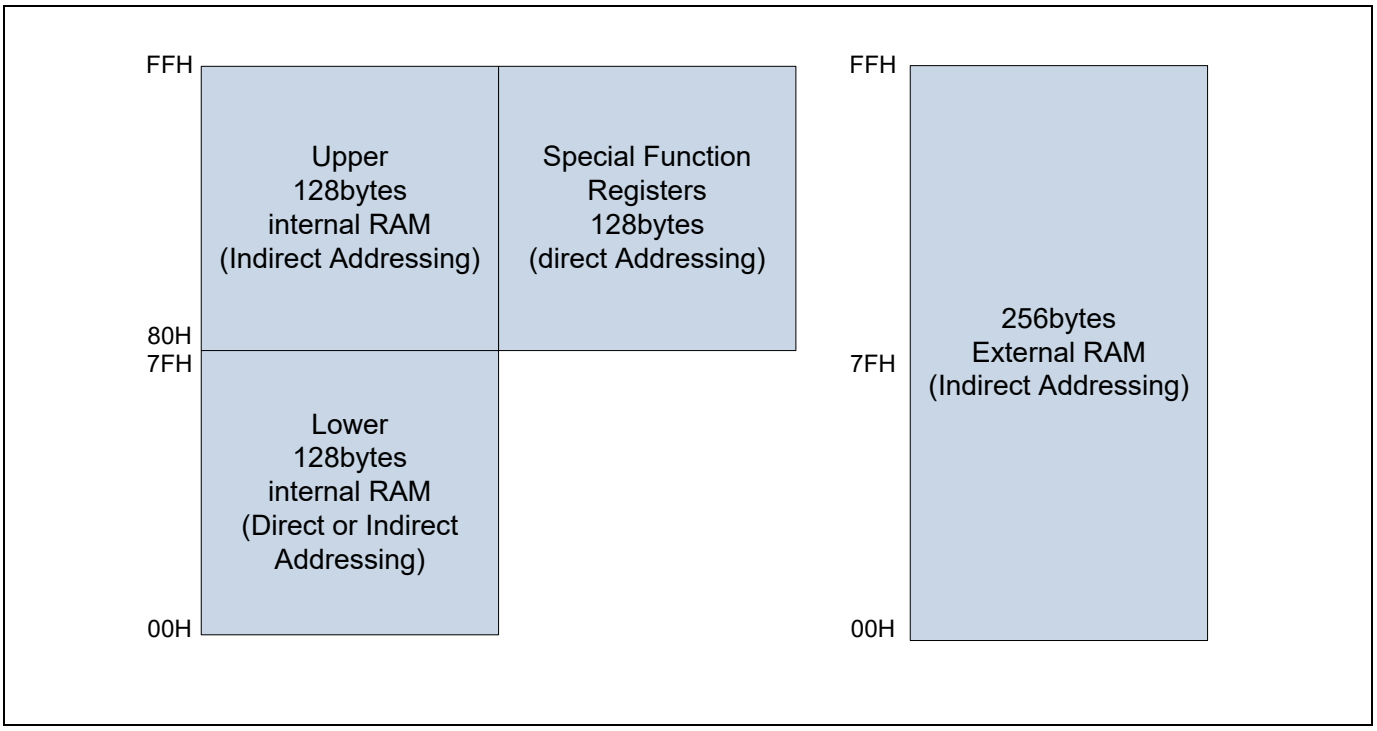


Figure 1.2 Data Memory map

There are 256bytes external memory space. Only indirect addressing can access the external data memory by using movx instruction.

The internal memory space is divided into three blocks, which are generally referred to as the lower 128, upper 128, and SFR space.

Internal Data memory addresses are always one byte wide, which implies an address space of only 256bytes. However, the addressing modes for internal RAM can in fact accommodate 384bytes, using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. Thus Figure 1.2 shows the upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128bytes of RAM are present in all 8051 devices as mapped in Figure 1.3. The lowest 32bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient used of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16bytes above the register banks form a block are bit-addressable memory space. The 8051 instruction set includes a wide selection of single-bit instructions, and the 128bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the lower 128 can be accessed by either direct or indirect addressing. These spaces are used for user RAM and stack pointer. The upper 128bytes RAM can only be accessed by indirect addressing.

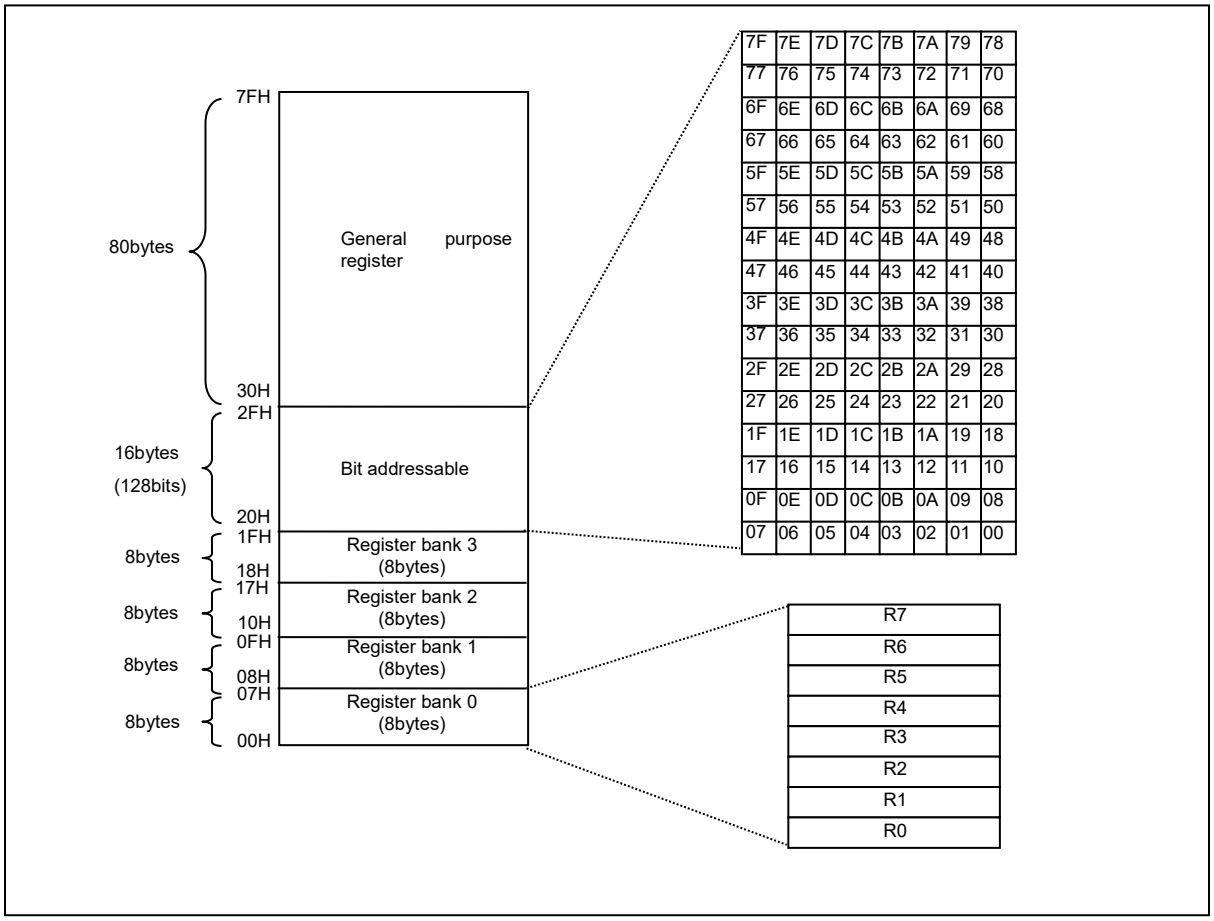


Figure 1.3 Low 128bytes RAM

8.3 XSFR

MC97F6108A has 256bytes XSFR. This area has no relation with RAM/FLASH. It can read and write through SFR with 8-bit unit.

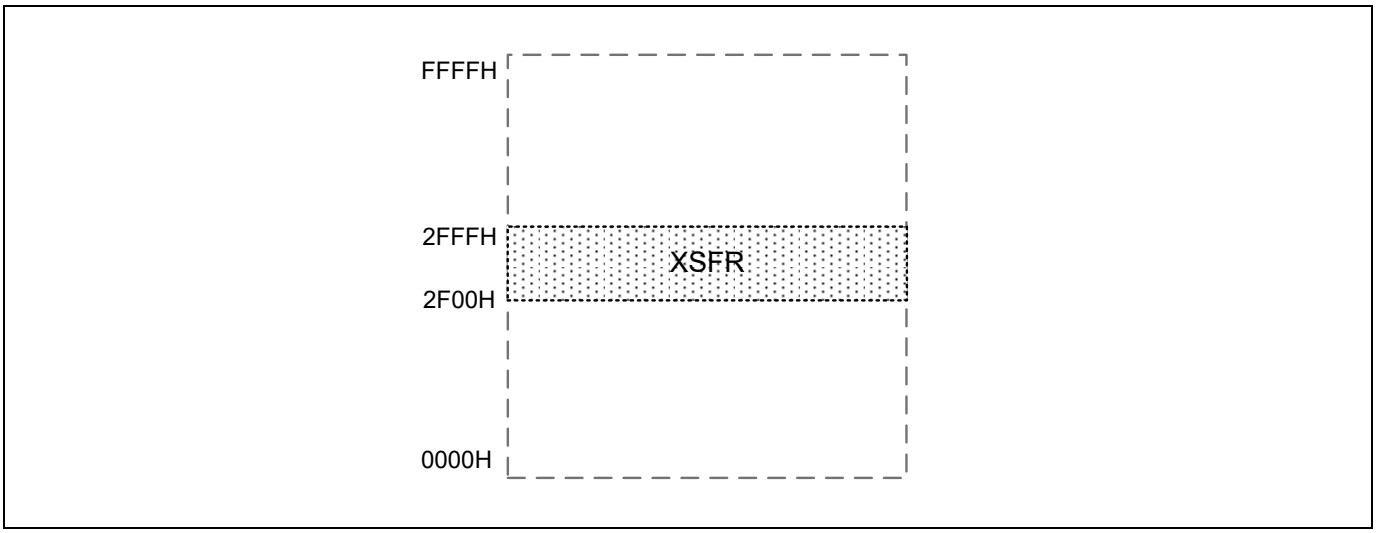


Figure 1.4 XDATA Memory area

8.4 SFR Map

8.4.1 SFR Map Summary

Addr	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
2F78								
2F70								
2F68								
2F60								
2F58								
2F50								
2F48	PSR0		PSR2	PSR3				
2F40								
2F38			CA_REGA	CA_REGB	CA_REGC			
2F30	CA_REG0	CA_REG1	CA_REG2	CA_REG3	CA_REG4	CA_REG5	CA_REG6	CA_REG7
2F28								
2F20								
2F18								
2F10	P2PU	P2OD	P2DB					
2F08	P1PU	P1OD	P1DB		OFFMAXLR	OFFMAXHR	OFFMINLR	OFFMINHR
2F00	P0PU	P0OD	P0DB		ATPMAXLR	ATPMAXHR	ATPMINLR	ATPMINHR
F8H	-	ATPCR	UCTRL1	UCRTL2	UCTRL3	USTAT	UBAUD	UDATA
F0H	B	ATPHR	FEARL	FEARM	FEARH	FEDR	FETR	
E8H	-	ATPLR	FEMR	FECR	FESR	FETCR	BUZCR	BUZDR
E0H	ACC	PPGCR2	PPGCR	PPGCR1	PPGDL	PPGDH	PPGPL	PPGPH
D8H	CFENAB	PPGPXH	I2CMR	I2CSR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR
D0H	PSW	PPGPXL		PPGCL/PPGL	PPGCH/PPGH	TMISR	I2CSAR1	I2CSAR
C8H	CFFLAG	DSTEP	T3CR	T3CR1	PWM3DRL CDR3L / T3L	PWM3DRH CDR3H / T3H	PWM3PRL T3DRL	PWM3PRH T3DRH
C0H	CFEDGE	USTEP	T2CR	T2CR1	PWM2DRL CDR2L / T2L	PWM2DRH CDR2H / T2H	PWM2PRL T2DRL	PWM2PRH T2DRH
B8H	-	OFFCR	T1CR	T1CR1	PWM1DRL CDR1L / T1L	PWM1DRH CDR1H / T1H	PWM1PRL T1DRL	PWM1PRH T1DRH
B0H	CFPOLA	CIENAB	T0CR	T0CR1	PWM0DRL CDR0L / T0L	PWM0DRH CDR0H / T0H	PWM0PRL T0DRL	PWM0PRH T0DRH
A8H	IE	IE1	IE2	IE3	CIFLAG	CIEDGE	CIPOLA	CIBOTH
A0H	CFBOTH		EO	EIENAB	EIFLAG	EIEDGE	EIPOLA	EIBOTH
98H		P2IO	IP1	IP1H	IP2	IP2H	IP3	IP3H
90H	P2	P1IO	IP	IPH	PCI	ADCM	ADCM1 /ADCRL	ADCRH
88H	P1	P0IO	SCCR	BCCR	BITR	WDTMR	WDTR /WDTCR	BODR
80H	P0	SP	DPL	DPH	DPL1	DPH1	RSFR	PCON

Table 8-1. SFR Map Summary

8.4.2 Compiler Compatible SFR

ACC (Accumulator) : E0H

7	6	5	4	3	2	1	0
ACC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ACC Accumulator

B (B Register) : F0H

7	6	5	4	3	2	1	0
B							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

B B Register

SP (Stack Pointer) : 81H

7	6	5	4	3	2	1	0
SP							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 07H

SP Stack Pointer

DPL (Data Pointer Low Byte) : 82H

7	6	5	4	3	2	1	0
DPL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPL Data Pointer Low Byte

DPH (Data Pointer High Byte) : 83H

7	6	5	4	3	2	1	0
DPH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPH Data Pointer High Byte

DPL1 (Data Pointer1 Low Byte) : 84H

7	6	5	4	3	2	1	0
DPL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPL1 Data Pointer1 Low Byte

DPH1 (Data Pointer1 High Byte) : 85H

7	6	5	4	3	2	1	0
DPH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPH1 Data Pointer1 High Byte

PSW (Program Status Word) : D0H

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CY	Carry Flag
AC	Auxiliary Carry Flag
F0	General Purpose User-Definable Flag
RS1	Register Bank Select bit 1
RS0	Register Bank Select bit 0
OV	Overflow Flag
F1	User-Definable Flag
P	Parity Flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of ‘1’ bits in the accumulator

EO (Extended Operation Register) : A2H

7	6	5	4	3	2	1	0
-	-	-	TRAP_EN	-	-	-	DPSEL.0
R	R	R	R/W	R	R	R	R/W

Initial value : 00H

TRAP_EN Select the instruction

0 Select MOVC @(DPTR++), A

1 Select Software TRAP instruction

DPSEL Select Banked Data Point Register

0 DPTR = {DPH, DPL}

1 DPTR1 = {DPH1, DPL1}

9 Input/Ouput Ports

9.1 Input/Output Ports

The MC97F6108A has 18 I/O ports (P0, P1, P2). Each port can be easily configured by software as I/O pin, internal pull up and open drain pin to meet various system configurations and design requirements.

Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	89H	R/W	00H	P0 Direction Register
P0PU	2F00H	R/W	20H	P0 Pull-up Resistor Selection Register
P0OD	2F01H	R/W	00H	P0 Open-drain Selection Register
P0DB	2F02H	R/W	00H	P0 De-bounce Enable Register
P1	88H	R/W	00H	P1 Data Register
P1IO	91H	R/W	00H	P1 Direction Register
P1PU	2F08H	R/W	00H	P1 Pull-up Resistor Selection Register
P1OD	2F09H	R/W	00H	P1 Open-drain Selection Register
P1DB	2F0AH	R/W	00H	P1 De-bounce Enable Register
P2	90H	R/W	00H	P2 Data Register
P2IO	99H	R/W	00H	P2 Direction Register
P2PU	2F10H	R/W	00H	P2 Pull-up Resistor Selection Register
P2OD	2F11H	R/W	00H	P2 Open-drain Selection Register
P2DB	2F12H	R/W	00H	P2 De-bounce Enable Register
PSR0	2F48H	R/W	00H	Port De-bounce selection register
PSR2	2F4AH	R/W	00H	Analog I/O port selection register
PSR3	2F4BH	R/W	00H	Analog I/O port selection register

Table 9-1. Register Map

9.1.1 Data Register (Px)

Data Register is a bidirectional I/O port. If ports are configured as output ports, data can be written to the corresponding bit of the Px. If ports are configured as input ports, the data can be read from the corresponding bit of the Port Px.

9.1.2 Direction Register (PxIO)

Each I/O pin can independently used as an input or an output through the PxIO register. Bits cleared in this read/write register will select the corresponding pin in Px to become an input, setting a bit sets the pin to output. All bits are cleared by a system reset.

9.1.3 Pull-up Resistor Selection Register (PxPU)

The on-chip pull-up resistor can be connected to them in 1-bit units with a pull-up resistor selection register (PxPU). The pull-up register selection controls the pull-up resister enable/disable of each port. When the corresponding bit is 1, the pull-up resister of the pin is enabled. When 0, the pull-up resister is disabled. All bits are cleared by a system reset.

9.1.4 Open-drain Selection Register (PxOD)

The open-drain selection register controls the open-drain enable/disable of each port. Ports become push-pull by a system reset. You should connect an internal resistor or an external resistor in open-drain output mode.

9.1.5 De-bounce Enable Register (PxDB)

P0 ~ P3 support de-bounce function. De-bounce time of each port has 1/2/4/8us

9.1.6 Port Selection Register (PSR0, PSR2, PSR3)

PSR0 : Port de-bounce selection register can select one of four de-bounce length of all port.
PSR2, PSR3 : Digital Input port selection register can select use of port's Digital input or not.

9.2 PORT P0

P0 (P0 Data Register) : 80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0[7:0] I/O Data

P0IO (P0 Direction Register) : 89H

7	6	5	4	3	2	1	0
P07IO	P06IO	P05IO	P04IO	P03IO	P02IO	P01IO	P00IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0IO[7:0] P0 data I/O direction.
0 Input
1 Output

P0PU (P0 Pull-up Resistor Selection Register) : 2F00H

7	6	5	4	3	2	1	0
P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 20H

P0PU[7:0] Configure pull-up resistor of P0 port
0 Disable
1 Enable

P0OD (P0 Open-drain Selection Register) : 2F01H

7	6	5	4	3	2	1	0
P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0OD[7:0] Configure open-drain of P0 port
0 Disable
1 Enable

P0DB (P0 De-bounce Enable Register) : 2F02H

7	6	5	4	3	2	1	0
P07DB	P06DB	P05DB	P04DB	P03DB	P02DB	P01DB	P00DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0DB[7:0] Configure de-bounce of P0 port
0 Disable
1 Enable

9.3 PORT P1

P1 (P1 Data Register) : 88H

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1[7:0] I/O Data

P1IO (P1 Direction Register) : 91H

7	6	5	4	3	2	1	0
P17IO	P16IO	P15IO	P14IO	P13IO	P12IO	P11IO	P10IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1IO[7:0] P1 data I/O direction.
0 Input
1 Output

P1PU (P1 Pull-up Resistor Selection Register) : 2F08H

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1PU[7:0] Configure pull-up resistor of P1 port
0 Disable
1 Enable

P1OD (P1 Open-drain Selection Register) : 2F09H

7	6	5	4	3	2	1	0
P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1OD[7:0] Configure open-drain of P1 port
0 Disable
1 Enable

P1DB (P1 De-bounce Enable Register) : 2F0AH

7	6	5	4	3	2	1	0
P17DB	P16DB	P15DB	P14DB	P13DB	P12DB	P11DB	P10DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1DB[7:0] Configure de-bounce of P1 port
0 Disable
1 Enable

9.4 PORT P2

P2 (P2 Data Register) : 90H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P21	P20
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

P2[1:0] I/O Data

P2IO (P2 Direction Register) : 99H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P21IO	P20IO
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

P2IO[1:0] P2 data I/O direction.
0 Input
1 Output

P2PU (P2 Pull-up Resistor Selection Register) : 2F10H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P21PU	P20PU
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

P2PU[1:0] Configure pull-up resistor of P2 port
0 Disable
1 Enable

P2OD (P2 Open-drain Selection Register) : 2F11H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P21OD	P20OD
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

P2OD[1:0] Configure open-drain of P2 port
0 Disable
1 Enable

P2DB (P2 De-bounce Enable Register) : 2F02H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P21DB	P20DB
-	-	-	-	-	-	R/W	R/W

Initial value : 00H

P2DB[1:0] Configure de-bounce of P2 port
0 Disable
1 Enable

9.5 PORT SELECT REGISTER PSR0, 2, 3

PSR0 (Port De-bounce selection register) : 2F48H

7	6	5	4	3	2	1	0
-	-	-	-	PSR03	PSR02	PSR01	PSR00
-	-	-	-	R/W	R/W	R/W	R/W

Initial value :00H

PSR0[3:2] External Reset De-bounce selection register

0	0	8us
0	1	16us
1	0	32us
1	1	64us

PSR0[1:0] Port De-bounce selection register

0	0	1us
0	1	2us
1	0	4us
1	1	8us

PSR2 (Analog I/O port selection register) : 2F4AH

7	6	5	4	3	2	1	0
PSR27	PSR26	PSR25	PSR24	PSR23	PSR22	PSR21	PSR20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

It is recommended to set this register when using port as Analog input such as ADC input, Comparator input, or OP-Amp input/output. And do not set this register when port is used as digital input.

PSR27 P1[7] Analog Input selection register

- 0 P1[7] digital input (default)
- 1 P1[7] AIN[7] input

PSR26 P1[0] Analog Input selection register

- 0 P1[0] digital input (default)
- 1 P1[0] AIN[6] input

PSR25 P2[0] Analog Input selection register

- 0 P2[0] digital input (default)
- 1 P2[0] AIN[5] input or AMP2O output

PSR24 P0[6] Analog Input selection register

- 0 P0[6] digital input (default)
- 1 P0[6] AIN[4] input

PSR23 P0[4] Analog Input selection register

- 0 P0[4] digital input (default)
- 1 P0[4] AIN[3] input

PSR22 P0[3] Analog Input selection register

0 P0[3] digital input (default)

1 P0[3] AIN[2] input

PSR21 P0[2] Analog Input selection register

0 P0[2] digital input (default)

1 P0[2] AIN[1] input

PSR20 P0[1] Analog Input selection register

0 P0[1] digital input (default)

1 P0[1] AIN[0] input

PSR3 (Analog I/O port selection register) : 2F4BH

7	6	5	4	3	2	1	0
-	-	PSR35	PSR34	PSR33	PSR32	PSR31	-
-	-	R/W	R/W	R/W	R/W	R/W	-

Initial value : 00H

It is recommended to set this register when using port as Analog input such as ADC input, Comparator input, or OP-Amp input/output. And do not set this register when port is used as digital input.

PSR35 P2[1] Analog Input selection register

0 P2[1] digital input (default)

1 P2[1] AMP1I input

PSR34 P1[6] Analog Input selection register

0 P1[6] digital input (default)

1 P1[6] CMP0_IN_N input

PSR33 P1[5] Analog Input selection register

0 P1[5] digital input (default)

1 P1[5] CMP0_IN_P input

PSR32 P1[4] Analog Input selection register

0 P1[4] digital input (default)

1 P1[4] CMP2_IN_P input

PSR31 P1[3] Analog Input selection register

0 P1[3] digital input (default)

1 P1[3] CMP1_IN_P input

Note) PSR3[0] must be kept '0'.

10 Interrupt Controller

10.1 Overview

The MC97F6108A supports up to 23 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have four levels of priority assigned to them. The interrupt controller has following features:

- receive the request from 23 interrupt source
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority are received simultaneously, the request of higher priority is serviced first and then lower priority is serviced.
- Each interrupt source can be controled by EA bit and each IEx bit
- Interrupt latency: 5~8 machine cycles in single interrupt system

The maskable interrupts are enabled through four pair of interrupt enable registers (IE, IE1, IE2, IE3). Bits of IE, IE1, IE2, IE3 register each individually enable/disable a particular interrupt source. Overall control is provided by EA (bit 7 of IE). When EA is set to '0', all interrupts are disabled: when EA is set to '1', interrupts are individually enabled or disabled through the other bits of the interrupt enable registers. The MC97F6108A supports a 4-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels by writing to IP or IPH or IP1 or IP1H or IP2 or IP2H or IP3 or IPH3.

Figure 3.1 shows the Interrupt Priority Level. Priority can be sets by writing to two bits of IPx and IPxH register. Each bit of IPx , IPxH corresponds to each interrupt and desides one of 4 priority levels of each interrupt. High level interrupt priority always has higher priority than low level interrupt. And Lower number interrupt has higher priority than higher number interrupt in the same level.

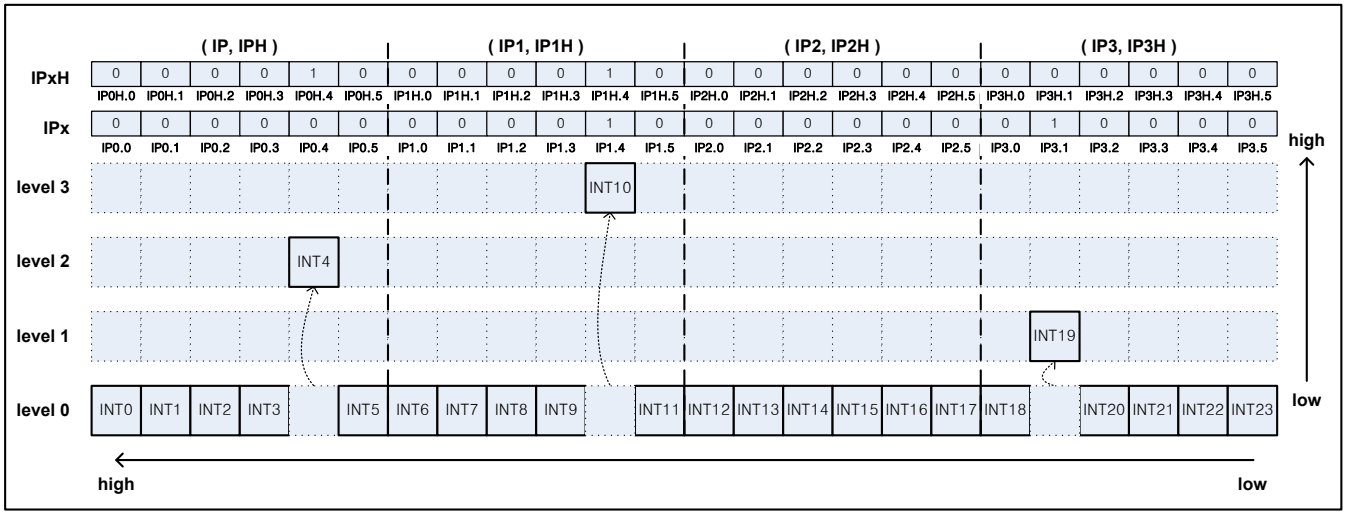


Figure 3.1 Interrupt Priority Level

10.2 External Interrupt

The external interrupt on EINT0, EINT1, EINT2 pins receive various interrupt request depending on the EIEDGE (External Interrupt Edge register) and EIPOLA (External Interrupt Polarity register).

Each external interrupt source has control setting bits. The EIFLAG (External interrupt flag register) register provides the status of each interrupts.

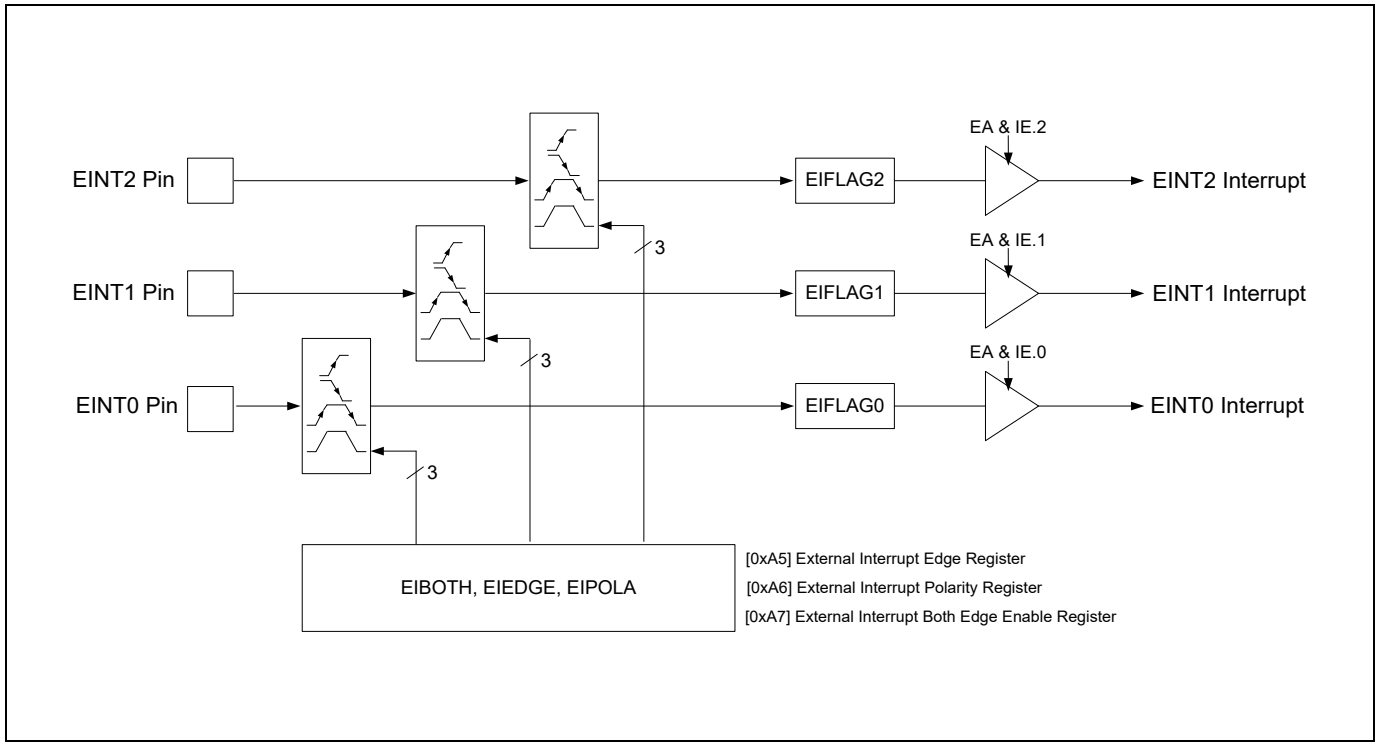


Figure 3.2 External Interrupt Description

10.3 Comparator interrupt and Comparator flag

Comparator output interrupt also receive various interrupt request depending on the CIEDGE (Comparator Interrupt Edge register) and CIPOLA (Comparator Interrupt Polarity register). Each comparator interrupt source has control setting bits. The CIFLAG (Comparator interrupt flag register) provides the status of each interrupts.

Comparators also have flags for controlling PPG. This flags are generated in the same manner as the comparator interrupt flags.

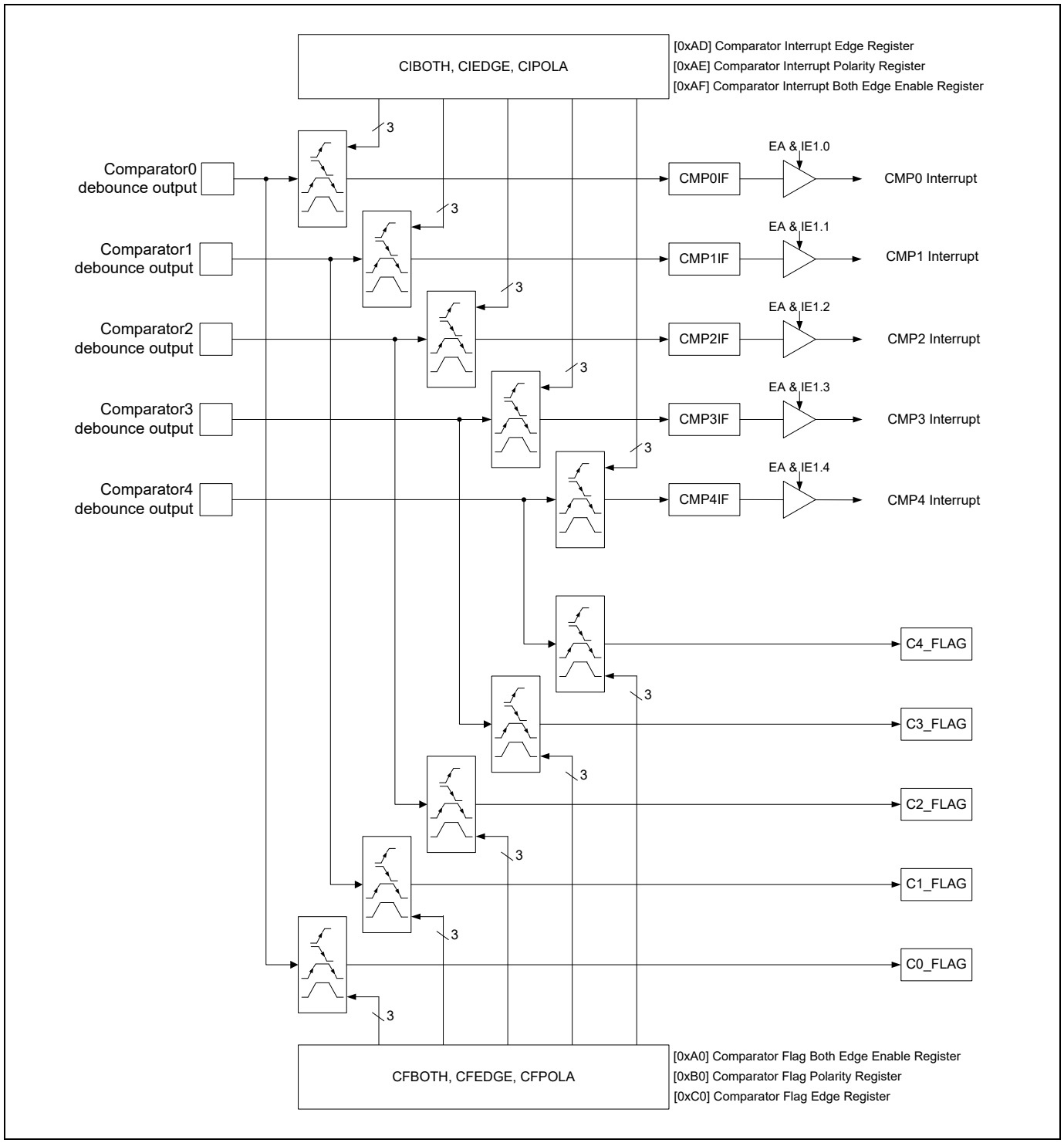


Figure 3.3 Comparator Interrupt and Comparator flag Description

10.4 Block Diagram

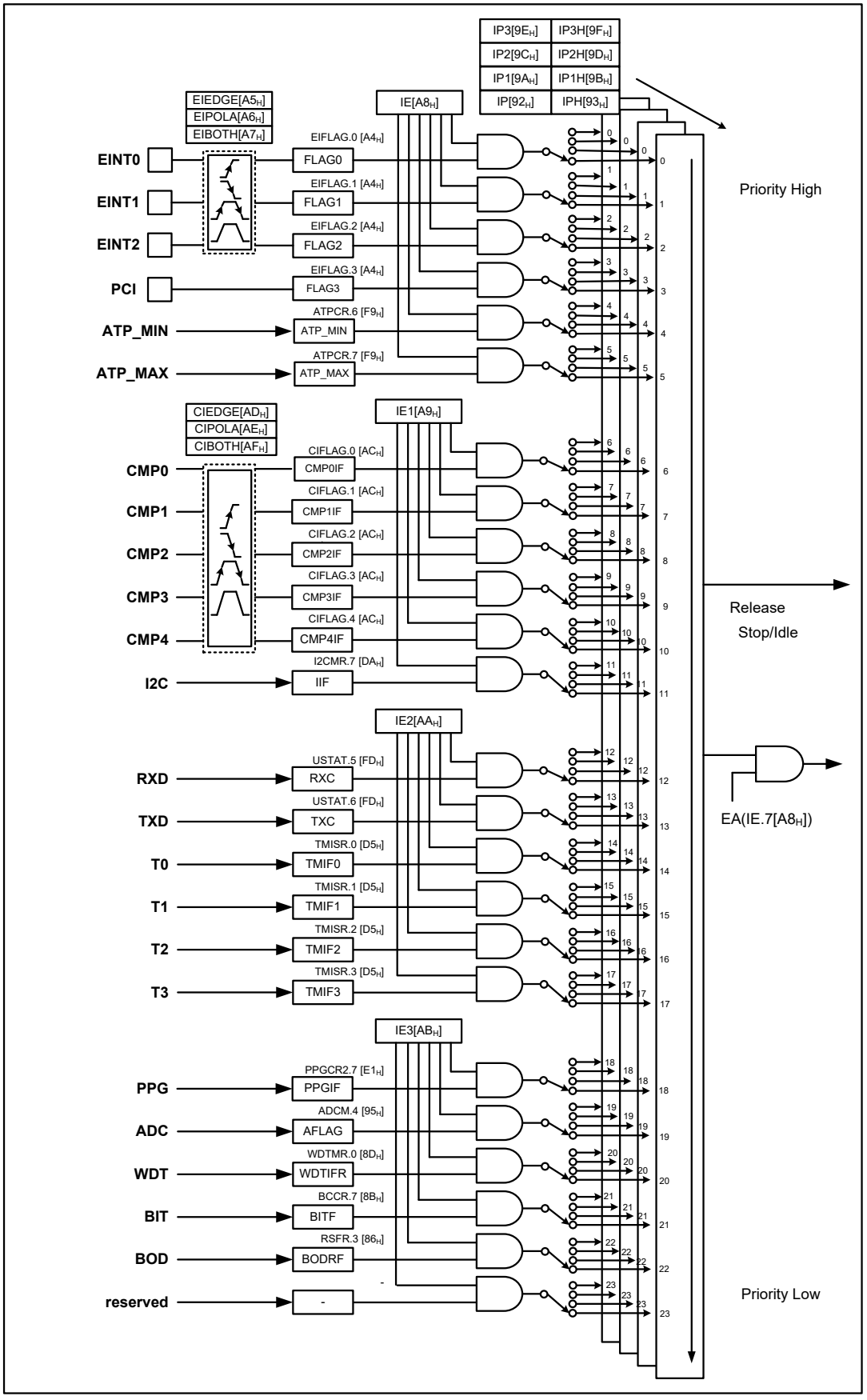


Figure 3.4 Block Diagram of Interrupt

10.5 Interrupt Vector Table

The interrupt controller supports 23 interrupt sources as shown in the Table 10-1 below. When interrupt becomes service, long call instruction (LCALL) is executed in the vector address. Interrupt request 23 has a decided priority order.

Interrupt Source	Symbol	Interrupt Enable Bit	Priority	Mask	Vector Address
Hardware Reset	RESETB	0	0	Non-Maskable	0000H
External Interrupt 0	INT0	IE.0	1	Maskable	0003H
External Interrupt 1	INT1	IE.1	2	Maskable	000BH
External Interrupt 2	INT2	IE.2	3	Maskable	0013H
PCI	INT3	IE.3	4	Maskable	001BH
ATP_MIN	INT4	IE.4	5	Maskable	0023H
ATP_MAX	INT5	IE.5	6	Maskable	002BH
CMP0	INT6	IE1.0	7	Maskable	0033H
CMP1	INT7	IE1.1	8	Maskable	003BH
CMP2	INT8	IE1.2	9	Maskable	0043H
CMP3	INT9	IE1.3	10	Maskable	004BH
CMP4	INT10	IE1.4	11	Maskable	0053H
I2C	INT11	IE1.5	12	Maskable	005BH
RXD	INT12	IE2.0	13	Maskable	0063H
TXD	INT13	IE2.1	14	Maskable	006BH
T0	INT14	IE2.2	15	Maskable	0073H
T1	INT15	IE2.3	16	Maskable	007BH
T2	INT16	IE2.4	17	Maskable	0083H
T3	INT17	IE2.5	18	Maskable	008BH
PPG	INT18	IE3.0	19	Maskable	0093H
ADC	INT19	IE3.1	20	Maskable	009BH
WDT	INT20	IE3.2	21	Maskable	00A3H
BIT	INT21	IE3.3	22	Maskable	00ABH
BOD	INT22	IE3.4	23	Maskable	00B3H
-	INT23	IE3.5	24	Maskable	00BBH

Table 10-1. Interrupt Vector Address Table

For mask-able interrupt execution, first EA bit must set ‘1’ and specific interrupt source must set ‘1’ by writing a ‘1’ to associated bit in the IEx. If interrupt request is received, specific interrupt request flag set ‘1’. And it remains ‘1’ until CPU accepts interrupt. After that, interrupt request flag will be cleared automatically.

10.6 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to '0' by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. After finishing the current instruction, at the next instruction to go interrupt service routine needs 5~8 machine cycle and the interrupt service task is terminated upon execution of an interrupt return instruction [RETI]. After generating interrupt, to go to interrupt service routine, the following process is progressed

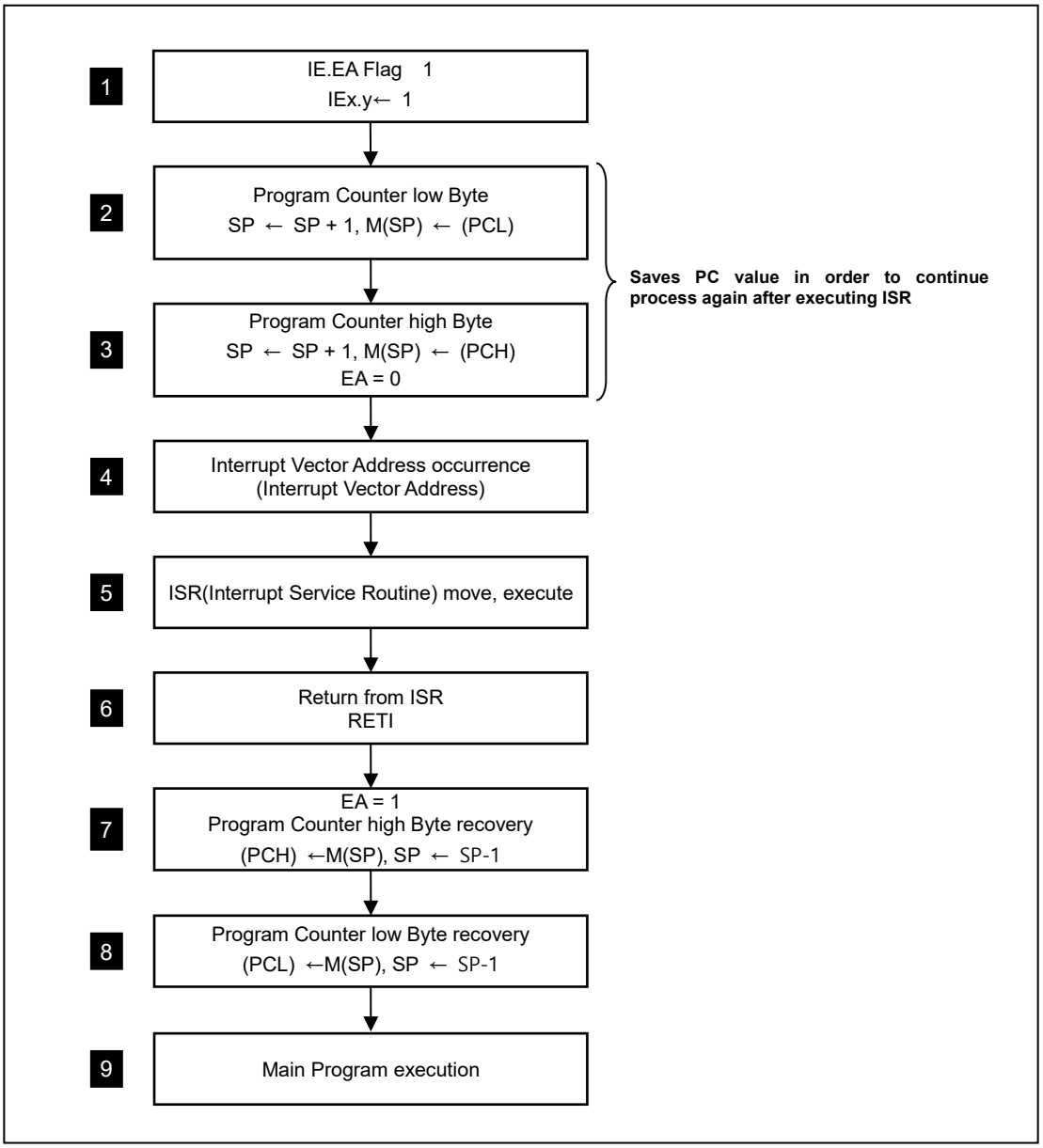


Figure 3.5 Interrupt Execution Sequence

10.7 Effective Timing after Controlling Interrupt bit

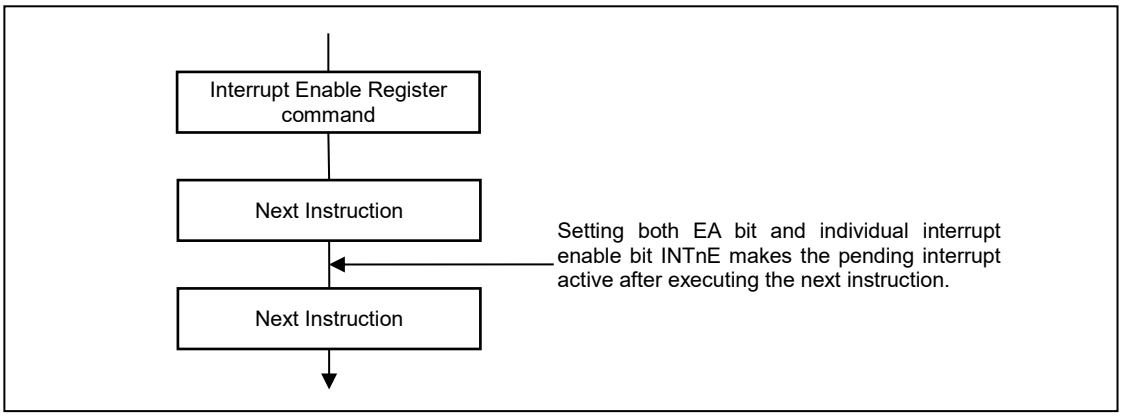


Figure 3.6 Interrupt Enable Register effective Timing

10.8 Multi Interrupt

If two requests of different priority are received simultaneously, the request of higher priority is serviced first and then lower priority is seviced. If requests of the interrupt are received at the same time simultaneously, an interrupt polling sequence determines by hardware which request is serviced.

However, multiple processing through software for special features is possible.

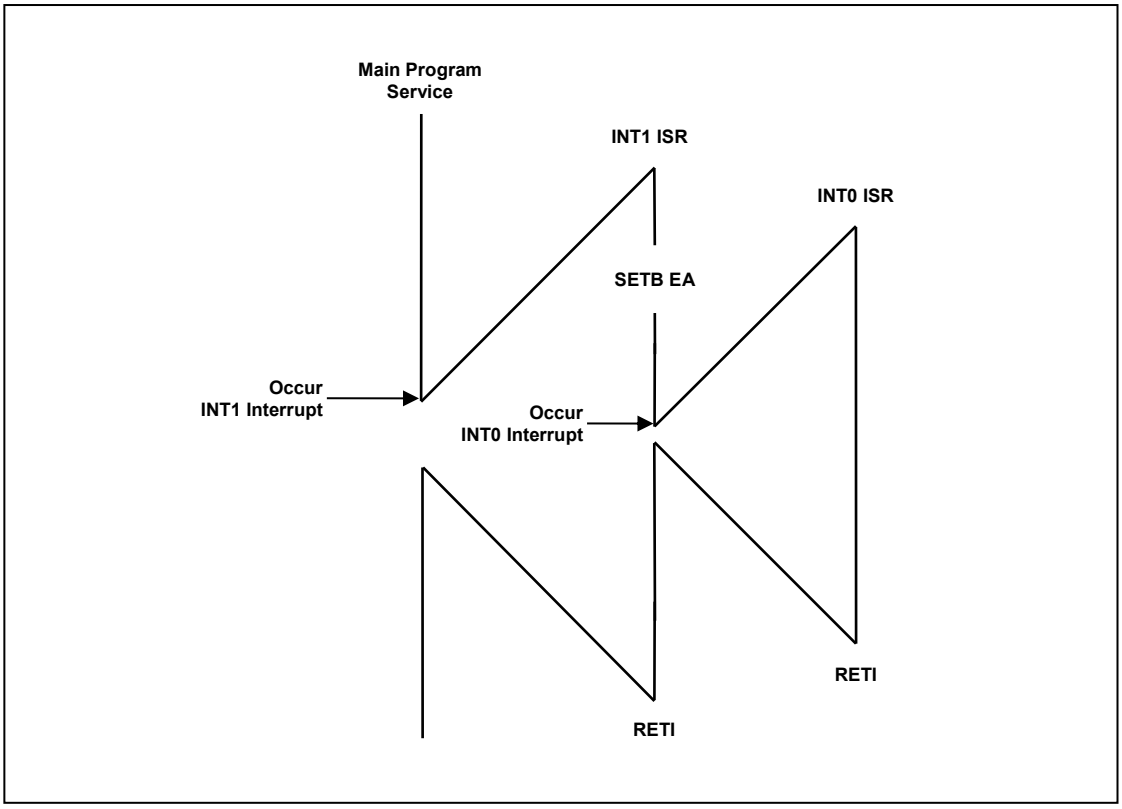


Figure 3.7 Execution of Multi Interrupt

Following example is shown to service INT0 routine during INT1 routine in Figure 3.7. In this example, INT0 interrupt level is higher than INT1 interrupt level. If some interrupt level is not higher than INT1 level, it can't service its interrupt service routine while INT1 ISR is serviced. INT0 ISR is serviced after INT1 ISR is finished.

Example) Software Multi Interrupt:

```
INT1:   MOV   IE, #03H      ; Enable INT0, INT1
        MOV   IP, #01H
        MOV   IPH, #00H     ; interrupt level of INT0 is 1 (INT1 is in level 0)
        SETB  EA            ; Enable global interrupt (necessary for multi interrupt)
```

10.9 Interrupt Enable Accept Timing

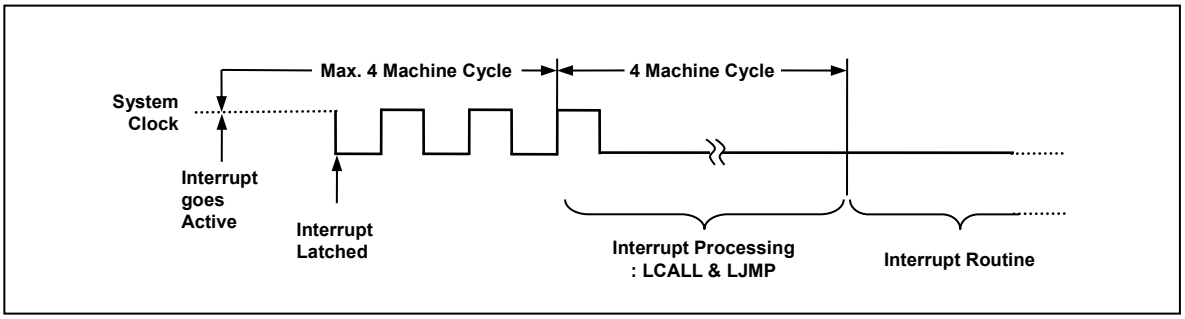


Figure 3.8 Interrupt Response Timing Diagram

10.10 Interrupt Service Routine Address

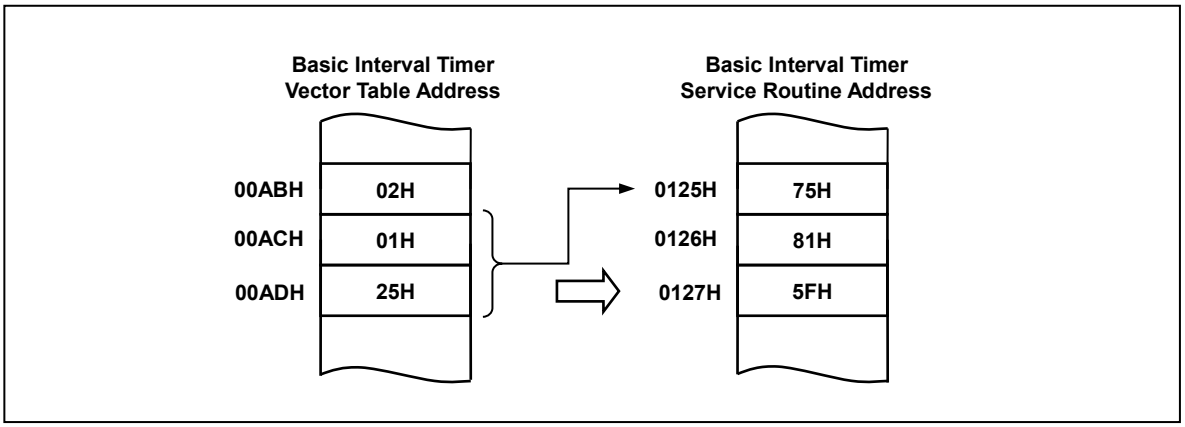
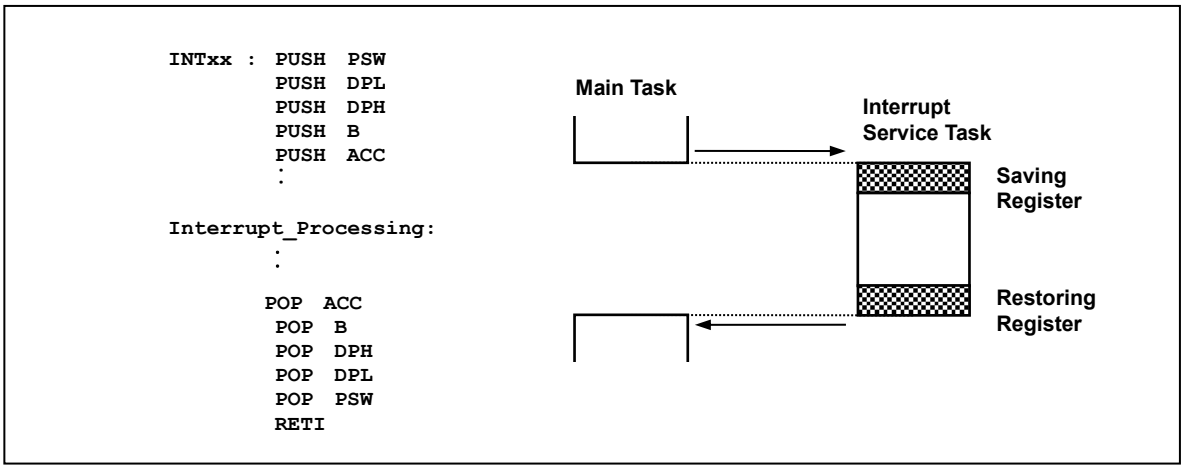


Figure 3.9 Correspondence between vector table address and the entry address of ISR

10.11 Saving/Restore General-Purpose Registers



```
INTxx :  PUSH  PSW
         PUSH  DPL
         PUSH  DPH
         PUSH  B
         PUSH  ACC
         :

Interrupt_Processing:
         :

        POP  ACC
         POP  B
         POP  DPH
         POP  DPL
         POP  PSW
         RETI
```


Figure 3.10 Saving/Restore Process Diagram & Sample Source

10.12 Interrupt Timing

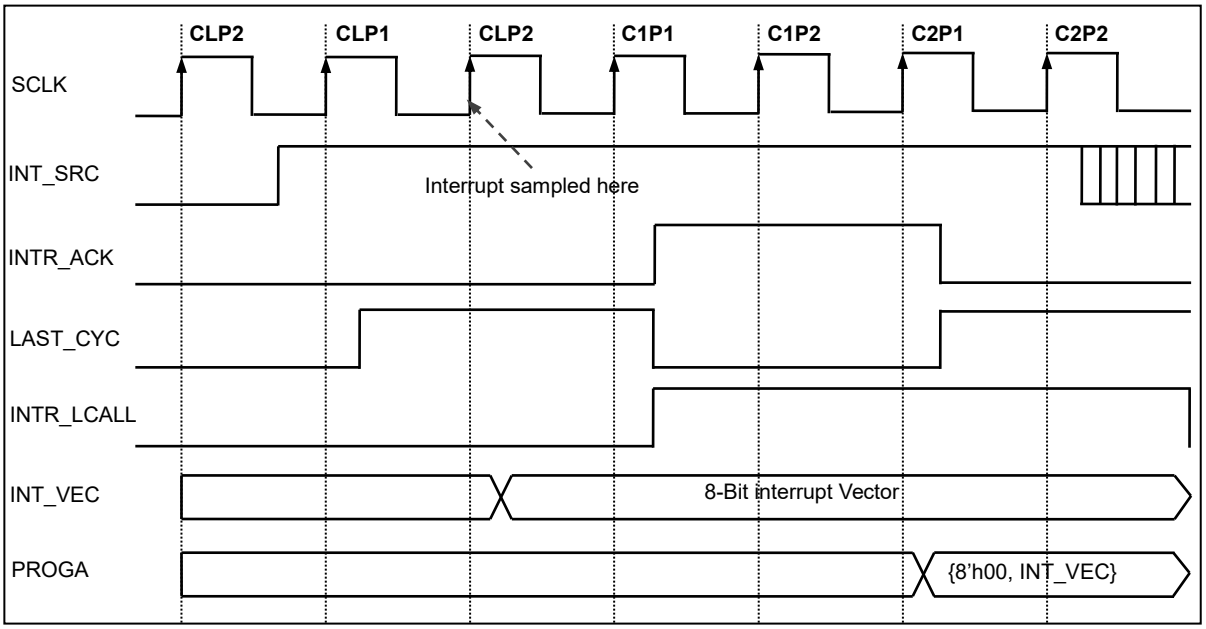


Figure 3.11 Timing chart of Interrupt Acceptance and Interrupt Return Instruction

Interrupt source sampled at last cycle of the command. When sampling interrupt source, it is decided to low 8-bit of interrupt vector. M8051W core makes interrupt acknowledge at first cycle of command, executes long call to jump interrupt routine as INT_VEC.

NOTE)

2. command cycle C?P?: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.13 Interrupt Register Description

Name	Address	Dir	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP	92H	R/W	00H	Interrupt Priority Register
IPH	93H	R/W	00H	Interrupt Priority Register High
IP1	9AH	R/W	00H	Interrupt Priority Register 1
IP1H	9BH	R/W	00H	Interrupt Priority Register 1 High
IP2	9CH	R/W	00H	Interrupt Priority Register 2
IP2H	9DH	R/W	00H	Interrupt Priority Register 2 High
IP3	9EH	R/W	00H	Interrupt Priority Register 3
IP3H	9FH	R/W	00H	Interrupt Priority Register 3 High

Table 10-2. Register Map

Name	Address	Dir	Default	Description
EIENAB	A3H	R/W	00H	External Interrupt Flag Enable Register
EIFLAG	A4H	R/W	00H	External Interrupt Flag Register
EIEDGE	A5H	R/W	00H	External Interrupt Flag Edge Register
EIPOLA	A6H	R/W	00H	External Interrupt Flag Polarity Register
EIBOTH	A7H	R/W	00H	External Interrupt Flag Both Edge Enable Register
CIENAB	B1H	R/W	00H	Comparator Interrupt Flag Enable Register
CIFLAG	ACH	R/W	00H	Comparator Interrupt Flag Register
CIEDGE	ADH	R/W	00H	Comparator Interrupt Flag Edge Register
CIPOLA	AEH	R/W	00H	Comparator Interrupt Flag Polarity Register
CIBOTH	AFH	R/W	00H	Comparator Interrupt Flag Both Edge Enable Register
CFENAB	D8H	R/W	00H	Comparator Flag Enable Register
CFFLAG	C8H	R/W	00H	Comparator Flag Register
CFEDGE	C0H	R/W	00H	Comparator Flag Edge Register
CFPOLA	B0H	R/W	00H	Comparator Flag Polarity Register
CFBOTH	A0H	R/W	00H	Comparator Flag Both Edge Enable Register
PCI	94H	R/W	00H	Pin Change Interrupt Enable Register

Table 10-3. Register Map (Continued)

10.14 Register description for Interrupt

IE (Interrupt Enable Register) : A8H

7	6	5	4	3	2	1	0
EA	-	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EA Enable or disable all interrupt bits
- 0 All Interrupt disable
- 1 All Interrupt enable

INT5E Enable or disable ATP_MAX Interrupt
- 0 ATP_MAX Interrupt Disable
- 1 ATP_MAX Interrupt Enable

INT4E Enable or disable ATP_MIN Interrupt
- 0 ATP_MIN Interrupt Disable
- 1 ATP_MIN Interrupt Enable

INT3E Enable or disable Pin Change Interrupt
- 0 Pin Change Interrupt Disable
- 1 Pin Change Interrupt Enable

INT2E Enable or disable External Interrupt 2
- 0 External interrupt 2 Disable
- 1 External interrupt 2 Enable

INT1E Enable or disable External Interrupt 1
- 0 External interrupt 1 Disable
- 1 External interrupt 1 Enable

INT0E Enable or disable External Interrupt 0
- 0 External interrupt 0 Disable
- 1 External interrupt 0 Enable

IE1 (Interrupt Enable Register 1) : A9H

7	6	5	4	3	2	1	0
-	-	INT11E	INT10E	INT9E	INT8E	INT7E	INT6E
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT11E Enable or disable I2C Interrupt
- 0 I2C Interrupt Disable
- 1 I2C Interrupt Enable

INT10E Enable or disable Comparator4 Interrupt
- 0 Comparator4 Interrupt Disable
- 1 Comparator4 Interrupt Enable

INT9E Enable or disable Comparator3 Interrupt
- 0 Comparator3 Interrupt Disable
- 1 Comparator3 Interrupt Enable

INT8E Enable or disable Comparator2 Interrupt
- 0 Comparator2 Interrupt Disable
- 1 Comparator2 Interrupt Enable

INT7E Enable or disable Comparator1 Interrupt
- 0 Comparator1 Interrupt Disable
- 1 Comparator1 Interrupt Enable

INT6E Enable or disable Comparator0 Interrupt
- 0 Comparator0f Interrupt Disable
- 1 Comparator0 Interrupt Enable

IE2 (Interrupt Enable Register 2) : AAH

7	6	5	4	3	2	1	0
-	-	INT17E	INT16E	INT15E	INT14E	INT13E	INT12E
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT17E Enable or disable Timer 3 Interrupt
- 0 Timer3 interrupt Disable
- 1 Timer3 interrupt Enable

INT16E Enable or disable Timer 2 Interrupt
- 0 Timer2 interrupt Disable
- 1 Timer2 interrupt Enable

INT15E Enable or disable Timer 1 Interrupt
- 0 Timer1 interrupt Disable
- 1 Timer1 interrupt Enable

INT14E Enable or disable Timer 0 Interrupt
- 0 Timer0 interrupt Disable
- 1 Timer0 interrupt Enable

INT13E Enable or disable USART TX Interrupt
- 0 USART TX interrupt Disable
- 1 USART TX interrupt Enable

INT12E Enable or disable USART RX Interrupt
- 0 USART RX interrupt Disable
- 1 USART RX interrupt Enable

IE3 (Interrupt Enable Register 1) : ABH

7	6	5	4	3	2	1	0
-	-	INT23E	INT22E	INT21E	INT20E	INT19E	INT18E
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT23E reserved
0
1

INT22E Enable or disable BOD Interrupt
0 BOD interrupt Disable
1 BOD interrupt Enable

INT21E Enable or disable BIT Interrupt
0 BIT interrupt Disable
1 BIT interrupt Enable

INT20E Enable or disable WDT Interrupt
0 WDT interrupt Disable
1 WDT interrupt Enable

INT19E Enable or disable ADC Interrupt
0 ADC interrupt Disable
1 ADC interrupt Enable

INT18E Enable or disable PPG Interrupt
0 PPG interrupt Disable
1 PPG interrupt Enable

IP (Interrupt Priority Register) : 92H

7	6	5	4	3	2	1	0
-	-	IP5	IP4	IP3	IP2	IP1	IP0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IPH (Interrupt Priority Register High) : 93H

7	6	5	4	3	2	1	0
-	-	IPH5	IPH4	IPH3	IPH2	IPH1	IPH0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP[5:0], IPH[5:0] Select Interrupt Priority.
Each IPH and IP corresponds to INT5~INT0.

IPH	IP	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP1 (Interrupt Priority Register 1) : 9AH

7	6	5	4	3	2	1	0
-	-	IP15	IP14	IP13	IP12	IP11	IP10
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1H (Interrupt Priority Register 1 High) : 9BH

7	6	5	4	3	2	1	0
-	-	IP1H5	IP1H4	IP1H3	IP1H2	IP1H1	IP1H0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1[5:0], IP1H[5:0] Select Interrupt Priority.
Each IP1H and IP1 corresponds to INT11~INT6.

IP1H	IP1	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP2 (Interrupt Priority Register 2) : 9CH

7	6	5	4	3	2	1	0
-	-	IP25	IP24	IP23	IP22	IP21	IP20
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 0H

IP2H (Interrupt Priority Register 2 High) : 9DH

7	6	5	4	3	2	1	0
-	-	IP2H5	IP2H4	IP2H3	IP2H2	I2PH1	IP2H0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 0H

IP2[3:0], IP2H[3:0] Select Interrupt Priority.
Each IP2H and IP2 corresponds to INT17~INT12.

IP2H	IP2	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP3 (Interrupt Priority Register 3) : 9EH

7	6	5	4	3	2	1	0
-	-	IP35	IP34	IP33	IP32	IP31	IP30
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 0H

IP3H (Interrupt Priority Register 3 High) : 9FH

7	6	5	4	3	2	1	0
-	-	IP3H5	IP3H4	IP3H3	IP3H2	I3PH1	IP3H0
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 0H

IP3[3:0], IP3H[3:0] Select Interrupt Priority.
Each IP3H and IP3 corresponds to INT18~INT23.

IP3H	IP3	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIENAB (External Interrupt Flag Enable Register) : A3H

7	6	5	4	3	2	1	0
-	-	-	-	-	ENAB2	ENAB1	ENAB0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 0H

ENAB2 Enable or Disable External Interrupt 2
0 Disable External Interrupt 2(default)
1 Enable External Interrupt 2

ENAB1 Enable or Disable External Interrupt 1
0 Disable External Interrupt 1(default)
1 Enable External Interrupt 1

ENAB0 Enable or Disable External Interrupt 0
0 Disable External Interrupt 0(default)
1 Enable External Interrupt 0

EIFLAG (External Interrupt Flag Register) : A4H

7	6	5	4	3	2	1	0
-	-	-	-	FLAG3	FLAG2	FLAG1	FLAG0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 0H

When an interrupt source is generated and EIENAB is set to '1', the flag is generated.
The flag can be cleared by writing a ‘0’ to bit. It is also cleared automatically before interrupt service routine is served.

FLAG3 When Pin Change Interrupt occurs this bit is set.
0 Pin Change Interrupt not occurred
1 Pin Change Interrupt occurred

FLAG2 When External Interrupt 2 occurs this bit is set.
0 External Interrupt 2 not occurred
1 External Interrupt 2 occurred

FLAG1 When External Interrupt 1 occurs this bit is set.
0 External Interrupt 1 not occurred
1 External Interrupt 1 occurred

FLAG0 When External Interrupt 0 occurs this bit is set.
0 External Interrupt 0 not occurred
1 External Interrupt 0 occurred

EIEDGE (External Interrupt Flag Edge Register) : A5H

7	6	5	4	3	2	1	0
-	-	–	-	-	EDGE2	EDGE1	EDGE0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 0H

EDGE2 Determines the type of External interrupt 2, edge or level sensitive.
- 0 Level (default)
- 1 Edge

EDGE1 Determines the type of External interrupt 1, edge or level sensitive.
- 0 Level (default)
- 1 Edge

EDGE0 Determines the type of External interrupt 0, edge or level sensitive.
- 0 Level (default)
- 1 Edge

EIPOLA (External Interrupt Flag Polarity Register) : A6H

7	6	5	4	3	2	1	0
-	-	-	-	-	POLA2	POLA1	POLA0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 0H

According to EIEDGE, this register acts differently. If EIEDGE is level type, external interrupt polarity have level value. If EIEDGE is edge type, external interrupt polarity have edge value.

POLA2 Determine the polarity of External Interrupt 2
- 0 When High level or rising edge, Interrupt occurs(default)
- 1 When Low level or falling edge, Interrupt occurs

POLA1 Determine the polarity of External Interrupt 1
- 0 When High level or rising edge, Interrupt occurs(default)
- 1 When Low level or falling edge, Interrupt occurs

POLA0 Determine the polarity of External Interrupt 0
- 0 When High level or rising edge, Interrupt occurs(default)
- 1 When Low level or falling edge, Interrupt occurs

EIBOTH (External Interrupt Flag Both Edge Enable Register) : A7H

7	6	5	4	3	2	1	0
-	-	-	-	-	BOTH2	BOTH1	BOTH0
-	-	-	-	-	R/W	R/W	R/W

Initial value : 0H

If BOTHx is written to '1', the corresponding external pin interrupt is enabled by both edges(no level).
And EIEDGE and EIPOLA register value are ignored.

BOTH2 Determine the type of External Interrupt 2
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

BOTH1 Determine the type of External Interrupt 1
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

BOTH0 Determine the type of External Interrupt 0
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CIENAB (Comparator Interrupt Flag Enable Register) : B1H

7	6	5	4	3	2	1	0
-	-	-	CIENAB4	CIENAB3	CIENAB2	CIENAB1	CIENAB0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CIENAB4 Enable or Disable Comparator4 Interrupt
- 0 Disable Comparator4 interrupt(default)
- 1 Enable Comparator4 interrupt

CIENAB3 Enable or Disable Comparator3 Interrupt
- 0 Disable Comparator3 interrupt(default)
- 1 Enable Comparator3 interrupt

CIENAB2 Enable or Disable Comparator2 Interrupt
- 0 Disable Comparator2 interrupt(default)
- 1 Enable Comparator2 interrupt

CIENAB1 Enable or Disable Comparator1 Interrupt
- 0 Disable Comparator1 interrupt(default)
- 1 Enable Comparator1 interrupt

CIENAB0 Enable or Disable Comparator0 Interrupt
- 0 Disable Comparator0 interrupt(default)
- 1 Enable Comparator0 interrupt

CIFLAG (Comparator Interrupt Flag Register) : ACH

7	6	5	4	3	2	1	0
-	-	-	CMP4IF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

When an interrupt source is generated and CIENAB is set to '1', the flag is generated.
The flag can be cleared by writing a ‘0’ to bit. It is also cleared automatically before interrupt service routine is served.

CMP4IF When Comparator4 Interrupt occurs this bit is set.
- 0 Comparator4 Interrupt not occurred
- 1 Comparator4 Interrupt occurred

CMP3IF When Comparator3 Interrupt occurs this bit is set.
- 0 Comparator3 Interrupt not occurred
- 1 Comparator3 Interrupt occurred

CMP2IF When Comparator2 Interrupt occurs this bit is set.
- 0 Comparator2 Interrupt not occurred
- 1 Comparator2 Interrupt occurred

CMP1IF When Comparator1 Interrupt occurs this bit is set.
- 0 Comparator1 Interrupt not occurred
- 1 Comparator1 Interrupt occurred

CMP0IF When Comparator0 Interrupt occurs this bit is set.
- 0 Comparator0 Interrupt not occurred
- 1 Comparator0 Interrupt occurred

CIEDGE (Comparator Interrupt Flag Edge Register) : ADH

7	6	5	4	3	2	1	0
-	-	-	CIEDGE4	CIEDGE3	CIEDGE2	CIEDGE1	CIEDGE0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CIEDGE4 Determines the type of Comparator4 interrupt, edge or level sensitive.
0 Level (default)
1 Edge

CIEDGE3 Determines the type of Comparator3 interrupt, edge or level sensitive.
0 Level (default)
1 Edge

CIEDGE2 Determines the type of Comparator2 interrupt, edge or level sensitive.
0 Level (default)
1 Edge

CIEDGE1 Determines the type of Comparator1 interrupt, edge or level sensitive.
0 Level (default)
1 Edge

CIEDGE0 Determines the type of Comparator0 interrupt, edge or level sensitive.
0 Level (default)
1 Edge

CIPOLA (Comparator Interrupt Flag Polarity Register) : AEH

7	6	5	4	3	2	1	0
-	-	-	CIPOLA4	CIPOLA3	CIPOLA2	CIPOLA1	CIPOLA0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

According to CIEDGE, this register acts differently. If CIEDGE is level type, comparator interrupt polarity have level value. If CIEDGE is edge type, comparator interrupt polarity have edge value.

CIPOLA4 Determine the polarity of Comparator4 Interrupt
0 When High level or rising edge, Interrupt occurs(default)
1 When Low level or falling edge, Interrupt occurs

CIPOLA3 Determine the polarity of Comparator3 Interrupt
0 When High level or rising edge, Interrupt occurs(default)
1 When Low level or falling edge, Interrupt occurs

CIPOLA2 Determine the polarity of Comparator2 Interrupt
0 When High level or rising edge, Interrupt occurs(default)
1 When Low level or falling edge, Interrupt occurs

CIPOLA1 Determine the polarity of Comparator1 Interrupt
0 When High level or rising edge, Interrupt occurs(default)
1 When Low level or falling edge, Interrupt occurs

CIPOLA0 Determine the polarity of Comparator0 Interrupt
0 When High level or rising edge, Interrupt occurs(default)
1 When Low level or falling edge, Interrupt occurs

CIBOTH (Comparator Interrupt Flag Both Edge Enable Register) : AFH

7	6	5	4	3	2	1	0
-	-	-	CIBOTH4	CIBOTH3	CIBOTH2	CIBOTH1	CIBOTH0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

If CIBOTHx is written to '1', the corresponding comparator interrupt is enabled by both edges(no level).
And CIEDGE and CIPOLA register value are ignored.

CIBOTH4 Determine the type of Comparator4 Interrupt
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CIBOTH3 Determine the type of Comparator3 Interrupt
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CIBOTH2 Determine the type of Comparator2 Interrupt
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CIBOTH1 Determine the type of Comparator1 Interrupt
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CIBOTH0 Determine the type of Comparator0 Interrupt
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CFENAB (Comparator Flag Enable Register) : D8H

7	6	5	4	3	2	1	0
-	-	-	CFENAB4	CFENAB3	CFENAB2	CFENAB1	CFENAB0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CFENAB4 Enable or Disable Comparator4 Flag generation
- 0 Disable Comparator4 Flag generation(default)
- 1 Enable Comparator4 Flag generation

CFENAB3 Enable or Disable Comparator3 Flag generation
- 0 Disable Comparator3 Flag generation(default)
- 1 Enable Comparator3 Flag generation

CFENAB2 Enable or Disable Comparator2 Flag generation
- 0 Disable Comparator2 Flag generation (default)
- 1 Enable Comparator2 Flag generation

CFENAB1 Enable or Disable Comparator1 Flag generation
- 0 Disable Comparator1 Flag generation (default)
- 1 Enable Comparator1 Flag generation

CFENAB0 Enable or Disable Comparator0 Flag generation
- 0 Disable Comparator0 Flag generation (default)
- 1 Enable Comparator0 Flag generation

CFFLAG (Comparator Flag Register) : C8H

7	6	5	4	3	2	1	0
-	-	-	C4_FLAG	C3_FLAG	C2_FLAG	C1_FLAG	C0_FLAG
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

When an comparator source is generated and CFENAB is set to '1', the flag is generated.
The flag can be cleared by writing a '0' to bit.
It is not cleared automatically.

C4_FLAG When Comparator4 occurs this bit is set.
- 0 Comparator4 not occurred
- 1 Comparator4 occurred

C3_FLAG When Comparator3 occurs this bit is set.
It is cleared automatically when PPG period matching.
- 0 Comparator3 not occurred
- 1 Comparator3 occurred

C2_FLAG When Comparator2 occurs this bit is set.
- 0 Comparator2 not occurred
- 1 Comparator2 occurred

C1_FLAG When Comparator1 occurs this bit is set.
- 0 Comparator1 not occurred
- 1 Comparator1 occurred

C0_FLAG When Comparator0 occurs this bit is set.
It is cleared automatically when PPG period matching.
- 0 Comparator0 not occurred
- 1 Comparator0 occurred

CFEDGE (Comparator Flag Edge Register) : C0H

7	6	5	4	3	2	1	0
-	-	-	CFEDGE4	CFEDGE3	CFEDGE2	CFEDGE1	CFEDGE0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CFEDGE4 Determines the type of Comparator4 flag, edge or level sensitive.
- 0 Level (default)
- 1 Edge

CFEDGE3 Determines the type of Comparator3 flag, edge or level sensitive.
- 0 Level (default)
- 1 Edge

CFEDGE2 Determines the type of Comparator2 flag, edge or level sensitive.
- 0 Level (default)
- 1 Edge

CFEDGE1 Determines the type of Comparator1 flag, edge or level sensitive.
- 0 Level (default)
- 1 Edge

CFEDGE0 Determines the type of Comparator0 flag, edge or level sensitive.
- 0 Level (default)
- 1 Edge

CFPOLA (Comparator Flag Polarity Register) : B0H

7	6	5	4	3	2	1	0
-	-	-	CFPOLA4	CFPOLA3	CFPOLA2	CFPOLA1	CFPOLA0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

According to CFEDGE, this register acts differently. If CFEDGE is level type, comparator flag polarity have level value. If CFEDGE is edge type, comparator flag polarity have edge value.

CFPOLA4 Determine the polarity of Comparator4 flag
- 0 When High level or rising edge, flag occurs(default)
- 1 When Low level or falling edge, flag occurs

CFPOLA3 Determine the polarity of Comparator3 flag
- 0 When High level or rising edge, flag occurs(default)
- 1 When Low level or falling edge, flag occurs

CFPOLA2 Determine the polarity of Comparator2 flag
- 0 When High level or rising edge, flag occurs(default)
- 1 When Low level or falling edge, flag occurs

CFPOLA1 Determine the polarity of Comparator1 flag
- 0 When High level or rising edge, flag occurs(default)
- 1 When Low level or falling edge, flag occurs

CFPOLA0 Determine the polarity of Comparator0 flag
- 0 When High level or rising edge, flag occurs(default)
- 1 When Low level or falling edge, flag occurs

CFBOTH (Comparator Flag Both Edge Enable Register) : A0H

7	6	5	4	3	2	1	0
-	-	-	CFBOTH4	CFBOTH3	CFBOTH2	CFBOTH1	CFBOTH0
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

If CFBOTHx is written to '1', the corresponding comparator flag is enabled by both edges(no level).
And CFEDGE and CFPOLA register value are ignored.

CFBOTH4 Determine the type of Comparator4 flag
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CFBOTH3 Determine the type of Comparator3 flag
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CFBOTH2 Determine the type of Comparator2 flag
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CFBOTH1 Determine the type of Comparator1 flag
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

CFBOTH0 Determine the type of Comparator0 flag
- 0 Both edge detection Disable (default)
- 1 Both edge detection Enable

PCI (Pin Change Interrupt Enable Register) : 94H

7	6	5	4	3	2	1	0
PCI7	PCI6	PCI5	PCI4	PCI3	PCI2	PCI1	PCI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PCI[7] Select PCI interrupt enable or disable of P1[7]
- 0 Disable (default)
- 1 Enable

PCI[6] Select PCI interrupt enable or disable of P1[6]
- 0 Disable (default)
- 1 Enable

PCI[5] Select PCI interrupt enable or disable of P1[5]
- 0 Disable (default)
- 1 Enable

PCI[4] Select PCI interrupt enable or disable of P1[4]
- 0 Disable (default)
- 1 Enable

PCI[3] Select PCI interrupt enable or disable of P1[3]
- 0 Disable (default)
- 1 Enable

PCI[2] Select PCI interrupt enable or disable of P1[2]
- 0 Disable (default)
- 1 Enable

PCI[1] Select PCI interrupt enable or disable of P1[1]
- 0 Disable (default)
- 1 Enable

PCI[0] Select PCI interrupt enable or disable of P1[0]
- 0 Disable (default)
- 1 Enable

11 Peripheral Hardware

11.1 Clock Generator

11.1.1 Overview

As shown in Figure 4.1, the clock generator produces the basic clock pulses which provide the system clock to CPU and peripheral hardware. The internal RC-OSC is only used as system clock. The default system clock is INT-OSC Oscillator and the default division rate is one.

In order to stabilize system internally, use 8kHz WDT-oscillator for BIT, WDT and ports de-bounce.

- Calibrated Internal RC Oscillator (16MHz)
  - . INT-RC OSC/1 (16MHz, Default system clock)
  - . INT-RC OSC/2 (8MHz)
  - . INT-RC OSC/4 (4MHz)
  - . INT-RC OSC/8 (2MHz)

11.1.2 Block Diagram

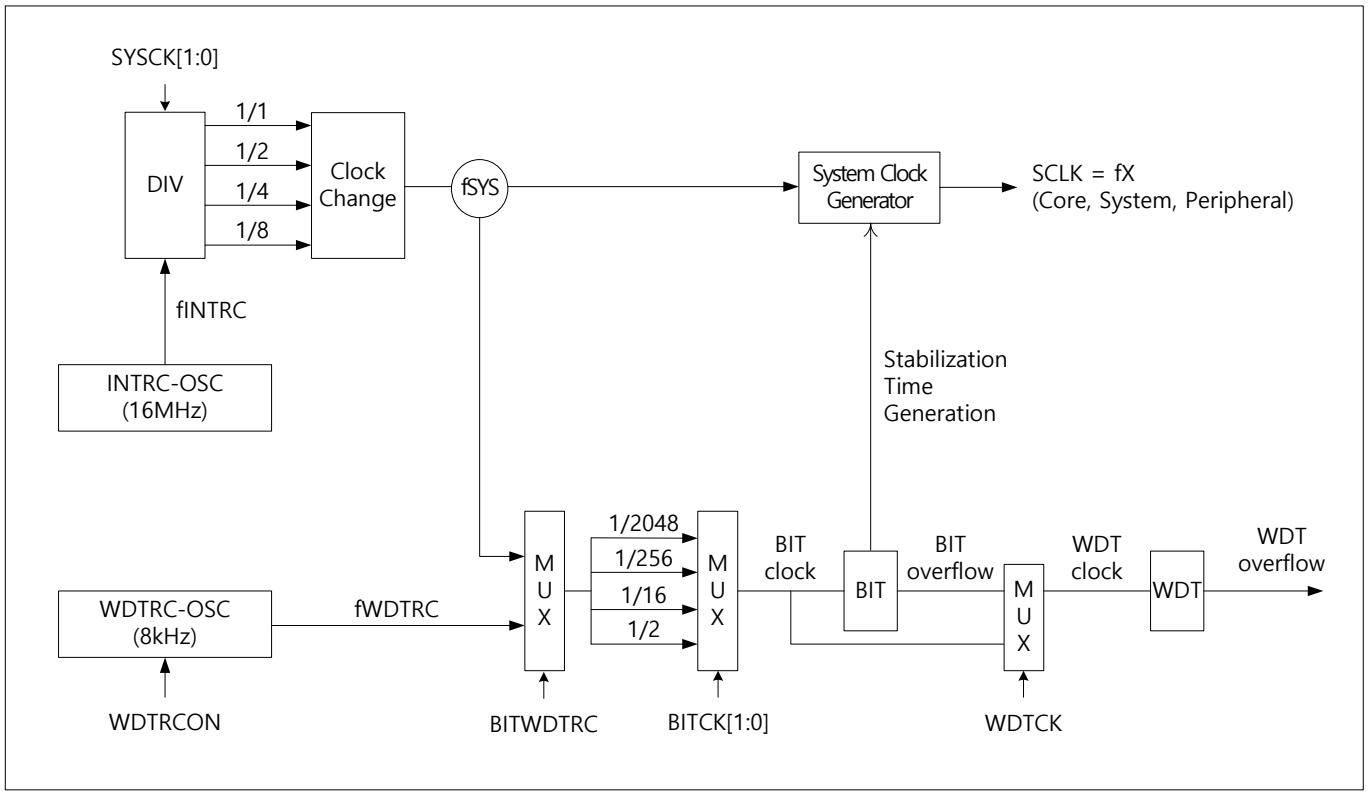


Figure 4.1 Clock Generator Block Diagram

11.1.3 Register Map

Name	Address	Dir	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register

Table 11-1. Register Map

11.1.4 Register description for Clock Generator

SCCR (System and Clock Control Register) : 8AH

7	6	5	4	3	2	1	0
WDTRCON	-	CHBS	-	OSCNF	-	SYSCK1	SYSCK0
0	-	0	-	0	-	0	0
R/W	-	R/W	-	R/W	-	R/W	R/W

Initial value : 00H

WDTRCON Select WDTRC oscillator on or off.
- 0 WDTRC off (default)
- 1 WDTRC on

CHBS Control the scheme of clock change. If this bit set to ‘0’, clock change is controlled by hardware. But if this set to ‘1’, clock change is controlled by software.
- 0 Clock changed by hardware during stop mode (default)
- 1 Clock changed by software

OSCNF OSC Noise Filter Enable
- 0 x1
- 1 x2

SYSCK[1:0] Determine division rate.
Note) To change by software, CHBS set to ‘1’

SYSCK1	SYSCK0	Description
0	0	fINTRC(default)
0	1	fINTRC/2
1	0	fINTRC/4
1	1	fINTRC/8

NOTE)

1. SCCR[2] & SCCR[4], must be kept '0'.
2. When clear CHBS, keep other bits in SCCR.

11.2 BIT

11.2.1 Overview

The MC97F6108A has one 8-bit Basic Interval Timer that is free-run and can't stop. Block diagram is shown in Figure 4.2. In addition, the Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt (BITF).

The MC97F6108A has these Basic Interval Timer (BIT) features:

- During Power On, BIT gives a stable clock generation time
- On exiting Stop mode, BIT gives a stable clock generation time
- As clock function, time interrupt occurrence

11.2.2 Block Diagram

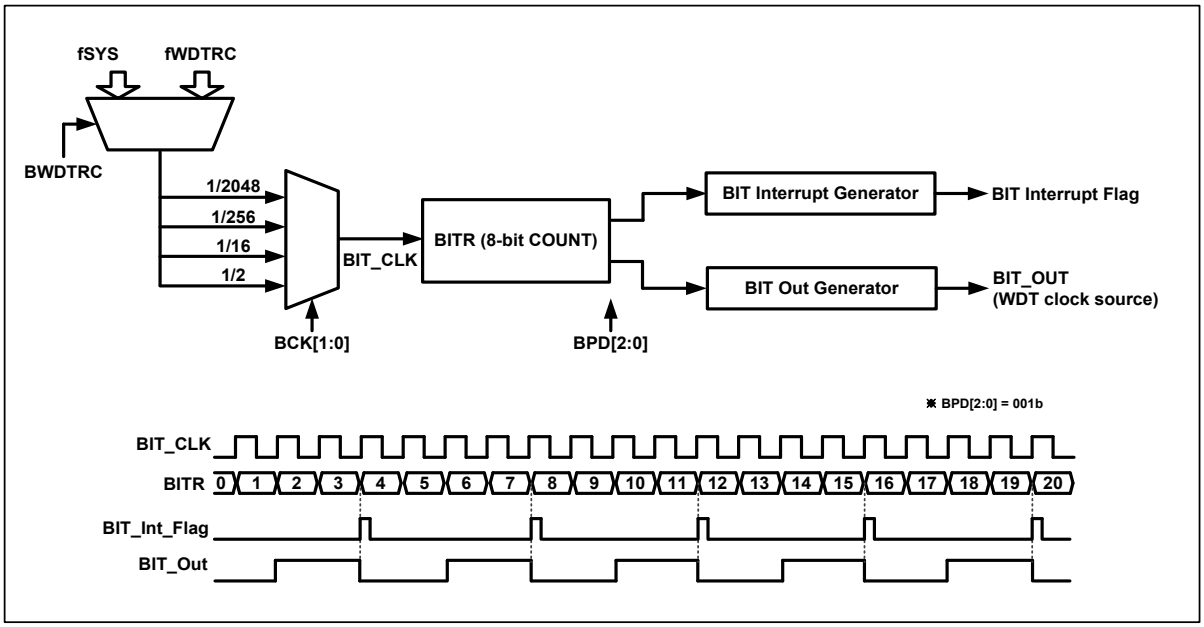


Figure 4.2 BIT Block Diagram

11.2.3 Register Map

Name	Address	Dir	Default	Description
BCCR	8BH	R/W	05H	BIT Clock Control Register
BITR	8CH	R	00H	Basic Interval Timer Register

Table 11-2. Register Map

11.2.4 Register description for Bit Interval Timer

BCCR (BIT Clock Control Register) : 8BH

7	6	5	4	3	2	1	0
BITF	BCK1	BCK0	BWDTRC	BCLR	BPD2	BPD1	BPD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 05H

BITF — When BIT Interrupt occurs, this bit becomes '1'. This bit is cleared automatically if BIT and global interrupt enable bit is set. For clearing bit, write '0' to this bit.

0 no generation

1 generation

BCK[1:0] — Select BIT Clock Source

BCK1	BCK0	
0	0	fBIT/2048(default)
0	1	fBIT/256
1	0	fBIT/16
1	1	fBIT/2

BWDTRC — Select BIT Clock Source to WDTRC

0 fSYS

1 fWDTRC

BCLR — If BCLR Bit is written to '1', BIT Counter is cleared as '0', After one machine cycle BCLR is cleared automatically.

0 Free Running

1 Clear Counter

BPD[2:0] — Select BIT overflow period (BIT Clock ≒7.8kHz, default)

BPD2	BPD1	BPD0	
0	0	0	0.256ms(BIT Clock * 2)
0	0	1	0.512ms
0	1	0	1.024ms
0	1	1	2.048ms
1	0	0	4.096ms
1	0	1	8.192ms (default)
1	1	0	16.384ms
1	1	1	32.768ms

BITR (Basic Interval Timer Register) : 8CH

7	6	5	4	3	2	1	0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
R	R	R	R	R	R	R	R

Initial value : 00H

BIT[7:0] BIT Counter

11.3 WDT

11.3.1 Overview

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or the like, and resumes the CPU to the normal state. The watchdog timer signal for detecting malfunction can be selected either a reset CPU or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. It is possible to use free running 8-bit timer mode (WDTRSON='0') or watch dog timer mode (WDTRSON='1') as setting WDTMR[6] bit. If writing WDTMR[5] to '1', WDT counter value is cleared and counts up. After 1 machine cycle, this bit has '0' automatically. The watchdog timer consists of 8-bit binary counter and the watchdog timer data register. When the value of 8-bit binary counter is equal to the 8 bits of WDTR, the interrupt request flag is generated. This can be used as Watchdog timer interrupt or reset the CPU in accordance with the bit WDTRSON.

WDT has BIT overflow output as default clock source. And by selecting WDTCK bit in WDTMR register, BIT clock source is selected as WDT clock source. The interval of watchdog timer interrupt is decided by BIT overflow period and WDTR set value. The equation is as below

WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTR Value+1)

11.3.2 Block Diagram

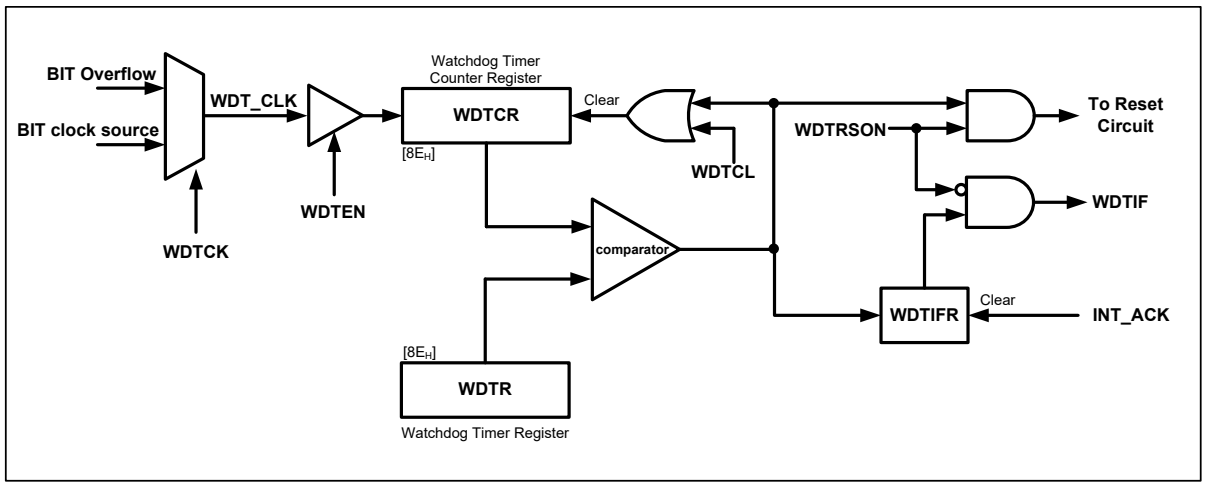


Figure 4.3 WDT Block Diagram

11.3.3 Register Map

Name	Address	Dir	Default	Description
WDTR	8EH	W	FFH	Watch Dog Timer Register
WDTCR	8EH	R	00H	Watch Dog Timer Counter Register
WDTMR	8DH	R/W	00H	Watch Dog Timer Mode Register

Table 11-3. Register Map

11.3.4 Register description for Watch Dog Timer

WDTR (Watch Dog Timer Register: Write Case) : 8EH

7	6	5	4	3	2	1	0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTR[7:0] Set a period
WDT Interrupt Interval=(BIT Interrupt Interval) x(WDTR Value+1)

Note) To guarantee proper operation, the data should be greater than 01H.

WDTCR (Watch Dog Timer Counter Register: Read Case) : 8EH

7	6	5	4	3	2	1	0
WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCR[7:0] WDT Counter

WDTMR (Watch Dog Timer Mode Register) : 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	WDTCK	-	-	-	WDTIFR
R/W	R/W	R/W	R/W	-	-	-	R/W

Initial value : 00H

WDTEN Control WDT operation
- 0 disable
- 1 enable

WDTRSON Control WDT Reset operation
- 0 Free Running 8-bit timer
- 1 Watch Dog Timer Reset ON

WDTCL Clear WDT Counter.
This bit is cleared automatically after 1 machine cycle
- 0 Free Run
- 1 Clear WDT Counter

WDTCK WDT Clock Source
- 0 BIT Overflow
- 1 BIT Clock Source

WDTIFR When WDT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal.
- 0 WDT Interrupt no generation
- 1 WDT Interrupt generation

11.3.5 WDT Interrupt Timing Waveform

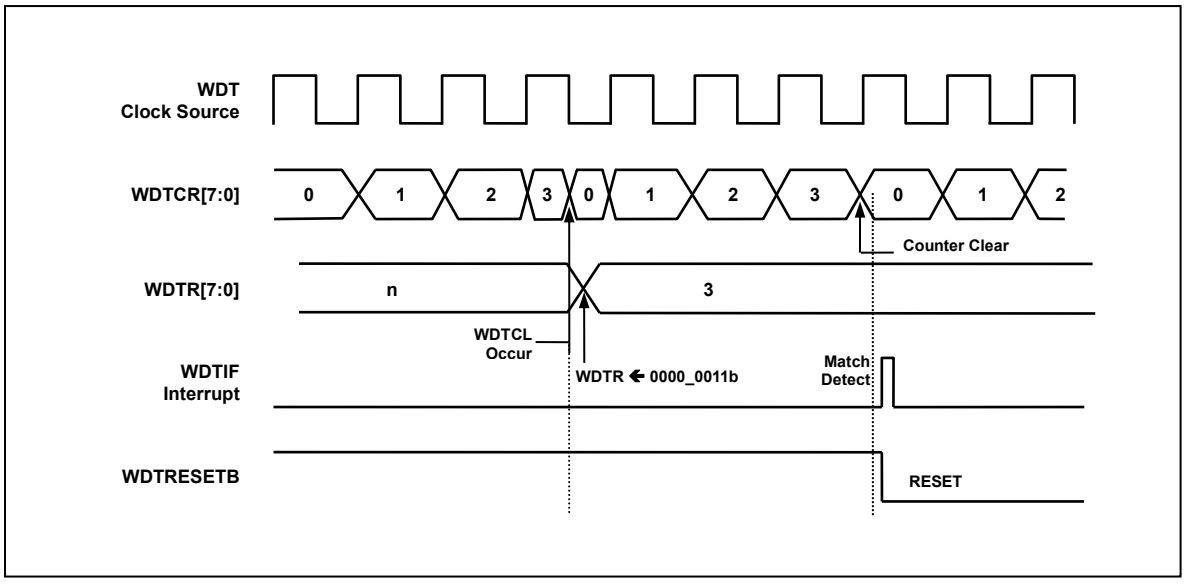


Figure 4.4 WDT Interrupt Timing Waveform

11.4 Timer/PWM

11.4.1 Overview

The 16-bit timer x(0~3) consists of Multiplexer, Timer Data Register High/Low, Timer Register High/Low, Timer Mode Control Register, PWM Duty High/Low, PWM Period High/Low Register It is able to use internal 16-bit timer/ counter without a port output function.

The 16-bit timer x can be clocked by internal or external clock source (EC0, EC1), the divided clock of the main clock selected from prescaler output.

11.4.2 16-Bit Timer/Counter Mode

In the 16-bit Timer/Counter Mode, If the TxH + TxL value and the TxDRH + TxDRL value are matched, Tx/PWMx port outputs. The output is 50:50 of duty square wave, the frequency is following

$$f_{COMP} = \frac{\text{Timer Clock Frequency}}{2 \times \text{Prescaler Value} \times (TxDR + 1)}$$

f_{COMP} is timer output frequency and TxDR is the 16 bits value of TxDRH and TxDRL.

To export the compare output as Tx/PWMx, the Tx_PE bit in the TxCR1 register must set to '1'.

The 16-bit Timer/Counter Mode is selected by control registers as shown in Figure 4.5.

When TxH, TxL are read, TxL should be read first. Because when TxL is read TxH is captured to buffer, and when TxH is read captured value of TxH is read.

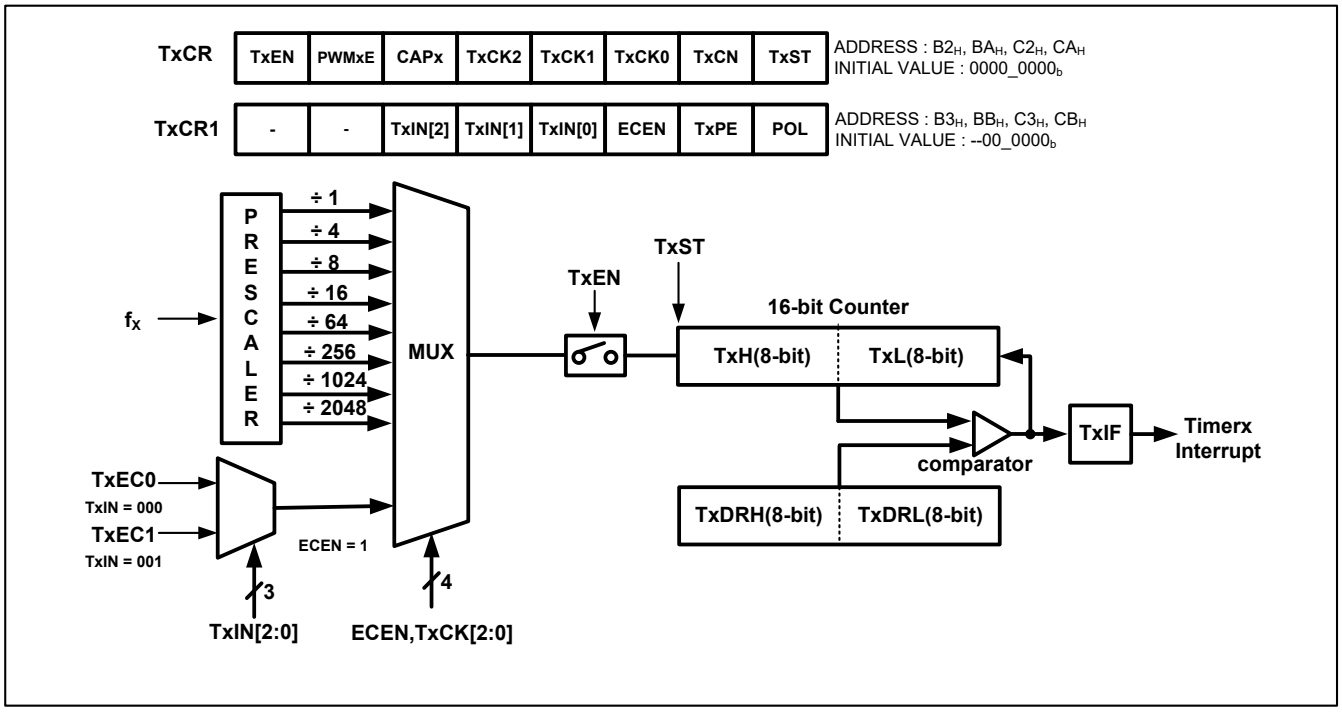


Figure 4.5 Timerx 16-bit Mode Block Diagram

11.4.3 16-Bit Capture Mode

The timer x(0~3) capture mode is set by CAPx as '1' in TxCR register. The clock is same source as Output Compare mode. The interrupt occurs at TxH, TxL and TxDRH, TxDRL matching time. The capture result is loaded into CDRxH, CDRxL. The TxH, TxL value is automatically cleared(0000_H) by hardware and restarts counter.

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. As the EIEDGE and EIPOLA and EIBOTH register setting, the external interrupt INTx function is chosen.

The CDRxH, PWMxDRH and TxH are in same address. In the capture mode, reading operation is read the CDRxH, not TxH because path is opened to the CDRxH. PWMxDRH will be changed in writing operation. The PWMxDRL, TxL, CDRxL has the same function.

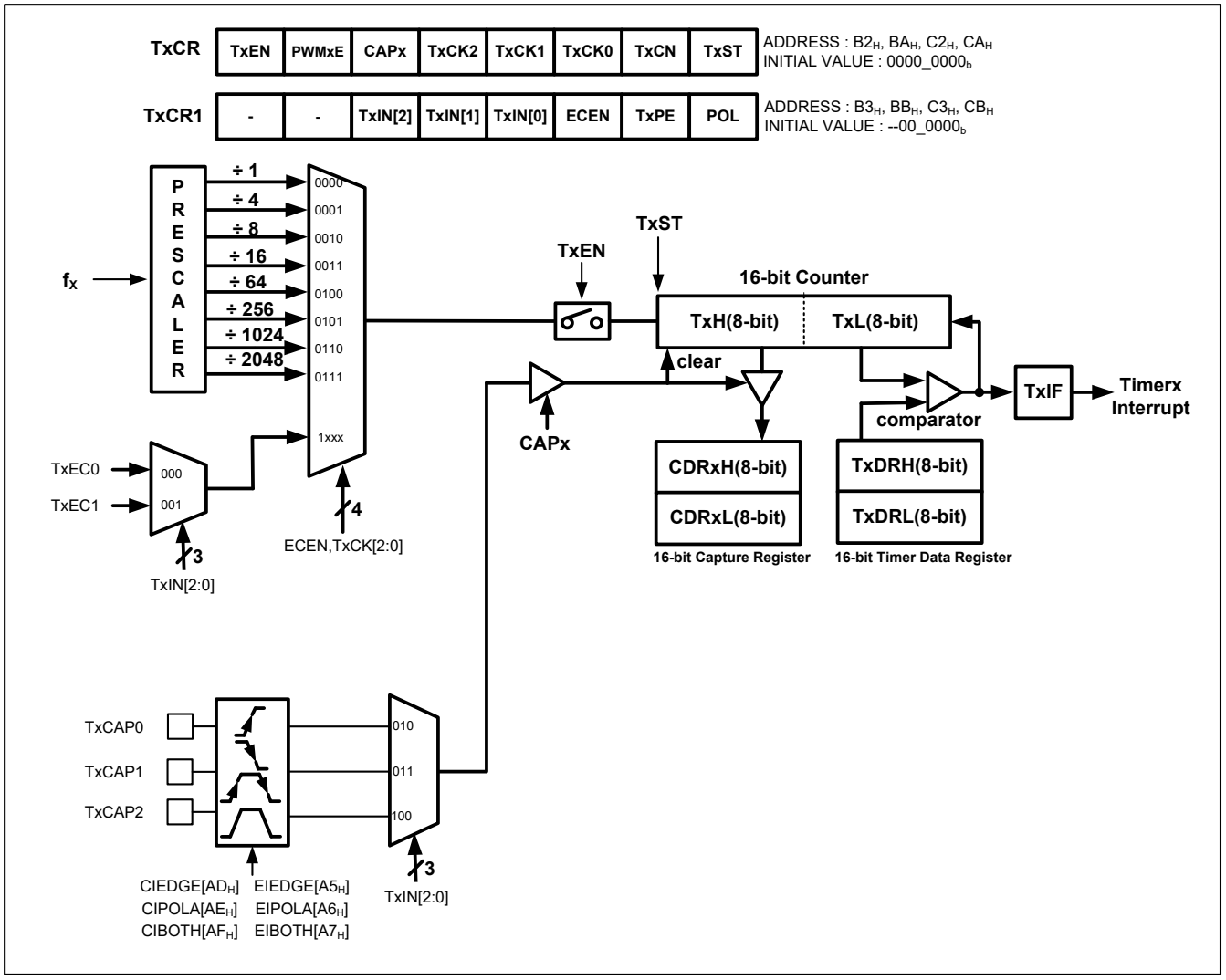


Figure 4.6 Timerx 16bit Capture Mode

11.4.4 Capture & Event Counter Source for Timer and PPG

Each timer x and PPG has different Capture sources and Event Counter sources as shown in Table 11-4

	TxEC0	TxEC1	TxCAP0	TxCAP1	TxCAP2
Timer 0	CPOUT0	EC1(P01)	EINT0(P02)	CMP0IF	EINT2(P04)
Timer 1	EC0(P10)	CPOUT1	EINT0(P02)	EINT1(P03)	CMP1IF
Timer 2	CPOUT2	EC1(P01)	EINT0(P02)	CMP2IF	EINT2(P04)
Timer 3	EC0(P10)	CPOUT3	EINT0(P02)	EINT1(P03)	CMP3IF
PPG	-	-	CMP3IF	CMP1IF	CMP4IF

Table 11-4. Capture & Event Counter source

11.4.5 PWM Mode

The timer x(0~3) has a PWM (pulse Width Modulation) function. In PWM mode, the Tx/PWMx output pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by set TX_PE to '1'. The PWM output mode is determined by the PWMxPRH, PWMxPRL, PWMxDRH and PWMxDRL. And you should configure PWMxE bit to "1" in TxCR register before write to PWM registers.

PWM Period = ({PWMxPRH, PWMxPRL} + 1) X Timerx Clock Period

PWM Duty = ({PWMxDRH, PWMxDRL} + 1) X Timerx Clock Period

Resolution	Frequency		
	TxCK[2:0]=000 (62.5ns)	TxCK[2:0]=001(250ns)	TxCK[2:0]=010(500ns)
16-bit	244.141Hz	61.035Hz	30.517Hz
15-bit	488.281Hz	122.07Hz	61.035Hz
10-bit	15.625kHz	3.906kHz	1.953kHz
9-bit	31.250kHz	7.812kHz	3.906kHz
8-bit	62.500kHz	15.625kHz	7.812kHz

Table 11-5. PWM Frequency vs. Resolution at 16 Mhz

In PWM mode, the duty value and counter matching enables the period value and counter comparison. After counter and the period value matching, counter restarts. If the duty value is set same to the period value, counter doesn't restart after the duty value and counter matching. It is highly recommended that the duty value is not set same to the period value.

PWM Period and Duty same output shown in

Example of PWM at 16MHz (Period = Duty).

The POL bit of TxCR register decides the polarity of duty cycle.

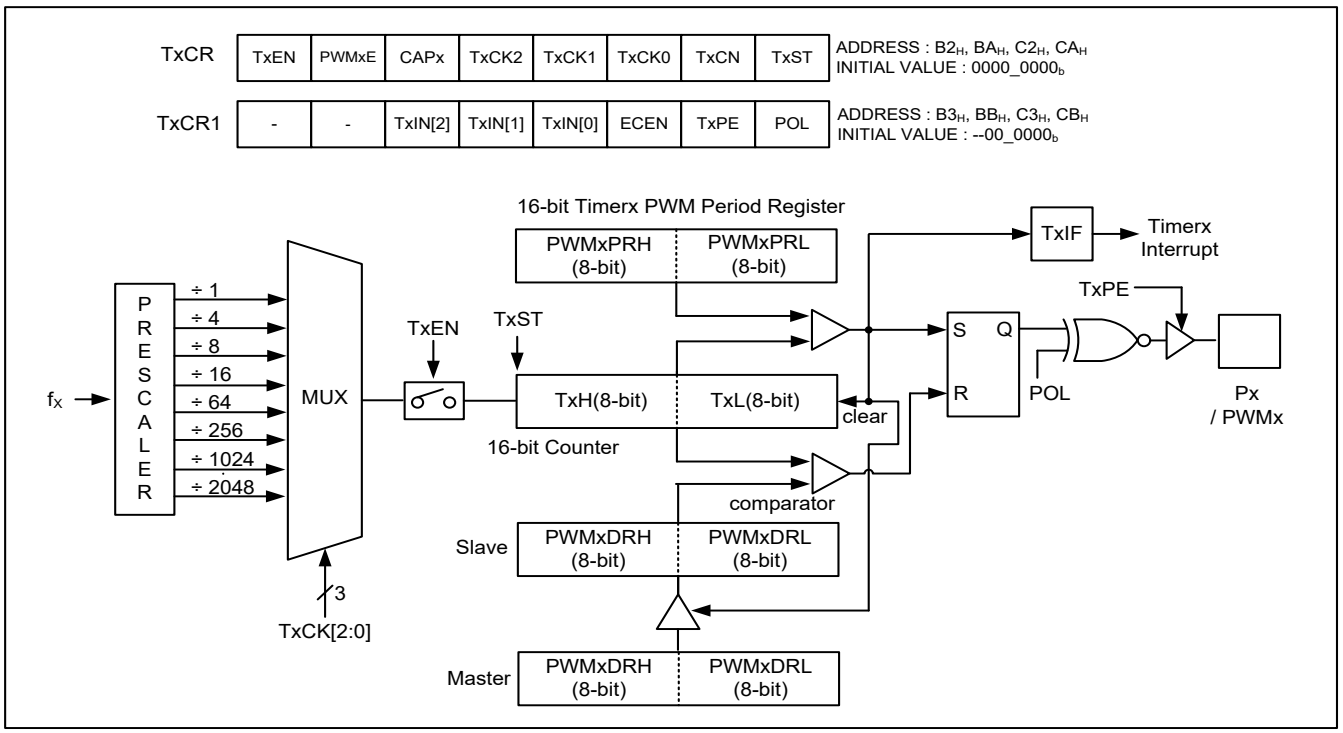


Figure 4.7 PWM Mode

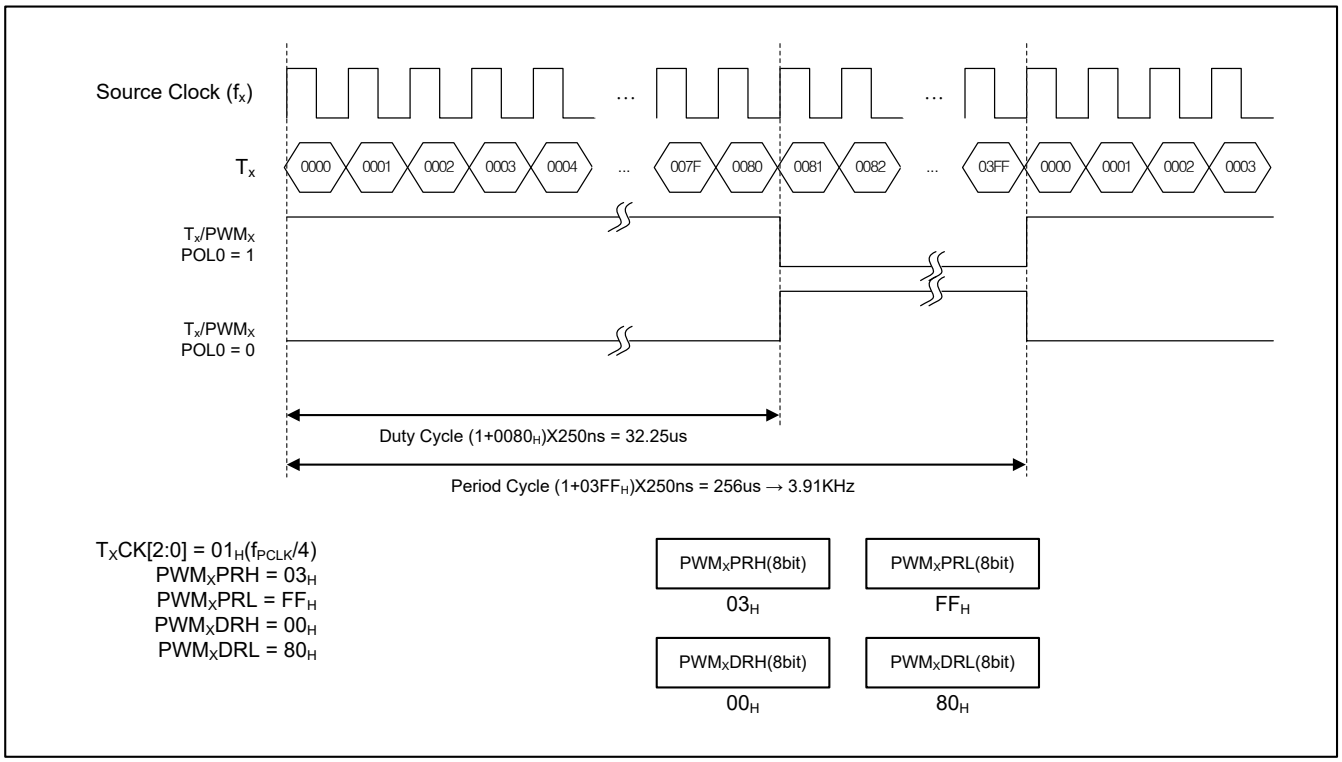


Figure 4.8 Example of PWM at 16MHz

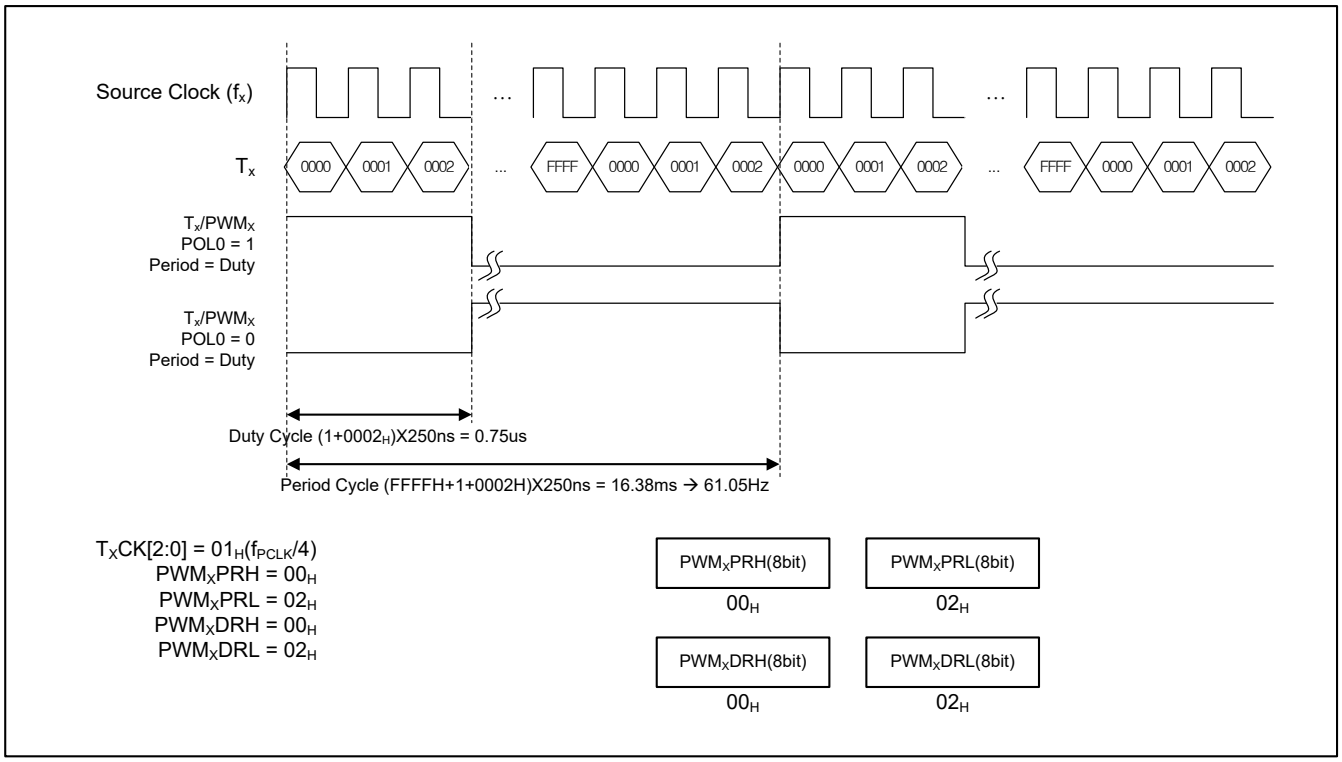


Figure 4.9 Example of PWM at 16MHz (Period = Duty)

11.4.6 Register Map

Name	Address	Dir	Default	Description
T0CR	$B2_H$	R/W	00_H	Timer 0 Mode Control Register
T0CR1	$B3_H$	R/W	00_H	Timer 0 Mode Control Register 1
T0L	$B4_H$	R	00_H	Timer 0 Low Register
PWM0DRL	$B4_H$	R/W	00_H	PWM 0 Duty Register Low
CDR0L	$B4_H$	R	00_H	Timer 0 Capture Data Register Low
T0H	$B5_H$	R	00_H	Timer 0 Register High
PWM0DRH	$B5_H$	R/W	00_H	PWM 0 Duty Register High
CDR0H	$B5_H$	R	00_H	Timer 0 Capture Data Register High
T0DRL	$B6_H$	R/W	FF_H	Timer 0 Data Register Low
PWM0PRL	$B6_H$	R/W	FF_H	PWM 0 Period Register Low
T0DRH	$B7_H$	R/W	FF_H	Timer 0 Data Register High
PWM0PRH	$B7_H$	R/W	FF_H	PWM 0 Period Register High
T1CR	BA_H	R/W	00_H	Timer 1 Mode Control Register
T1CR1	BB_H	R/W	00_H	Timer 1 Mode Control Register 1
T1L	BC_H	R	00_H	Timer 1 Register Low
PWM1DRL	BC_H	R/W	00_H	PWM 1 Duty Register Low
CDR1L	BC_H	R	00_H	Timer 1 Capture Data Register Low
T1H	BD_H	R	00_H	Timer 1 Register High
PWM1DRH	BD_H	R/W	00_H	PWM 1 Duty Register High
CDR1H	BD_H	R	00_H	Timer 1 Capture Data Register High
T1DRL	BE_H	R/W	FF_H	Timer 1 Data Register Low
PWM1PRL	BE_H	R/W	FF_H	PWM 1 Period Register Low
T1DRH	BF_H	R/W	FF_H	Timer 1 Data Register High
PWM1PRH	BF_H	R/W	FF_H	PWM 1 Period Register High
T2CR	$C2_H$	R/W	00_H	Timer 2 Mode Control Register
T2CR1	$C3_H$	R/W	00_H	Timer 2 Mode Control Register 1
T2L	$C4_H$	R	00_H	Timer 2 Register Low
PWM2DRL	$C4_H$	R/W	00_H	PWM 2 Duty Register Low
CDR2L	$C4_H$	R	00_H	Timer 2 Capture Data Register Low
T2H	$C5_H$	R	00_H	Timer 2 Register High
PWM2DRH	$C5_H$	R/W	00_H	PWM 2 Duty Register High
CDR2H	$C5_H$	R	00_H	Timer 2 Capture Data Register High
T2DRL	$C6_H$	R/W	FF_H	Timer 2 Data Register Low
PWM2PRL	$C6_H$	R/W	FF_H	PWM 2 Period Register Low
T2DRH	$C7_H$	R/W	FF_H	Timer 2 Data Register High
PWM2PRH	$C7_H$	R/W	FF_H	PWM 2 Period Register High
T3CR	CA_H	R/W	00_H	Timer 3 Mode Control Register
T3CR1	CB_H	R/W	00_H	Timer 3 Mode Control Register 1
T3L	CC_H	R	00_H	Timer 3 Register Low
PWM3DRL	CC_H	R/W	00_H	PWM 3 Duty Register Low
CDR3L	CC_H	R	00_H	Timer 3 Capture Data Register Low

Table 11-6. Register Map

Name	Address	Dir	Default	Description
T3H	CD_H	R	00_H	Timer 3 Register High
PWM3DRH	CD_H	R/W	00_H	PWM 3 Duty Register High
CDR3H	CD_H	R	00_H	Timer 3 Capture Data Register High
T3DRL	CE_H	R/W	FF_H	Timer 3 Data Register Low
PWM3PRL	CE_H	R/W	FF_H	PWM 3 Period Register Low
T3DRH	CF_H	R/W	FF_H	Timer 3 Data Register High
PWM3PRH	CF_H	R/W	FF_H	PWM 3 Period Register High
TMISR	$D5_H$	W	00_H	Timer x Interrupt Flag Clear Register

Table 11-7. Register Map (Continued)

11.4.7 Register description for Timer/Counter x

TxCR (Timer 0~3 Mode Control Register): T0CR(B2H), T1CR(BAH), T2CR(C2H), T3CR(CAH)

7	6	5	4	3	2	1	0
TxEN	PWMxE	CAPx	TxCK2	TxCK1	TxCK0	TxCN	TxST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

TxEN Control Timer X
- 0 0
- 1 Timer X enable

PWMxE Control PWM enable
- 0 PWM disable
- 1 PWM enable

CAPx Control Timer X capture mode.
- 0 Timer/Counter mode
- 1 Capture mode

TxCK[2:0] Select clock source of Timer X. Fx is the frequency of main system

TxCK2	TxCK1	TxCK0	description
0	0	0	f_X
0	0	1	f_X /4
0	1	0	f_X /8
0	1	1	f_X /16
1	0	0	f_X /64
1	0	1	f_X /256
1	1	0	f_X /1024
1	1	1	f_X /2048

TxCN Control Timer X Count pause/continue.
- 0 Temporary count stop
- 1 Continue count

TxST Control Timer X start/stop
- 0 Counter stop
- 1 Clear counter and start

NOTE)

1. set TxST bit after write to Tx, PWM, CDRx registers.

TxCR1 (Timer 0~3 Mode Control Register 1) : B3H, BBH, C3H, CBH

7	6	5	4	3	2	1	0
-	-	TxIN[2]	TxIN[1]	TxIN[0]	ECEN	Tx_PE	POL
-	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

TxIN[2:0] Select Event Counter and External Interrupt for Capture mode

TxIN2	TxIN1	TxIN0	description
0	0	0	TxEC0
0	0	1	TxEC1
0	1	0	TxCAP0
0	1	1	TxCAP1
1	0	0	TxCAP2
1	0	1	-
1	1	0	-
1	1	1	-

ECEN Control Event Counter

0 Event Counter disable

1 Event Counter enable

Tx_PE Control Timer X Output port

0 Timer X Output disable

1 Timer X Output enable

POL Configure PWM polarity

0 Negative (Duty Match: Clear)

1 Positive (Duty Match: Set)

TxL (Timer 0~3 Register Low, Read Case) : B4H, BCH, C4H, CCH

7	6	5	4	3	2	1	0
TxL7	TxL6	TxL5	TxL4	TxL3	TxL2	TxL1	TxL0
R	R	R	R	R	R	R	R

Initial value : 00_H

TxL[7:0] TxL Counter Period Low data.

CDRxL (Capture 0~3 Data Register Low, Read Case) : B4H, BCH, C4H, CCH

7	6	5	4	3	2	1	0
CDRxL07	CDRxL06	CDRxL05	CDRxL04	CDRxL03	CDRxL02	CDRxL01	CDRxL00
R	R	R	R	R	R	R	R

Initial value : 00_H

CDRxL[7:0] Tx Capture Low data.

PWMxDRL (PWM 0~3 Duty Register Low, Write Case) : B4H, BCH, C4H, CCH

7	6	5	4	3	2	1	0
PWMxDRL7	PWMxDRL6	PWMxDRL5	PWMxDRL4	PWMxDRL3	PWMxDRL2	PWMxDRL1	PWMxDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PWMxDRL[7:0] Tx PWM Duty Low data

Note) Reading and writing is possible only when PWMxE = 1 and TxST = 0

TxH (Timer 0~3 Register High, Read Case) : B5H, BDH, C5H, CDH

7	6	5	4	3	2	1	0
TxH7	TxH6	TxH5	TxH4	TxH3	TxH2	TxH1	TxH0
R	R	R	R	R	R	R	R

Initial value : 00_H

TxH[7:0] TxH Counter Period High data.

CDRxH (Capture 0~3 Data High Register, Read Case) : B5H, BDH, C5H, CDH

7	6	5	4	3	2	1	0
CDRxH07	CDRxH06	CDRxH05	CDRxH04	CDRxH03	CDRxH02	CDRxH01	CDRxH00
R	R	R	R	R	R	R	R

Initial value : 00_H

CDRxH[7:0] Tx Capture High data

PWMxDRH (PWM0~3 Duty Register High, Write Case) : B5H, BDH, C5H, CDH

7	6	5	4	3	2	1	0
PWMxDRH7	PWMxDRH6	PWMxDRH5	PWMxDRH4	PWMxDRH3	PWMxDRH2	PWMxDRH1	PWMxDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PWMxDRH[7:0] Tx PWM Duty High data

Note) Reading and writing is effective only when PWMxE = 1 and TxST = 0

TxDRL (Timer 0~3 Data Register Low, Write Case) : B6H, BEH, C6H, CEH

7	6	5	4	3	2	1	0
TxDRL7	TxDRL6	TxDRL5	TxDRL4	TxDRL3	TxDRL2	TxDRL1	TxDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

TxDRL[7:0] TxL Compare Low data

Note) Be sure to clear PWMxE before loading this register.

PWMxPRL (PWM 0~3 Period Register Low, Write Case) : B6H, BEH, C6H, CEH

7	6	5	4	3	2	1	0
PWMxPRL7	PWMxPRL6	PWMxPRL5	PWMxPRL4	PWMxPRL3	PWMxPRL2	PWMxPRL1	PWMxPRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

PWMxPRL[7:0] Tx PWM Period Low data

Note) Reading and writing is effective only when PWMxE = 1 and TxST = 0

TxDRH (Timer 0~3 Data Register High, Write Case) : B7H, BFH, C7H, CFH

7	6	5	4	3	2	1	0
TxDRH7	TxDRH6	TxDRH5	TxDRH4	TxDRH3	TxDRH2	TxDRH1	TxDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

TxDRH[7:0] TxH Compare High data

Note) Be sure to clear PWMxE before loading this register.

PWMxPRH (PWM 0~3 Period Register High, Write Case) : B7H, BFH, C7H, CFH

7	6	5	4	3	2	1	0
PWMxPRH7	PWMxPRH6	PWMxPRH5	PWMxPRH4	PWMxPRH3	PWMxPRH2	PWMxPRH1	PWMxPRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

PWMxPRH[7:0] Tx PWM Period High data

Note) Reading and writing is effective only when PWMxE = 1 and TxST = 0

11.4.8 Timer Interrupt Status Register (TMISR)

11.4.8.1 Register description for TMISR

TMISR (Timer Interrupt Status Register) : D5H

7	6	5	4	3	2	1	0
-	-	-	-	TMIF3	TMIF2	TMIF1	TMIF0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 0H

When TIMERx Interrupt occurs, this bit becomes ‘1’. This bit is cleared automatically if TIMERx and global interrupt enable bit is set. For clearing bit, write ‘1’ to this bit.

TMIF3	0	No generation
	1	generation
TMIF2	0	No generation
	1	generation
TMIF1	0	No generation
	1	generation
TMIF0	0	No generation
	1	generation

11.5 BUZZER DRIVER

11.5.1 Overview

The Buzzer consists of 6 bit counter, buzzer data register (BUZDR), and buzzer control register (BUZCR). The Square Wave is outputted through P30/BUZO pin. In buzzer data register BUZDR[5:0] controls the buzzer frequency and BUZDIV[1:0] selects fBUZ divided by DIV block . In buzzer control register (BUZCR), BUCK[2:0] selects source clock divided by prescaler

$$f_{BUZO} = \frac{f_{BUZ}}{2 \times BUZDIV \times (BUZDATA + 1)} (Hz)$$

11.5.2 Buzzer Frequency at f_{BUZ} = 1MHz

BUZDATA [5:0]	BUZDIV[1:0]			
	00 (fbuz/8)	01 (fbuz/16)	10 (fbuz/32)	11 (fbuz/64)
0	62.500	31.250	15.625	7.813
1	31.250	15.625	7.813	3.906
2	20.833	10.417	5.208	2.604
3	15.625	7.813	3.906	1.953
4	12.500	6.250	3.125	1.563
5	10.417	5.208	2.604	1.302
6	8.929	4.464	2.232	1.116
7	7.813	3.906	1.953	0.977
8	6.944	3.472	1.736	0.868
9	6.250	3.125	1.563	0.781
10	5.682	2.841	1.420	0.710
11	5.208	2.604	1.302	0.651
12	4.808	2.404	1.202	0.601
13	4.464	2.232	1.116	0.558
14	4.167	2.083	1.042	0.521
15	3.906	1.953	0.977	0.488
16	3.676	1.838	0.919	0.460
17	3.472	1.736	0.868	0.434
18	3.289	1.645	0.822	0.411
19	3.125	1.563	0.781	0.391
20	2.976	1.488	0.744	0.372
21	2.841	1.420	0.710	0.355
22	2.717	1.359	0.679	0.340
23	2.604	1.302	0.651	0.326
24	2.500	1.250	0.625	0.313
25	2.404	1.202	0.601	0.300
26	2.315	1.157	0.579	0.289
27	2.232	1.116	0.558	0.279
28	2.155	1.078	0.539	0.269
29	2.083	1.042	0.521	0.260
30	2.016	1.008	0.504	0.252
31	1.953	0.977	0.488	0.244

BUZDATA [5:0]	BUZDIV[1:0]			
	00 (fbuz/8)	01 (fbuz/16)	10 (fbuz/32)	11 (fbuz/64)
32	1.894	0.947	0.473	0.237
33	1.838	0.919	0.460	0.230
34	1.786	0.893	0.446	0.223
35	1.736	0.868	0.434	0.217
36	1.689	0.845	0.422	0.211
37	1.645	0.822	0.411	0.206
38	1.603	0.801	0.401	0.200
39	1.563	0.781	0.391	0.195
40	1.524	0.762	0.381	0.191
41	1.488	0.744	0.372	0.186
42	1.453	0.727	0.363	0.182
43	1.420	0.710	0.355	0.178
44	1.389	0.694	0.347	0.174
45	1.359	0.679	0.340	0.170
46	1.330	0.665	0.332	0.166
47	1.302	0.651	0.326	0.163
48	1.276	0.638	0.319	0.159
49	1.250	0.625	0.313	0.156
50	1.225	0.613	0.306	0.153
51	1.202	0.601	0.300	0.150
52	1.179	0.590	0.295	0.147
53	1.157	0.579	0.289	0.145
54	1.136	0.568	0.284	0.142
55	1.116	0.558	0.279	0.140
56	1.096	0.548	0.274	0.137
57	1.078	0.539	0.269	0.135
58	1.059	0.530	0.265	0.132
59	1.042	0.521	0.260	0.130
60	1.025	0.512	0.256	0.128
61	1.008	0.504	0.252	0.126
62	0.992	0.496	0.248	0.124
63	0.977	0.488	0.244	0.122

Table 11-8. Buzzer Frequency at f_{BUZ} = 1MHz

11.5.3 BLOCK DIAGRAM

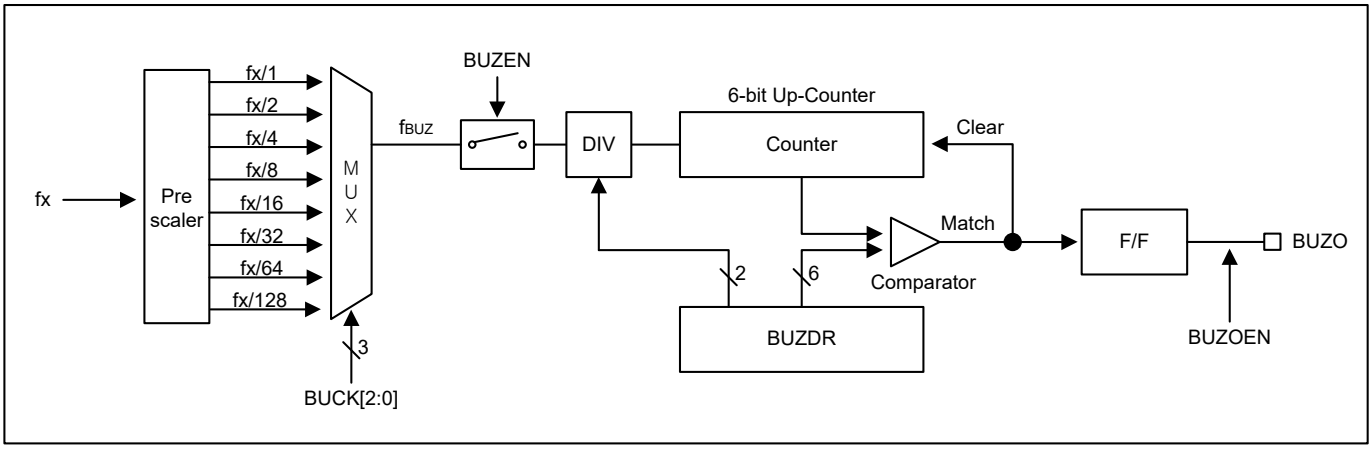


Figure 4.10 BUZZER Driver Block Diagram

11.5.4 REGISTER MAP

Name	Address	Dir	Default	Description
BUZCR	EEH	R/W	00H	BUZZER Control Register
BUZDR	EFH	R/W	FFH	BUZZER Data Register

Table 11-9. Register Map

11.5.5 REGISTER DESCRIPTION FOR BUZZER DRIVER

BUZDR (Buzzer Data Register) : EFH

7	6	5	4	3	2	1	0
BUZDIV1	BUZDIV2	BUZDATA5	BUZDATA4	BUZDATA3	BUZDATA2	BUZDATA1	BUZDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

BUZDIV[1:0] Buzzer Clock Divider

BZDIV1	BUZDIV2	description
0	0	$f_{BUZ}/8$
0	1	$f_{BUZ}/16$
1	0	$f_{BUZ}/32$
1	1	$f_{BUZ}/64$

BUZDATA[5:0] These bits control the BUZZER frequency
Its resolution is 00H to 3FH

BUZCR (Buzzer Control Register) : EEH

7	6	5	4	3	2	1	0
BUZOEN	-	-	-	BUCK2	BUCK1	BUCK0	BUZEN
R/W	-	-	-	R/W	R/W	R/W	R/W

Initial value : 00_H

BUZOEN Control Buzzer output port

0 Buzzer Output disable

1 Buzzer Output enable

BUCK[2:0] Buzzer driver source clock selection

BUCK2	BUCK1	BUCK0	description
0	0	0	fx/1
0	0	1	fx/2
0	1	0	fx/4
0	1	1	fx/8
1	0	0	fx/16
1	0	1	fx/32
1	1	0	fx/64
1	1	1	fx/128

BUZEN Buzzer driver operation control

0 Buzzer driver disable

1 Buzzer driver enable

11.6 USART

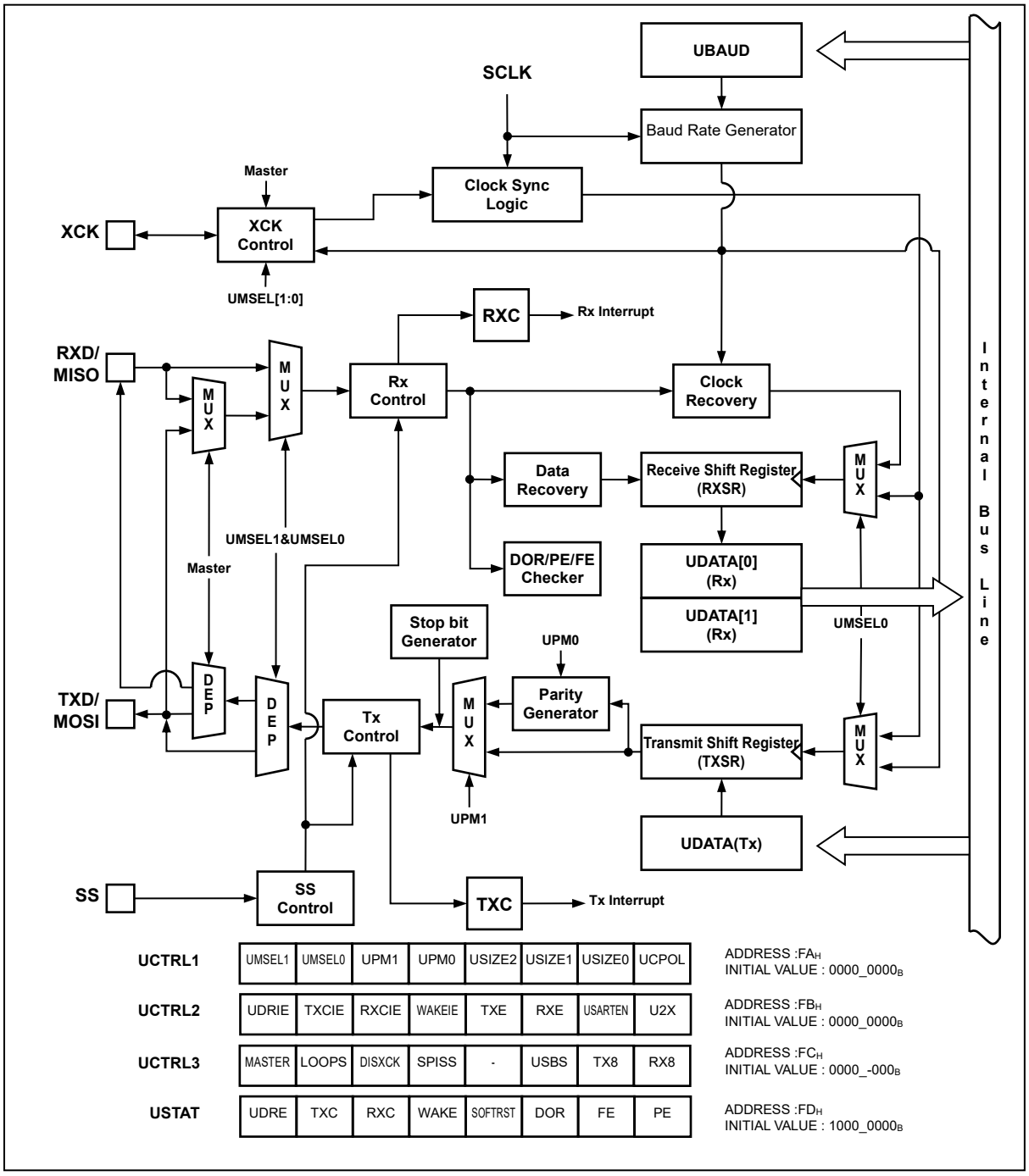
11.6.1 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous and SPI Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, 3)
- LSB First or MSB First Data Transfer @SPI mode
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5,6,7,8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous Communication Mode

USART has three main parts of Clock Generator, Transmitter and Receiver. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the Receiver includes a parity checker, a shift register, a two level receive FIFO (UDATA) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

11.6.2 Block Diagram



UCTRL1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1	USIZE0	UCPOL	ADDRESS :FA_H INITIAL VALUE : 0000_0000_B
UCTRL2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS :FB_H INITIAL VALUE : 0000_0000_B
UCTRL3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS :FC_H INITIAL VALUE : 0000_-000_B
USTAT	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS :FD_H INITIAL VALUE : 1000_0000_B

Figure 4.11 USART Block Diagram

11.6.3 Clock Generation

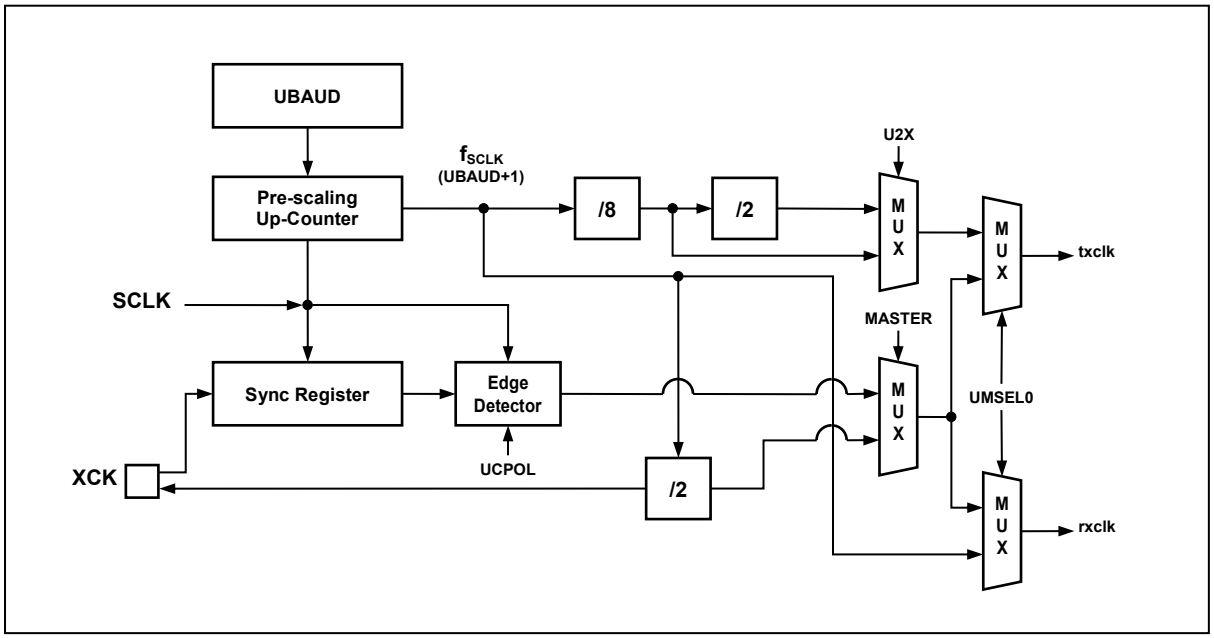


Figure 4.12 Clock Generation Block Diagram

The Clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation and those are Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous. The clock generation scheme for Master SPI and Slave SPI mode is the same as Master Synchronous and Slave Synchronous operation mode. The UMSELn bit in UCTRL1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRL2 register. The MASTER bit in UCTRL2 register controls whether the clock source is internal (Master mode, output port) or external (Slave mode, input port). The XCK pin is only active when the USART operates in Synchronous or SPI mode.

Table below contains equations for calculating the baud rate (in bps).

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (U2X=0)	$\text{Baud Rate} = \frac{\text{fSCLK}}{16(\text{UBAUD}+1)}$
Asynchronous Double Speed Mode (U2X=1)	$\text{Baud Rate} = \frac{\text{fSCLK}}{8(\text{UBAUD}+1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUD}+1)}$

Table 11-11. Equations for Calculating Baud Rate Register Setting

11.6.4 External Clock (XCK)

External clocking is used by the synchronous or spi slave modes of operation.

External clock input from the XCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum frequency of the external XCK pin is limited by the following equation.

$$fXCK = \frac{fSCLK}{4}$$

where fXCK is the frequency of XCK and fSCLK is the frequency of main system clock (SCLK).

11.6.5 Synchronous mode Operation

When synchronous or spi mode is used, the XCK pin will be used as either clock input (slave) or clock output (master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input on RXD (MISO in spi mode) pin is sampled at the opposite XCK clock edge of the edge in the data output on TXD (MOSI in spi mode) pin is changed.

The UCPOL bit in UCTRL1 register selects which XCK clock edge is used for data sampling and which is used for data change. As shown in the figure below, when UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge.

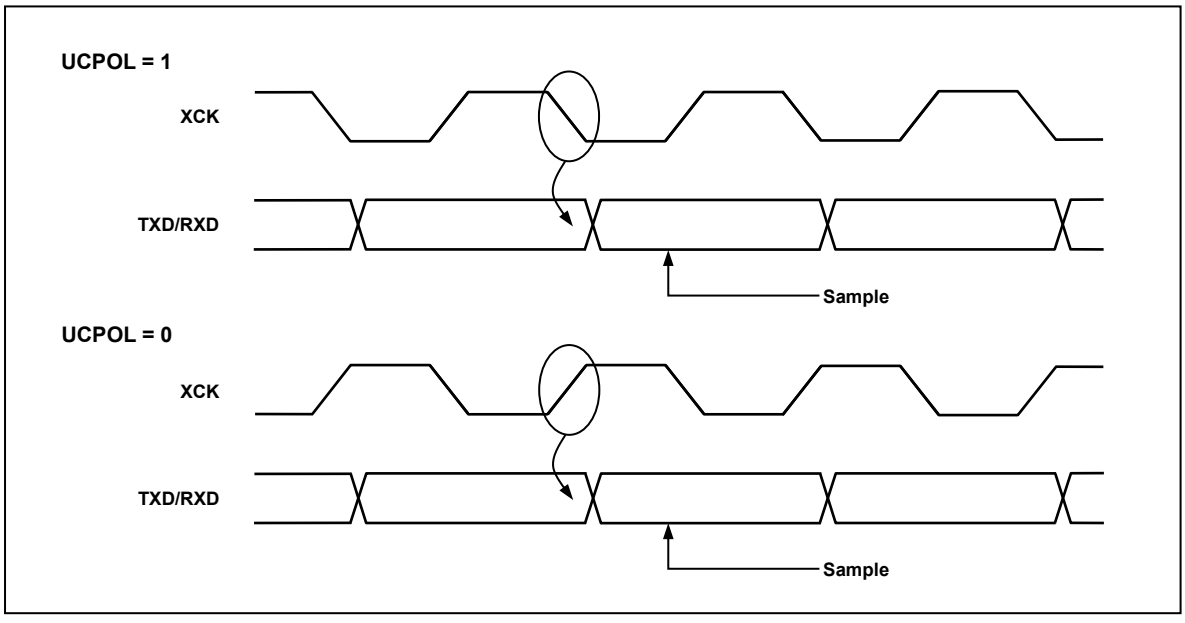


Figure 4.13 Synchronous Mode XCKn Timing

11.6.6 Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.

The USART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit (MSB). If enabled the parity bit is inserted after the data bits, before the stop bits. A high to low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The next figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

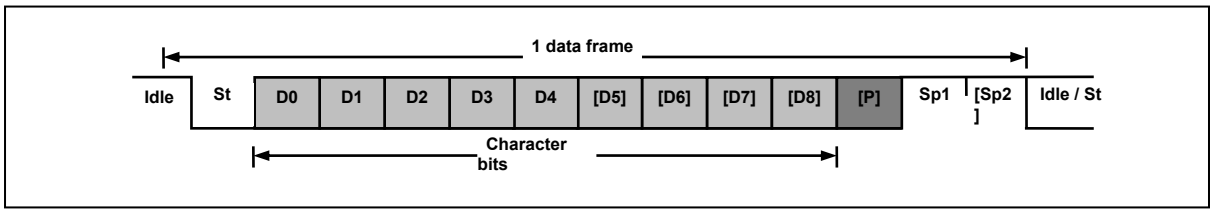


Figure 4.14 frame format

1 data frame consists of the following bits

- Idle No communication on communication line (TxD/RxD)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the USART is set by the USIZE[2:0], UPM[1:0] and USBS bits in UCTRL1 register. The Transmitter and Receiver use the same setting.

11.6.7 Parity bit

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

P_{even} : Parity bit using even parity

P_{odd} : Parity bit using odd parity

D_n : Data bit n of the character

11.6.8 USART Transmitter

The USART Transmitter is enabled by setting the TXE bit in UCTRL1 register. When the Transmitter is enabled, the normal port operation of the TXD pin is overridden by the serial output pin of USART. The baud-rate, operation mode and frame format must be setup once before doing any transmissions. If synchronous or spi operation is used, the clock on the XCK pin will be overridden and used as transmission clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

11.6.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATA register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame at the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0]=7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading transmit buffer (UDATA register).

11.6.8.2 Transmitter flag and interrupt

The USART Transmitter has 2 flags which indicate its state. One is USART Data Register Empty (UDRE) and the other is Transmit Complete (TXC). Both flags can be interrupt sources.

UDRE flag indicates whether the transmit buffer is ready to be loaded with new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the Data Register Empty Interrupt Enable (UDRIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Data Register Empty Interrupt is generated while UDRE flag is set.

The Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC bit in USTAT register.

When the Transmit Complete Interrupt Enable (TXCIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Transmit Complete Interrupt is generated while TXC flag is set.

11.6.8.3 Parity Generator

The Parity Generator calculates the parity bit for the sending serial frame data. When parity bit is enabled (UPM[1]=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the sending frame.

11.6.8.4 Disabling Transmitter

Disabling the Transmitter by clearing the TXE bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD pin is used as normal General Purpose I/O (GPIO) or primary function pin.

11.6.9 USART Receiver

The USART Receiver is enabled by setting the RXE bit in the UCTRL1 register. When the Receiver is enabled, the normal pin operation of the RXD pin is overridden by the USART as the serial input pin of the Receiver. The baud-rate, mode of operation and frame format must be set before serial reception. If synchronous or spi operation is used, the clock on the XCK pin will be used as transfer clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

11.6.9.1 Receiving Rx data

When USART is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RXD pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2nd stop bit in the frame, the 2nd stop bit is ignored by the Receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UDATA register.

If 9-bit characters are used (USIZE[2:0] = 7) the ninth bit is stored in the RX8 bit position in the UCTRL3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the UDATA register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATA register. This is because the error flags are stored in the same FIFO position of the receive buffer.

11.6.9.2 Receiver flag and interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXC) flag indicates whether there are unread data present in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the Receiver is disabled (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

When the Receive Complete Interrupt Enable (RXCIE) bit in the UCTRL2 register is set and Global Interrupt is enabled, the USART Receiver Complete Interrupt is generated while RXC flag is set.

The USART Receiver has three error flags which are Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). These error flags can be read from the USTAT register. As data received are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from UDATA register, read the USTAT register first which contains error flags.

The Frame Error (FE) flag indicates the state of the first stop bit. The FE flag is zero when the stop bit was correctly detected as one, and the FE flag is one when the stop bit was incorrect, ie detected as zero. This flag can be used for detecting out-of-sync conditions between data frames.

The Data OverRun (DOR) flag indicates data loss due to a receive buffer full condition. A DOR occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The Parity Error (PE) flag indicates that the frame in the receive buffer had a Parity Error when received. If Parity Check function is not enabled (UPM[1]=0), the PE bit is always read zero.

Note) The error flags related to receive operation are not used when USART is in SPI mode.

11.6.9.3 Parity Checker

If Parity Bit is enabled (UPM[1]=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

11.6.9.4 Disabling Receiver

In contrast to Transmitter, disabling the Receiver by clearing RXE bit makes the Receiver inactive immediately. When the Receiver is disabled the Receiver flushes the receive buffer and the remaining data in the buffer is all reset. The RXD pin is not overridden the function of USART, so RXD pin becomes normal GPIO or primary function pin.

11.6.9.5 Asynchronous Data Reception

To receive asynchronous data frame, the USART includes a clock and data recovery unit. The Clock Recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXD pin.

The Data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXD pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate for normal mode, and 8 times the baud rate for Double Speed mode (U2X=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the Double Speed mode.

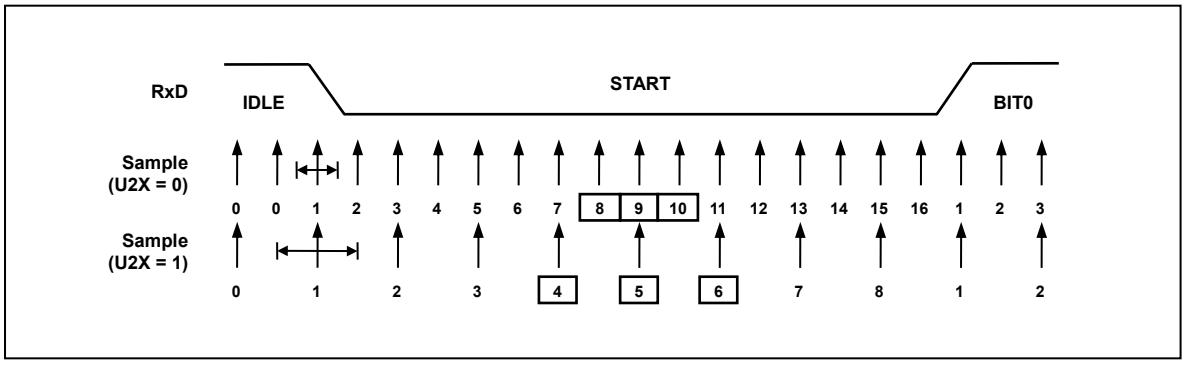


Figure 4.15 Start Bit Sampling

When the Receiver is enabled (RXE=1), the clock recovery logic tries to find a high to low transition on the RXD line, the start bit condition. After detecting high to low transition on RXD line, the clock recovery logic uses samples 8,9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the Receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for Normal mode and 8 times for Double Speed mode. And uses sample 8, 9, and 10 to decide data value for Normal mode, samples 4, 5, and 6 for Double Speed mode. If more than 2 samples have low levels, the received bit is considered to a logic 0 and more than 2 samples have high levels, the received bit is considered to a logic 1. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

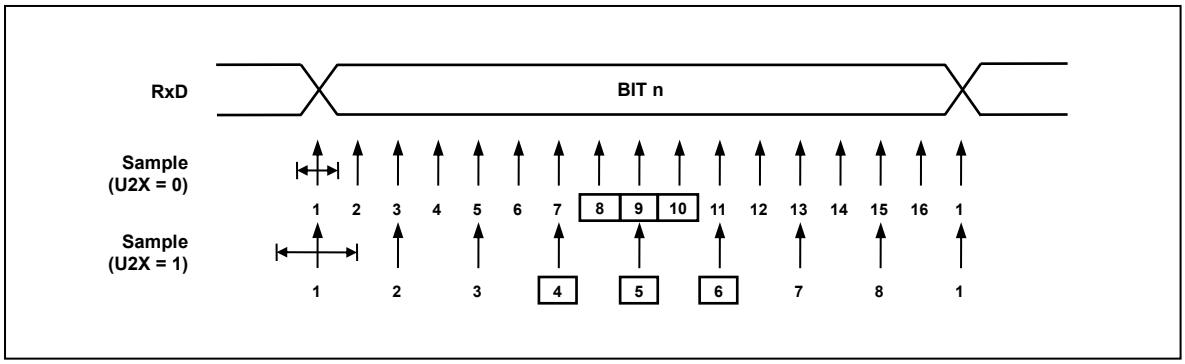


Figure 4.16 Sampling of Data and Parity Bit

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a Frame Error flag is set. After deciding first stop bit, whether a valid stop bit is received or not, the Receiver goes idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

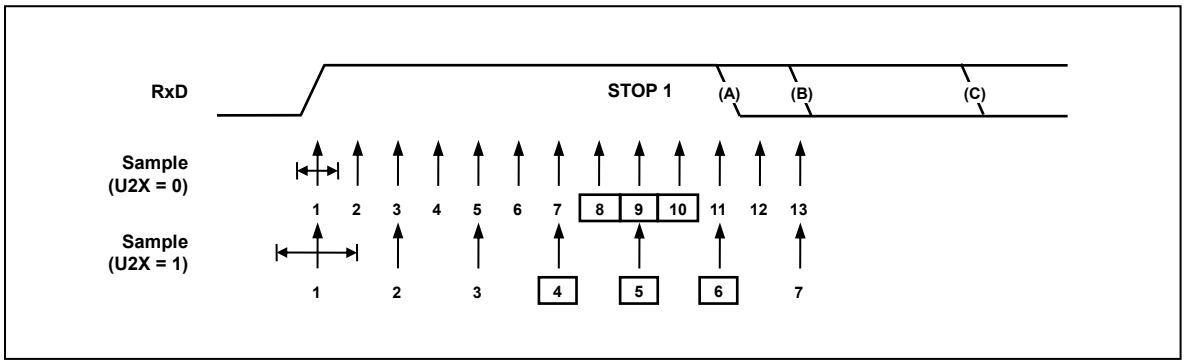


Figure 4.17 Stop Bit Sampling and Next Start Bit Sampling

11.6.10 SPI Mode

The USART can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full duplex, three-wire synchronous data transfer
- Master or Slave operation
- Supports all four SPI modes of operation (mode0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (UMSEL[1:0]=3), the Slave Select (SS) pin becomes active low input in slave mode operation, or can be output in master mode operation if SPISS bit is set.

Note that during SPI mode of operation, the pin RXD is renamed as MISO and TXD is renamed as MOSI for compatibility to other SPI devices.

11.6.10.1 SPI Clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (UCPOL) and a clock phase control bit (UCPHA) to select one of four clock formats for data transfers. UCPOL selectively insert an inverter in series with the clock. UCPHA chooses between two different clock phase relationships between the clock and data. Note that UCPHA and UCPOL bits in UCTRL1 register have different meanings according to the UMSEL[1:0] bits which decides the operating mode of USART.

Table below shows four combinations of UCPOL and UCPHA for SPI mode 0, 1, 2, and 3.

SPI Mode	UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

Table 11-12. CPOL Funtionality

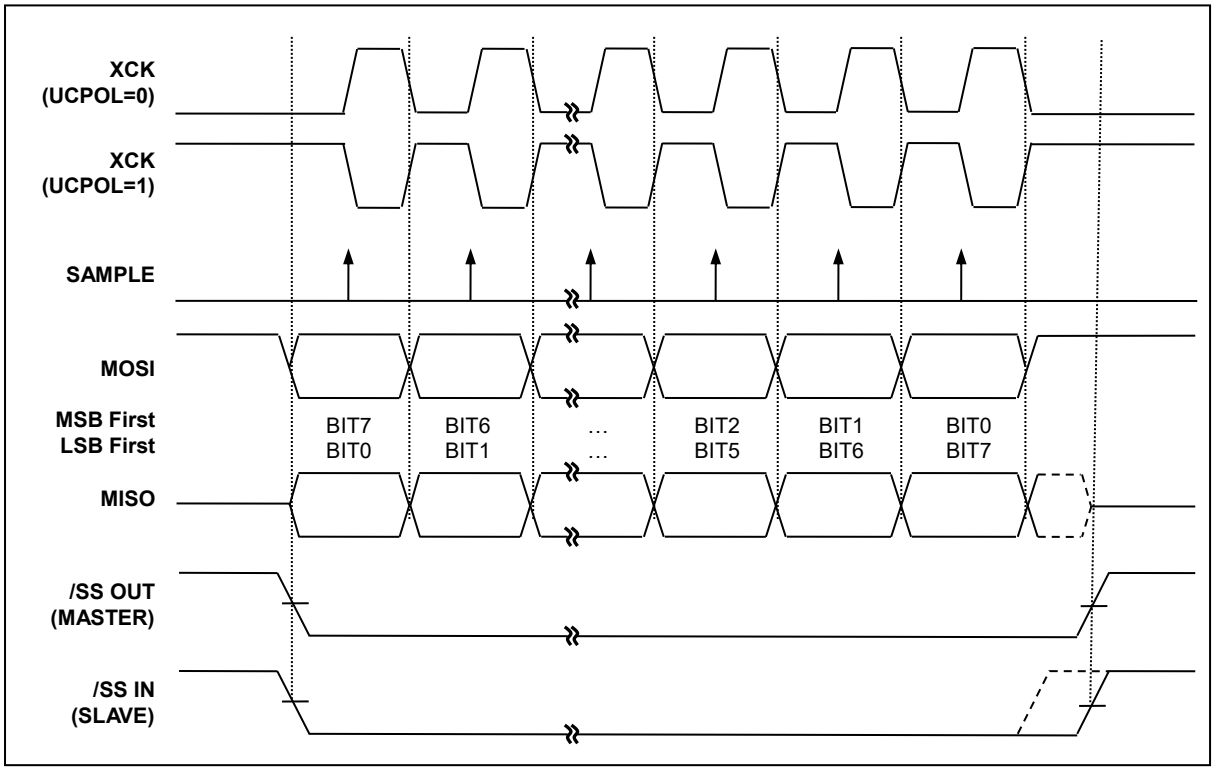


Figure 4.18 SPI Clock Formats when UCPHA=0

When UCPHA=0, the slave begins to drive its MISO output with the first data bit value when SS goes to active low. The first XCK edge causes both the master and the slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the second XCK edge, the USART shifts the second data bit value out to the MOSI and MISO outputs of the master and slave, respectively. Unlike the case of UCPHA=1, when UCPHA=0, the slave's SS input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SS input.

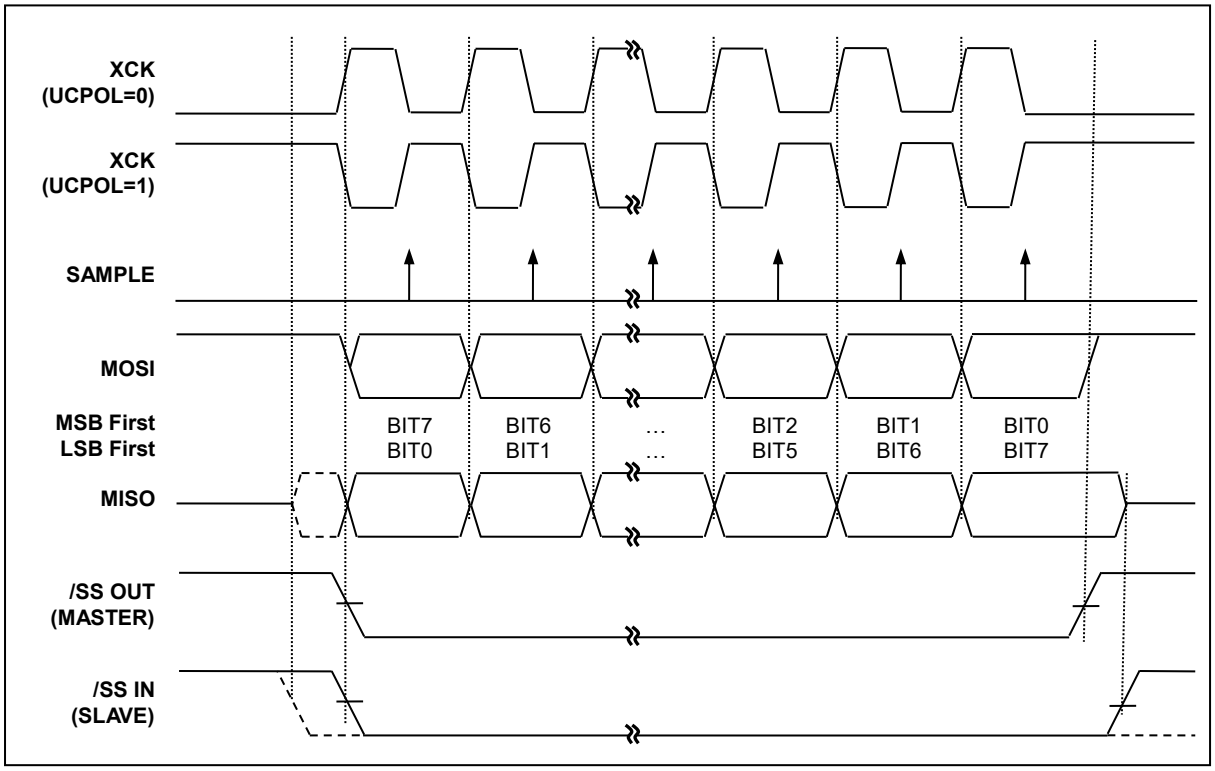


Figure 4.19 SPI Clock Formats when UCPHA=1

When UCPHA=1, the slave begins to drive its MISO output when SS goes active low, but the data is not defined until the first XCK edge. The first XCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next XCK edge causes both the master and slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the third XCK edge, the USART shifts the second data bit value out to the MOSI and MISO output of the master and slave respectively. When UCPHA=1, the slave's SS input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USART resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USART Data Register Empty flag (UDRE=1) and then writing a byte of data to the UDATA Register. In master mode of operation, even if transmission is not enabled (TXE=0), writing data to the UDATA register is necessary because the clock XCK is generated from transmitter block.

11.6.11 Register Map

Name	Address	Dir	Default	Description
UCTRL1	FAH	R/W	00H	USART Control 1 Register
UCTRL2	FBH	R/W	00H	USART Control 2 Register
UCTRL3	FCH	R/W	00H	USART Control 3 Register
USTAT	FDH	R	80H	USART Status Register
UBAUD	FEH	R/W	FFH	USART Baud Rate Generation Register
UDATA	FFH	R/W	FFH	USART Data Register

Table 11-13. Register Map

11.6.12 Register description for USART

UCTRL1 (USART Control 1 Register) : FAH

7	6	5	4	3	2	1	0
UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1 UDORD	USIZE0 UCPHA	UCPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

UMSEL[1:0] Selects operation mode of USART

UMSEL1	UMSEL0	Operating Mode
0	0	Asynchronous Mode (Normal Uart)
0	1	Synchronous Mode (Synchronous Uart)
1	0	Reserved
1	1	SPI Mode

UPM[1:0] Selects Parity Generation and Check methods

UPM1	UPM0	Parity mode
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USIZE[2:0] Selects the length of data bits in frame.

USIZE2	USIZE1	USIZE0	Data length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9 bit

UDORD This bit is in the same bit position with USIZE1. Select the data transmission sequence in SPI mode.

0	LSB First
1	MSB First

UCPOL Selects polarity of XCK in synchronous or spi mode

0	TXD change @Rising Edge, RXD change @Falling Edge
1	TXD change @ Falling Edge, RXD change @ Rising Edge

UCPHA This bit is in the same bit position with USIZE0. In SPI mode, along with UCPOL bit, selects one of two clock formats for different kinds of synchronous serial peripherals. Leading edge means first XCK edge and trailing edge means 2nd or last clock edge of XCK in one XCK pulse. And Sample means detecting of incoming receive bit, Setup means preparing transmit data.

UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

UCTRL2 (USART Control 2 Register) : FBH

7	6	5	4	3	2	1	0
UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

UDRIE	Interrupt enable bit for USART Data Register Empty.	
	0	Interrupt from UDRE is inhibited (use polling)
	1	When UDRE is set, request an interrupt
TXCIE	Interrupt enable bit for Transmit Complete.	
	0	Interrupt from TXC is inhibited (use polling)
	1	When TXC is set, request an interrupt
RXCIE	Interrupt enable bit for Receive Complete	
	0	Interrupt from RXC is inhibited (use polling)
	1	When RXC is set, request an interrupt
WAKEIE	Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RXD goes to LOW level an interrupt can be requested to wake-up system.	
	0	Interrupt from Wake is inhibited
	1	When WAKE is set, request an interrupt
TXE	Enables the transmitter unit.	
	0	Transmitter is disabled
	1	Transmitter is enabled
RXE	Enables the receiver unit.	
	0	Receiver is disabled
	1	Receiver is enabled
USARTEN	Activate USART module by supplying clock.	
	0	USART is disabled (clock is halted)
	1	USART is enabled
U2X	This bit only has effect for the asynchronous operation and selects receiver sampling rate.	
	0	Normal asynchronous operation
	1	Double Speed asynchronous operation

UCTRL3 (USART Control 3 Register) : FCH

7	6	5	4	3	2	1	0
MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00_H

MASTER Selects master or slave in SPI or Synchronous mode operation and controls the direction of XCK pin.

0 Slave mode operation and XCK is input pin.

1 Master mode operation and XCK is output pin

LOOPS Controls the Loop Back mode of USART, for test mode

0 Normal operation

1 Loop Back mode

DISXCK In Synchronous mode of operation, selects the waveform of XCK output.

0 XCK is free-running while USART is enabled in synchronous master mode.

1 XCK is active while any frame is on transferring.

SPISS Controls the functionality of SS pin in master SPI mode.

0 SS pin is normal GPIO or other primary function

1 SS output to other slave device

USBS Selects the length of stop bit in Asynchronous or Synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

TX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the UDATA register.

0 MSB (9^{th} bit) to be transmitted is ‘0’

1 MSB (9^{th} bit) to be transmitted is ‘1’

RX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer.

0 MSB (9^{th} bit) received is ‘0’

1 MSB (9^{th} bit) received is ‘1’

USTAT (USART Status Register) : FDH

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE
R/W	R/W	R/W	R/W	R/W	R	R	R

Initial value : 80_H

UDRE — The UDRE flag indicates if the transmit buffer (UDATA) is ready to be loaded with new data. If UDRE is '1', it means the transmit buffer is empty and can hold one or two new data. This flag can generate an UDRE interrupt. Writing '0' to this bit position will clear UDRE flag.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC — This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXC interrupt is executed. It is also cleared by writing '0' to this bit position. This flag can generate a TXC interrupt.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

RXC — This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXC flag can be used to generate a RXC interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE — This flag is set when the RX pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKE interrupt. This bit is set only when in asynchronous mode of operation. NOTE

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated.

SOFTRST — This is an internal reset and only has effect on USART. Writing '1' to this bit initializes the internal logic of USART and is auto cleared.

0 No operation

1 Reset USART

DOR — This bit is set if a Data Overrun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data Overrun

1 Data Overrun detected

FE — This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read.

0 No Frame Error

1 Frame Error detected

PE — This bit is set if the next character in the receive buffer has a Parity Error when received while Parity Checking is enabled. This bit is valid until the receive buffer is read.

0 No Parity Error

1 Parity Error detected

Note) When the WAKE function of USART is used as a release source from STOP mode, it is required to clear this bit in the RX interrupt service routine. Else the device will not wake-up from STOP mode again by the change of RX pin.

UBAUD (USART Baud-Rate Generation Register) : FEH

7	6	5	4	3	2	1	0
UBAUD7	UBAUD6	UBAUD5	UBAUD4	UBAUD3	UBAUD2	UBAUD1	UBAUD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

UBAUD [7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate XCK clock in synchronous or spi mode. To prevent malfunction, do not write '0' in asynchronous mode, and do not write '0' or '1' in synchronous or spi mode.

UDATA (USART Data Register) : FFH

7	6	5	4	3	2	1	0
UDATA7	UDATA6	UDATA5	UDATA4	UDATA3	UDATA2	UDATA1	UDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

UDATA [7:0] The USART Transmit Buffer and Receive Buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the UDATA register. Reading the UDATA register returns the contents of the Receive Buffer.

Write this register only when the UDRE flag is set. In spi or synchronous master mode, write this register even if TX is not enabled to generate clock, XCK.

11.6.13 Baud Rate setting (example)

	U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
Baud Rate	**fOSC=2.00MHz**				**fOSC=4.00MHz**			
2400	51	0.2%	103	0.2%	103	0.2%	207	0.2%
4800	25	0.2%	51	0.2%	51	0.2%	103	0.2%
9600	12	0.2%	25	0.2%	25	0.2%	51	0.2%
14.4K	8	-3.5%	16	2.1%	16	2.1%	34	-0.8%
19.2K	6	-7.0%	12	0.2%	12	0.2%	25	0.2%
28.8K	3	8.5%	8	-3.5%	8	-3.5%	16	2.1%
38.4K	2	8.5%	6	-7.0%	6	-7.0%	12	0.2%
57.6K	1	8.5%	3	8.5%	3	8.5%	8	-3.5%
76.8K	1	-18.6%	2	8.5%	2	8.5%	6	-7.0%
115.2K	-	-	1	8.5%	1	8.5%	3	8.5%
230.4K					-	-	1	8.5%
250K					-	-	1	0.0%
Baud Rate	**fOSC=8.00MHz**				**fOSC=16.00MHz**			
2400	207	0.2%	-	-				
4800	103	0.2%	207	0.2%	207	0.2%	-	-
9600	51	0.2%	103	0.2%	103	0.2%	207	0.2%
14.4K	34	-0.8%	68	0.6%	68	0.7%	138	-0.1%
19.2K	25	0.2%	51	0.2%	51	0.2%	103	0.2%
28.8K	16	2.1%	34	-0.8%	34	-0.8%	68	0.7%
38.4K	12	0.2%	25	0.2%	25	0.2%	51	0.2%
57.6K	8	-3.5%	16	2.1%	16	2.2%	34	-0.8%
76.8K	6	-7.0%	12	0.2%	12	0.2%	25	0.2%
115.2K	3	8.5%	8	-3.5%	8	-3.6%	16	2.2%
230.4K	1	8.5%	3	8.5%	3	8.6%	8	-3.6%
250K	1	0.0%	3	0.0%	2	33.4%	7	0.0%
0.5M	-	-	1	0.0%	1	0.0%	3	0.0%

Table 11-14. Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

11.7 I2C

11.7.1 Overview

The I2C is one of industrial standard serial communication protocols, and which uses 2 bus lines Serial Data Line (SDA) and Serial Clock Line (SCL) to exchange data. Because both SDA and SCL lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

- Compatible with I2C bus standard
- Multi-master operation
- Up to 400kHz data transfer speed
- 7 bit address
- Support 2 slave addresses
- Both master and slave operation
- Bus busy detection

11.7.2 Block Diagram

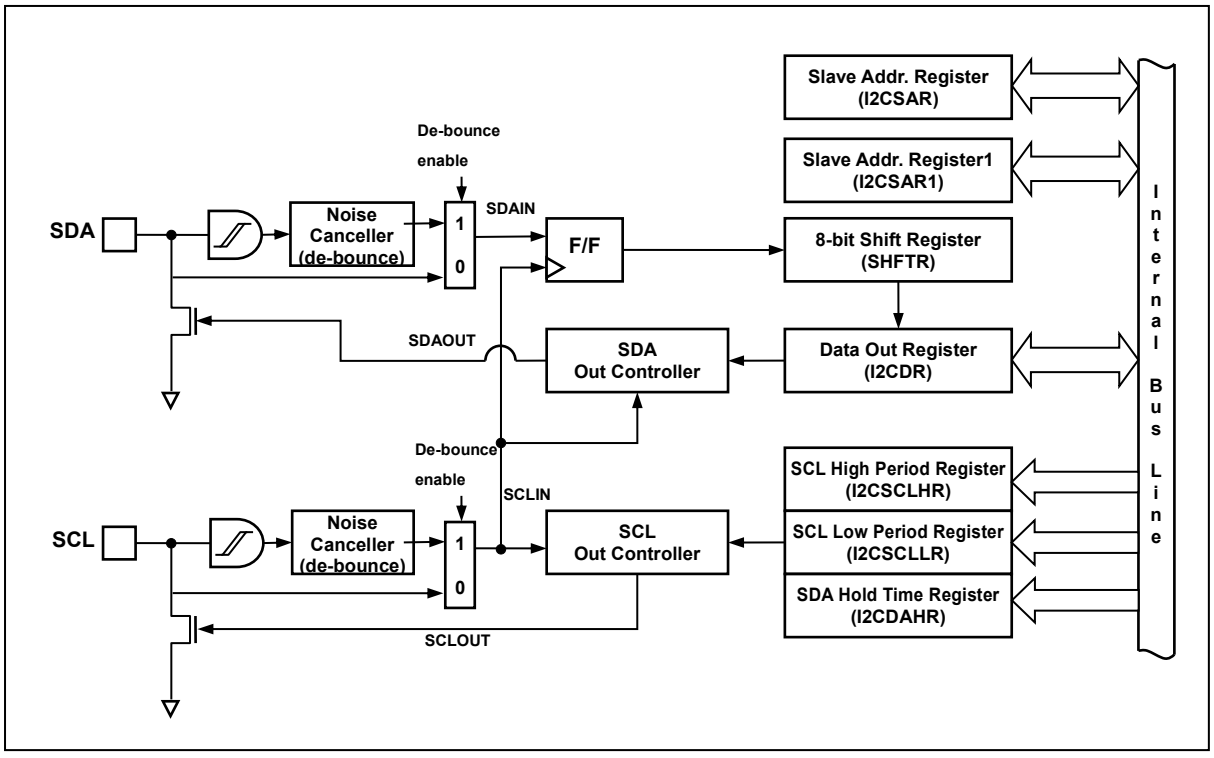


Figure 4.20 I2C Block Diagram

11.7.3 I2C Bit Transfer

The data on the SDA line must be stable during HIGH period of the clock, SCL. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

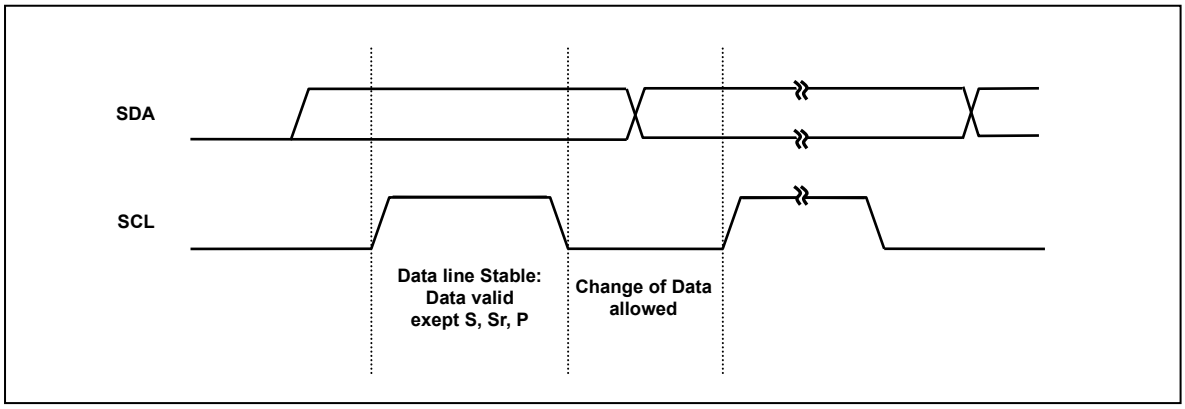


Figure 4.21 Bit Transfer on the I2C-Bus

11.7.4 Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCL, SDA lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

- A high to low transition on the SDA line while SCL is high defines a START (S) condition.
- A low to high transition on the SDA line while SCL is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, ie, the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

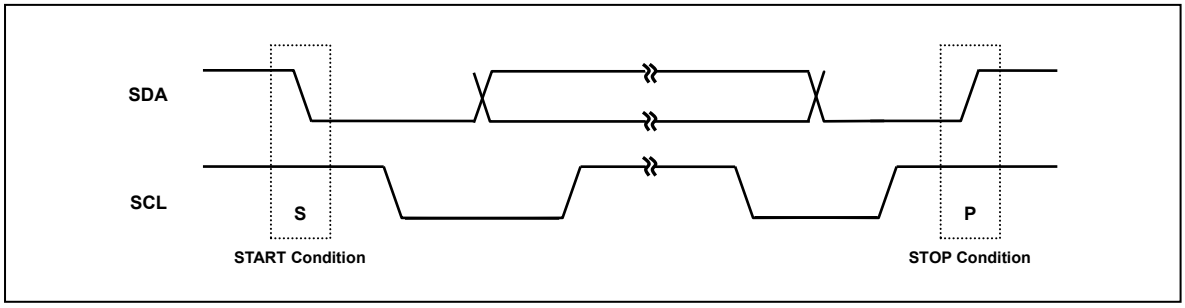


Figure 4.22 START and STOP Condition

11.7.5 Data Transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

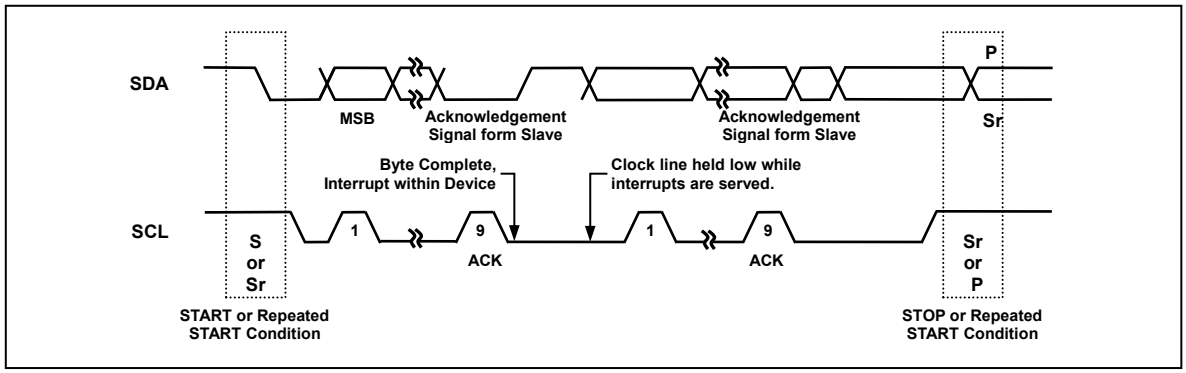


Figure 4.23 Data Transfer on the I2C-Bus

11.7.6 Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it's performing some real time function, the data line must be left HIGH by the slave. And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDA line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

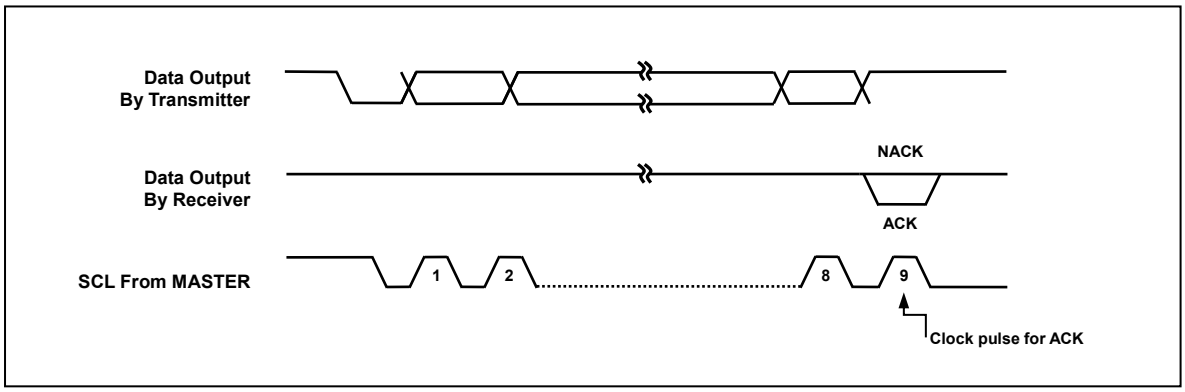


Figure 4.24 Acknowledge on the I2C-Bus

11.7.7 Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and it will hold the SCL line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

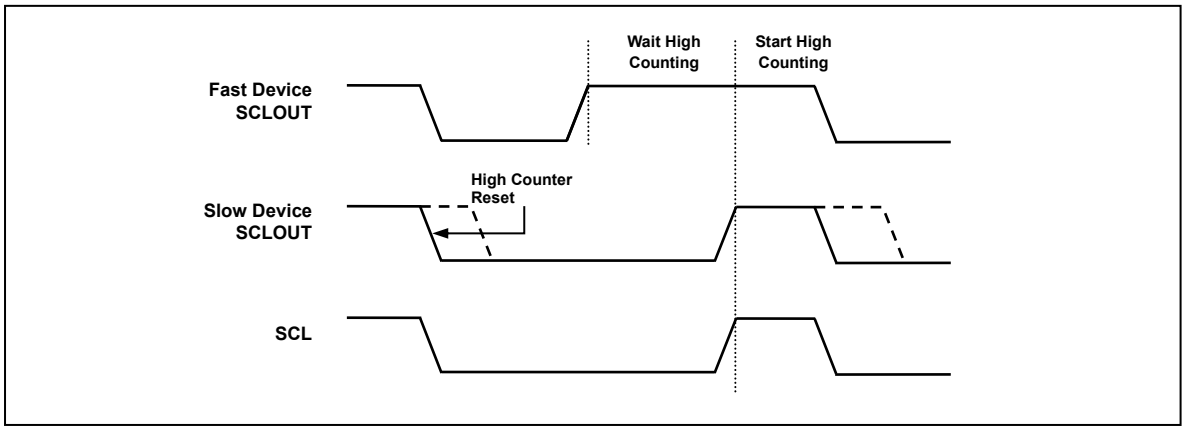


Figure 4.25 Clock Synchronization during Arbitration Procedure

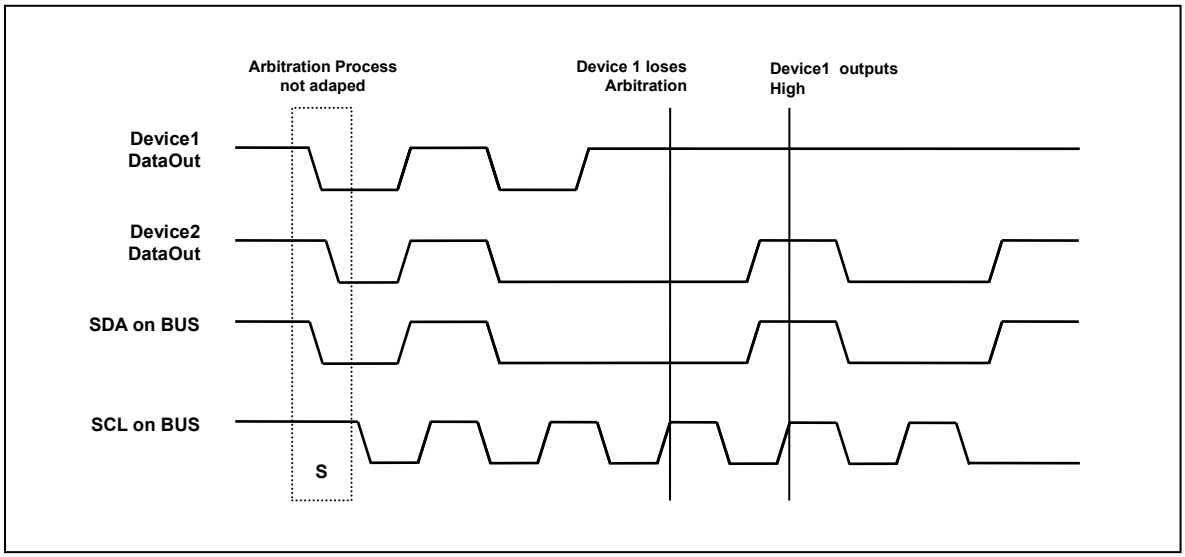


Figure 4.26 Arbitration Procedure of Two Masters

11.7.8 I2C Block operation

The I2C block as peripheral design is independently operating with main CPU operation. The operation of I2C block does a byte unit of I2C frame. After finishing a byte operation (transmit/receive data and clock) on I2C bus system, I2C block generate I2C interrupt for next byte operation. The I2C Interrupt service manage I2C block with the SFR registers, data load/read register (I2CDR) from/to I2C bus system, block control register (I2CMR), the state register (I2CSR) contained operation result. An operation unit of I2C H/W block generates/ receive 9 SCL clock that are for 8 bits data and an ACK. I2C block send / receive ACK signal at 9^{th} clock of SCL according to I2C specification.

The I2C application software initialize I2C block condition depended on clock system, I2C devices condition after system power on.

An application S/W prepares I2C bus communication resource on RAM buffers. If it is to set the start flag in I2CMR register. I2C block start to generate start signal and send a Slave address to slave device. All steps of I2C communication service except start signal and slave address is done by H/W block and I2C Interrupt service. Therefore main application software can reduce time resource while I2C Data write/read operation.

I2C block design supports both functions of master/ Slave on the same block. In case of Masker device it generate SCL clock to slave device and the case of slave mode receive SCL clock from master device.

I2C block decide SDA data direction with the data direction bit (R/$\overline{W}$) of device address in both cases of master and slave mode(TMODE bit 0-> Receive, 1-> Transmit)

Note) When an I2C interrupt is generated by I2C block, IIF flag in I2CMR register is set and it is cleared by writing any value to I2CSR. When I2C interrupt occurs, the SCL line is hold LOW for reading/writing I2CDR register and control I2CMR until writing any value to I2CSR. When the IIF flag is set, the I2CSR contains a value for the state of the I2C bus. According to the value in I2CSR, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver.

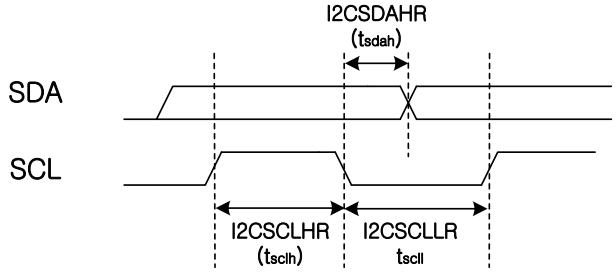
I2C block initialization process

After power ON, it is necessary to have to initialize I2C block for that I2C Block provide I2C Slave device service

① I2C block will start operation (operation clock active) by setting IICEN bit on I2CMR register.
I2CMR = IICEN; // I2C block enable

② Reset I2C block by setting RESET bit on I2CMR register.
I2CMR = RESET; // Reset I2C block by S/W

③ Depended on I2C devices, it shall define I2C SCL max clock and write the value of SCL Low /high time and SDA hold time on I2CSCLLR, I2CSCLHR, I2CSDAHR as following diagram



The timing values are calculated as the follow formula

t_{scll} = t_{sysclk} (4xI2CSCLLR +1) → SCL clock low time

t_{sclh} = t_{sysclk} (4xI2CSCLHR +3) → SCL clock High time

t_{sdah} = t_{sysclk} (I2CSDAHR +1) → SDA data hold time after falling edge of SCL

t_{sysclk} = system clock timing

Ex))

In case of I2C clock (100kHz) and system clock(4MHz), each of t_{scll}, t_{sclh} times is 5us and tsdah is 2.5us.

I2CSCLLR = 5; I2CSCLHR = 4; I2CSDAHR = 9;

④ It is to decide I2C Slave device address and write the address to I2CSAR
I2CSAR = SELF_ADDRESS;

⑤ Finally be ready to get I2C data from I2C bus system as slave device by setting I2C interrupt enable, I2C block enable, ACK enable bits on I2CMR register
I2CMR = IICEN+INTEN+ACKEN; // I2C interrupt enable

I2C interrupt Service

I2C Interrupt service will use for next management action and data load/read from I2C block after I2C H/W block operation (as I2C Master/ Slave device). Because I2C block acts I2C data receiving/writing as a byte unit, I2C block make I2C interrupt for next action of I2C block. While the interrupt happen, I2C block serve the state of I2C bus condition and operation result to I2CSR register. Interrupt service look both registers of I2CMR and I2CSR and do next steps (Save a data from I2CDR, load a data to I2CDR, make STOP condition or Re-start so on).

I2C Interrupt occur at after the following cases

1) As I2C Master Device
- sending a byte on I2CDR register after setting Start bit. (GCALL interrupt)
- sending a byte on I2CDR register after write to I2CSR.(TEND interrupt)
- receiving a byte on I2CDR after write to I2CSR (TEND interrupt)
- Occurring an arbitration loss (MLOST interrupt)
- detecting Stop condition (STOP interrupt)

2) As I2C Slave device
- getting start condition and same device address from a Master (SSEL interrupt)
- sending a byte on I2CDR register after write to I2CSR.(TEND interrupt)
- receiving a byte on I2CDR after write to I2CSR (TEND interrupt)
- detecting Stop condition (STOP interrupt)

Depended on above results I2C service provide services to read/write data from/to I2CDR, generate STOP condition, make next I2C Block action by writing a data to I2CSR register.
Bus arbitration of I2C block processes from I2C bus start condition to last data of I2C data frame. If getting an arbitration loss (MLOST interrupt), I2C interrupt service make I2C block Reset for bus free.

Master transmitter

Main software is to have write/read data to/from slave I2C device. The software has to be ready to get number of data with internal RAM or sending data on internal RAM according to I2C bus protocol type of Slave device. It writes Salve Address to I2CDR register in I2C Block and then if it set START bit on I2CMR register I2C block send slave address with SCL clock to slave device. I2C Block takes master mode (MASTER bit -> 1) and take the read/write state (TMODE bit , read(0), write(1)) according to the data direction bit (R/$\overline{W}$) of device address.
The following is examples software for the case of master mode

Master write

```
I2CMR   = IICEN+INTEN;                    // set I2C block( enable IIC block, I2C interrupt)
I2CDR   =   Slave Address + Write mode;   // load target Salve Address
I2CMR   |= SRT;                           // generate start condition and send slave address
```

I2C Interrupt Service

```
If(Master Mode) and (TMODE)
    If( ACK and GCALL or ACK and TEND )
        If ( Not End of Data )
            I2CDR   =   NEXT DATA;        // load target Salve Address
            I2CSR = 0xFF;                 // Byte transmit start
    ELSE
        I2CMR = IICEN+INTEN+STP;          // STOP generation
    ELSE
        Initialize I2C block              // if have ACK error, any error
End of I2C interrupt service
```

Master Read (without sub address of Slave device)

```
I2CMR   =   IICEN+INTEN;              // start generate
I2CDR   =   Slave Address + Read mode;          // load target Salve Address
I2CMR   |= SRT;                       // generate start condition
```

I2C Interrupt Service

```
If (Master mode) and ( /TMODE)
    If( ACK and GCALL )
        I2CMR |= ACKEN                // After receive data, generate ACK
        I2CSR = 0xFF;                 // Byte transmit start
    ELSE
 if ACK and TEND )
  If ( Not End of Data )
     If(LAST Data)
        I2CMR &= ~ACKEN               // After receive data, generate ACK
      I2C_buffer = I2CDR              // read
      I2CSR = 0xFF;                   // Byte transmit start
 ELSE
  If( ~ACK and TEND)
      I2CMR = IICEN+INTEN+STP;        // STOP generation
   I2CSR = 0xFF;                      // Byte transmit start
  ELSE
    Initialize I2C block              // if have ACK error, any error
End of I2C interrupt service
```

Slave Receiver

I2C Block that is under IIC enable and INTEN enable on I2CMR is monitoring I2C bus lines for being a start condition and self-address with I2CSAD. To have both signals of start signal and getting self-address, I2C block generate I2C interrupt with the status bits (SSEL, BUSY RXACK, SLAVE mode ..) after sending ACK signal. At the time I2C block control SCL line to low state for ready to get/handle next i2c data. If I2C block by I2C interrupt service is ready for next step, it is to release the SCL line to high state for getting next SCL clock from the master. I2C Block decide bus direction (data receive/transmission) by data direction (R/$\overline{W}$) bit in Slave address from master. The state of bus direction is on TMOD bit on I2CSR register. If the master generate Stop condition I2C block receive STOP condition and generate I2C interrupt. I2C interrupt service write any data to I2CSR and finish Slave operation. I2C interrupt service and state register condition is diagrammed in Figure xxxx.

I2C Interrupt service

```
I2C Slave service
  if((Getting SSEL and send ACK)      // received Self-address form master
    if(TMODE)                         // data direction (R/W)
        I2CDR=I2C_TXData              // Transmission mode, Load data
    else
     I2C_RXData =I2CDR
  else
     if (Get STOP condition)
     else
       if (TMODE)                     // data direction (R/W)
           I2CDR= I2C_TXData          // Transmission mode, Load data
         else
           I2C_RXData =I2CDR          // Save received Data
     I2CSR=0xff;
```

11.7.9 Register Map

Name	Address	Dir	Default	Description
I2CMR	DAH	R/W	00H	I2C Mode Control Register
I2CSR	DBH	R	00H	I2C Status Register
I2CSCLLR	DCH	R/W	3FH	SCL Low Period Register
I2CSCLHR	DDH	R/W	3FH	SCL High Period Register
I2CSDAHR	DEH	R/W	01H	SDA Hold Time Register
I2CDR	DFH	R/W	FFH	I2C Data Register
I2CSAR	D7H	R/W	00H	I2C Slave Address Register
I2CSAR1	D6H	R/W	00H	I2C Slave Address Register 1

Table 11-15. Register Map

11.7.10 I2C Register description

I2C Registers are composed of I2C Mode Control Register (I2CMR), I2C Status Register (I2CSR), SCL Low Period Register (I2CSCLLR), SCL High Period Register (I2CSCLHR), SDA Hold Time Register (I2CSDAHR), I2C Data Register (I2CDR), and I2C Slave Address Register (I2CSAR).

11.7.11 Register description for I2C

I2CMR (I2C Mode Control Register) : DAH

7	6	5	4	3	2	1	0
IIF	IICEN	RESET	INTEN	ACKEN	MASTER	STOP	START
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

IIF This is interrupt flag bit.
- 0 No interrupt is generated or interrupt is cleared
- 1 An interrupt is generated

IICEN Enable I2C Function Block (by providing clock)
- 0 I2C is inactive
- 1 I2C is active

RESET Initialize internal registers of I2C.
- 0 No operation
- 1 Initialize I2C, auto cleared

INTEN Enable interrupt generation of I2C.
- 0 Disable interrupt, operates in polling mode
- 1 Enable interrupt

ACKEN Controls ACK signal generation at ninth SCL period.
Note) ACK signal is output (SDA=0) for the following 3 cases.
When received address packet equals to SLA bits in I2CSAR
When received address packet equals to value 0x00 with GCALL enabled
When I2C operates as a receiver (master or slave)
- 0 No ACK signal is generated (SDA=1)
- 1 ACK signal is generated (SDA=0)

MASTER Represent operating mode of I2C
- 0 I2C is in slave mode
- 1 I2C is in master mode

STOP When I2C is master, generates STOP condition.
- 0 No operation
- 1 STOP condition is to be generated

START When I2C is master, generates START condition.
- 0 No operation
- 1 START or repeated START condition is to be generated

I2CSR (I2C Status Register) : DBH

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
R	R	R	R	R	R	R	R

Initial value : 00H

GCALL This bit has different meaning depending on whether I2C is master or slave. Note 1)
When I2C is a master, this bit represents whether it received AACK (Address ACK) from slave.
When I2C is a slave, this bit is used to indicate general call.
0 No AACK is received (Master mode)
1 AACK is received (Master mode)
0 Received address is not general call address (Slave mode)
1 General call address is detected (Slave mode)

TEND This bit is set when 1-Byte of data is transferred completely. Note 1)
0 1 byte of data is not completely transferred
1 1 byte of data is completely transferred

STOP This bit is set when STOP condition is detected. Note 1)
0 No STOP condition is detected
1 STOP condition is detected

SSEL This bit is set when I2C is addressed by other master. Note 1)
0 I2C is not selected as slave
1 I2C is addressed by other master and acts as a slave

MLOST This bit represents the result of bus arbitration in master mode. Note 1)
0 I2C maintains bus mastership
1 I2C has lost bus mastership during arbitration process

BUSY This bit reflects bus status.
0 I2C bus is idle, so any master can issue a START condition
1 I2C bus is busy

TMODE This bit is used to indicate whether I2C is transmitter or receiver.
0 I2C is a receiver
1 I2C is a transmitter

RXACK This bit shows the state of ACK signal.
0 No ACK is received
1 ACK is generated at ninth SCL period

Note) These bits can be source of interrupt.

When an I2C interrupt occurs except for STOP interrupt, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, the TEND, STOP, SSEL, LOST, RXACK bits are cleared.

I2CSCLLR (SCL Low Period Register) : DCH

7	6	5	4	3	2	1	0
SCLL7	SCLL6	SCLL5	SCLL4	SCLL3	SCLL2	SCLL1	SCLL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLL[7:0] This register defines the LOW period of SCL when I2C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :$t_{SCLK}\times$ (4 × SCLL + 1) where t_{SCLK} is the period of SCLK.

I2CSCLHR (SCL High Period Register) : DDH

7	6	5	4	3	2	1	0
SCLH7	SCLH6	SCLH5	SCLH4	SCLH3	SCLH2	SCLH1	SCLH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLH[7:0] This register defines the HIGH period of SCL when I2C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula :$t_{SCLK}\times$ (4 × SCLH + 3) where t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C in master mode (fI2C) is calculated by the following equation.

$$fI2C = \frac{1}{tSCLK \times (4(SCLL + SCLH) + 4)}$$

I2CSDAHR (SDA Hold Time Register) : DEH

7	6	5	4	3	2	1	0
SDAH7	SDAH6	SDAH5	SDAH4	SDAH3	SDAH2	SDAH1	SDAH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

SDAH[7:0] This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after $t_{SCLK}\times$ SDAH. In master mode, load half the value of SCLL to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after $t_{SCLK}\times$ (SDAH + 1). So, to insure normal operation in slave mode, the value $t_{SCLK}\times$ (SDAH + 1) must be smaller than the period of SCL.

I2CDR (I2C Data Register) : DFH

7	6	5	4	3	2	1	0
ICD7	ICD6	ICD5	ICD4	ICD3	ICD2	ICD1	ICD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

ICD[7:0] When I2C is configured as a transmitter, load this register with data to be transmitted. When I2C is a receiver, the received data is stored into this register.

I2CSAR (I2C Slave Address Register) : D7H

7	6	5	4	3	2	1	0
SLA7	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	GCALLEN
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I2C module when I2C operates in slave mode.

GCALLEN This bit decides whether I2C allows general call address or not when I2C operates in slave mode.

0 Ignore general call address

1 Allow general call address

I2CSAR1 (I2C Slave Address Register 1) : D6H

7	6	5	4	3	2	1	0
SLA7	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	GCALLEN
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I2C module when I2C operates in slave mode.

GCALLEN This bit decides whether I2C allows general call address or not when I2C operates in slave mode.

0 Ignore general call address

1 Allow general call address

11.8 12-Bit A/D Converter

11.8.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 12-bit digital value. The A/D module has eight analog inputs. The output of the multiplex is the input into the converter, which generates the result via successive approximation. The A/D module has four registers which are the control register ADCM (A/D Converter Mode Register), ADCM1 (A/D Converter Mode Register 1) and A/D result register ADCHR (A/D Converter Result High Register) and ADCLR (A/D Converter Result Low Register). It is selected for the corresponding channel to be converted by setting ADSEL[3:0]. To executing A/D conversion, ADST bit sets to '1'. The register ADCHR and ADCLR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCHR and ADCLR, the A/D conversion status bit AFLAG is set to '1', and the A/D interrupt is set. While processing A/D conversion, AFLAG bit is read as '0'. If using STBY (power down) bit, the ADC is disabled. Also interrupt of internal timer, external event can start ADC regardless of interrupt occurrence.

ADC Conversion Time = ADCLK * 60cycles

After STBY bit is reset (ADC power enable) and it is restarted, during some cycle, ADC conversion value may have an inaccurate value.

Note) When using the ADC, set to input mode the direction of the port to be used. (PxIO)

11.8.2 Block Diagram

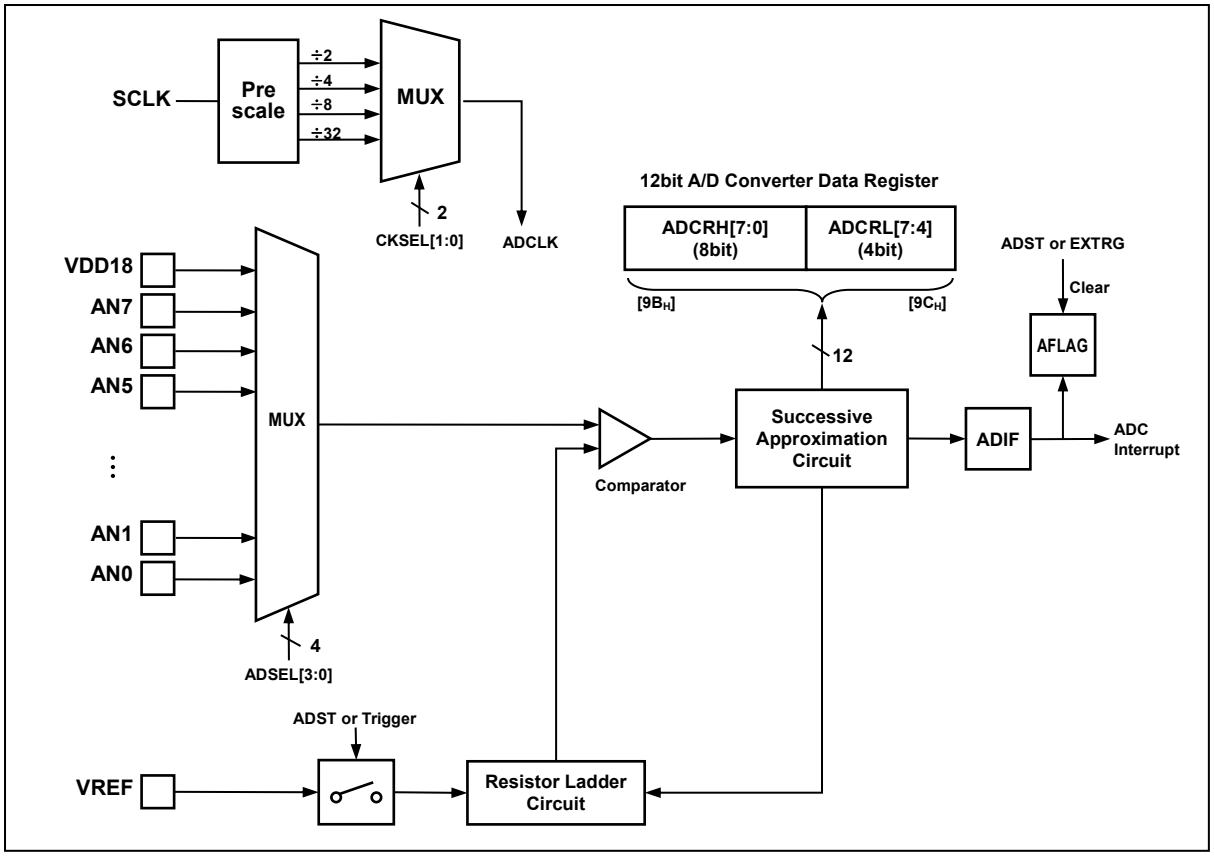


Figure 4.27 ADC Block Diagram

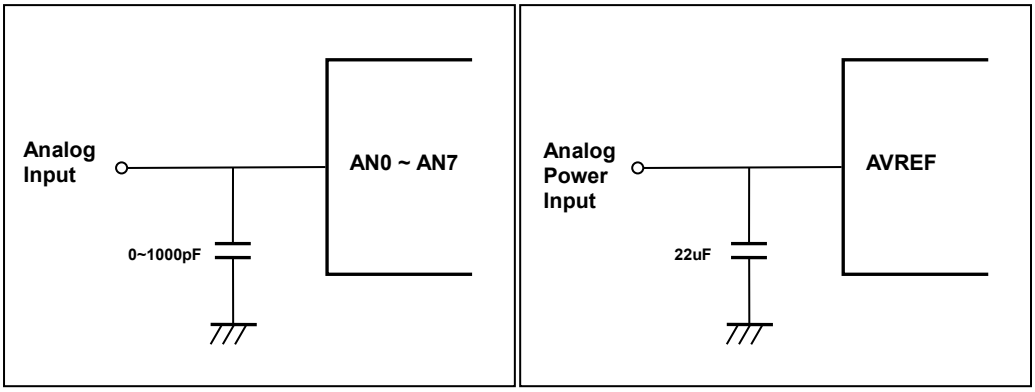


Figure 4.28 A/D Analog Input Pin & A/D Power(AVDD) Pin Connecting Capacitor

11.8.3 ADC Operation

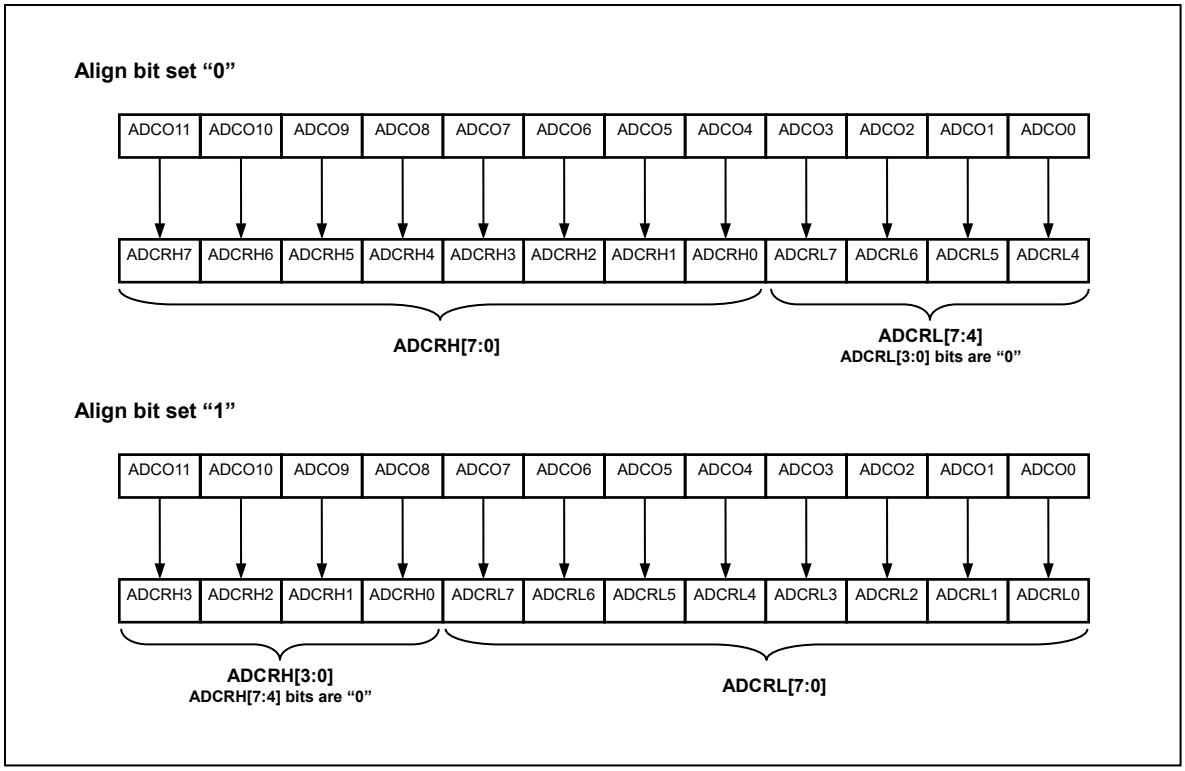


Figure 4.29 ADC Operation for Align bit

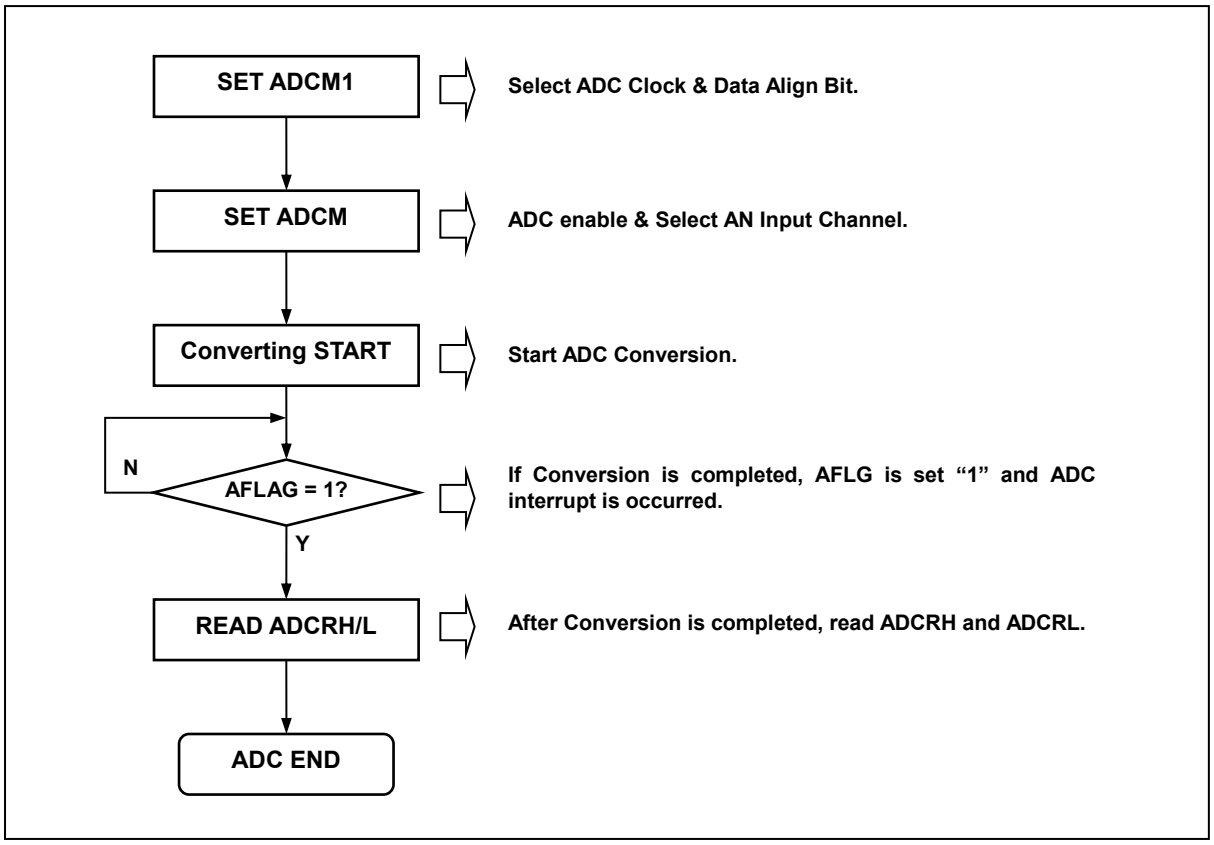


Figure 4.30 Converter Operation Flow

11.8.4 Register Map

Name	Address	Dir	Default	Description
ADCM	95H	R/W	8FH	A/D Converter Mode Register
ADCRL	96H	R	xxH	A/D Converter Result High Register
ADCRH	97H	R	xxH	A/D Converter Result Low Register
ADCM1	96H	R/W (STBY=1)	01H	A/D Converter Mode 1 Register
ADCM1	96H	W (STBY=0)	01H	A/D Converter Mode 1 Register

Table 11-16. Register Map

11.8.5 Register description

The ADC Register consists of A/D Converter Mode Register (ADCM), A/D Converter Result High Register (ADCRH), A/D Converter Result Low Register (ADCRL), A/D Converter Mode 2 Register (ADCM2)..

Note) when STBY bit is set to '1', ADCM1 can be read. If ADC enables, it is possible only to write ADCM1. When reading, ADCRL is read.

11.8.6 Register description for ADC

ADCM (A/D Converter Mode Register) : 95H

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Initial value : 8FH

STBY Control operation of A/D standby (power down)
- 0 ADC module enable
- 1 ADC module disable (power down)

ADST Control A/D Conversion stop/start.
- 0 ADC Conversion Stop
- 1 ADC Conversion Start

REFSEL A/D Converter reference selection
- 0 Internal Reference (VDD)
- 1 External Reference(AVREF)

AFLAG A/D Converter operation state
- 0 During A/D Conversion
- 1 A/D Conversion finished

ADSEL[3:0] A/D Converter input selection

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	Channel0(AN0)
0	0	0	1	Channel1(AN1)
0	0	1	0	Channel2(AN2)
0	0	1	1	Channel3(AN3)
0	1	0	0	Channel4(AN4)
0	1	0	1	Channel5(AN5)
0	1	1	0	Channel6(AN6)
0	1	1	1	Channel7(AN7)
1	1	1	1	Channel15(VDD18), default

Note) 1. When using ports as ADC input port, it is recommended to set corresponding PSR2 register to prevent current leakage or unexpected function, because analog value enters to digital circuit.
2. AMP2O connected to the AN5 input switch.
3. If AMP2O enabled(A12STO_EN=1), ADC converts internal analog AMP output.
4. If AMP2O disabled(A12STO_EN=0), ADC converts external analog input.
5. AN8~AN14 does not receive analog input.

ADCRH (A/D Converter Result High Register) : 97H

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value : xxH

ADDM[11:4] MSB align, A/D Converter High result (8-bit), default

ADDL[11:8] LSB align, A/D Converter High result (4-bit)

ADCRL (A/D Converter Result Low Register) : 96H

7	6	5	4	3	2	1	0
ADDM3 ADDL7	ADDM2 ADDL6	ADDM1 ADDL5	ADDM0 ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R	R	R	R

Initial value : xxH

ADDM[3:0] MSB align, A/D Converter Low result (4-bit), default

ADDL[7:0] LSB align, A/D Converter Low result (8-bit)

ADCM1 (A/D Converter Mode Register) : 96H

	7	6	5	4	3	2	1	0
	EXTRG	TSEL2	TSEL1	TSEL0	-	ALIGN	CKSEL1	CKSEL0
STBY=1	RW(	RW	RW	RW	-	RW	RW	RW
STBY=0	W	W	W	W	-	W	W	W

Initial value : 01H

EXTRG A/D external Trigger

A/D conversion Start by external Trigger, and Stop by clearing this bit

0 A/D conversion Stop and External Trigger disable

1 External Trigger enable

TSEL[2:0] A/D Trigger Source selection

TSEL2	TSEL1	TSEL0	Description
0	0	0	External Interrupt 0
0	0	1	External Interrupt 1
0	1	0	-
0	1	1	-
1	0	0	Timer0 interrupt
1	0	1	Timer1 interrupt
1	1	0	Timer2 interrupt
1	1	1	Timer3 interrupt

ALIGN A/D Converter data align selection.

0 MSB align (ADCRH[7:0], ADCRL[7:4]), default

1 LSB align (ADCRH[3:0], ADCRL[7:0])

CKSEL[1:0] A/D Converter Clock selection

CKSEL1	CKSEL0	ADC Clock	ADC VDD
0	0	fx/2	Test Only
0	1	fx/4, default	3V~5V
1	0	fx/8	2.7V~3V
1	1	fx/32	2.4V~2.7V

Note) 1. fx : system clock
2. ADC clock have to be used 3MHz under

11.9 PPG (Programmable Pulse Generator)

11.9.1 Overview

MC97F6108A has programmable pulse generator (PPG) supporting capture mode operations based on 16-bit timer. PPG basic fuction is similar to 16-bit timer's but PPG operates as PWM mode only. When PPGEN set to '1', PPG output goes to high. And initial value of P05PU(pullup register of PPGO port) is '1'.
MC97F6108A has five analog comparators, and their output controls PPG operation respectively. Output from internal comparator 0 whose input is from an external analog pin can be used to synchronize the pulse output and the pulse controls IGBT

The following are the PPG main features.

- One shot pulse.
- Comparator 0 : When PPGMD=1, Comparator 0 start the PPG.
- Comparator 1 : Disable the PPG output until flag is cleared.
- Comparator 2 : Increase or decrease the PPG period in Auto period mode.
- Comparator 3 : Disable the PPG output (only one period).
- Comparator 4 : Detect zero crossing
- PPGPXH, PPGPXL registers set the maximem period of PPG.
- PPG output port(P05) default pullup (P0PU[5] = 1)
- PPG default output value is 'high'.
  Duty matching reset the PPGO, period matching set the PPGO.
- 3 comparator interrupt flags are used as PPG capture sources. (CMP3IF, CMP1IF and CMP4IF).

- PPG off time can be controled by PPG off time min and max registers.
- PPG period high/low values are written in the period registers at the same time.
- PPG duty high/low values are written in the duty registers at the same time.
- It is possible to read and write to the PPG duty and period registers.

11.9.2 PPG block diagram

PPGCR	PPGEN	-	CAPE	PPGCK2	PPGCK1	PPGCK0	PPGCN	PPGST
PPGCR1	-	-	PPGIN[2]	PPGIN[1]	PPGIN[0]	-	PPG_PE	-
PPGCR2	PPGIF	TPPGOen	PPGMD	DETC3EN	DETC1EN	-	-	-

ADDRESS : $E2_H$
INITIAL VALUE : 0000_0000_b

ADDRESS : $E3_H$
INITIAL VALUE : 0000_0000_b

ADDRESS : $E1_H$
INITIAL VALUE : 0000_0000_b

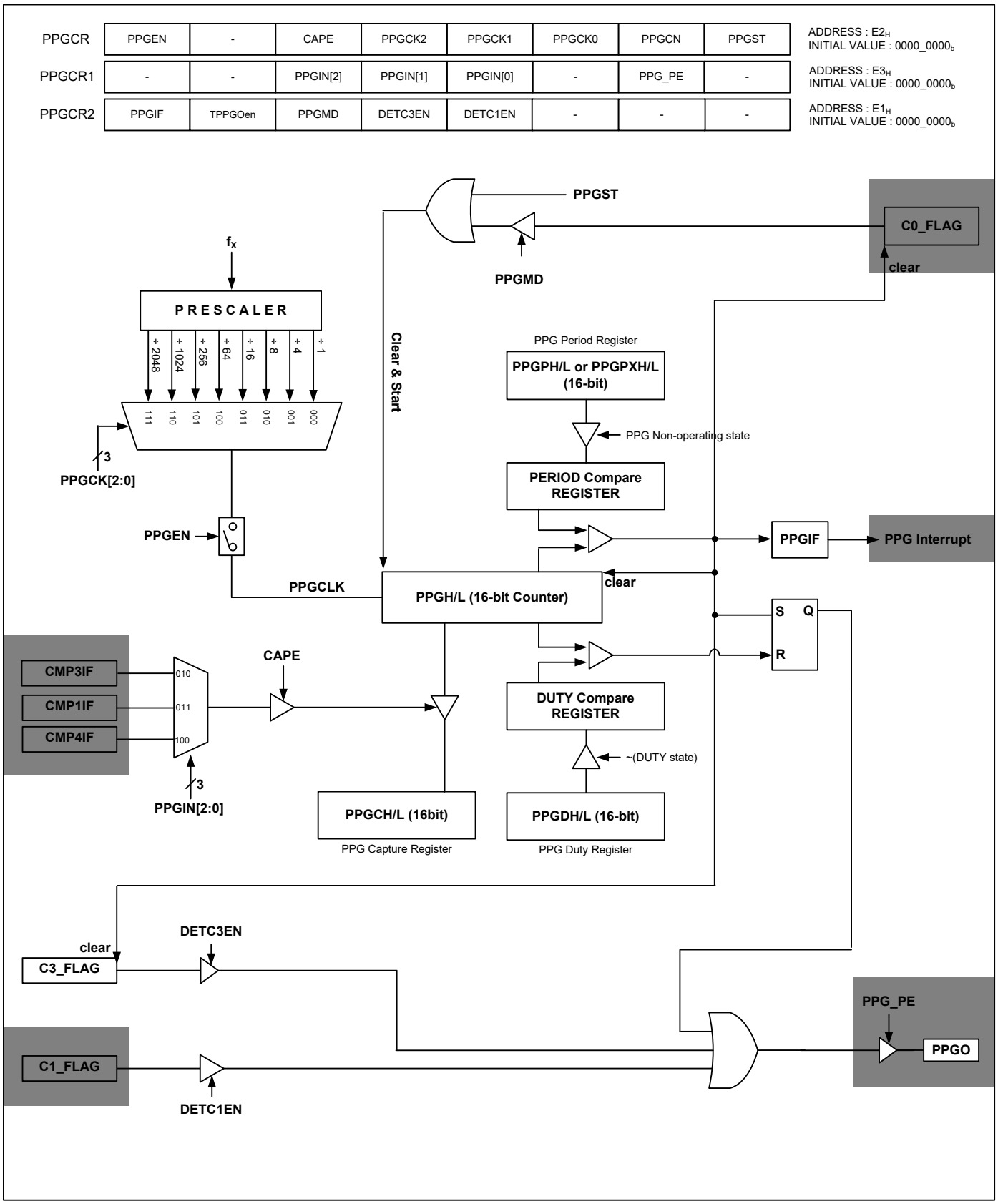


Figure 4.31 PPG block diagram

11.9.3 PPG Start & One Shot Pulse

PPG start sources are PPGST bit of PPGCR register and C0_FLAG (comparator 1 flag). If PPGMD = '0', only PPGST starts the counter, if PPGMD = '1', PPGST or C0_FLAG starts the counter. After one period C0_FLAG is cleared Automatically.

PPG period is determined by the PPGPH, PPGPL and duty is determined by PPGDH, PPGDL..

Period = ({PPGPH, PPGPL} + 1) X PPG Clock Period

Duty = ({PPGDH, PPGDL} + 1) X PPG Clock Period

If PPG starts counting, PPG compares PPGH, PPGL value with PPGDH, PPGDL value. If the PPGH, PPGL value and the PPGDH, PPGDL value matches, PPG compares PPGH, PPGL value with PPGPH, PPGPL value.

If the PPGH, PPGL value and the PPGPH, PPGPL value matches, PPG counter is cleared and does not start again. Only start sources can start counter again. This is one shot pulse operation.

The duty value and counter matching enables the period value and counter comparison. PPGPH/L registers are loaded to period compare register when the PPG is not operate. PPGDH/L registers are loaded to compare registers when the PPG is not in the duty state. It is highly recommended that the duty value is not set same to the period value.

Default value of PPGO is high, it is toggled when duty matching and period matching(duty matching : reset, period matching : set). To export the pulse output as PPG port, set the PPG_PE bit in the PPGCR1 register.

When PPGH, PPGL are read, PPGL should be read first. Because when PPGL is read PPGH is captured to buffer, and when PPGH is read captured value of PPGH is read.

Note)

1. The duty value should not set same to the period value.

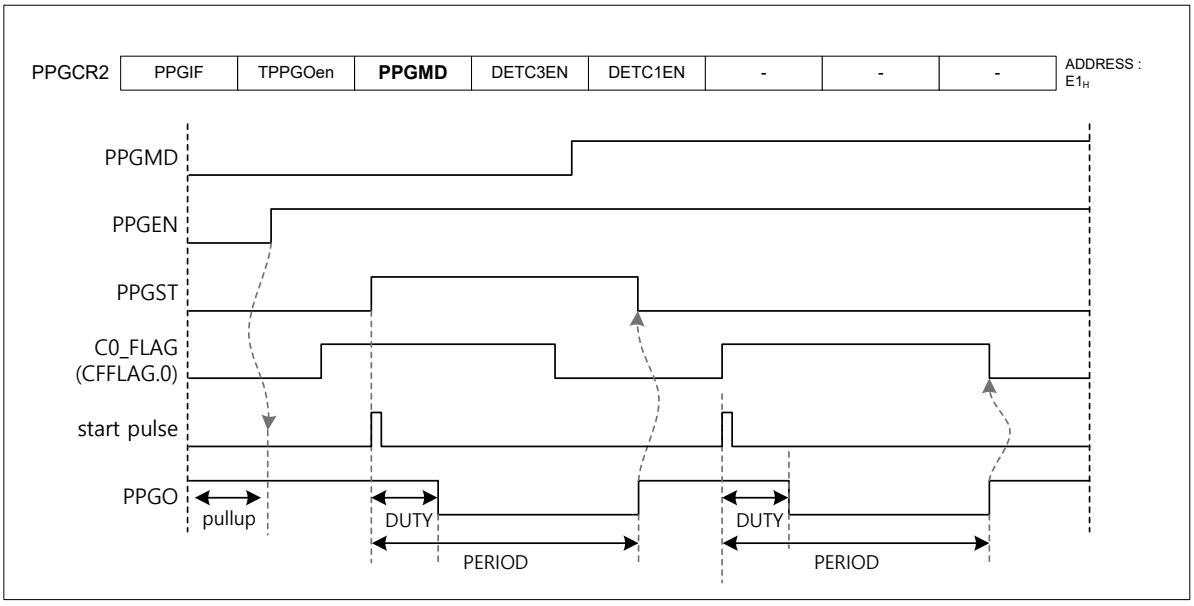


Figure 4.32 PPG start & one shot pulse

11.9.4 PPG Period/Duty Write

When writing a value to the PPG period registers, write to the PPGPH first, then write to the PPGPL. As shown in the figure below, when writing to the PPGPH, the value is saved to a buffer. When writing to the PPGPL, the saved value and PPGPL is loaded to the compare registers. PPG duty registers (PPGDH,PPGDL) have the same write function.

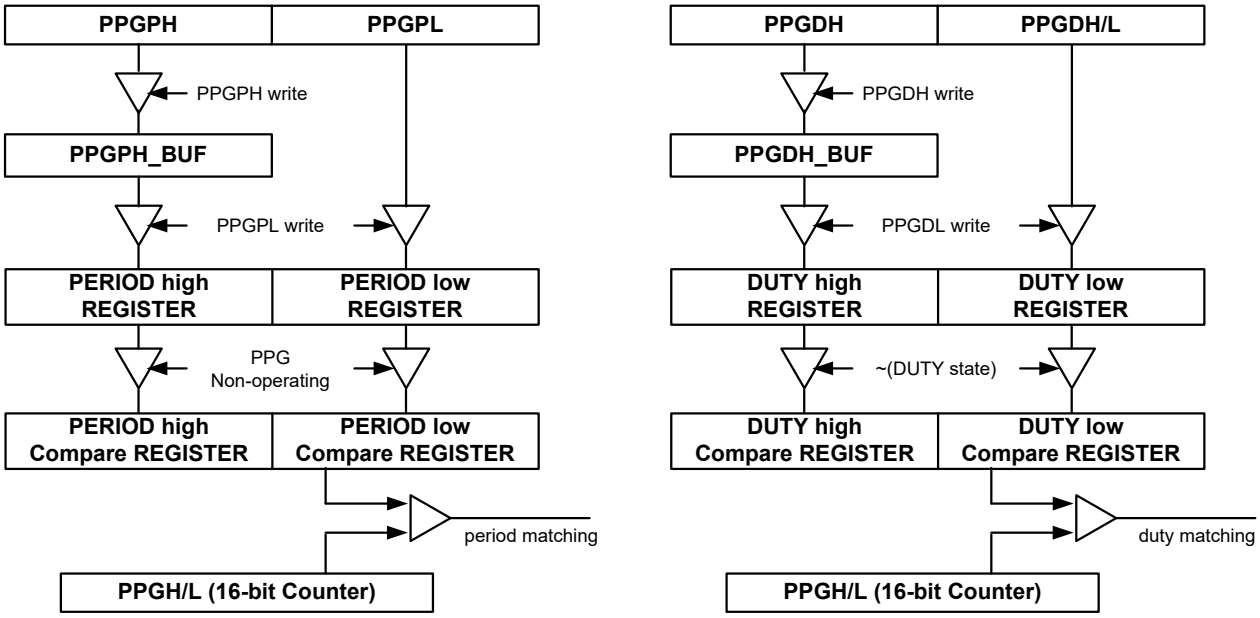


Figure 4.33 PPG Period/Duty Write

PPG period high/low register is entered to the period high/low compare register, when the PPG is not operate.
PPG duty high/low register is entered to the duty high/low compare register, when PPG is not in the duty state.

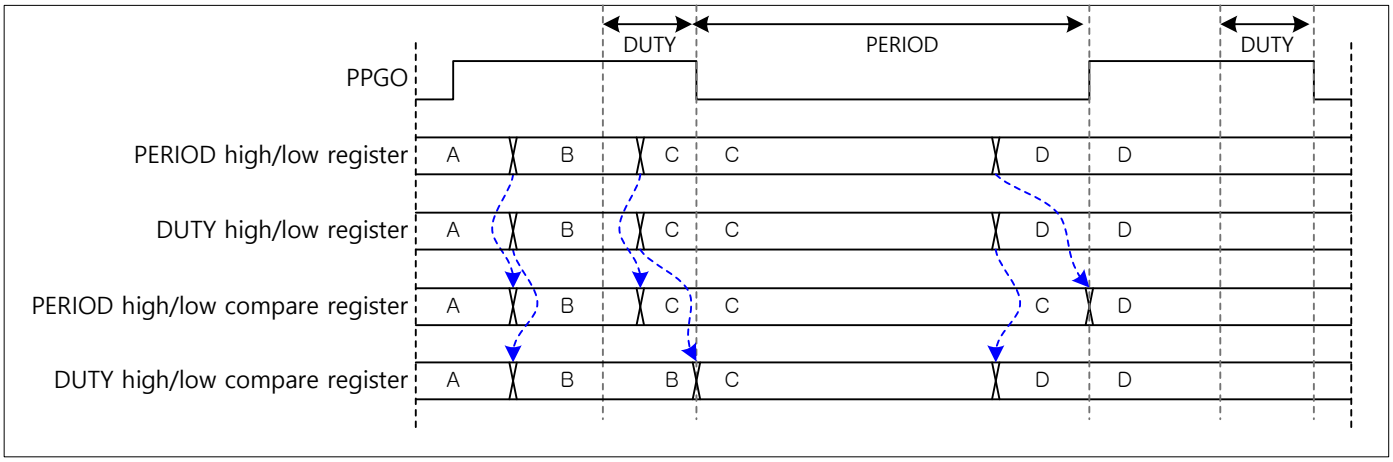


Figure 4.34 PPG Period/Duty Load to compare registers

11.9.5 Capture Mode

The PPG capture mode is set by CAPE as '1' in PPGCR register. The capture result is loaded into PPGCH, PPGCL. The counter(PPGH,PPGL) does not stop when counter is captured. Capture sources are CMP3IF, CMP1IF, CMP4IF. function is chosen. The capture source is selected by setting PPGIN[2:0] in the PPGCR1 register.

The PPGCH and PPGH are in same address. In the capture mode, reading operation is read the PPGCH, not PPGH because path is opened to the PPGCH. The PPGL, PPGCL has the same function.

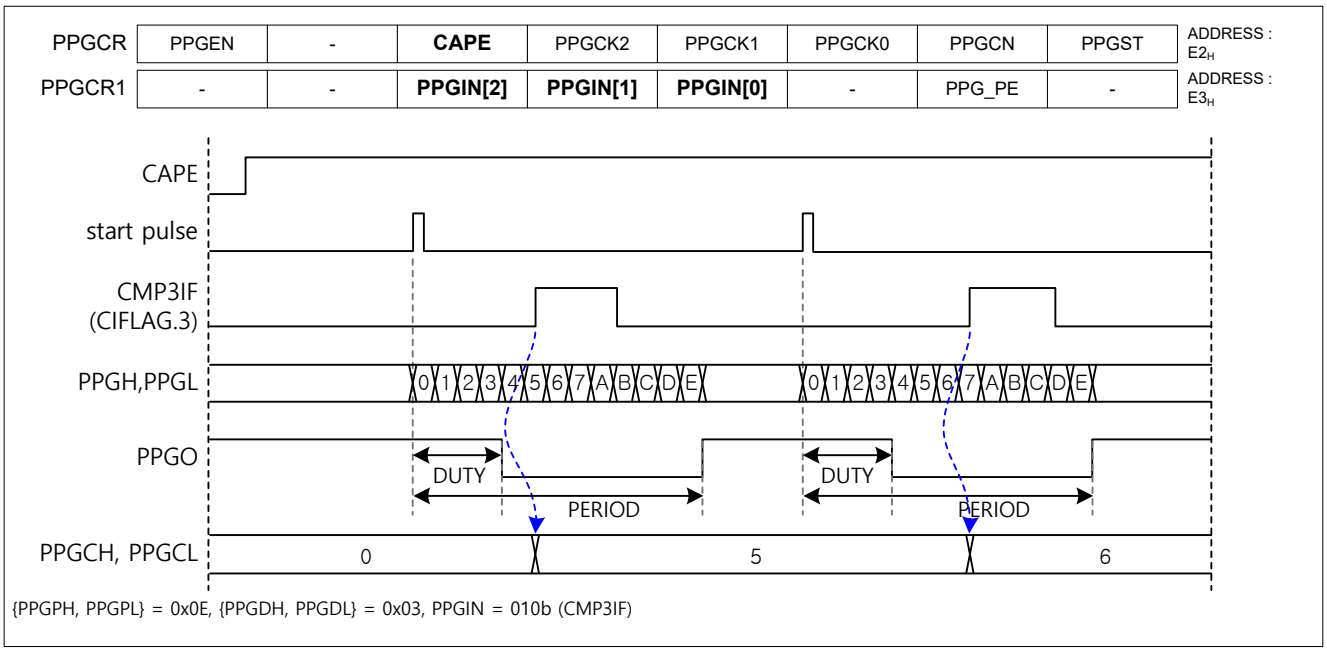


Figure 4.36 Capture mode

11.9.6 Disable PPG output by comparator 1

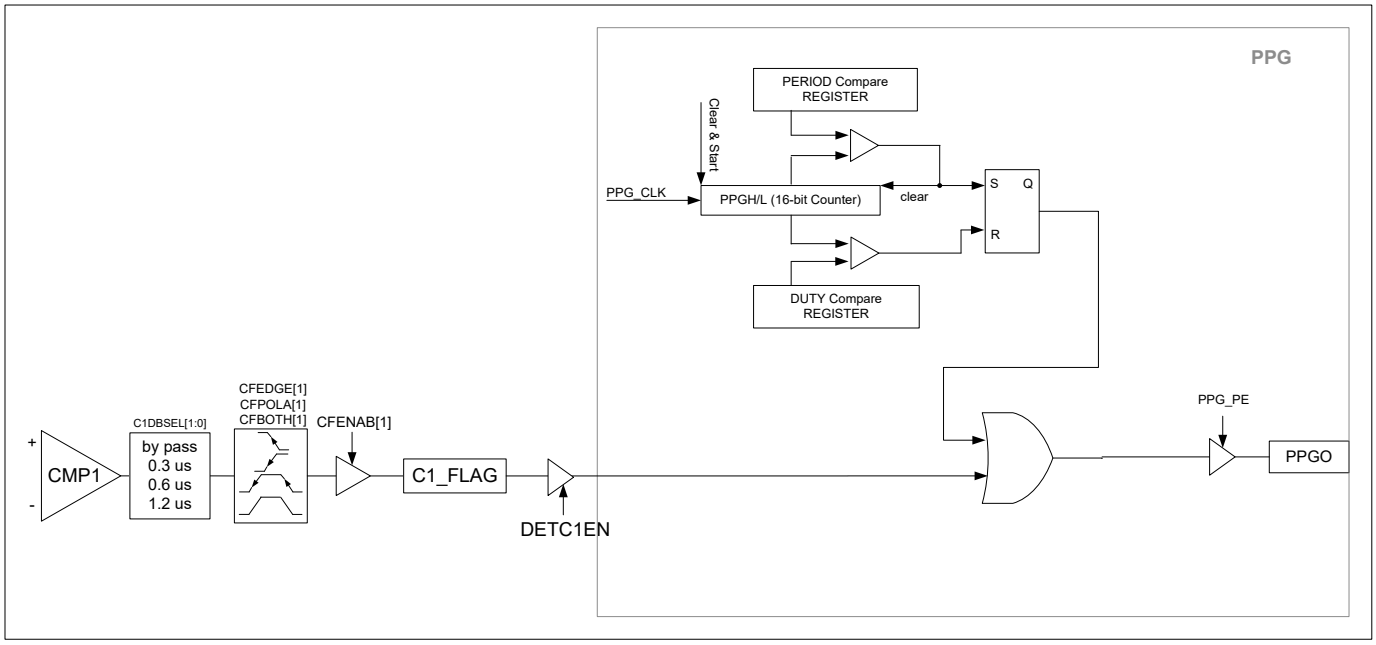


Figure 4.37 Disable PPG output by comparator 1

PPG output is disabled by comparator 1 flag(C1_FLAG) with enable bit DETC1EN in the PPGCR2 register. When DETC1EN is set and comparator 1 output occurs, PPG outputs it's default value. C1_FLAG is cleared by writting '0'. Once PPG output is disabled by C1_FLAG, C0_FLAG is not generated until C1_FLAG is cleared.

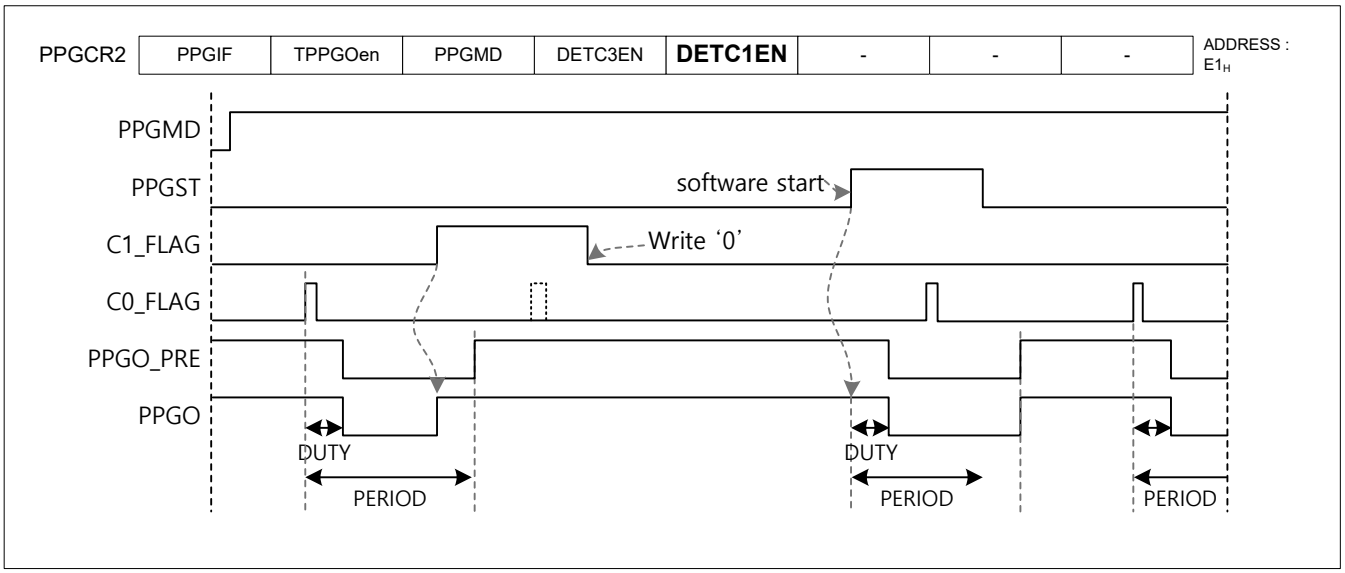


Figure 4.38 Disable PPG output by comparator 1 (C1_FLAG)

11.9.7 Disable PPG output by comparator 3

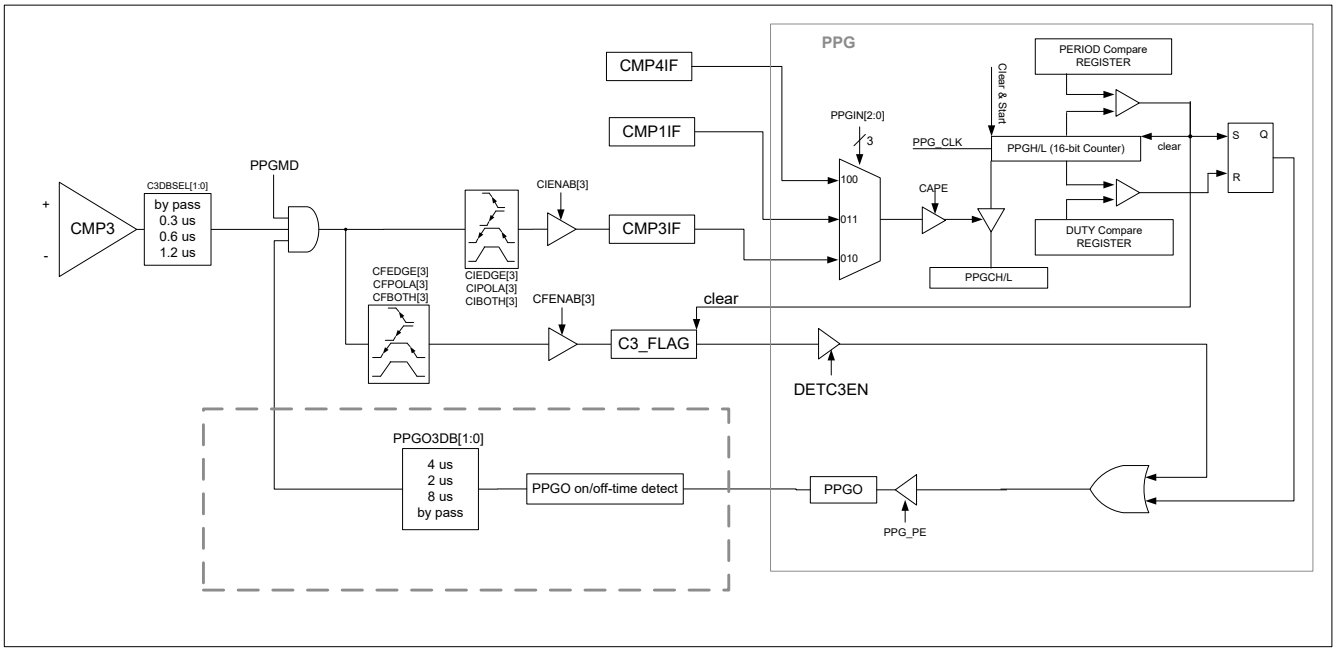


Figure 4.39 Disable PPG output by comparator 3 block diagram

PPG output is disabled by comparator 3 flag(C3_FLAG) with enable bit DETC3EN in the PPGCR2 register. When DETC3EN is set and comparator 3 output occurs, PPG outputs it's default value. Then C3_FLAG is cleared when PPG output is disabled. C3_FLAG can be cleared by writting '0'.

Specified period of time after the PPGO goes to low and PPG is disabled (PPGO = 'high') C3_FLAG is not generated. The time is determined by PPGO3DB register in CA_REGC. To generate C3_FLAG, PPGMD need to be '1'.

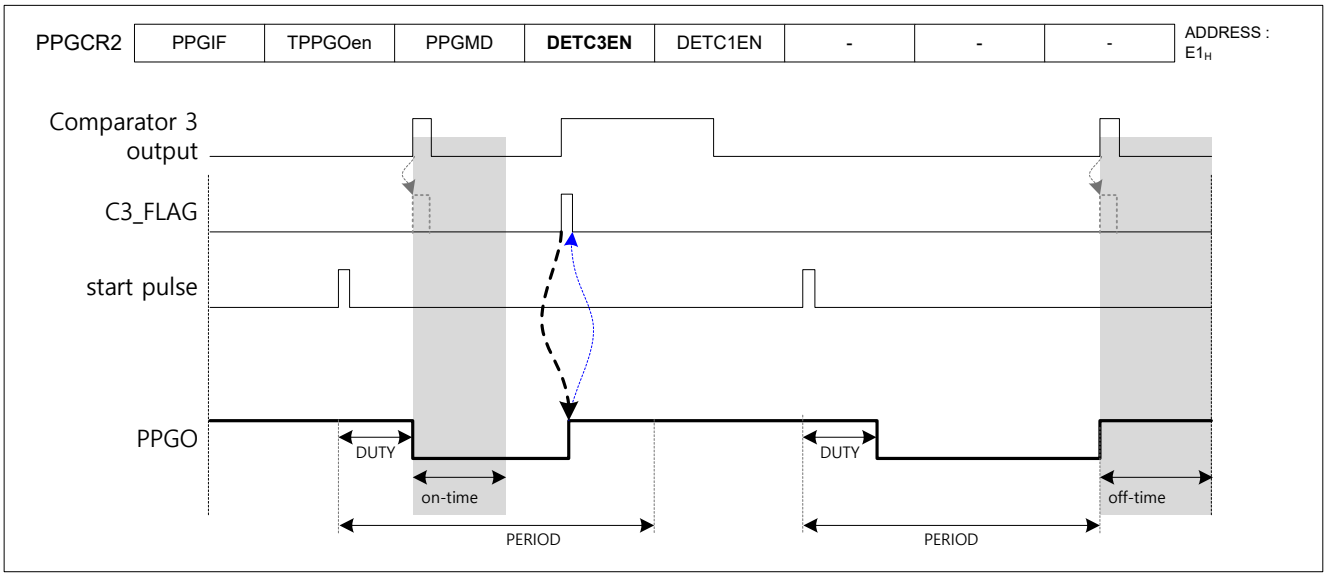


Figure 4.40 Disable PPG output by comparator 3 (C3_FLAG)

11.9.8 PPG period limitation

PPG max period is determined by the PPGPXH, PPGPXL. After duty matching, period comparation is enabled until period matching. If PPG counter value and PPGPXH, PPGPXL matches before PPG counter value and PPGPH, PPGPL matches, the counter is cleared and waits for the start signal.

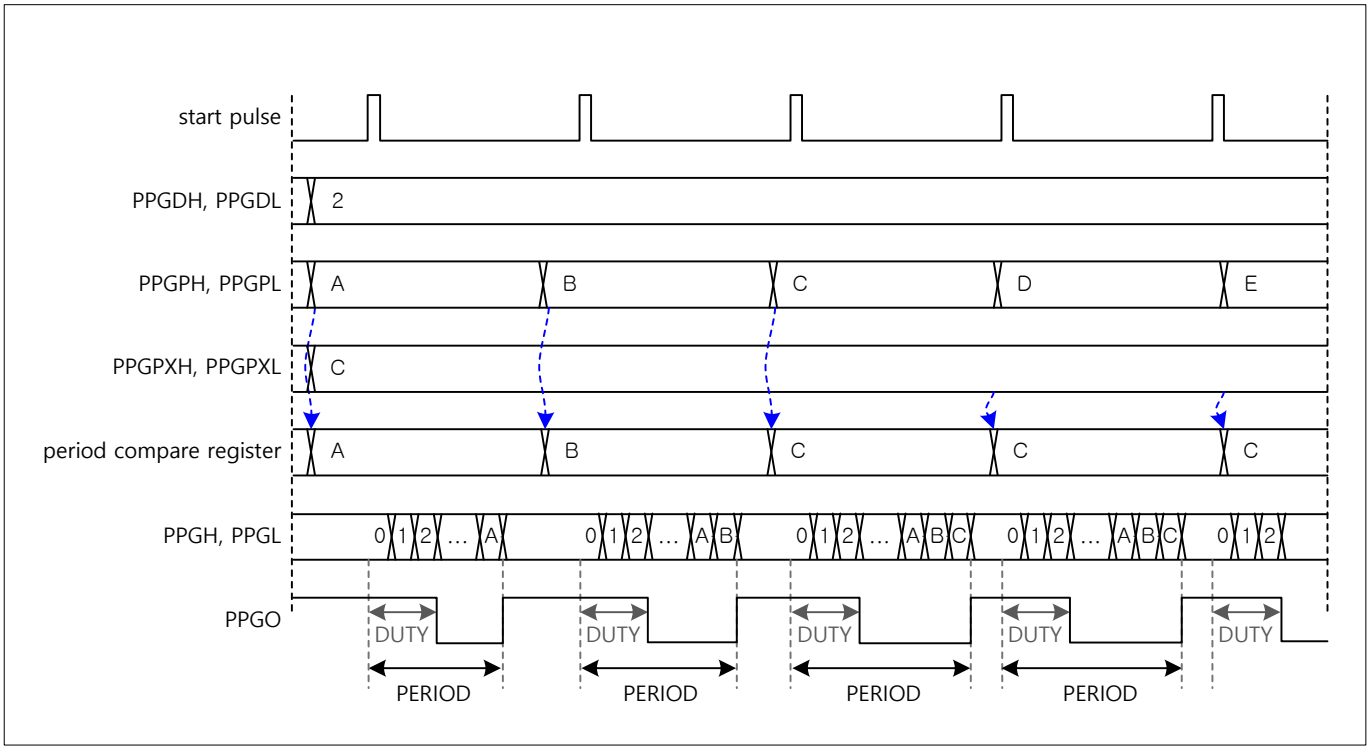


Figure 4.41 PPG period limitation

11.9.9 Auto Period Mode by comparator 2

When ATPEN = '1', PPG operates in auto period mode and PPG period is changed automatically. When comparator 2 outputs high, ATPHR,ATPLR (the period register in auto period mode) is decreased by DSTEP and it is applied to the next cycle. If comparator 2 output does not occur during one cycle, the ATPHR/ATPLR is set to the following three value, current value(ATPHR,ATPLR), initially set value(PPGPH,PPGPL), and increased value by USTEP. The three value is selectable by ATPSEL[1:0] in the ATPCR register. DSTEP and USTEP is 8-bit register which represent the amount of increase or decrease.

In duty state Comparator2 is not detected.

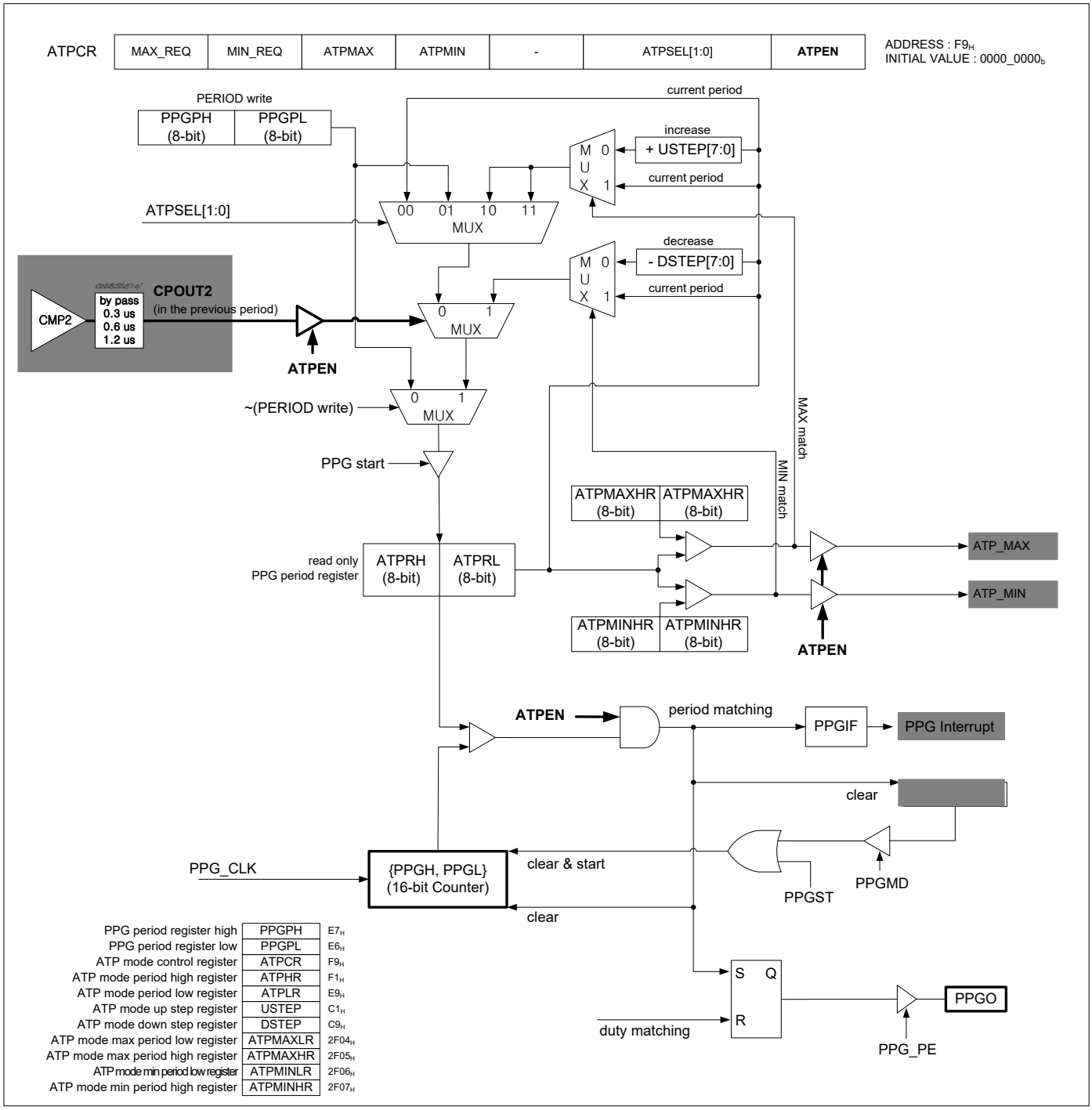


Figure 4.42 Auto Period Mode block diagram

11.9.9.1 PPG period decrease

When comparator 2 outputs high, ATPHR,ATPLR (the period register in auto period mode) is decreased by DSTEP and it is applied to the next cycle.

Note) In duty state Comparator2 is not detected.

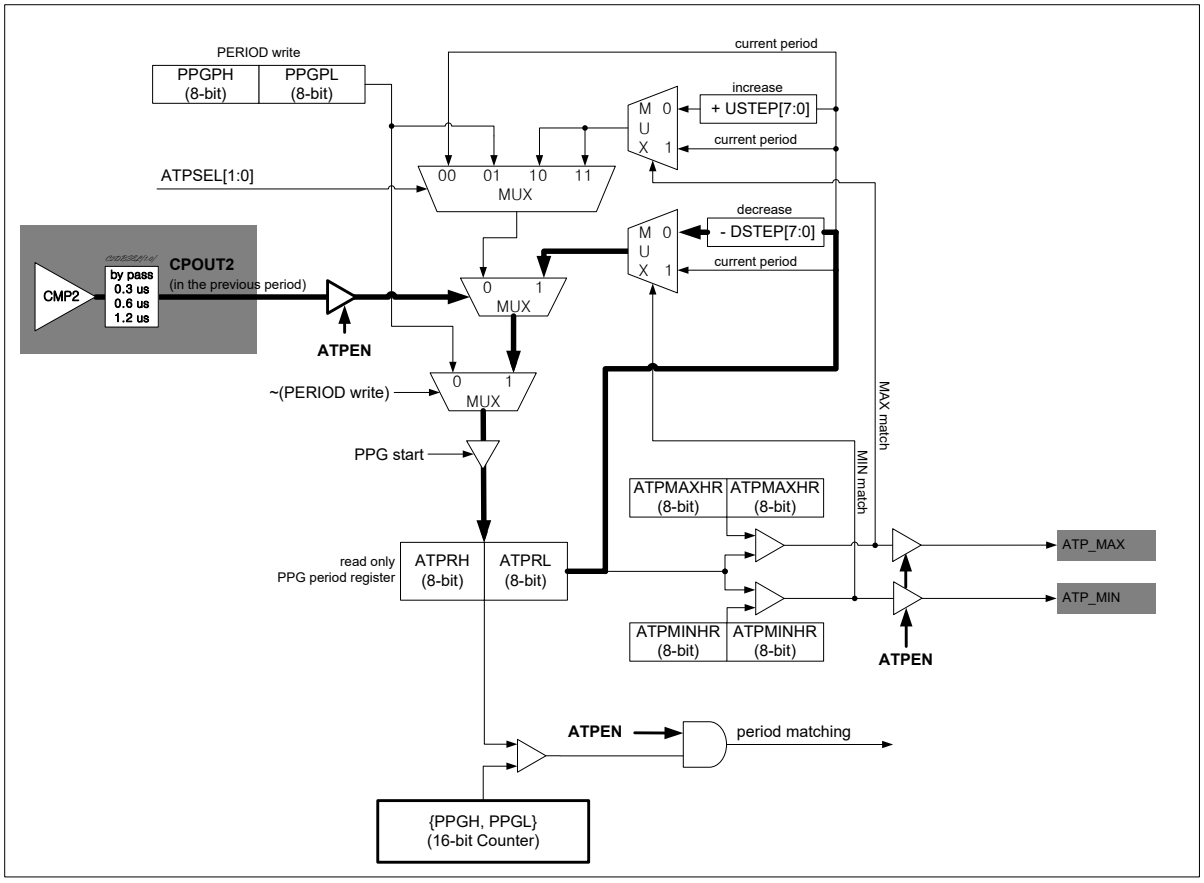


Figure 4.43 Period decrement in auto period mode block diagram

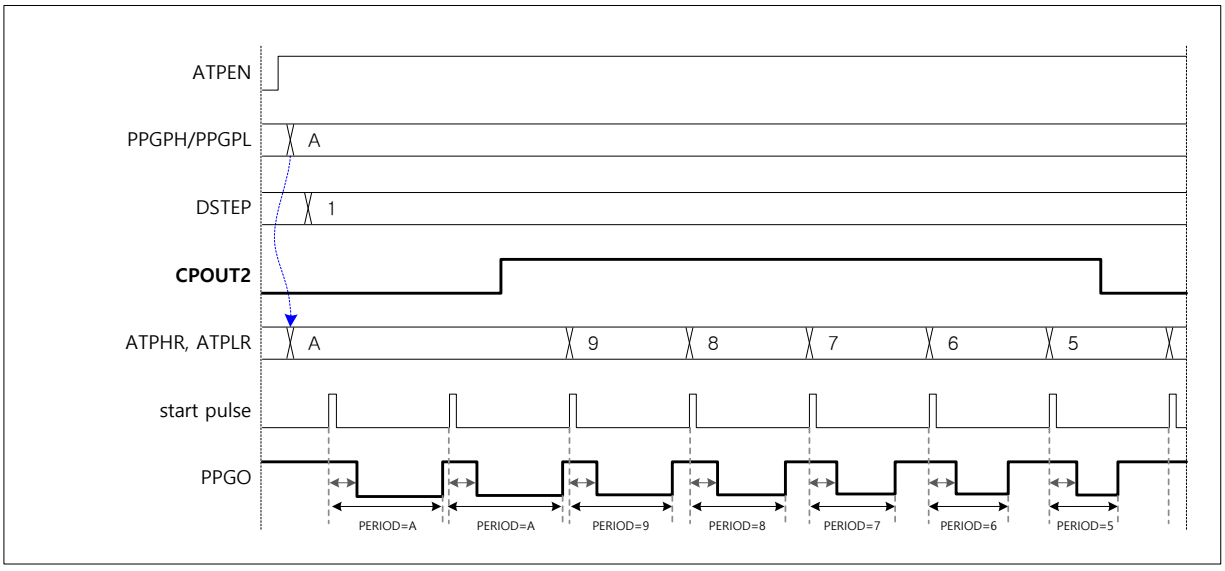


Figure 4.44 Period decrement in auto period mode

11.9.9.2 PPG period when ATPSEL = 2'b00

When comparator 2 outputs low and ATPSEL = 2'b00, ATPHR,ATPLR (the period register in auto period mode) is maintained as the current period.

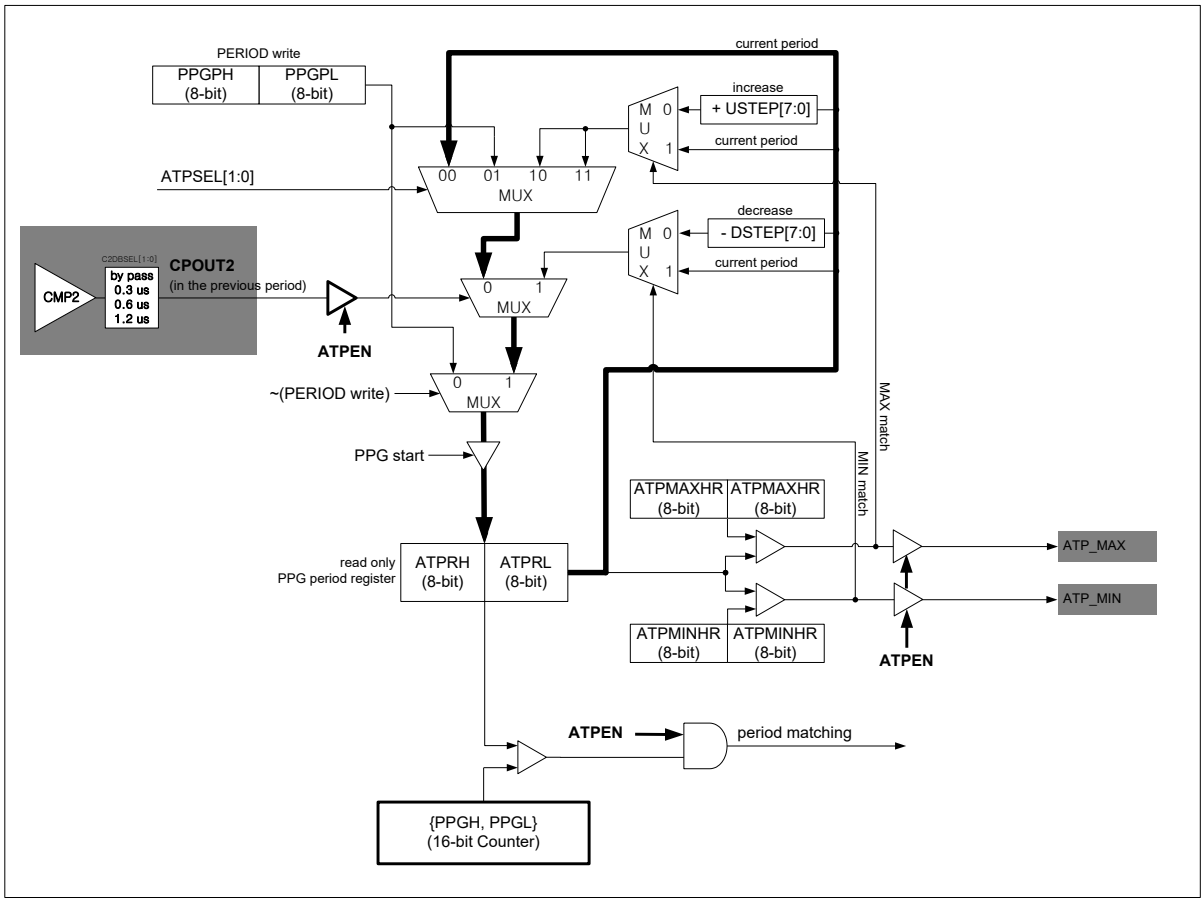


Figure 4.45 Auto period mode block diagram (ATPSEL = 2'b00)

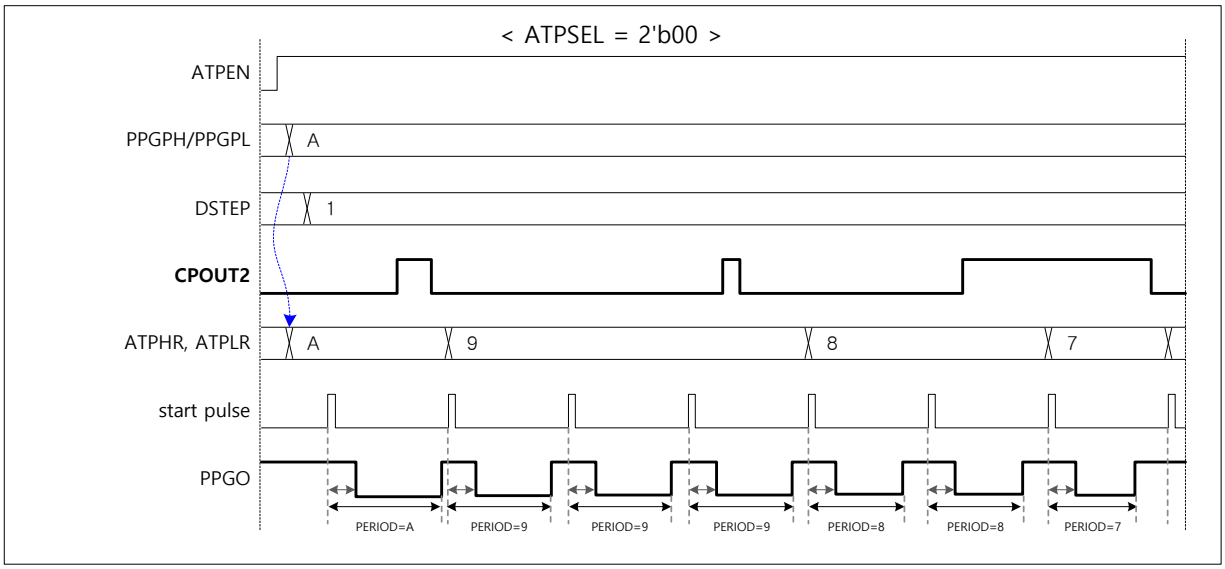


Figure 4.46 Auto period mode (ATPSEL = 2'b00)

11.9.9.3 PPG period when ATPSEL = 2'b01

When comparator 2 outputs low and ATPSEL = 2'b01, ATPHR,ATPLR (the period register in auto period mode) is PPGPH,PPGPL.

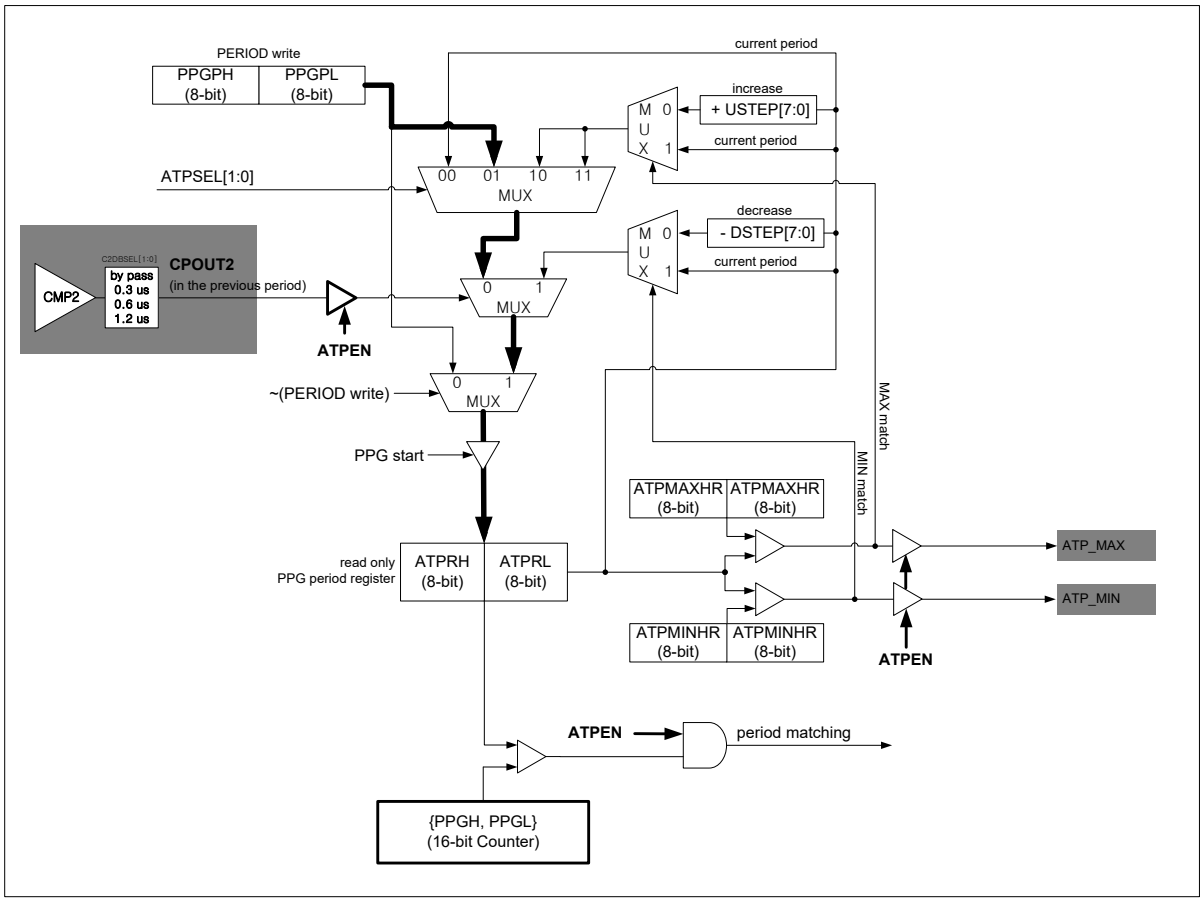


Figure 4.47 Auto period mode block diagram (ATPSEL = 2'b01)

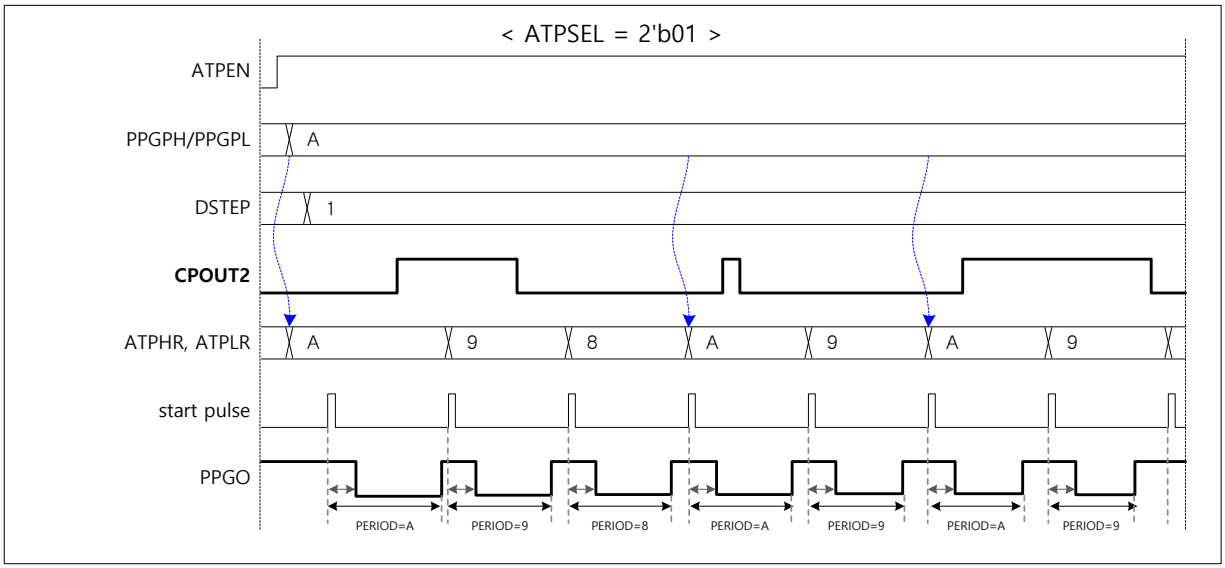


Figure 4.48 Auto period mode (ATPSEL = 2'b01)

11.9.9.4 PPG period when ATPSEL = 2'b1x

When comparator 2 outputs low and ATPSEL = 2'b1x, ATPHR,ATPLR (the period register in auto period mode) is increased by USTEP and it is applied to the next cycle.

Note) ATPEN should be set to '1' before writing a value in a period register (ATPHR, ATPLR). Otherwise the cycle starts with the increased period value by USTEP.

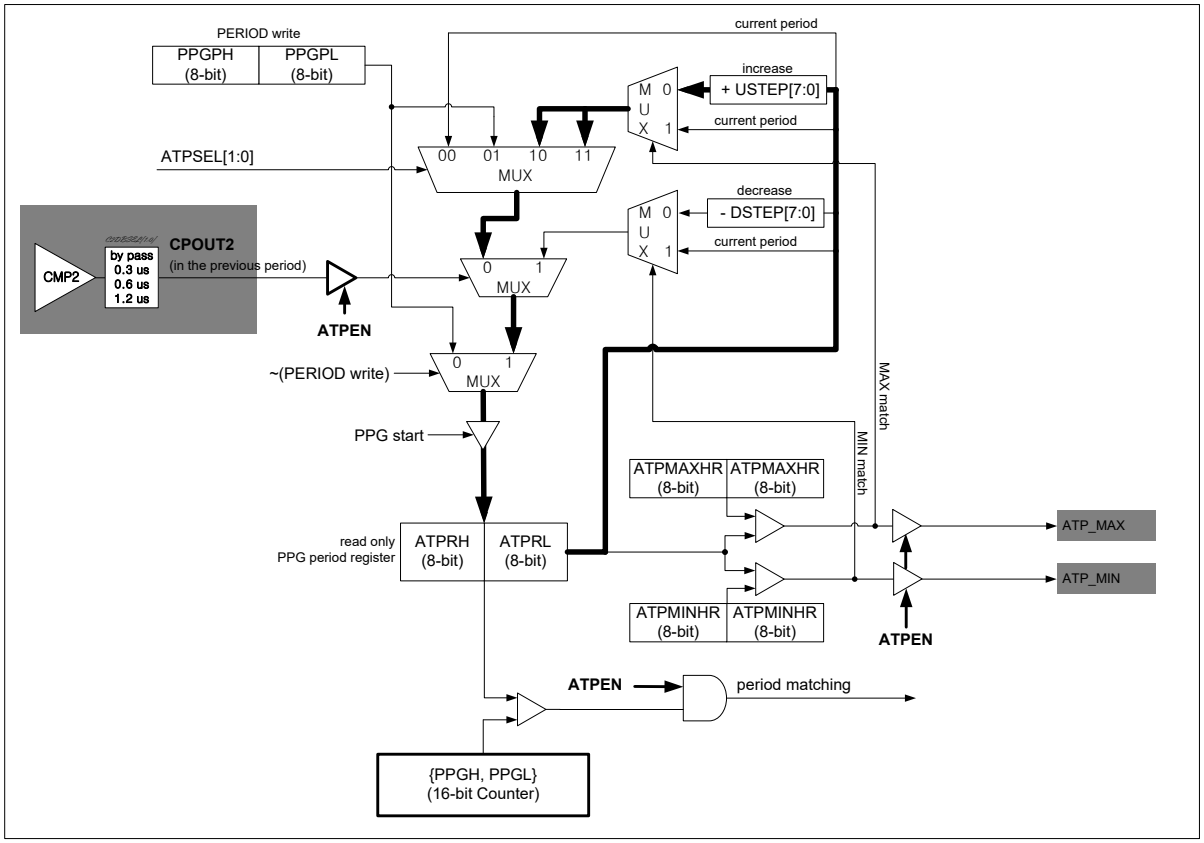


Figure 4.49 Auto period mode block diagram (ATPSEL = 2'b1x)

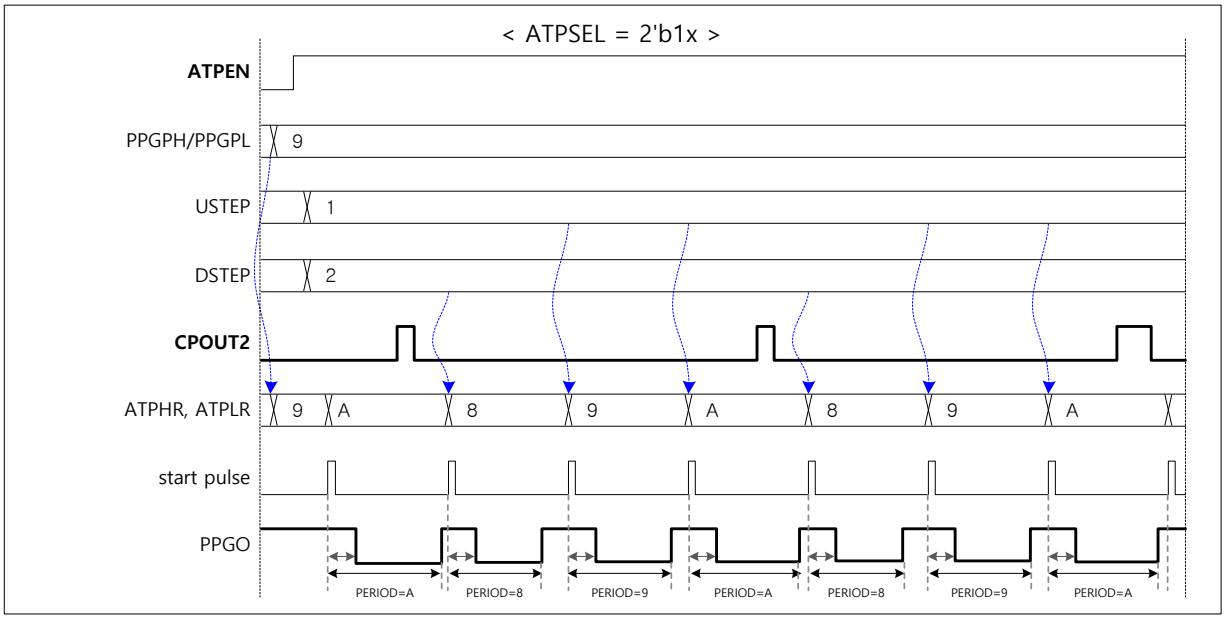


Figure 4.50 Auto period mode (ATPSEL = 2'b1x)

11.9.9.5 PPG period when writing

When writing to PPGPH/PPGPL, written period value is loaded into the ATPHR/ATPLR in the next cycle and other function is ignored.

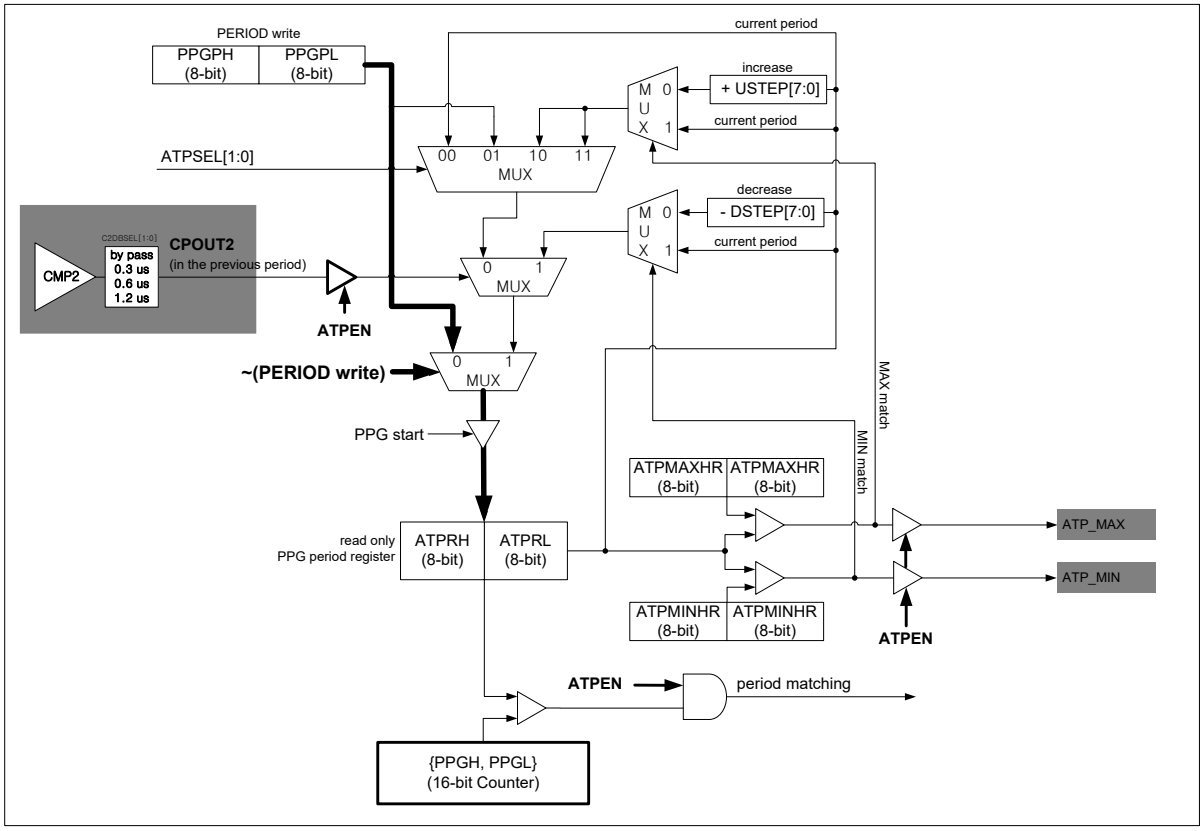


Figure 4.51 PPG period block diagram when writing

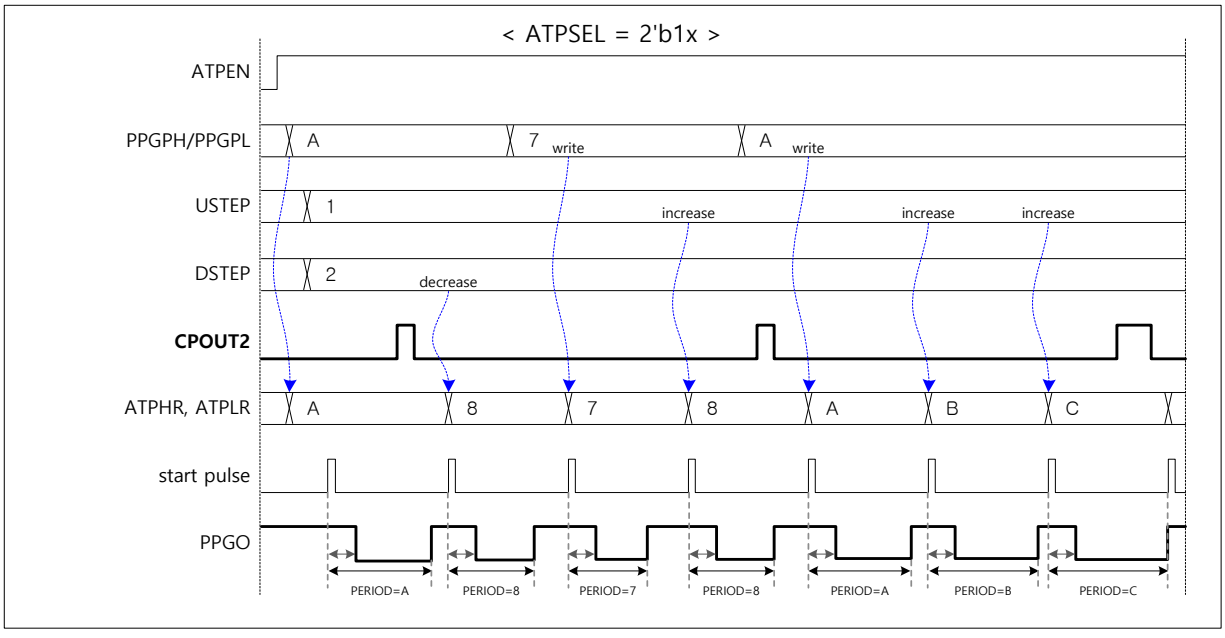
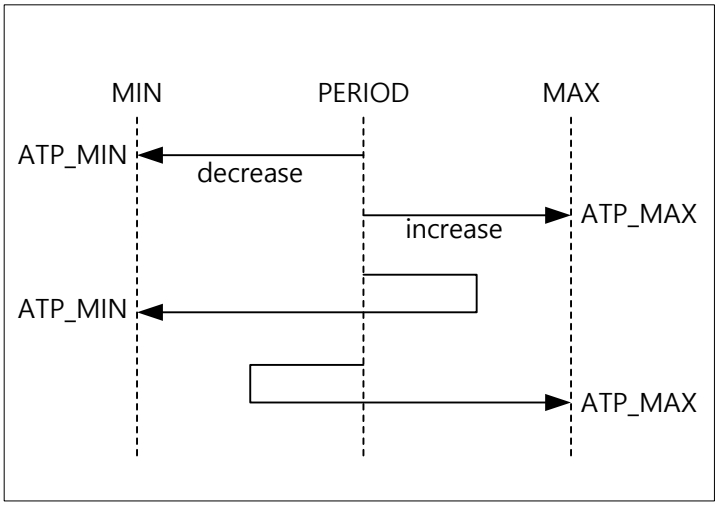


Figure 4.52 PPG period when writing (ATPSEL = 2'b1x)

11.9.9.6 PPG period min/max limitation

When ATPHR/ATPLR is increasing, if ATPHR/ATPLR value and ATPMAXHR/ATPMAXLR matches, ATP_MAX (max period matching flag) set to '1'. When ATPHR/ATPLR is decreasing, if ATPHR/ATPLR value and ATPMINHR/ATPMINLR matches, ATP_MIN (min matching flag) set to '1'.



ATP_MAX and ATP_MIN is assigned to interrupt vector 4 and 5. In the auto period mode, the PPG period is not greater than the max value, not less than the min value. But when writing to the PPGPH/PPGPL, even if PPGPH/PPGPL is outside the range of the min and max value, PPGPH/PPGPL is loaded to the ATPHR/ATPLR, and ATP_MAX or ATP_MIN set to '1'.

ATP_MAX and ATP_MIN is cleared by hardward before interrupt service routine is served or by writing '0' .

When ATP_MAX or ATP_MIN occur, PPG current period is maintained to the next cycle.

Figure 4.53 Max & min period limitation

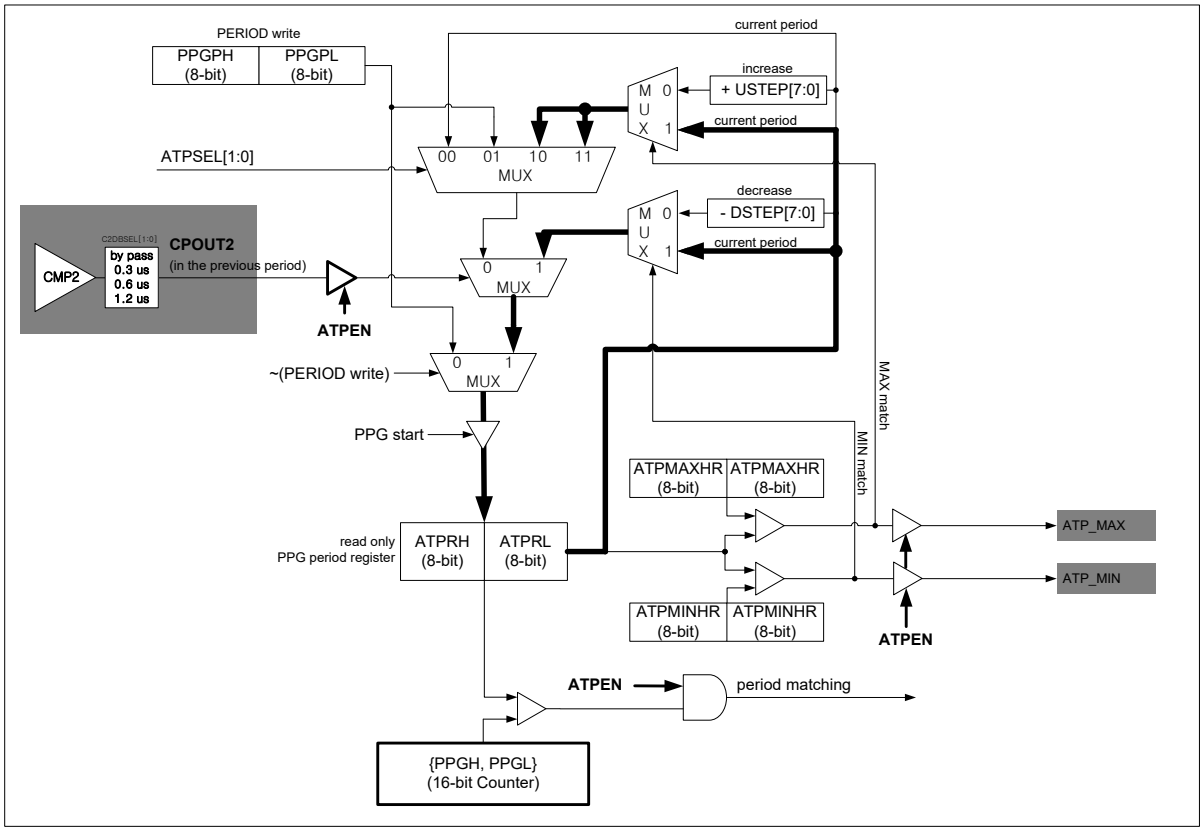


Figure 4.54 PPG period block diagram when period min/max matching

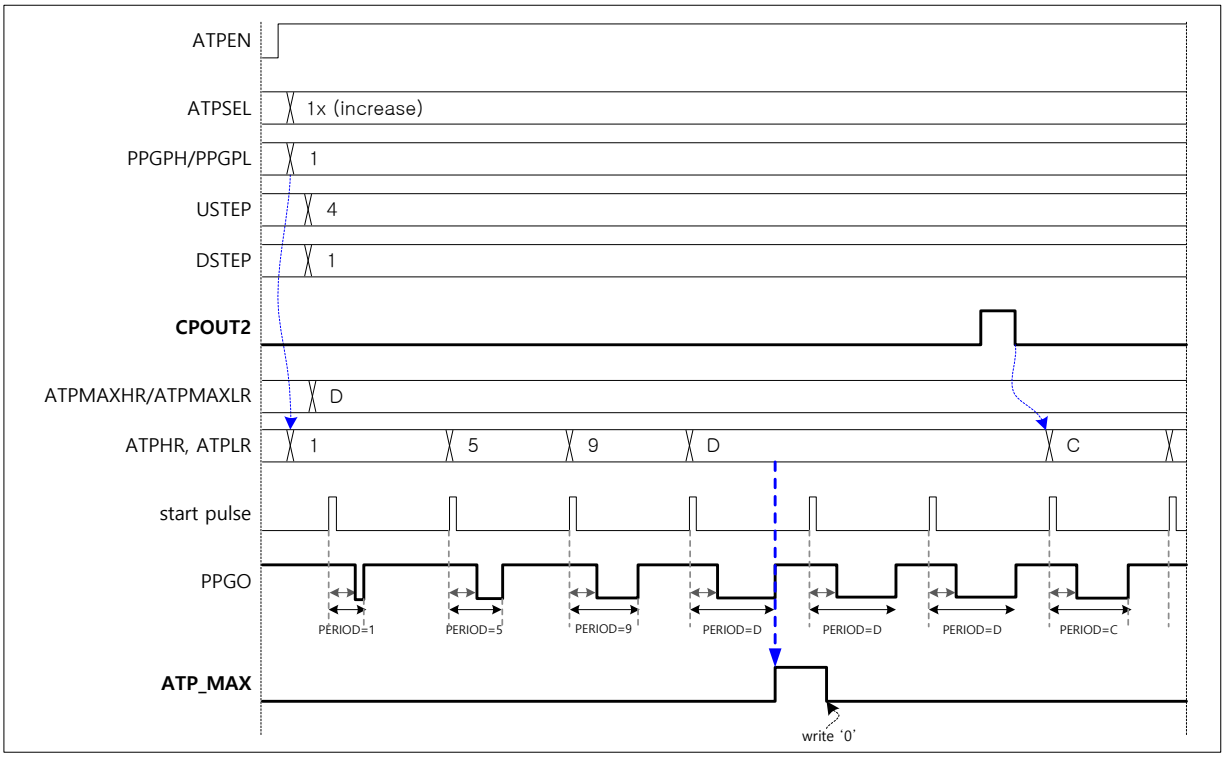


Figure 4.55 When max period matching

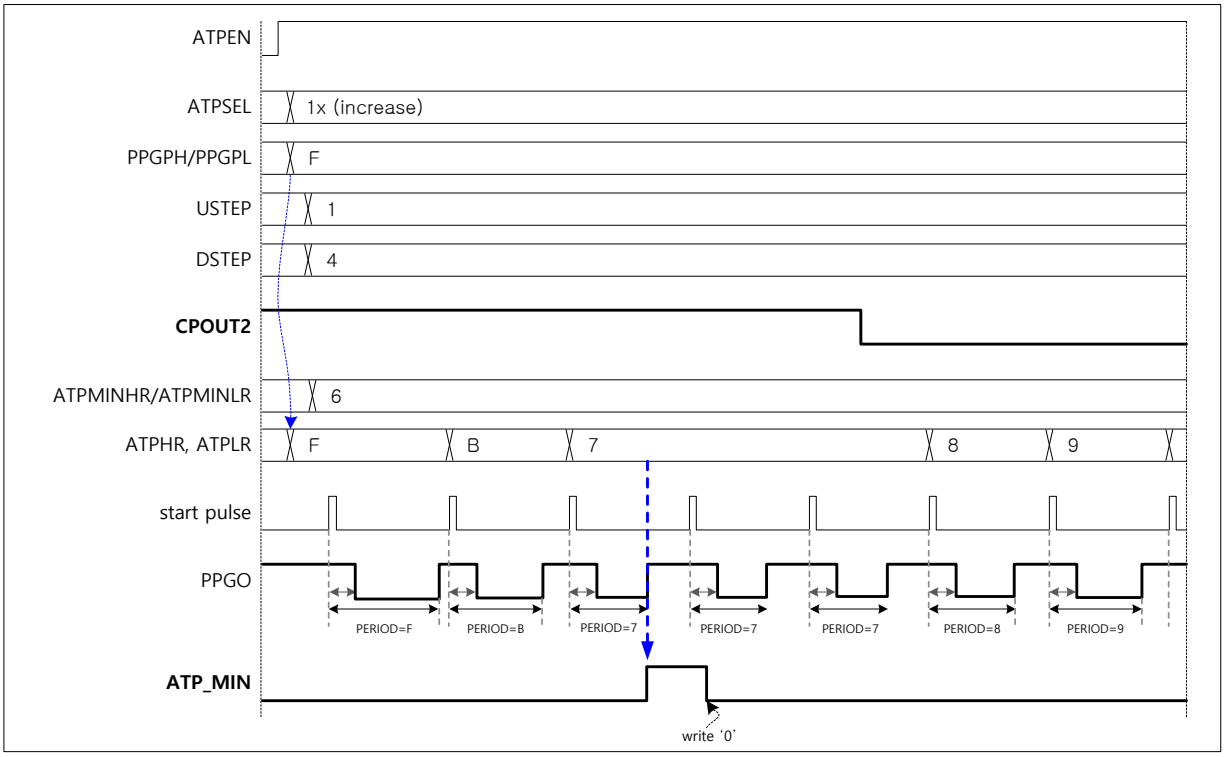


Figure 4.56 When min period matching

11.9.9.7 PPG off-time max/min limitation

PPG off-time can be limited by OFFCR register setting to prevent that PPG off-time is too long or too short. If PPG off-time is too long and start source is not received, PPG start automatically after a certain period of time. PPG max off-time is set to the OFFMAXHR,OFFMAXLR registers. Off-time max limitation is enabled by OFFMAX in the OFFCR register.

To prevent PPG off-time is too short, C0_FLAG is not generated during specified time after PPG off. PPG min off-time is set to the OFFMINHR,OFFMINLR registers and off-time min limitation is enabled by OFFMIN bit in the OFFCR register.

Note) 1. Do not change the off-time max/min registers, while OFMAX='1' or OFMIN='1'.
2. Off-time max/min registers must satisfy the following condition.
{OFFMAXHR,OFFMAXLR} > {OFFMINHR,OFFMINLR}

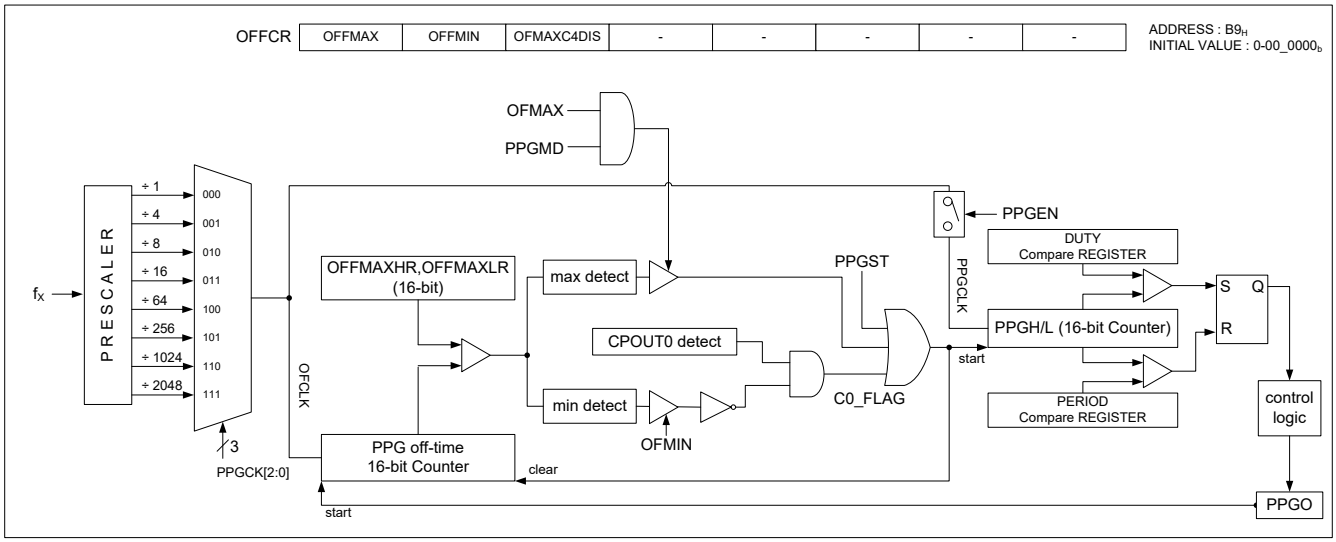


Figure 4.57 PPG off-time max/min limitation block diagram

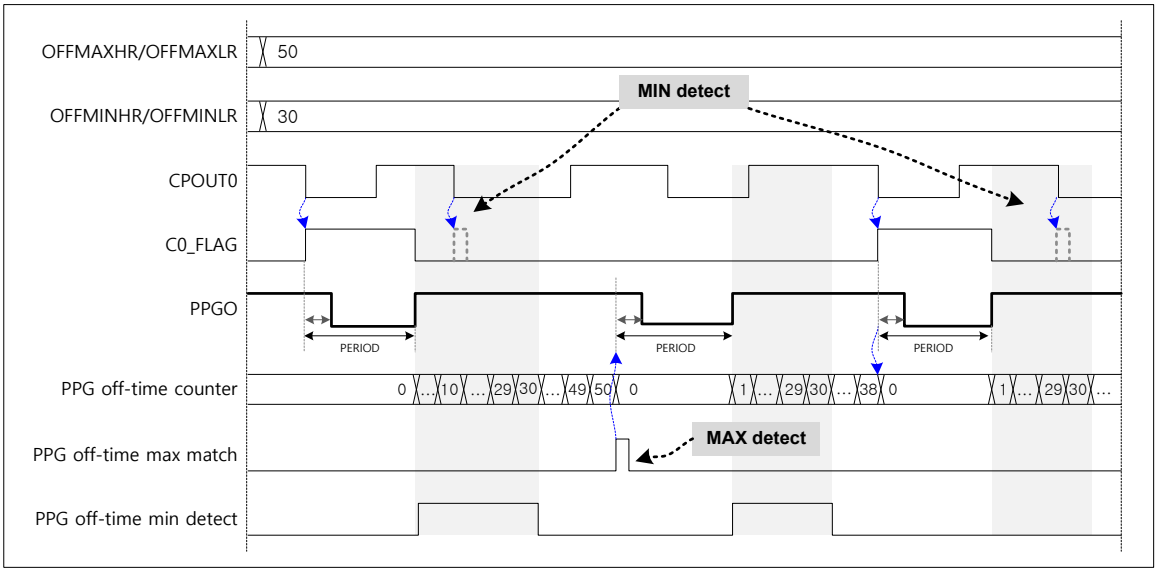


Figure 4.58 PPG off-time max/min limitation

11.9.10 Register Map

Name	Address	Dir	Default	Description
PPGCR	$E2_H$	R/W	00_H	PPG Control Register
PPGCR1	$E3_H$	R/W	00_H	PPG Control Register 1
PPGCR2	$E1_H$	R/W	00_H	PPG Control Register 2
PPGL	$D3_H$	R	00_H	PPG Low Register
PPGCL	$D3_H$	R	00_H	PPG Capture Data Register Low
PPGH	$D4_H$	R	00_H	PPG Register High
PPGCH	$D4_H$	R	00_H	PPG Capture Data Register High
PPGDL	$E4_H$	R/W	00_H	PPG Duty Register Low
PPGDH	$E5_H$	R/W	00_H	PPG Duty Register High
PPGPL	$E6_H$	R/W	FF_H	PPG Period Register Low
PPGPH	$E7_H$	R/W	FF_H	PPG Period Register High
PPGPXH	$D9_H$	R/W	00_H	PPG Max Period Register High
PPGPXL	$D1_H$	R/W	00_H	PPG Max Period Register Low
ATPCR	$F9_H$	R/W	00_H	Auto Period Mode Control Register
USTEP	$C1_H$	R/W	00_H	Auto Period Mode Up Step Register
DSTEP	$C9_H$	R/W	00_H	Auto Period Mode Down Step Register
ATPLR	$E9_H$	R	FF_H	Auto Period Mode Period Low Register
ATPHR	$F1_H$	R	FF_H	Auto Period Mode Period High Register
ATPMAXLR	$2F04_H$	R/W	FF_H	Auto Period Mode Max Period High Register
ATPMAXHR	$2F05_H$	R/W	FF_H	Auto Period Mode Max Period Low Register
ATPMINLR	$2F06_H$	R/W	00_H	Auto Period Mode Min Period High Register
ATPMINHR	$2F07_H$	R/W	00_H	Auto Period Mode Min Period Low Register
OFFCR	$B9_H$	R/W	00_H	PPG Off Time Control Register
OFFMAXLR	$2F0C_H$	R/W	FF_H	PPG Max Off-Time Low Register
OFFMAXHR	$2F0D_H$	R/W	FF_H	PPG Max Off-Time High Register
OFFMINLR	$2F0E_H$	R/W	00_H	PPG Min Off-Time Low Register
OFFMINHR	$2F0F_H$	R/W	00_H	PPG Min Off Time High Register

Table 11-17. Register Map

11.9.11 Register description for PPG

PPGCR (PPG Control Register): E2H

7	6	5	4	3	2	1	0
PPGEN	-	CAPE	PPGCK2	PPGCK1	PPGCK0	PPGCN	PPGST
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PPGEN Enable PPG
- 0 PPG disable
- 1 PPG enable

CAPE capture mode enable
- 0 Disable capture mode
- 1 Enable capture mode

PPGCK[2:0] Select clock source of PPG. Fx is the frequency of main system

PPGCK2	PPGCK1	PPGCK0	description
0	0	0	f_X
0	0	1	f_X /4
0	1	0	f_X /8
0	1	1	f_X /16
1	0	0	f_X /64
1	0	1	f_X /256
1	1	0	f_X /1024
1	1	1	f_X /2048

PPGCN Control PPG Count pause/continue.
- 0 Temporary count stop
- 1 Continue count

PPGST Control PPG start/stop
- 0 Counter stop
- 1 Clear counter and start

Note) set PPGST bit after write to other PPG registers.

PPGCR1 (PPG Control Register 1) : E3H

7	6	5	4	3	2	1	0
-	-	PPGIN[2]	PPGIN[1]	PPGIN[0]	-	PPG_PE	-
-	-	R/W	R/W	R/W	-	R/W	-

Initial value : 00_H

PPGIN[2:0] Select Event Counter and comparator Interrupt for Capture mode

PPGIN2	PPGIN1	PPGIN0	description
0	0	0	-
0	0	1	-
0	1	0	CMP3IF
0	1	1	CMP1IF
1	0	0	CMP4IF
1	0	1	-
1	1	0	-
1	1	1	-

PPG_PE Control PPG Output port

0 PPG Output disable

1 PPG Output enable

0 Negative (Duty Match: Clear)

1 Positive (Duty Match: Set)

NOTE)

1. PPGCR1.0 must be kept 0.

PPGCR2 (PPG Control Register 1) : E1H

7	6	5	4	3	2	1	0
PPGIF	TPPGOen	PPGMD	DETC3EN	DETC1EN	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-

Initial value : 00_H

PPGIF If PPG Interrupt occurs, the flag becomes '1'. The flag can be cleared by writing a '1' to bit. It is also cleared automatically after interrupt service routine is served.

0 PPG interrupt not occurred

1 PPG interrupt occurred

TPPGOen PPG output to P07 port

0 disable

1 enable

PPGMD PPG start source control

0 Only PPGST can start PPG

1 PPGST or comparator 0 interrupt flag can start PPG

DETC3EN Enable PPG output disable by comparator 3 interrupt (one period)

0 Disable

1 Enable

DETC1EN Enable PPG output disable by comparator 1 interrupt

0 Disable

1 Enable

PPGL (PPG Register Low, Read Case) : D3H

7	6	5	4	3	2	1	0
PPGL7	PPGL6	PPGL5	PPGL4	PPGL3	PPGL2	PPGL1	PPGL0
R	R	R	R	R	R	R	R

Initial value : 00_H

PPGL[7:0] PPGL Counter Low data.

PPGCL (PPG Capture Data Register Low, Read Case) : D3H

7	6	5	4	3	2	1	0
PPGCL7	PPGCL6	PPGCL5	PPGCL4	PPGCL3	PPGCL2	PPGCL1	PPGCL0
R	R	R	R	R	R	R	R

Initial value : 00_H

PPGCL[7:0] PPG Capture Low data.

PPGH (PPG Register High, Read Case) : D4H

7	6	5	4	3	2	1	0
PPGH7	PPGH6	PPGH5	PPGH4	PPGH3	PPGH2	PPGH1	PPGH0
R	R	R	R	R	R	R	R

Initial value : 00_H

PPGH[7:0] PPGH Counter Period High data.

PPGCH (PPG Capture Data High Register, Read Case) : D4H

7	6	5	4	3	2	1	0
PPGCH7	PPGCH6	PPGCH5	PPGCH4	PPGCH3	PPGCH2	PPGCH1	PPGCH0
R	R	R	R	R	R	R	R

Initial value : 00_H

PPGCH[7:0] PPG Capture High data

PPGDL (PPG Duty Register Low) : E4H

7	6	5	4	3	2	1	0
PPGDL7	PPGDL6	PPGDL5	PPGDL4	PPGDL3	PPGDL2	PPGDL1	PPGDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PPGDL[7:0] PPG Duty Low data

PPGDH (PPG Duty Register High) : E5H

7	6	5	4	3	2	1	0
PPGDH7	PPGDH6	PPGDH5	PPGDH4	PPGDH3	PPGDH2	PPGDH1	PPGDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PPGDH[7:0] PPG Duty High data

PPGPL (PPG Period Register Low) : E6H

7	6	5	4	3	2	1	0
PPGPL7	PPGPL6	PPGPL5	PPGPL4	PPGPL3	PPGPL2	PPGPL1	PPGPL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

PPGPL[7:0] PPG Period Low data

PPGPH (PPG Period Register High) : E7H

7	6	5	4	3	2	1	0
PPGPH7	PPGPH6	PPGPH5	PPGPH4	PPGPH3	PPGPH2	PPGPH1	PPGPH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FF_H

PPGPH[7:0] PPG Period High data

PPGPXH (PPG Max Period Register High) : D9H

7	6	5	4	3	2	1	0
PPGPXH7	PPGPXH6	PPGPXH5	PPGPXH4	PPGPXH3	PPGPXH2	PPGPXH1	PPGPXH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PPGPXH[7:0] PPG max period register high

PPGPXL (PPG Max Period Register Low) : D1H

7	6	5	4	3	2	1	0
PPGPXL7	PPGPXL6	PPGPXL5	PPGPXL4	PPGPXL3	PPGPXL2	PPGPXL1	PPGPXL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

PPGPXL[7:0] PPG max period register low

ATPCR (Auto Period Mode Control Register) : F9H

7	6	5	4	3	2	1	0
MAX_REQ	MIN_REQ	ATPMAX	ATPMIN	-	ATPSEL1	ATPSEL0	ATPEN
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00_H

MAX_REQ — If PPG period max matching occurs, the flag becomes '1'. The flag is cleared before interrupt service routine is served. It is also cleared by writing a '1'.
- 0 PPG period max matching is not occurred
- 1 PPG period max matching is occurred

MIN_REQ — If PPG period min matching occurs, the flag becomes '1'. The flag is cleared before interrupt service routine is served. It is also cleared by writing a '1'.
- 0 PPG period min matching is not occurred
- 1 PPG period min matching is occurred

ATPMAX — Enable or Disable max period matching in the auto period mode
- 0 Disable
- 1 Enable

ATPMIN — Enable or Disable min period matching in the auto period mode
- 0 Disable
- 1 Enable

ATPSEL[1:0] — Configure PPG period, when the output of comparator 2 has not occurred, in the auto period mode.
- 00 {ATPHR,ATPLR} = {ATPHR,ATPLR}
- 01 {ATPHR,ATPLR} = {PPGPWMPH,PPGPWMPL}
- 1x {ATPHR,ATPLR} = {ATPHR,ATPLR} + USTEP

ATPEN — Auto period mode enable
- 0 disable
- 1 Enable

USTEP (Auto Period Mode Up Step Register) : C1H

7	6	5	4	3	2	1	0
USTEP7	USTEP6	USTEP5	USTEP4	USTEP3	USTEP2	USTEP1	USTEP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

USTEP[7:0] Auto Period Mode Up Step Register

DSTEP (Auto Period Mode Down Step Register) : C9H

7	6	5	4	3	2	1	0
DSTEP7	DSTEP6	DSTEP5	DSTEP4	DSTEP3	DSTEP2	DSTEP1	DSTEP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

DSTEP[7:0] Auto Period Mode Down Step Register

ATPLR (Auto Period Mode Period Low Register) : E9H

7	6	5	4	3	2	1	0
ATPLR7	ATPLR6	ATPLR5	ATPLR4	ATPLR3	ATPLR2	ATPLR1	ATPLR0
R	R	R	R	R	R	R	R

Initial value : FF_H

ATPLR[7:0] Auto Period Mode Period Low Register

ATPHR (Auto Period Mode Period High Register) : F1H

7	6	5	4	3	2	1	0
ATPHR7	ATPHR6	ATPHR5	ATPHR4	ATPHR3	ATPHR2	ATPHR1	ATPHR0
R	R	R	R	R	R	R	R

Initial value : FF_H

ATPHR[7:0] Auto Period Mode Period High Register

ATPMAXLR (Auto Period Mode Max Period Low Register) : 2F04H

7	6	5	4	3	2	1	0
ATPMAXLR7	ATPMAXLR6	ATPMAXLR5	ATPMAXLR4	ATPMAXLR3	ATPMAXLR2	ATPMAXLR1	ATPMAXLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

ATPMAXLR[7:0] Auto Period Mode Max Period Low Register

ATPMAXHR (Auto Period Mode Max Period High Register) : 2F05H

7	6	5	4	3	2	1	0
ATPMAXHR7	ATPMAXHR6	ATPMAXHR5	ATPMAXHR4	ATPMAXHR3	ATPMAXHR2	ATPMAXHR1	ATPMAXHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

ATPMAXHR[7:0] Auto Period Mode Max Period High Register

ATPMINLR (Auto Period Mode Min Period Low Register) : 2F06H

7	6	5	4	3	2	1	0
ATPMINLR7	ATPMINLR6	ATPMINLR5	ATPMINLR4	ATPMINLR3	ATPMINLR2	ATPMINLR1	ATPMINLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

ATPMINLR[7:0] Auto Period Mode Min Period Low Register

ATPMINHR (Auto Period Mode Min Period High Register) : 2F07H

7	6	5	4	3	2	1	0
ATPMINHR7	ATPMINHR6	ATPMINHR5	ATPMINHR4	ATPMINHR3	ATPMINHR2	ATPMINHR1	ATPMINHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

ATPMINHR[7:0] Auto Period Mode Min Period High Register

OFFCR (PPG Off-Time Control Register) : B9H

7	6	5	4	3	2	1	0
OFMAX	OFMIN	0	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-

Initial value : 00_H

OFMAX Enable or Disable PPG max off-time.
0 Disable
1 Enable

OFMIN Enable or Disable PPG min off time
0 Disable
1 Enable

OFFMAXLR (PPG Off Time Max Period Low Register) : 2F0CH

7	6	5	4	3	2	1	0
OFFMAXLR7	OFFMAXLR6	OFFMAXLR5	OFFMAXLR4	OFFMAXLR3	OFFMAXLR2	OFFMAXLR1	OFFMAXLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

OFFMAXLR[7:0] PPG Off Time Max Period Low Register

OFFMAXHR (PPG Off Time Max Period High Register) : 2F0DH

7	6	5	4	3	2	1	0
OFFMAXHR7	OFFMAXHR6	OFFMAXHR5	OFFMAXHR4	OFFMAXHR3	OFFMAXHR2	OFFMAXHR1	OFFMAXHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

OFFMAXHR[7:0] PPG Off Time Max Period High Register

OFFMINLR (PPG Off Time Min Period Low Register) : 2F0EH

7	6	5	4	3	2	1	0
OFFMINLR7	OFFMINLR6	OFFMINLR5	OFFMINLR4	OFFMINLR3	OFFMINLR2	OFFMINLR1	OFFMINLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

OFFMINLR[7:0] PPG Off Time Min Period Low Register

OFFMINHR (PPG Off Time Min Period High Register) : 2F0FH

7	6	5	4	3	2	1	0
OFFMINHR7	OFFMINHR6	OFFMINHR5	OFFMINHR4	OFFMINHR3	OFFMINHR2	OFFMINHR1	OFFMINHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

OFFMINLR[7:0] PPG Off Time Min Period High Register

11.10 Analog Comparator & OP-AMP

11.10.1 overview

The MC97F6108A has five analog comparators and two operational amplifiers whose input is external analog inputs and whose output is source of internal peripheral circuits.

Each comparator has output noise canceler, de-bounce and interrupt circuits. De-bounce length for each comparator is selectable by setting CxDBSEL[1:0]. The comparator's interrupt vector address and control registers are described in interrupt controller section. In addition to the interrupt flags, each comparator has comparator flags. Interrupt flags work only for interrupt and they are cleared by hardware and software. But comparator flags are connected to PPG block and cleared only by writing '0', some of them are cleared by PPG operation.

Comparator's de-bounce output and flags are connected to PPG, interrupt control block and timers and comparator output port CMPXO(P01). By setting CMPOSL[2:0] register, one comparator is selected to the port.

Comparator's internal reference voltage level is selectable by software and CMPEN enables all comparators.

AMP1 output enters into the input of AMP2 and comparator 3. AMP2 output is connected to AMP2O port through the voltage divider circuit. If voltage divider option is selected by setting AOSEL and AORA_EN=1, AMP2 output is divided. If AORA_EN=0 and AOSEL=111111b, the voltage divider is disabled. AMP2 output is directly connected to ADC block when AMP2O_EN = 1 and ADC channel 7 is selected. OPAEN enables OP-AMP operation.

OINPen, C0INNen, C0INPen, C1INPen and C2INPen enable each port to receive external analog source and AMP1O_EN and AMP2O_EN enable AMP1 and AMP2 output to each port. In order to prevent the analog input entering to digital circuit, PSR register need to be set.

11.10.2 comparator and OP-AMP description

- Comparator 0:
  - "+" and "–" input are external analog port (CMP0_IN_P, CMP0_IN_N).
  - Output(CPOUT0) generates interrupt flag(CMP0IF) and comparator flag(C0_FLAG).
  - CPOUT0 is connected to the timer 0 event counter source.
  - CMP0IF is connected to a capture source of the timer 0.
  - C0_FLAG is used to start PPG.

- Comparator 1:
  - "+" input is external analog port (CMP1_IN_P).
  - "-" input is internal Vref selected by setting C1NVSL[3:0] register.
  - Output(CPOUT1) generates interrupt flag(CMP1IF) and comparator flag(C1_FLAG).
  - CPOUT1 is connected to the timer 1 event counter source.
  - CMP1IF is connected to a capture source of the timer 1 and PPG.
  - C1_FLAG is used to disable PPG output.

- Comparator 2
  - "+" input is external analog port (CMP2_IN_P).
  - "-" input is internal Vref selected by setting C2NVSL[3:0] register.
  - Output(CPOUT2) generates interrupt flag(CMP2IF) and comparator flag(C2_FLAG).
  - CPOUT2 is connected to the timer 2 event counter source.
  - CMP2IF is connected to a capture source of the timer 2.
  - CPOUT2 is connected to the PPG to control the period of PPG in auto period mode.

- Comparator 3
  - "+" input is the output of the AMP1
  - "-" input is internal Vref selected by setting C3NVSL[3:0] register.
  - Output(CPOUT3) generates interrupt flag(CMP3IF) and comparator flag(C3_FLAG).
  - CPOUT3 is connected to the timer 3 event counter source.
  - CMP3IF is connected to a capture source of the timer 3 and PPG.
  - C3_FLAG is used to disable PPG output.

- Comparator 4
  - "+" input is external analog port shared with comparator1(CMP1_IN_P).
  - "-" input is internal Vref selected by setting C4NVSL[2:0] register.
  - Output(CPOUT4) generates interrupt flag(CMP4IF) and comparator flag(C4_FLAG).
  - CMP4IF is connected to a capture source of the PPG.

- AMP1
  - "+" input is external analog port AMP1I.
  - Output is connected to the comparator 3 and AMP2 input.
  - The gain is selected by setting DGCAL1[2:0] register.

- AMP2
  - "+" input is output of AMP1.
  - Output is connected to AMP2O port and AN5 (ADC channel 5).
  - The gain is selected by setting DGCAL2[2:0] register.

11.10.3 block diagram

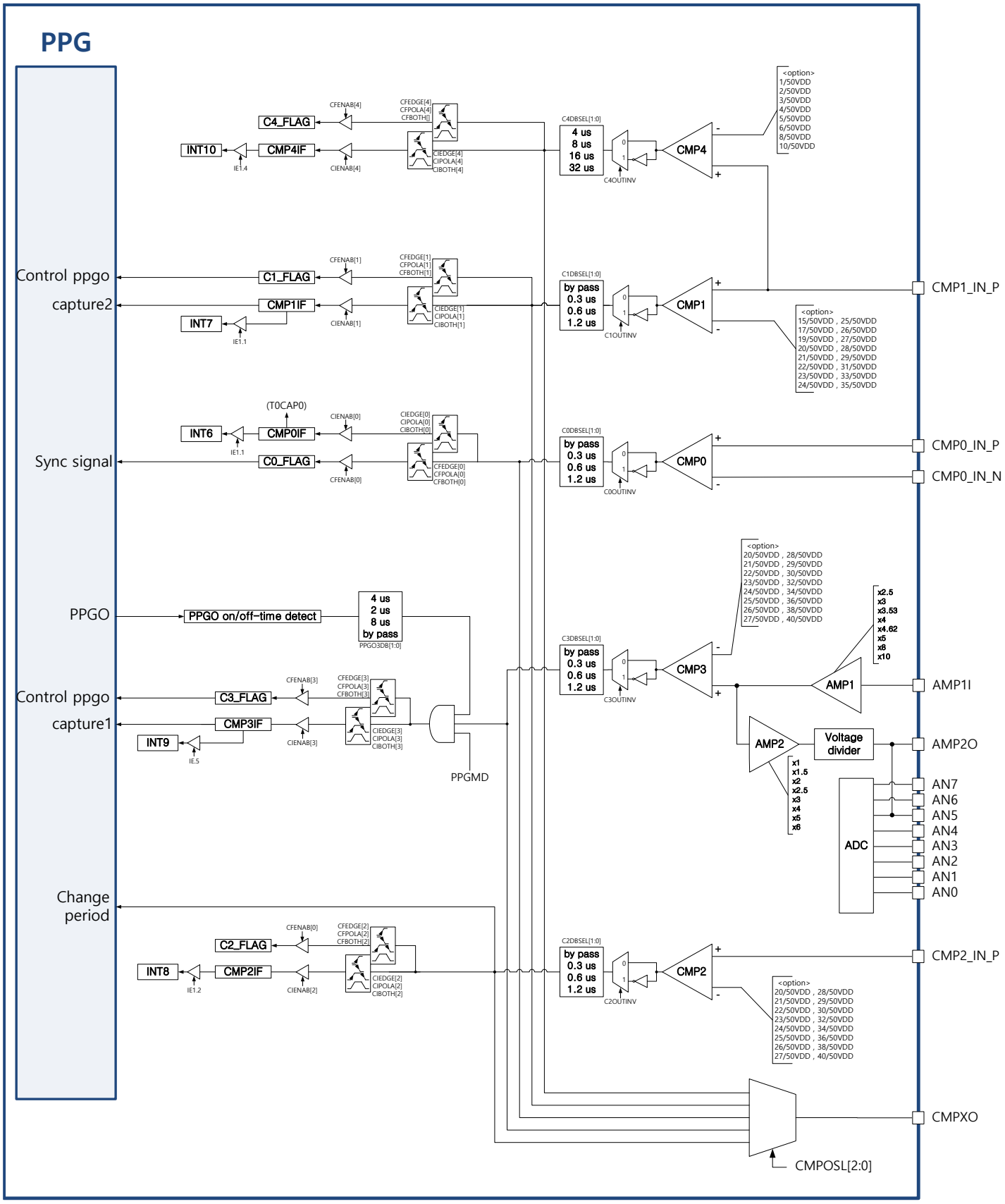


Figure 4.59 Analog comparator & OP-AMP block diagram

11.10.4 Register description

CA_REG1 (Comparator Amp Register 1) : 2F31H

7	6	5	4	3	2	1	0
-	-	-	-	OPAEN	AORA_EN	AMP2EI_EN	OINPen
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 00_H

OPAEN Control OP-AMP
- 0 OP-AMP disable
- 1 OP-AMP enable

AORA_EN Control AMP2 output voltage divider
- 0 Disable
- 1 enable

AMP2EI_EN Control AMP1IB port
- 0 AMP2 external input from P17 port disable (AMP1O connected)
- 1 AMP2 external input from P17 port enable (AMP1O disconnected)

OINPen Control AMP1I port
- 0 AMP1 input from AMP1I port disable
- 1 AMP1 input from AMP1I port enable

CA_REG2 (Comparator Amp Register 2) : 2F32H

7	6	5	4	3	2	1	0
AMP2O_EN	AMP1O_EN	AOSEL[5]	AOSEL[4]	AOSEL[3]	AOSEL[2]	AOSEL[1]	AOSEL[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

AMP2O_EN Control AMP2 output
- 0 AMP2 output disable
- 1 AMP2 output enable

AMP1O_EN Control AMP1 output
- 0 AMP1 output disable
- 1 AMP2 output enable

AOSEL[5:0] AMP2 output voltage divider selection
- 000000 AMP2STO x 6/69
- 000001 AMP2STO x 7/69
- 000010 AMP2STO x 8/69
- 000011 AMP2STO x 9/69
- 000100 AMP2STO x 10/69
- ~ ~
- 111010 AMP2STO x 64/69
- 111011 AMP2STO x 65/69
- 111100 AMP2STO x 66/69
- 111101 AMP2STO x 67/69
- 111110 AMP2STO x 68/69
- 111111 AMP2STO x 69/69

CA_REG3 (Comparator Amp Register 3) : 2F33H

7	6	5	4	3	2	1	0
-	DGCAL2[2]	DGCAL2[1]	DGCAL2[0]	-	DGCAL1[2]	DGCAL1[1]	DGCAL1[0]
-	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00_H

DGCAL2[2:0]	AMP2 DC gain selection	
	000	x 1
	001	x 1.5
	010	x 2
	011	x 2.5
	100	x 3
	101	x 4
	110	x 5
	111	x 6
DGCAL1[2:0]	AMP1 DC gain selection	
	000	x 2.5
	001	x 3
	010	x 3.53
	011	x 4
	100	x 4.62
	101	x 5
	110	x8
	111	X10

NOTE)

1. Since the calibration is conducted by the gain of 4x5 (CA_REG3 = 63h), +-4mV offset spec is guaranteed only for the gain of 4x5 (CA_REG3 = 63h).

CA_REG4 (Comparator Amp Register 4) : 2F34H

7	6	5	4	3	2	1	0
CMPEN	HYSL_EN3	HYSL_EN2	HYSL_EN1	C2INPen	C1INPen	C0INPen	C0INNen
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

CMPEN Control comparator
- 0 All Comparator Disable
- 1 All Comparator Enable

HYSL_EN3 Control Comparator 3 hysteresis
- 0 Disable
- 1 Enable

HYSL_EN2 Control Comparator 1, 2, 4 hysteresis
- 0 Disable
- 1 Enable

HYSL_EN1 Control Comparator 0 hysteresis
- 0 Disable
- 1 Enable

C2INPen Control CMP2_IN_P port
- 0 Comparator 2 input from CMP2_IN_P port disable
- 1 Comparator 2 input from CMP2_IN_P port enable

C1INPen Control CMP1_IN_P port
- 0 Comparator 1 input from CMP1_IN_P port disable
- 1 Comparator 1 input from CMP1_IN_P port enable

C0INPen Control CMP0_IN_P port
- 0 Comparator 0 input from CMP0_IN_P port disable
- 1 Comparator 0 input from CMP0_IN_P port enable

C0INNen Control CMP0_IN_N port
- 0 Comparator 0 input from CMP0_IN_N port disable
- 1 Comparator 0 input from CMP0_IN_N port enable

CA_REG5 (Comparator Amp Register 5) : 2F35H

7	6	5	4	3	2	1	0
-	C4NVSL[2]	C4NVSL[1]	C4NVSL[0]	C1NVSL[3]	C1NVSL[2]	C1NVSL[1]	C1NVSL[0]
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

C4NVSL[2:0] Comparator 1 input reference voltage selection

000	VDD x 1/50
001	VDD x 2/50
010	VDD x 3/50
011	VDD x 4/50
100	VDD x 5/50
101	VDD x 6/50
110	VDD x 8/50
111	VDD x 10/50

C1NVSL[3:0] Comparator 1 input reference voltage selection

0000	VDD x 15/50
0001	VDD x 17/50
0010	VDD x 19/50
0011	VDD x 20/50
0100	VDD x 21/50
0101	VDD x 22/50
0110	VDD x 23/50
0111	VDD x 24/50
1000	VDD x 25/50
1001	VDD x 26/50
1010	VDD x 27/50
1011	VDD x 28/50
1100	VDD x 29/50
1101	VDD x 31/50
1110	VDD x 33/50
1111	VDD x 35/50

CA_REG6 (Comparator Amp Register 6) : 2F36H

7	6	5	4	3	2	1	0
C3NVSL[3]	C3NVSL[2]	C3NVSL[1]	C3NVSL[0]	C2NVSL[3]	C2NVSL[2]	C2NVSL[1]	C2NVSL[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

C3NVSL[3:0] Comparator 3 input reference voltage selection

0000	VDD x 20/50
0001	VDD x 21/50
0010	VDD x 22/50
0011	VDD x 23/50
0100	VDD x 24/50
0101	VDD x 25/50
0110	VDD x 26/50
0111	VDD x 27/50
1000	VDD x 28/50
1001	VDD x 29/50
1010	VDD x 30/50
1011	VDD x 32/50
1100	VDD x 34/50
1101	VDD x 36/50
1110	VDD x 38/50
1111	VDD x 40/50

C2NVSL[3:0] Comparator 2 input reference voltage selection

0000	VDD x 20/50
0001	VDD x 21/50
0010	VDD x 22/50
0011	VDD x 23/50
0100	VDD x 24/50
0101	VDD x 25/50
0110	VDD x 26/50
0111	VDD x 27/50
1000	VDD x 28/50
1001	VDD x 29/50
1010	VDD x 30/50
1011	VDD x 32/50
1100	VDD x 34/50
1101	VDD x 36/50
1110	VDD x 38/50
1111	VDD x 40/50

CA_REG7 (Comparator Amp Register 7) : 2F37H

7	6	5	4	3	2	1	0
-	-	-	-	CMPXOen	CMPOSL[2]	CMPOSL[1]	CMPOSL[0]
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 00_H

CMPXOen Control CMPXO port
- 0 Comparator 0~4 output to CMPXO port disable
- 1 Comparator 0~4 output to CMPXO port enable

CMPOSL[2:0] Comparator 0~3 output selection
- 000 Comparator 0
- 001 Comparator 1
- 010 Comparator 2
- 011 Comparator 3
- 100 Comparator 4

CA_REGA (Comparator Amp Register A) : 2F3AH

7	6	5	4	3	2	1	0
C3DBSEL[1]	C3DBSEL[0]	C2DBSEL[1]	C2DBSEL[0]	C1DBSEL[1]	C1DBSEL[0]	C0DBSEL[1]	C0DBSEL[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

C3DBSEL[1:0] Comparator 3 de-bounce selection
- 00 By pass
- 01 0.3us
- 10 0.6us
- 11 1.2us

C2DBSEL[1:0] Comparator 2 de-bounce selection
- 00 By pass
- 01 0.3us
- 10 0.6us
- 11 1.2us

C1DBSEL[1:0] Comparator 1 de-bounce selection
- 00 By pass
- 01 0.3us
- 10 0.6us
- 11 1.2us

C0DBSEL[1:0] Comparator 0 de-bounce selection
- 00 By pass
- 01 0.3us
- 10 0.6us
- 11 1.2us

CA_REGB (Comparator Amp Register B) : 2F3BH

7	6	5	4	3	2	1	0
C4DBSEL[1]	C4DBSEL[0]	-	C4OUTINV	C3OUTINV	C2OUTINV	C1OUTINV	C0OUTINV
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00_H

C4DBSEL[1:0] Comparator 4 de-bounce selection
- 00 4us
- 01 8us
- 10 16us
- 11 32us

C4OUTINV Control Comparator 4 output
- 0 Non invert
- 1 Invert

C3OUTINV Control Comparator 3 output
- 0 Non invert
- 1 Invert

C2OUTINV Control Comparator 2 output
- 0 Non invert
- 1 Invert

C1OUTINV Control Comparator 1 output
- 0 Non invert
- 1 Invert

C0OUTINV Control Comparator 0 output
- 0 Non invert
- 1 Invert

CA_REGC (Comparator Amp Register C) : 2F3CH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PPGO3DB[1]	PPGO3DB[0]
-	-	-	-	-	-	R/W	R/W

Initial value : 00_H

PPGO3DB[1:0] PPGO de-bounce selection.
- 00 4us
- 01 2us
- 10 8us
- 11 By pass

12 Power Down Operation

12.1 Overview

The MC97F6108A has three power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides three kinds of power saving functions, IDLE, STOP1 and STOP2 mode. In three modes, program is stopped.

To go to STOP1 mode, WDTRC should be set by writing '1' to WDTRCON bit in SCCR register.

12.2 Peripheral Operation in IDLE/STOP Mode

Peripheral	IDLE Mode	STOP1 Mode	STOP2 Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain	Retain
Basic Interval Timer	Operates Continuously	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Operates Continuously	Stop
Timer	Operates Continuously	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)
I2C	Operates Continuously	Stop	Stop
Internal OSC (16MHz)	Oscillation	Stop	Stop
Internal RCOSC (8kHz)	Oscillation	Oscillation	Stop
I/O Port	Retain	Retain	Retain
Control Register	Retain	Retain	Retain
Address Data Bus	Retain	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, SPI (External clock), External Interrupt (with PCI), WDT, BIT, BOD, TIMER(EC)	By RESET, SPI (External clock), External Interrupt (with PCI), BOD TIMER(EC)

Table 12-1. Peripheral Operation during Power Down Mode

12.3 IDLE mode

The power control register is set to '01h' to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

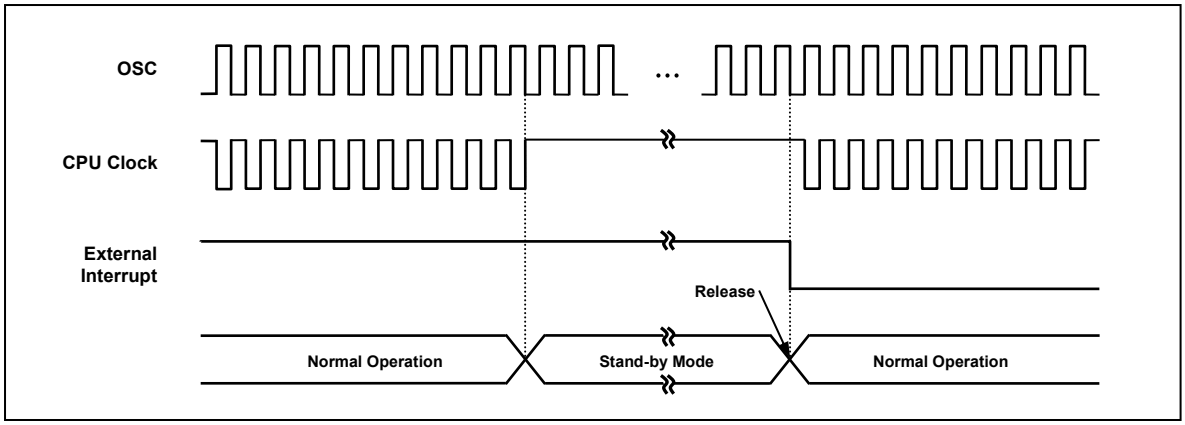


Figure 5.1 IDLE Mode Release Timing by External Interrupt

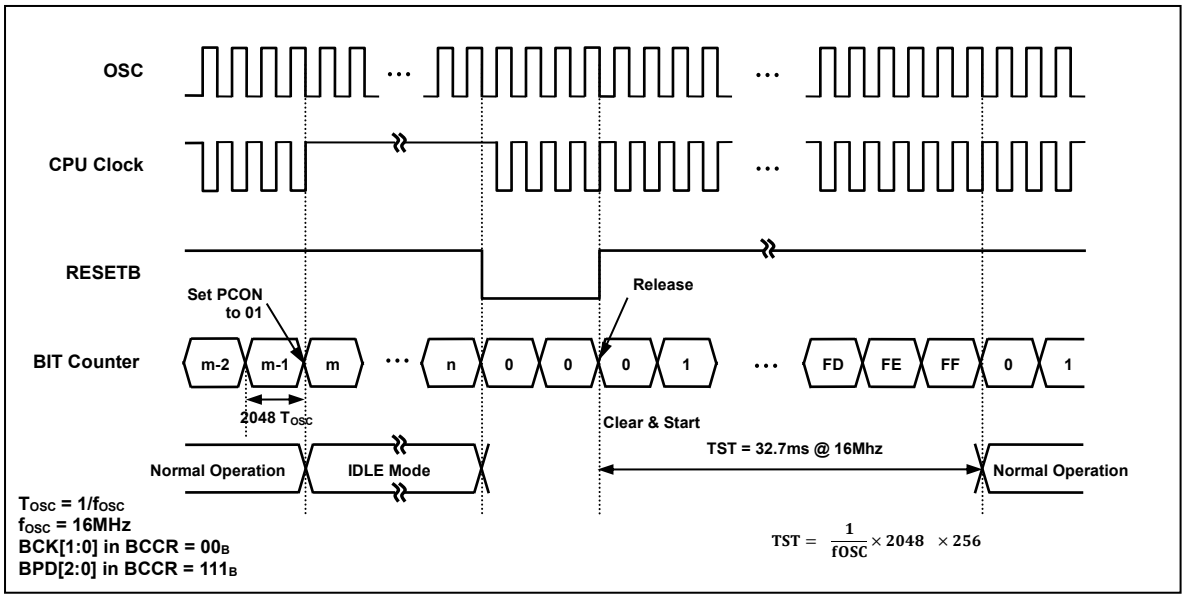


Figure 5.2 IDLE Mode Release Timing by /RESET

(Ex) MOV PCON, #0000_0001b ; setting of IDLE mode : set the bit of STOP and IDLE Control register (PCON)

12.4 STOP mode

The power control register is set to ‘03h’ to enter the STOP Mode. In the stop mode, the main oscillator, system clock and peripheral clock is stopped, but watch timer continue to operate if WDTRCON bit in SCCR register is written to '1'. With the clock frozen, all functions are stooped, but the on-chip RAM and control registers are held.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 5.3 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescaler divide ratio to have long enough time (more than 20msec). this guarantees that oscillator has started and stabilized.

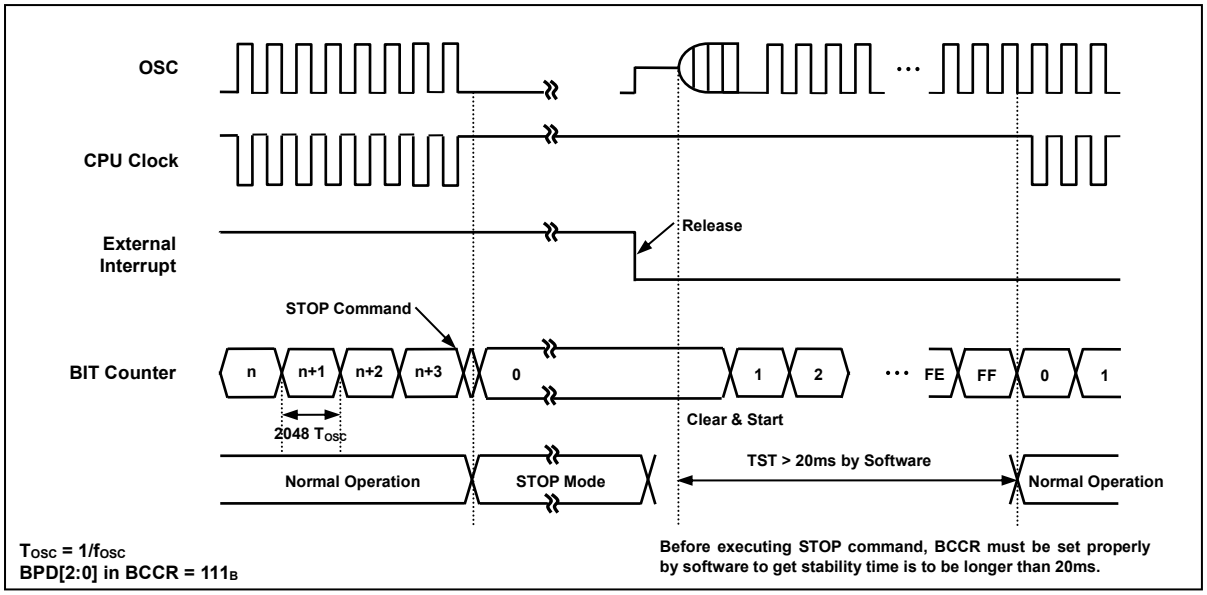


Figure 5.3 STOP Mode Release Timing by External Interrupt

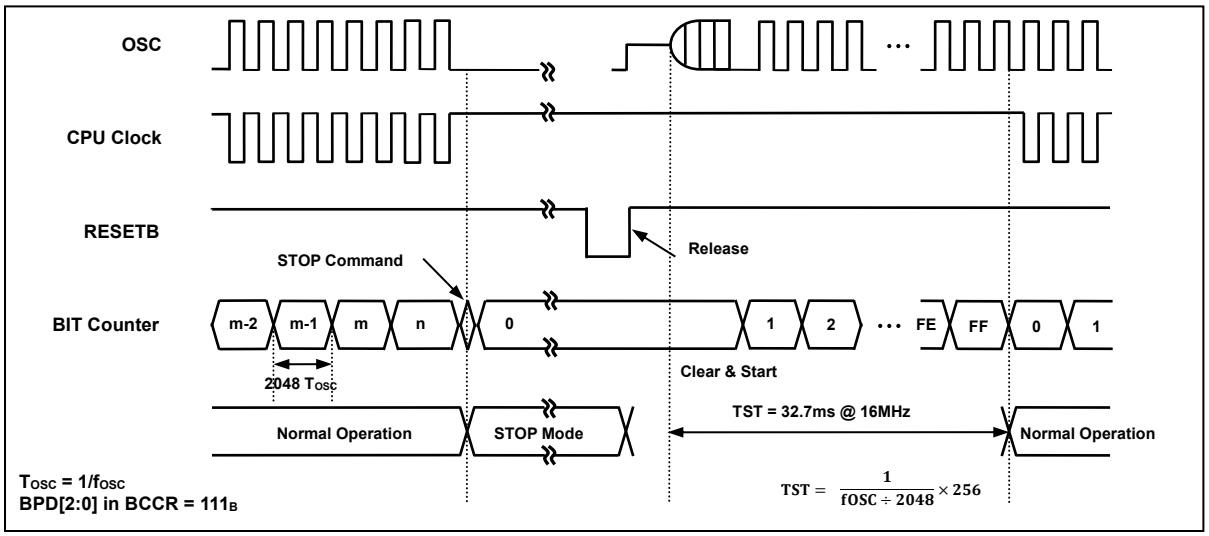


Figure 5.4 STOP Mode Release Timing by /RESET

12.5 Release Operation of STOP1, 2 Mode

After STOP1, 2 mode is released, the operation begins according to content of related interrupt register just before STOP1, 2 mode start. Interrupt Enable Flag of All (EA) of IE should be set to `1`. Released by only interrupt which each interrupt enable flag = `1`, and jump to the relevant interrupt service routine.

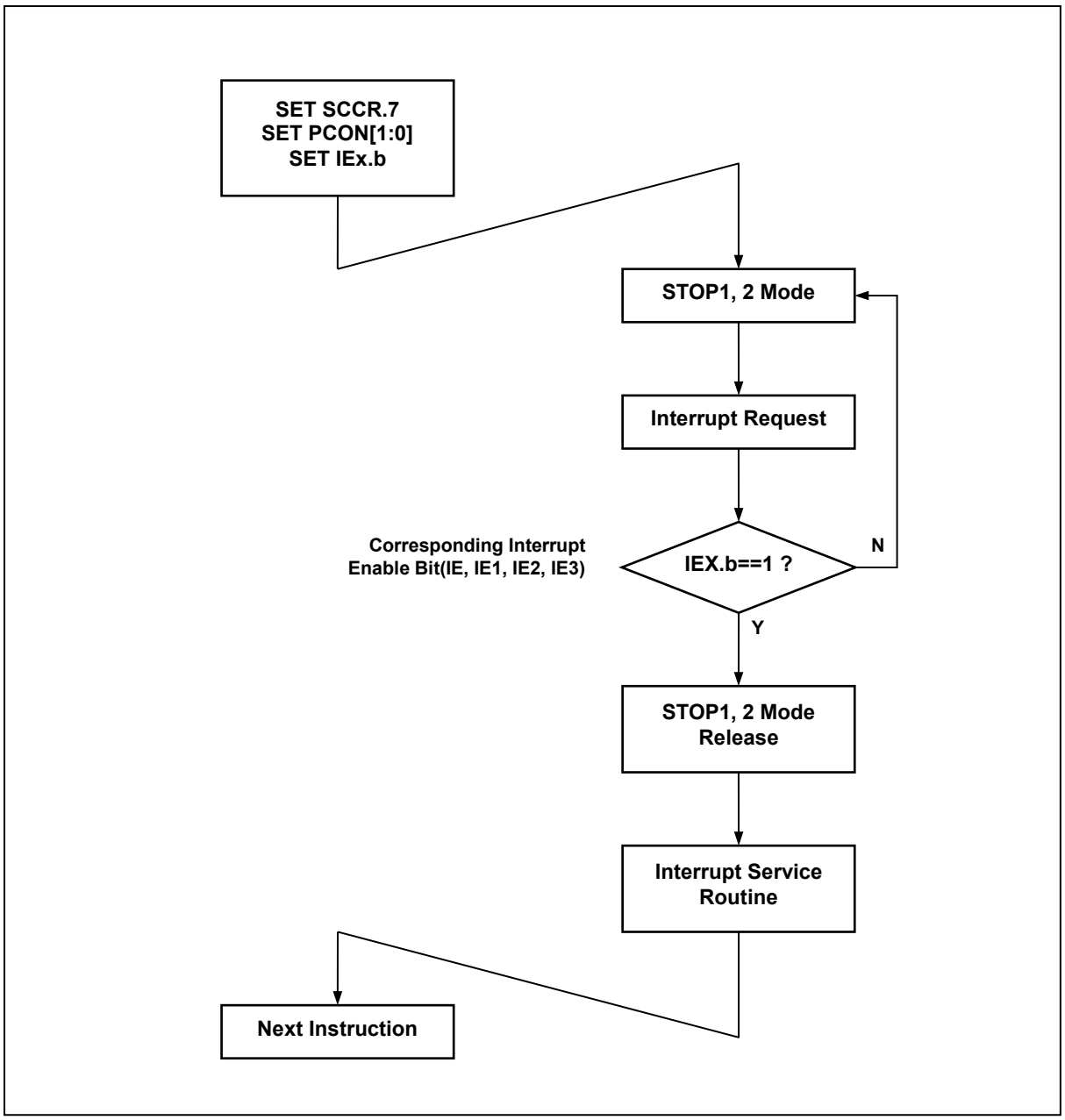


Figure 5.5 STOP1, 2 Mode Release Flow

12.6 Register Map

Name	Address	Dir	Default	Description
PCON	87H	R/W	00H	Power Control Register

Table 12-2. Register Map

12.7 Register description for Power Down Operation

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

01H IDLE mode enable

03H STOP1, 2 mode enable

NOTE)

1. To enter IDLE mode, PCON must be set to ‘01H’.
2. To STOP1,2 mode, PCON must be set to ‘03H’.
3. (In STOP1,2 mode, PCON register is cleared automatically by interrupt or reset)
4. When PCON is set to ‘03H’, if SCCR[7] is set to ‘1’, it enters the STOP1 mode. if SCCR[7] is cleared to ‘0’, it enters the STOP2 mode
5. The different thing in STOP 1,2 is only clock operation of internal 8kHz-OSC during STOP mode operating.

13 RESET

13.1 Overview

The MC97F6108A has reset by external RESETB pin. The following is the hardware setting value.

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Peripheral Registers refer
Brown-Out Detector	Enable

Table 13-1. Reset state

13.2 Reset source

The MC97F6108A has seven types of reset generation procedures. The following is the reset sources.

- External RESETB (In the case of RSTEN = '1')
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- BOD Reset (In the case of BODLS ≠ `000 `)
- LVR Reset
- OCD2 Reset

13.3 Block Diagram

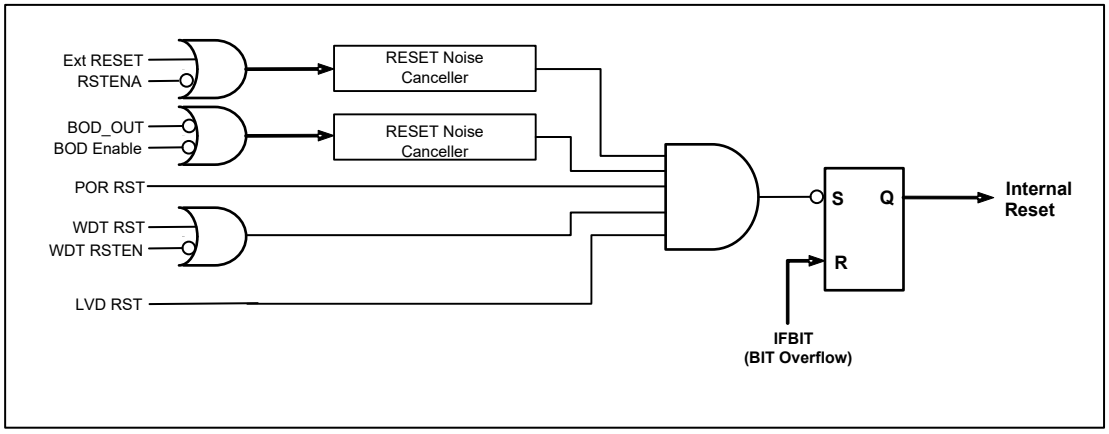


Figure 6.1 RESET Block Diagram

13.4 RESET Noise Canceller

The Figure 6.2 is the Noise canceller diagram for Noise cancel of RESET. It has the Noise cancel value of about 8us (@V_{DD}=5V) to the low input of System Reset.

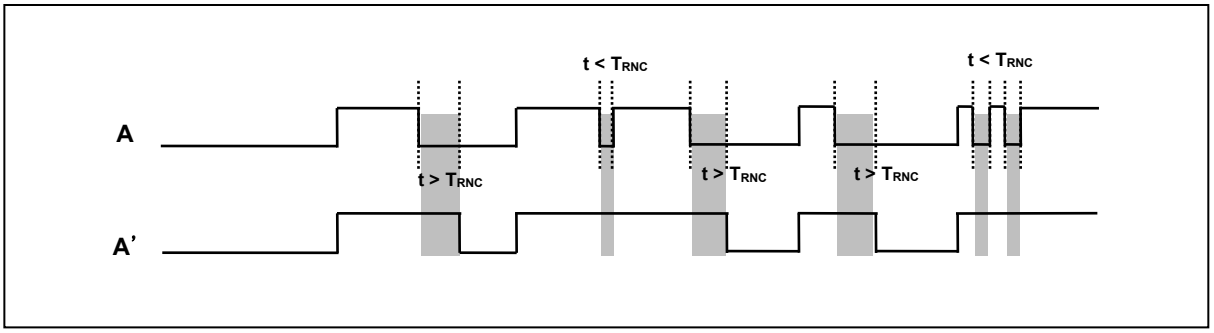


Figure 6.2 Reset noise canceller time diagram

13.5 Power ON RESET

When rising device power, the POR (Power ON Reset) have a function to reset the device. If using POR, it executes the device RESET function instead of the RESET IC or the RESET circuits. And External RESET PIN is able to use as Normal input pin.

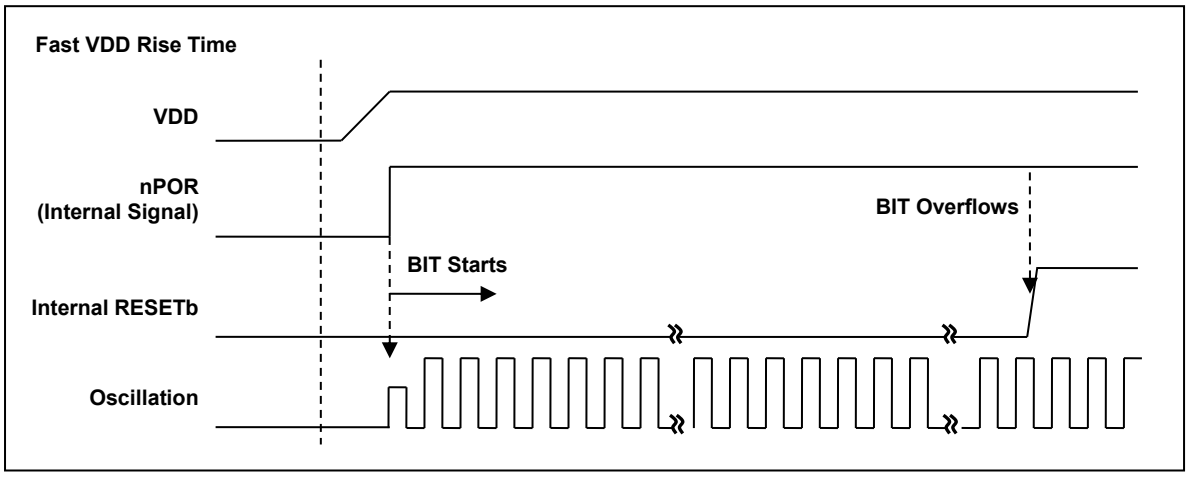


Figure 6.3 Fast VDD rising time

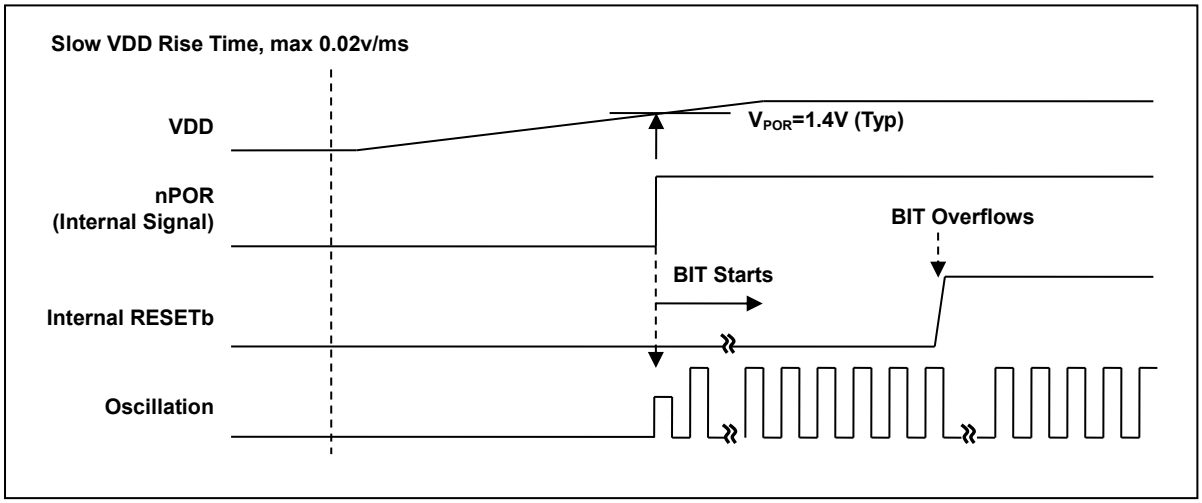


Figure 6.4 Internal RESET Release Timing On Power-Up

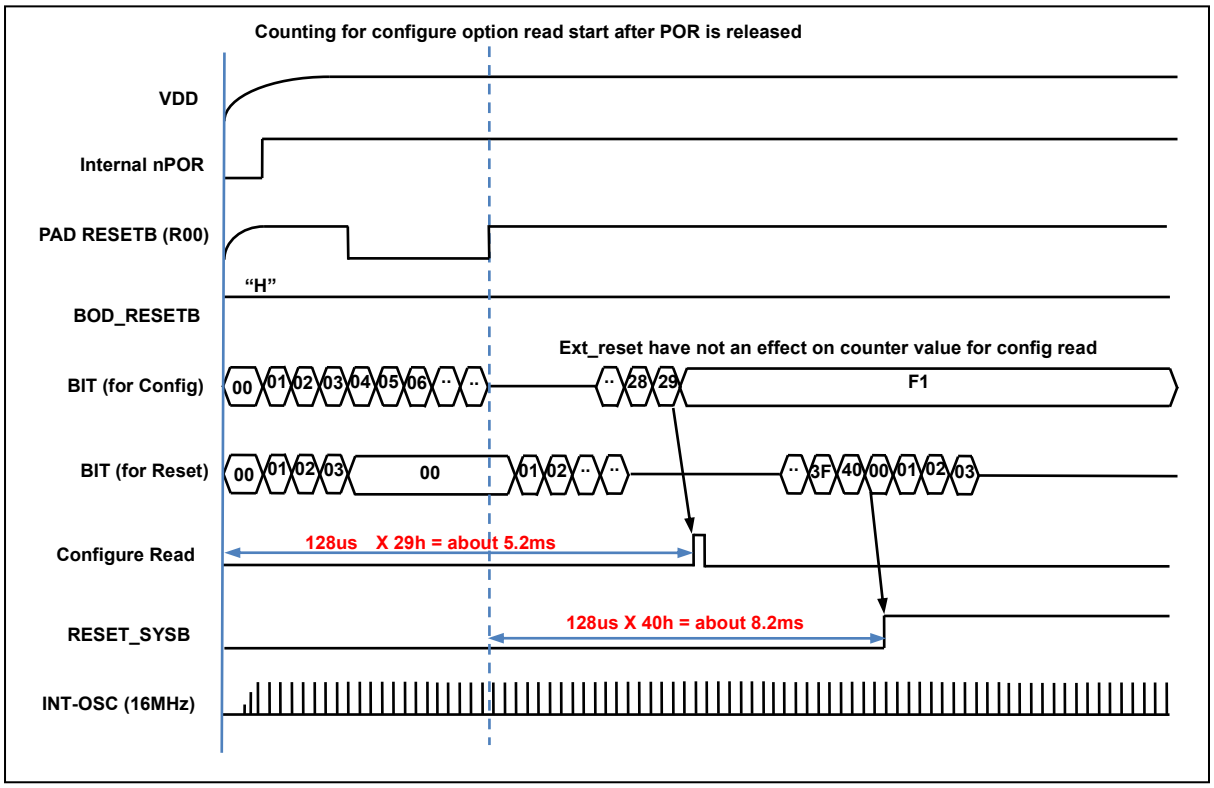


Figure 6.5 Configuration timing when Power-on

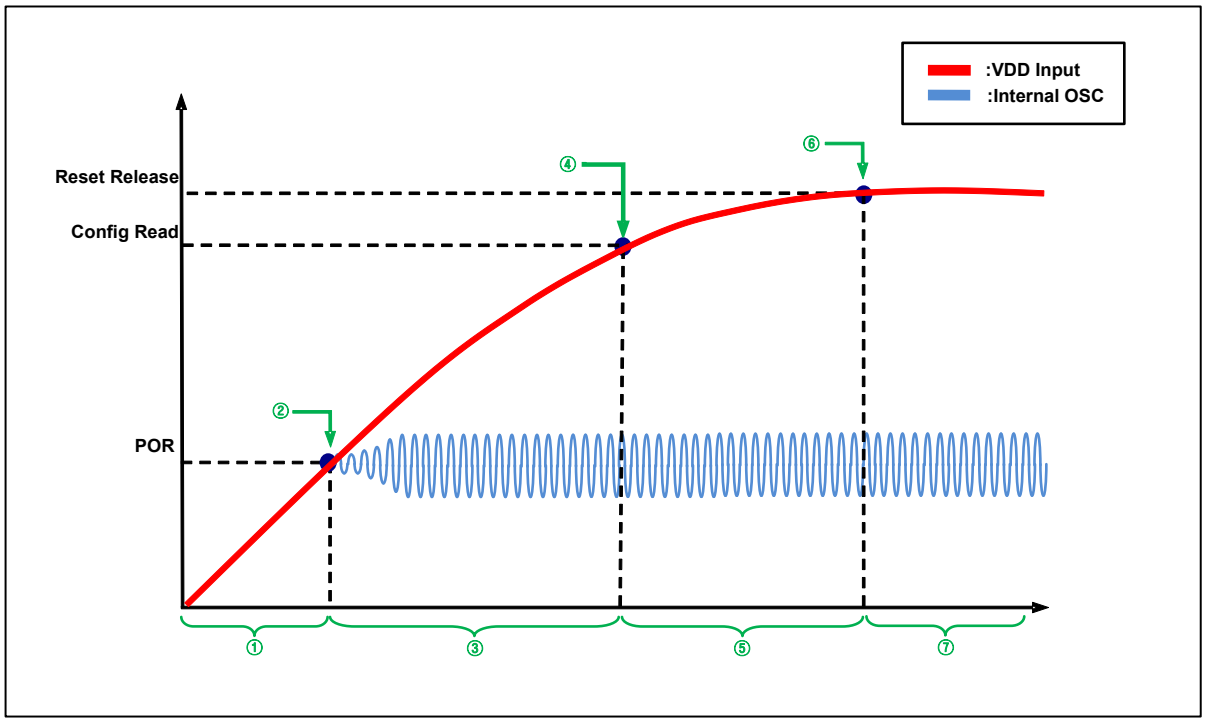


Figure 6.6 Boot Process Wave Form

Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection -Internal OSC (16MHz) ON	-about 1.2V ~ 1.6V
③	- (INT-OSC16MHz/2)×29h Delay section (=5ms) -VDD input voltage must rise over than flash operating voltage for Configure option read	-Slew Rate >= 0.025V/ms
④	- Configure option read point	-about 1.5V ~ 1.6V -Configure option value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-8ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after8ms, after External Reset Release (External reset) ii) 8ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

Table 13-2. Boot Process Description

13.6 External RESETB Input

The External RESETB is the input to a Schmitt trigger. A reset in accomplished by holding the reset pin low for at least 8us over, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset state becomes '1', it needs the stabilization time with 8ms and after the stable state, the internal RESET becomes '1'. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

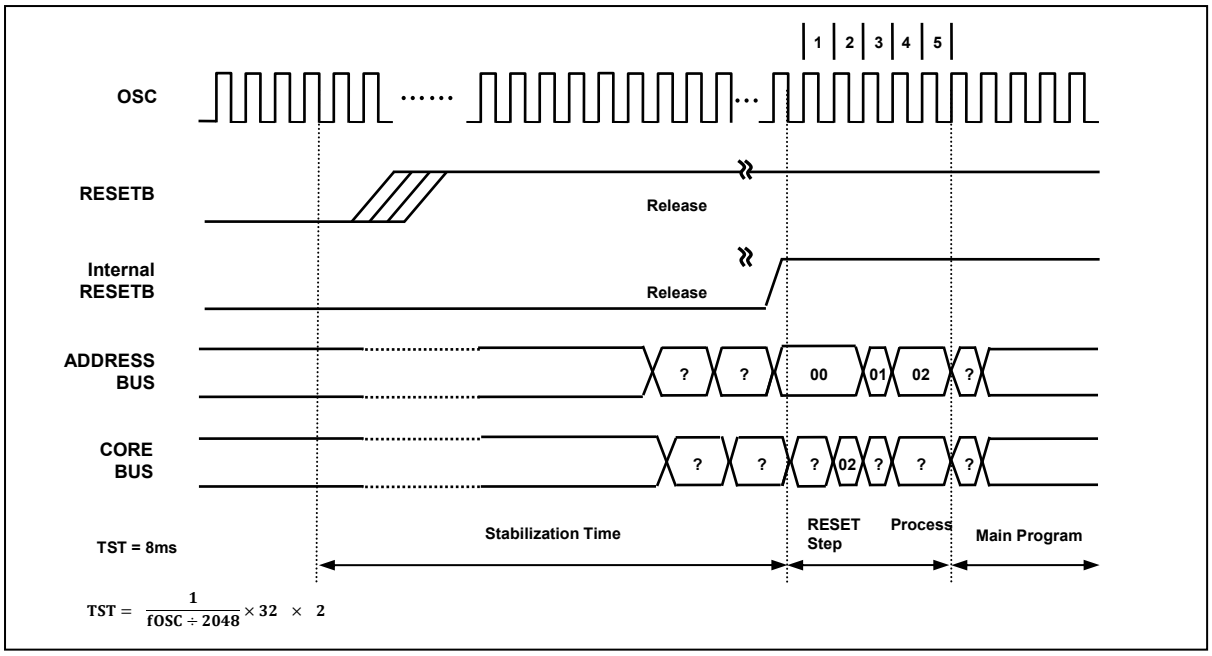


Figure 6.7 Timing Diagram after RESET

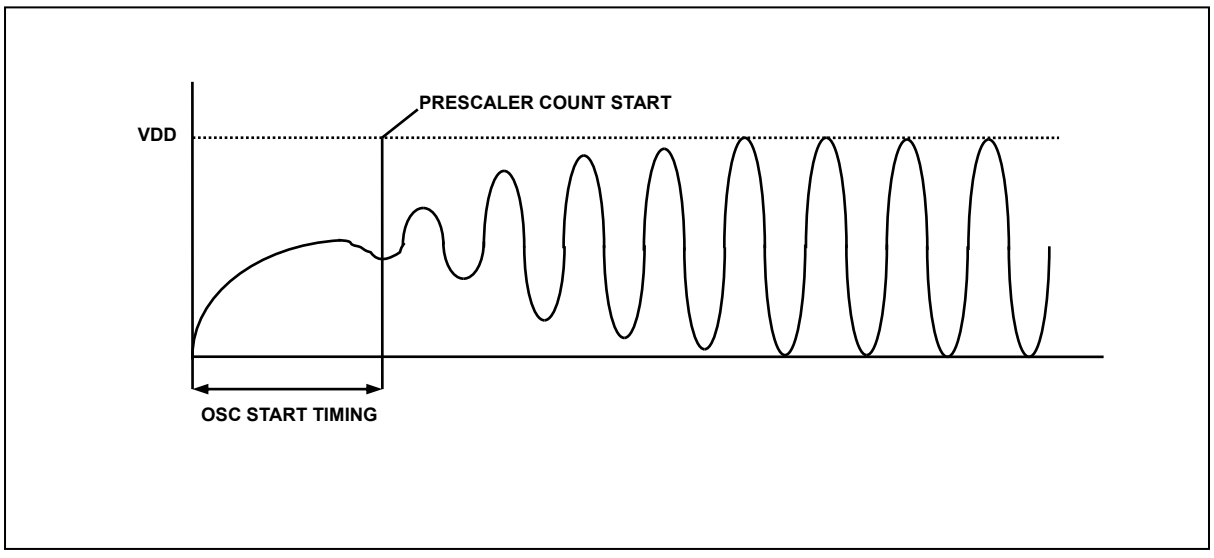


Figure 6.8 Oscillator generating waveform example

Note)

1. as shown Figure 6.8, the stable generating time is not included in the start-up time.

13.7 Brown Out Detector Processor

The MC97F6108A has an On-chip Brown-out detection circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by BODLS[2:0]. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, it is nessecerry to disable BOD by select BODLS[2:0]=000b to off by software.

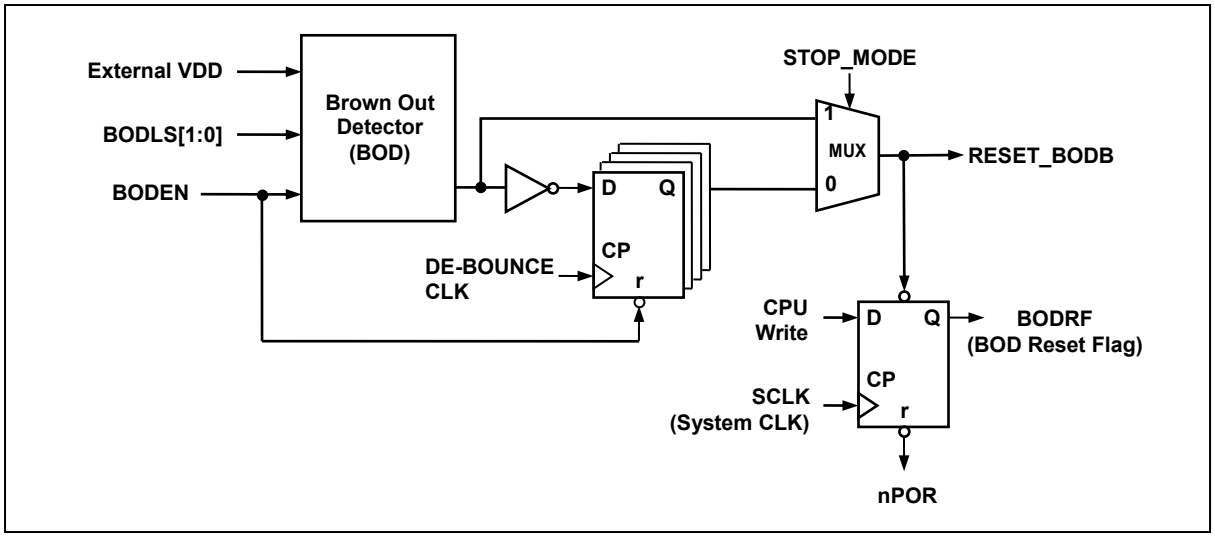


Figure 6.9 Block Diagram of BOD

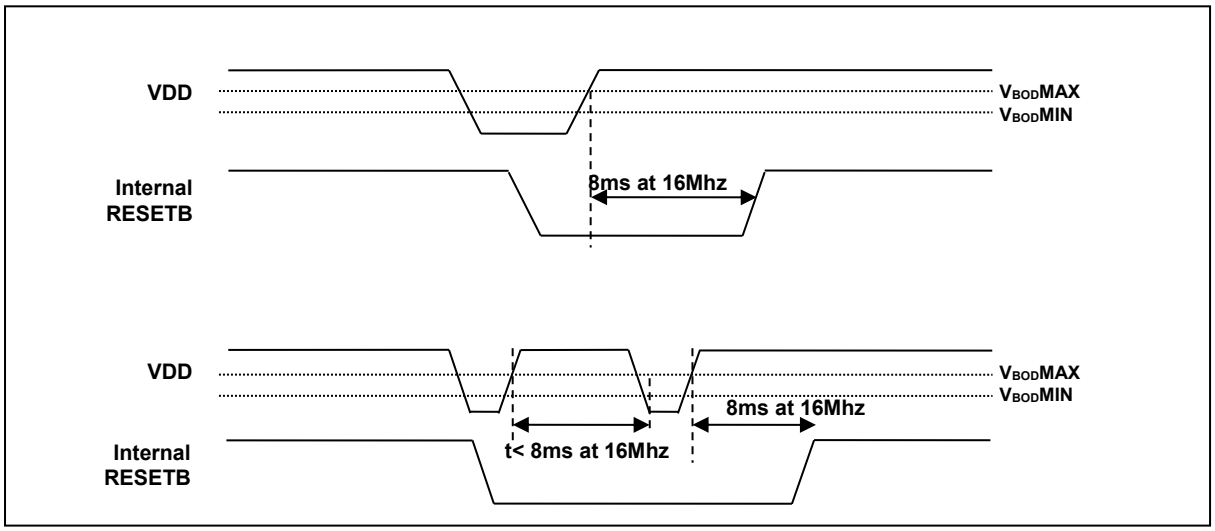


Figure 6.10 Internal Reset at the power fail situation

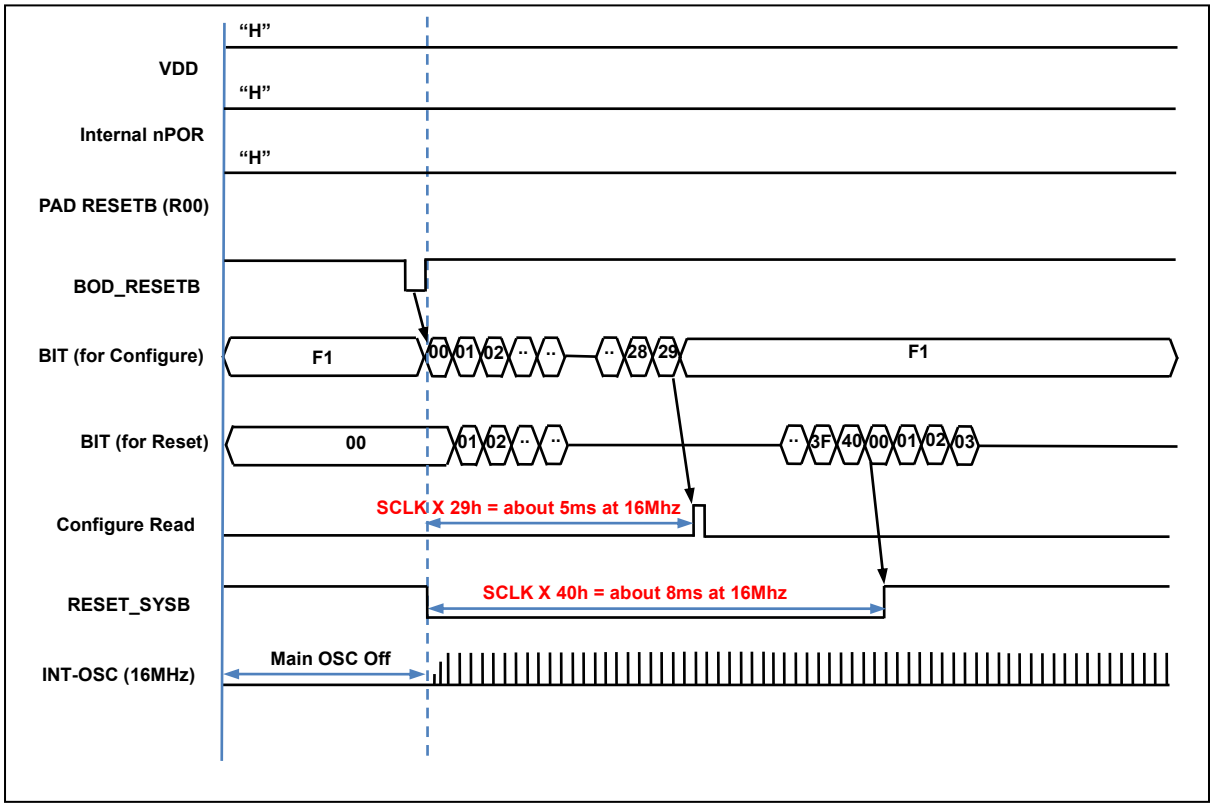


Figure 6.11 Configuration timing when BOD RESET

13.8 Register Map

Name	Address	Dir	Default	Description
RSFR	86H	R/W	88H	Reset Source Flag register
BODR	8FH	R/W	01H	BOD Control register

Figure 6.12 Register Map

13.9 Register description for Reset Operation

RSFR (Reset Source Flag register) : 86H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	BODRF	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-

Initial value : 88H

PORF Power-On Reset flag bit. The bit is reset by writing '0' to this bit.

0 No detection

1 Detection

EXTRF External Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.

0 No detection

1 Detection

WDTRF Watch Dog Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.

0 No detection

1 Detection

OCDRF On-Chip Debug2 Reset flag bit. The bit is reset by writing '0' to this bit or by Power ON reset.

0 No detection

1 Detection

BODRF Brown-Out Reset & Interrupt flag bit. The bit is reset by writing '0' to this bit or by Power ON reset or by BOD ack signal

0 No detection

1 Detection

BODR (BOD Control Register) : 8FH

7	6	5	4	3	2	1	0
-	BODINTON	-	-	ENBODST	BODLS2	BODLS1	BODLS0
-	R/W	-	-	R/W	R/W	R/W	R/W

Initial value : 01H

BODINTON Select BOD reset or Interrupt

0 Reset

1 Interrupt

ENBODST Select STOP mode BOD enable or disable

0 disable

1 enable

BODLS[2:0] BOD level Voltage

BODLS2	BODLS1	BODLS0	Description
0	0	0	BOD disable
0	0	1	2.2V (default)
0	1	0	2.5V
0	1	1	2.7V
1	0	0	3.2V
1	0	1	3.7V
1	1	0	4.2V
1	1	1	reserved

14 On-chip Debug System

14.1 Overview

14.1.1 Description

On-chip debug System (OCD2) of MC97F6108A can be used for programming the non-volatile memories and on-chip debugging. Detailed descriptions for programming via the OCD2 interface can be found in the following chapter.

Figure 7.1 shows a block diagram of the OCD2 interface and the On-chip Debug system.

14.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash Memory
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, Fuses, and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by OCD2 Dongle
- Operating frequency
  - Supports the maximum frequency of the target MCU

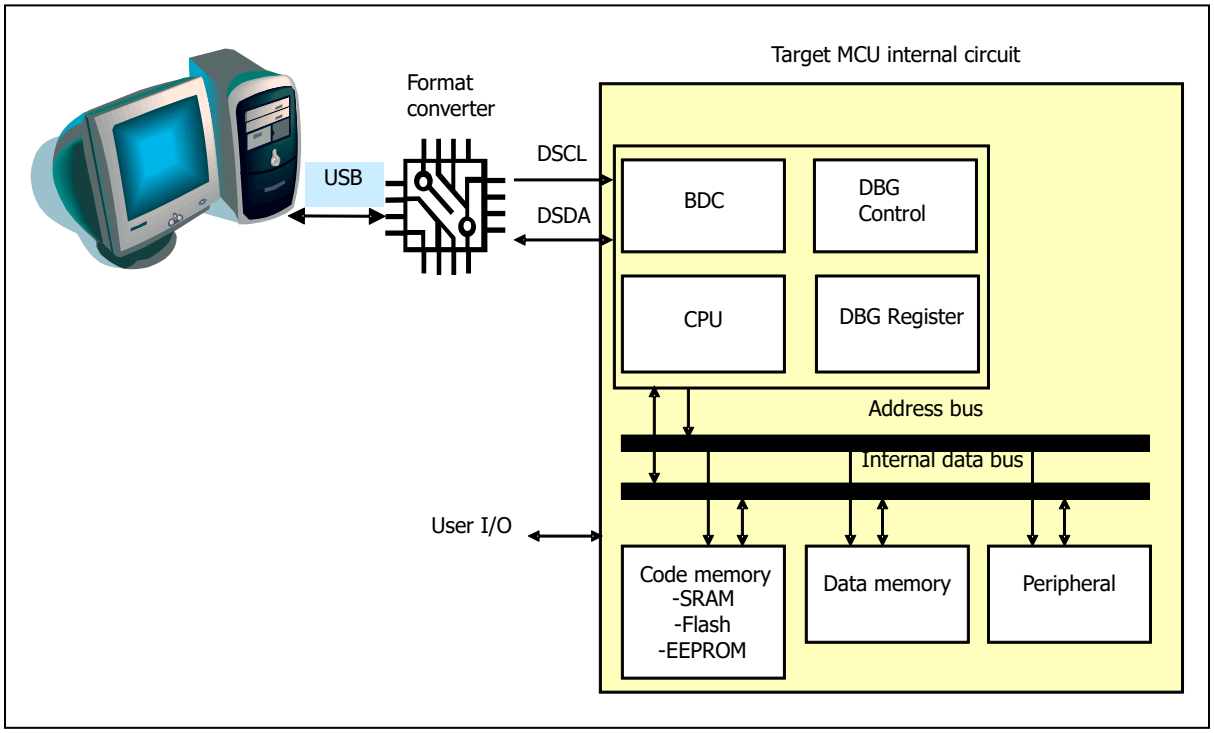


Figure 7.1 Block Diagram of On-chip Debug System

14.2 Two-pin external interface

14.2.1 Basic transmission packet

- 10-bit packet transmission using two-pin interface.
- packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start condition and stop condition notify the start and the stop of background debugger command respectively.

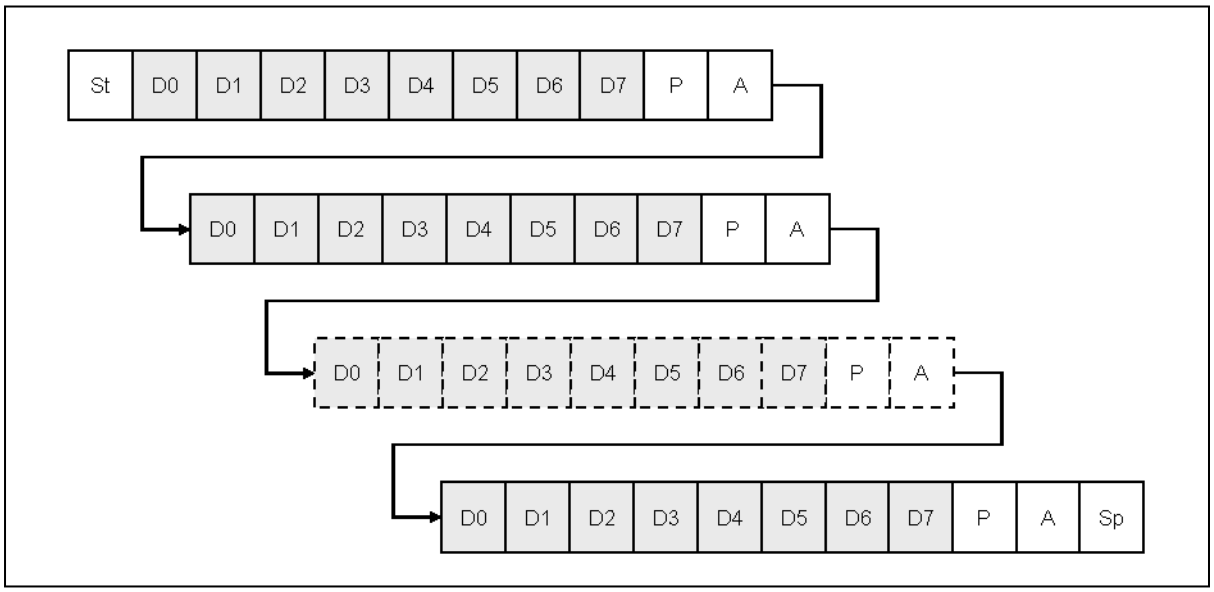


Figure 7.2 10-bit transmission packet

14.2.2 Packet transmission timing

14.2.2.1 Data transfer

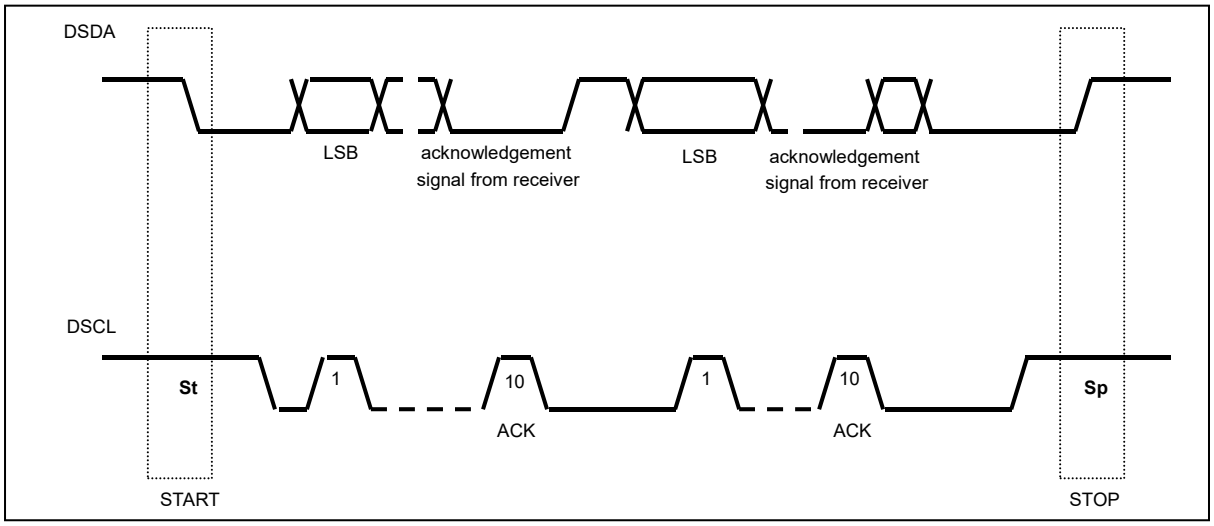


Figure 7.3 Data transfer on the twin bus

14.2.2.2 Bit transfer

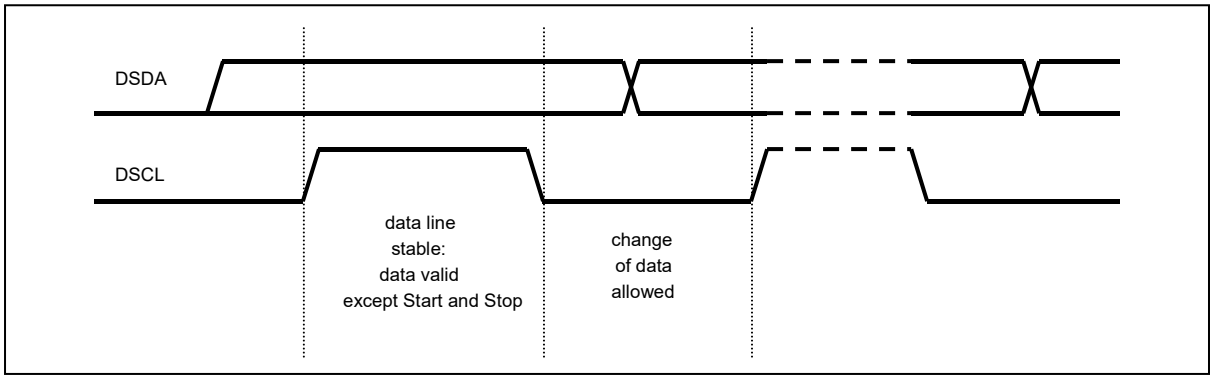


Figure 7.4 Bit transfer on the serial bus

14.2.2.3 Start and stop condition

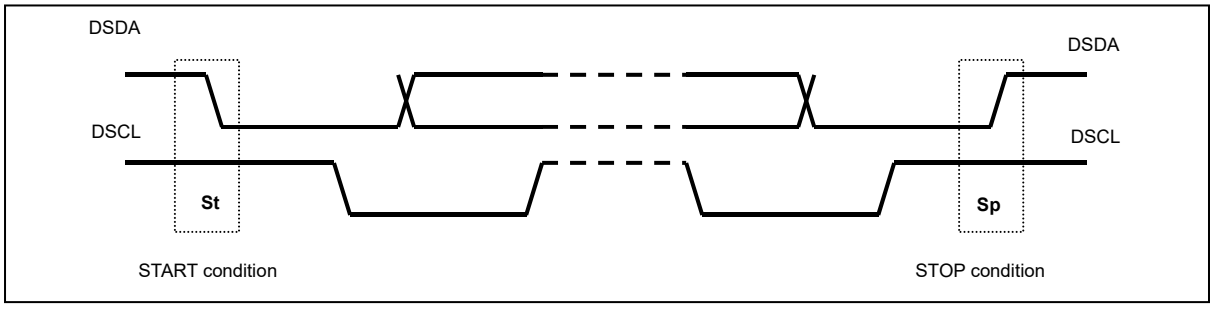


Figure 7.5 Start and stop condition

14.2.2.4 Acknowledge bit

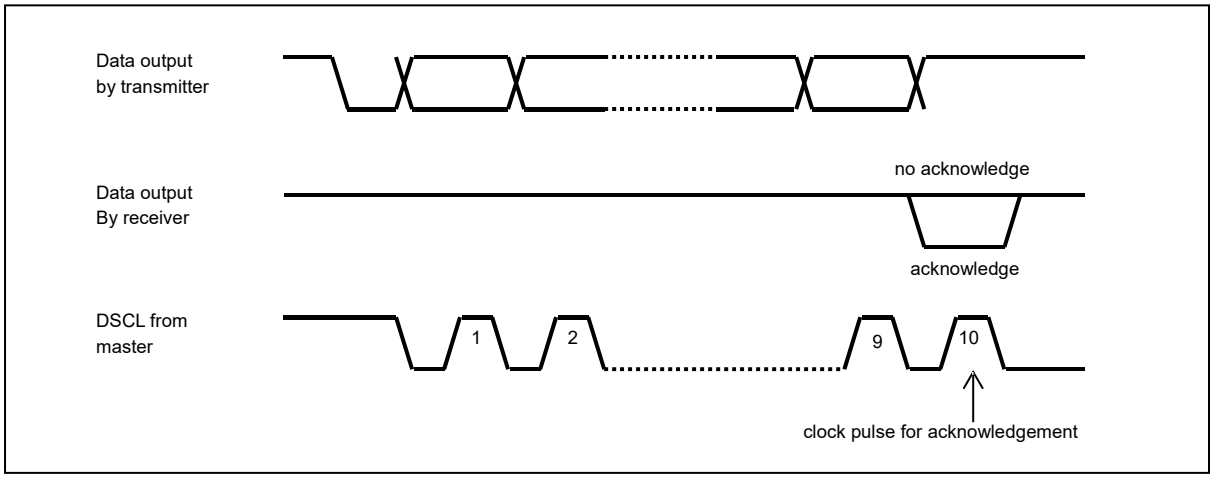


Figure 7.6 Acknowledge on the serial bus

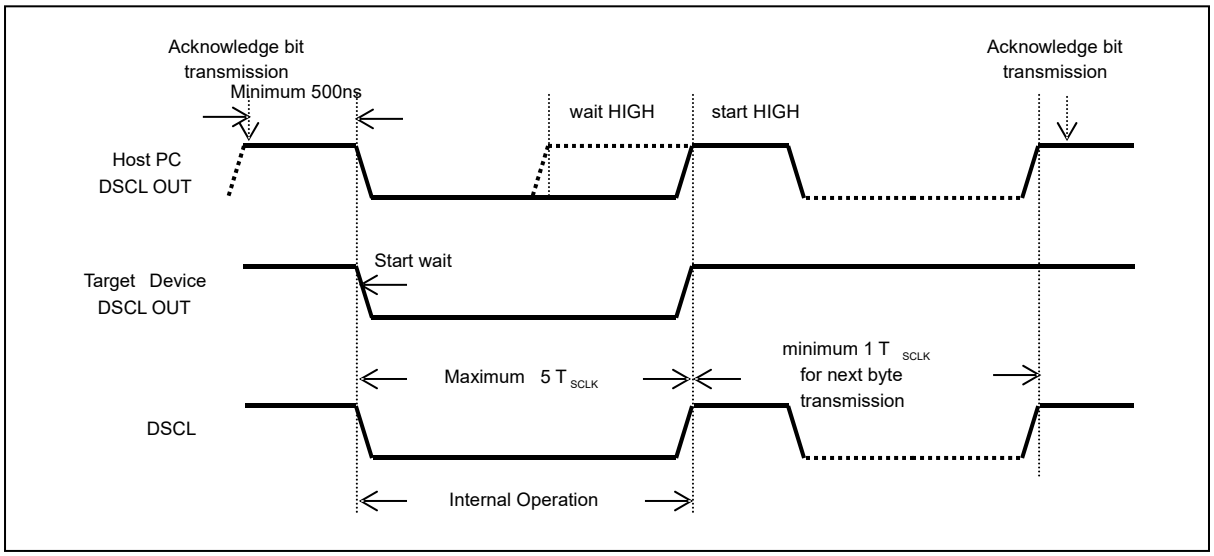


Figure 7.7 Clock synchronization during wait procedure

14.2.3 Connection of transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

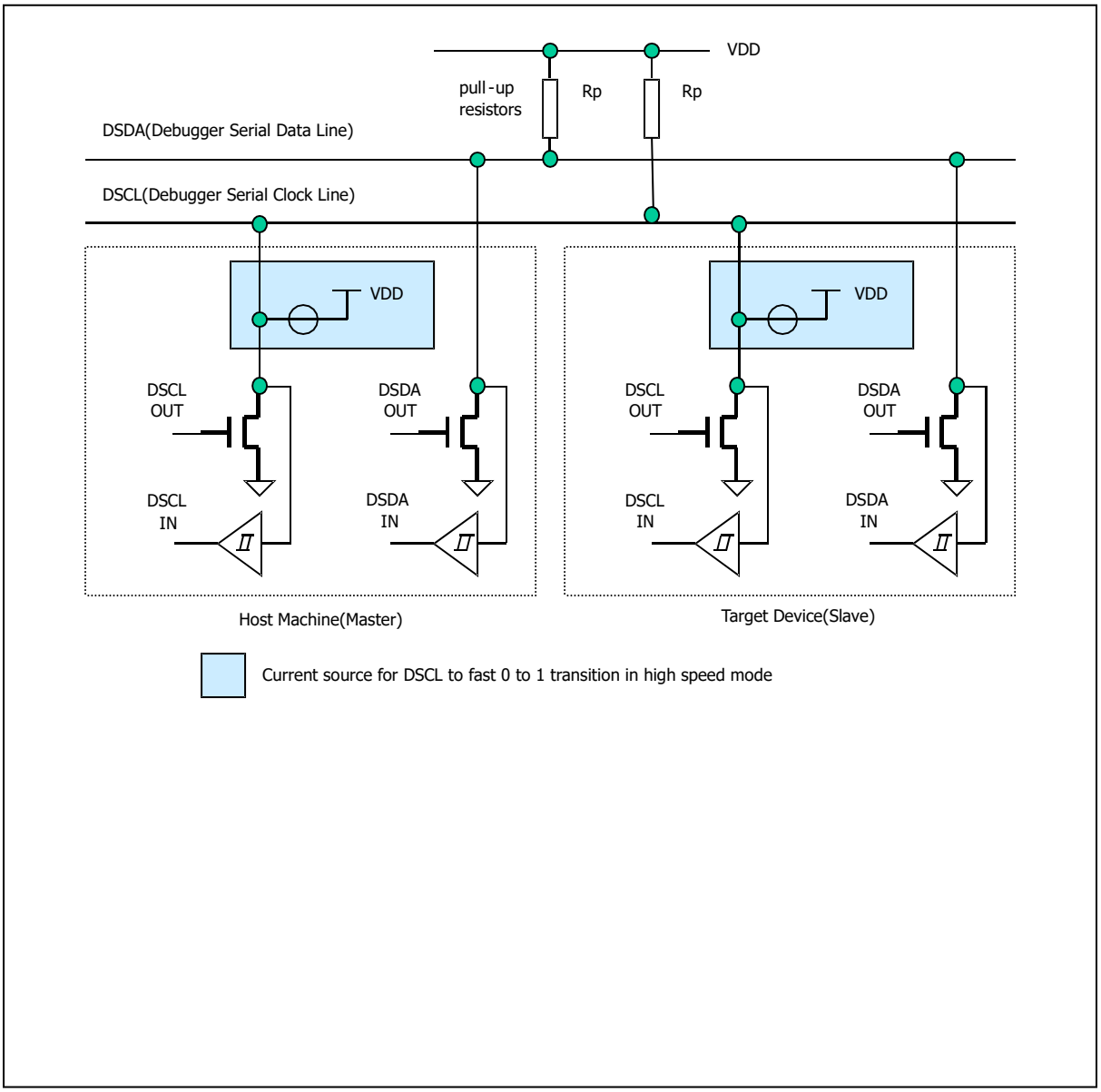


Figure 7.8 Connection of transmission

15 Memory Programming

15.1 Overview

15.1.1 Description

MC97F6108A has flash memory to which a program can be written, erased, and overwritten while mounted on the board.

Serial ISP modes and byte-parallel ROM writer mode are supported.

15.1.2 Features

- Flash Size : 8Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000 program/erase cycles at typical voltage and temperature for flash memory
- Security feature

15.2 Flash Control and status register

Registers to control Flash are Mode Register (FEMR), Control Register (FECR), Status Register (FESR), Time Control Register (FETCR), Address Low Register (FEARL), Address Middle Register (FEARM), address High Register (FEARH) and Data Register (FEDR). They are mapped to SFR area and can be accessed only in programming mode.

15.2.1 Register Map

Name	Address	Dir	Default	Description
FEMR	EAH	R/W	00H	Flash Mode Register
FECR	EBH	R/W	03H	Flash Control Register
FESR	ECH	R/W	80H	Flash Status Register
FETCR	EDH	R/W	00H	Flash Time Control Register
FEARL	F2H	R/W	00H	Flash Address Low Register
FEARM	F3H	R/W	00H	Flash Address Middle Register
FEARH	F4H	R/W	00H	Flash Address High Register
FEDR	F5H	R/W	00H	Flash Data Register

Table 15-1. Register Map

15.2.2 Register Description for Flash

FEMR (Flash Mode Register) : EAH

7	6	5	4	3	2	1	0
FSEL	-	PGM	ERASE	PBUFF	OTPE	VFY	FEEN
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSEL Select flash memory.
- 0 Deselect flash memory
- 1 Select flash memory

PGM Enable program or program verify mode with VFY
- 0 Disable program or program verify mode
- 1 Enable program or program verify mode

ERASE Enable erase or erase verify mode with VFY
- 0 Disable erase or erase verify mode
- 1 Enable erase or erase verify mode

PBUFF Select page buffer
- 0 Deselect page buffer
- 1 Select page buffer

OTPE Select OTP area instead of program memory
- 0 Deselect OTP area
- 1 Select OTP area

VFY Set program or erase verify mode with PGM or ERASE

Program Verify: PGM=1, VFY=1

Erase Verify: ERASE=1, VFY=1

FEEN Enable program and erase of Flash. When inactive, it is possible to read as normal mode
- 0 Disable program and erase
- 1 Enable program and erase

FECR (Flash Control Register) : EBH

7	6	5	4	3	2	1	0
AEF	-	EXIT1	EXIT0	WRITE	READ	nFERST	nPBRST
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

AEF Enable flash bulk erase mode
- 0 Disable bulk erase mode of Flash memory
- 1 Enable bulk erase mode of Flash memory

EXIT[1:0] Exit from program mode. It is cleared automatically after 1 clock

EXIT1	EXIT0	Description
0	0	Don't exit from program mode
0	1	Don't exit from program mode
1	0	Don't exit from program mode
1	1	Exit from program mode

WRITE Start to program or erase of Flash. It is cleared automatically after 1 clock
- 0 No operation
- 1 Start to program or erase of Flash

READ Start auto-verify of Flash. It is cleared automatically after 1 clock
- 0 No operation
- 1 Start auto-verify of Flash

nFERST Reset Flash control logic. It is cleared automatically after 1 clock
- 0 No operation
- 1 Reset Flash control logic.

nPBRST Reset page buffer with PBUFF. It is cleared automatically after 1 clock

PBUFF	nPBRST	Description
0	0	Page buffer reset
1	0	Write checksum reset

WRITE and READ bits can be used in program, erase and verify mode with FEAR registers. Read or writes for memory cell or page buffer uses read and write enable signals from memory controller. Indirect address mode with FEAR is only allowed to program, erase and verify

FESR (Flash Status Register) : ECH

7	6	5	4	3	2	1	0
PEVBSY	VFYGOOD	-	-	ROMINT	WMODE	EMODE	VMODE
R	R/W	R	R	R/W	R	R	R

Initial value : 80H

PEVBSY Operation status flag. It is cleared automatically when operation starts. Operations are program, erase or verification
- 0 Busy (Operation processing)
- 1 Complete Operation

VFYGOOD Auto-verification result flag.
- 0 Auto-verification fails
- 1 Auto-verification successes

ROMINT Flash interrupt request flag. Auto-cleared when program/erase/verify starts. Active in program/erase/verify completion
- 0 No interrupt request.
- 1 Interrupt request.

WMODE Write mode flag

EMODE Erase mode flag

VMODE Verify mode flag

FEARL (Flash address low Register) : F2H

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
W	W	W	W	W	W	W	W

Initial value : 00H

ARL[7:0] Flash address low

FEARM (Flash address middle Register) : F3H

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
W	W	W	W	W	W	W	W

Initial value : 00H

ARM[7:0] Flash address middle

FEARH (Flash address high Register) : F4H

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
W	W	W	W	W	W	W	W

Initial value : 00H

ARH[7:0] Flash address high

FEAR registers are used for program, erase and auto-verify. In program and erase mode, it is page address and ignored the same least significant bits as the number of bits of page address. In auto-verify mode, address increases automatically by one.

FEARs are write-only register. Reading these registers returns 24-bit checksum result

FEDR (Flash control Register) : F5H

7	6	5	4	3	2	1	0
FEDR7	FEDR6	FEDR5	FEDR4	FEDR3	FEDR2	FEDR1	FEDR0
W	W	W	W	W	W	W	W

Initial value : 00H

FEDR[7:0] Flash and EEPROM data

Data register. In no program/erase/verify mode, READ/WRITE of FECR read or write data from FLASH to this register or from this register to Flash.

The mode entrance sequence is to write 0xA5 and 0x5A to it in order.

FETCR (Flash Time control Register) : EDH

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

TCR[7:0] Flash Time control

Program and erase time is controlled by setting FETCR register. Program and erase timer uses 10-bit counter. It increases by one at each divided system clock frequency(=SCLK/128). It is cleared when program or erase starts. Timer stops when 10-bit counter is same to FETCR. PEVBSY is cleared when program, erase or verify starts and set when program, erase or verify stops.

Max program/erase time at 16Mhz system clock : (255+1) * 2 * (62.5ns * 128) = 4.096ms

In the case of 10% of error rate of counter source clock, program or erase time is 3.6~4.5ms

* * Program/erase time calculation

for page write or erase, Tpe = (TCON+1) * 2 * (SCLK * 128)

for bulk erase, Tbe = (TCON+1) * 4 * (SCLK * 128)

	Min	Typ	Max	Unit
program/erase Time	2.4	2.5	2.6	ms

Table 15-2. Program/erase Time

※ Recommended program/erase time at 16MHz (FETCR = 9Dh)

15.3 Memory map

15.3.1 Flash Memory Map

Program memory uses 8-Kbyte of Flash memory. It is read by byte and written by byte or page. One page is 32-byte

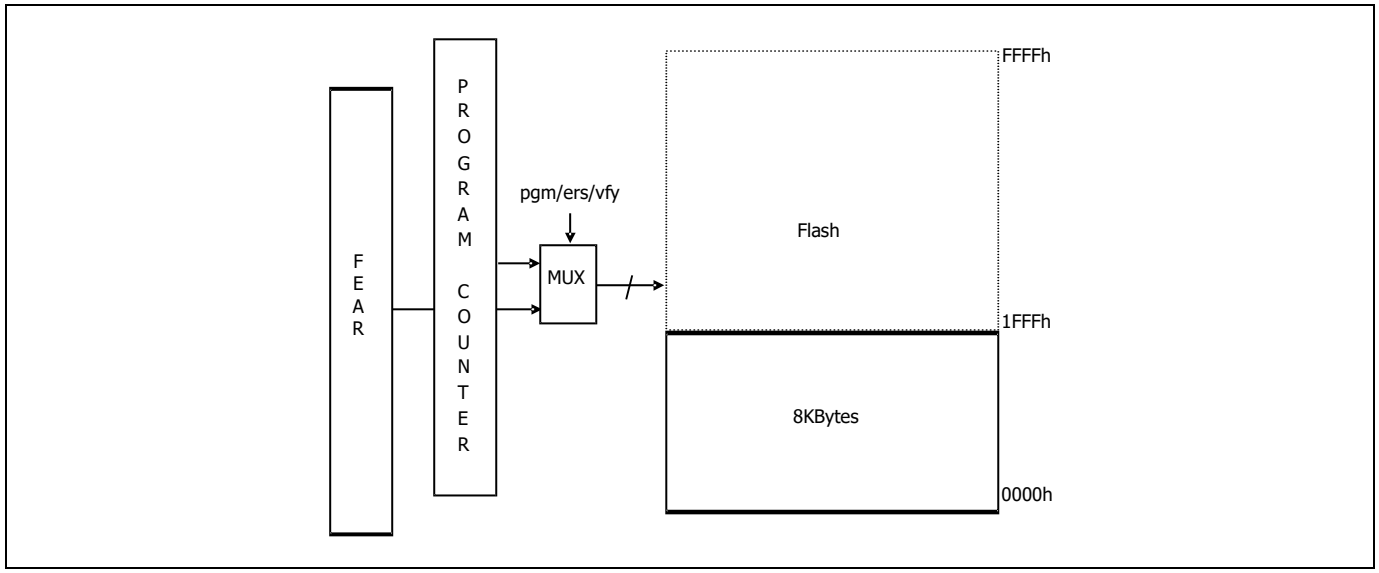


Figure 8.1 Flash Memory Map

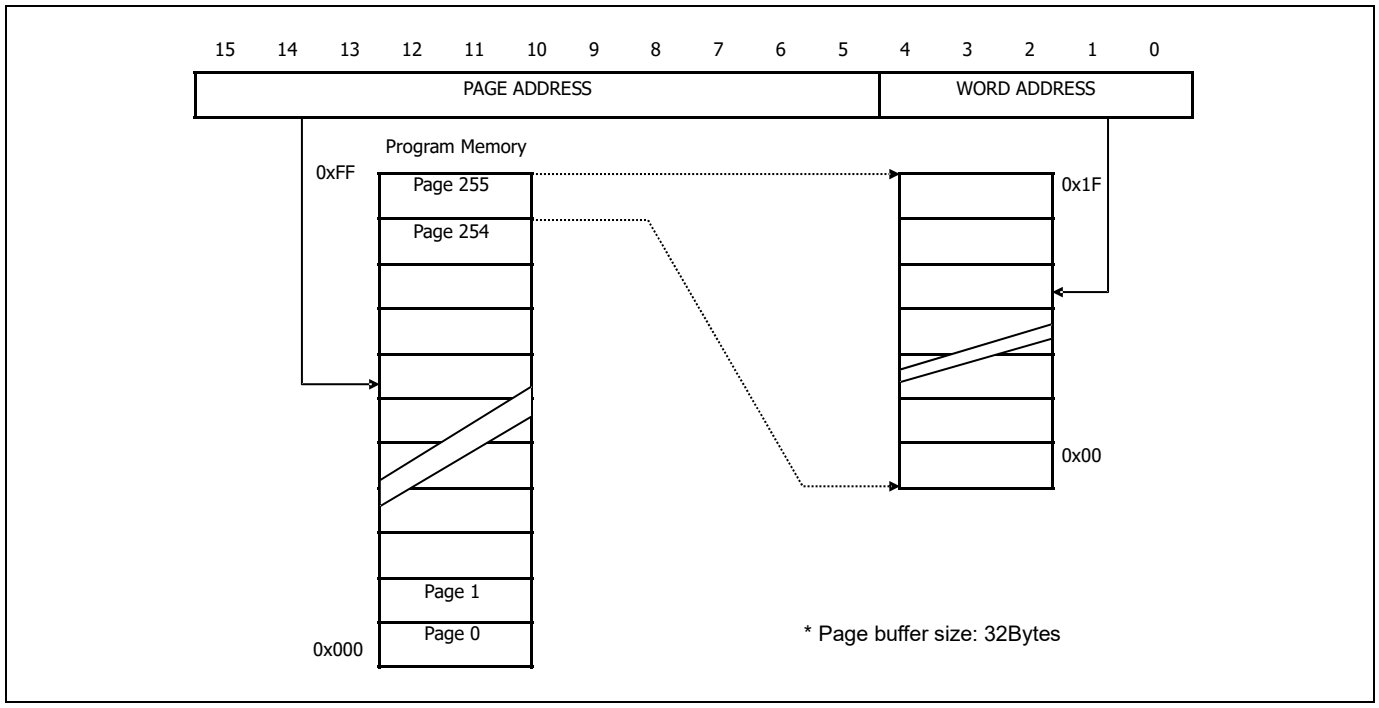


Figure 8.2 Address configuration of Flash memory

15.4 Serial In-System Program Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger.

15.4.1 Flash operation

Configuration (This Configuration is just used for follow description)

7	6	5	4	3	2	1	0
-	FEMR[4] &[1]	FEMR[5] &[1]	-	-	FEMR[2]	FECR[6]	FECR[7]
-	ERASE&VFY	PGM&VFY	-	-	OTPE	AEE	AEF

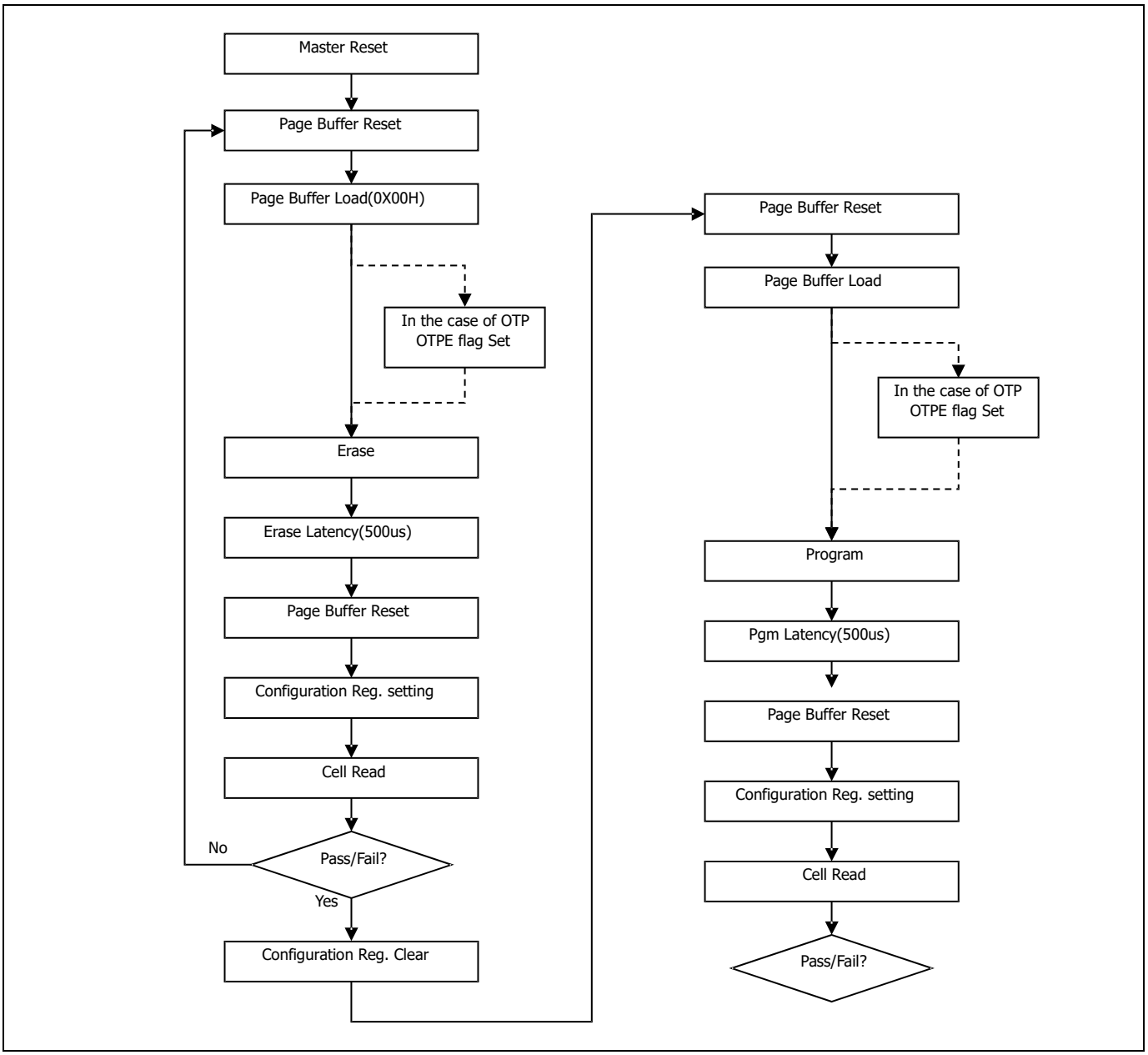


Figure 8.3 The sequence of page program and erase of Flash memory

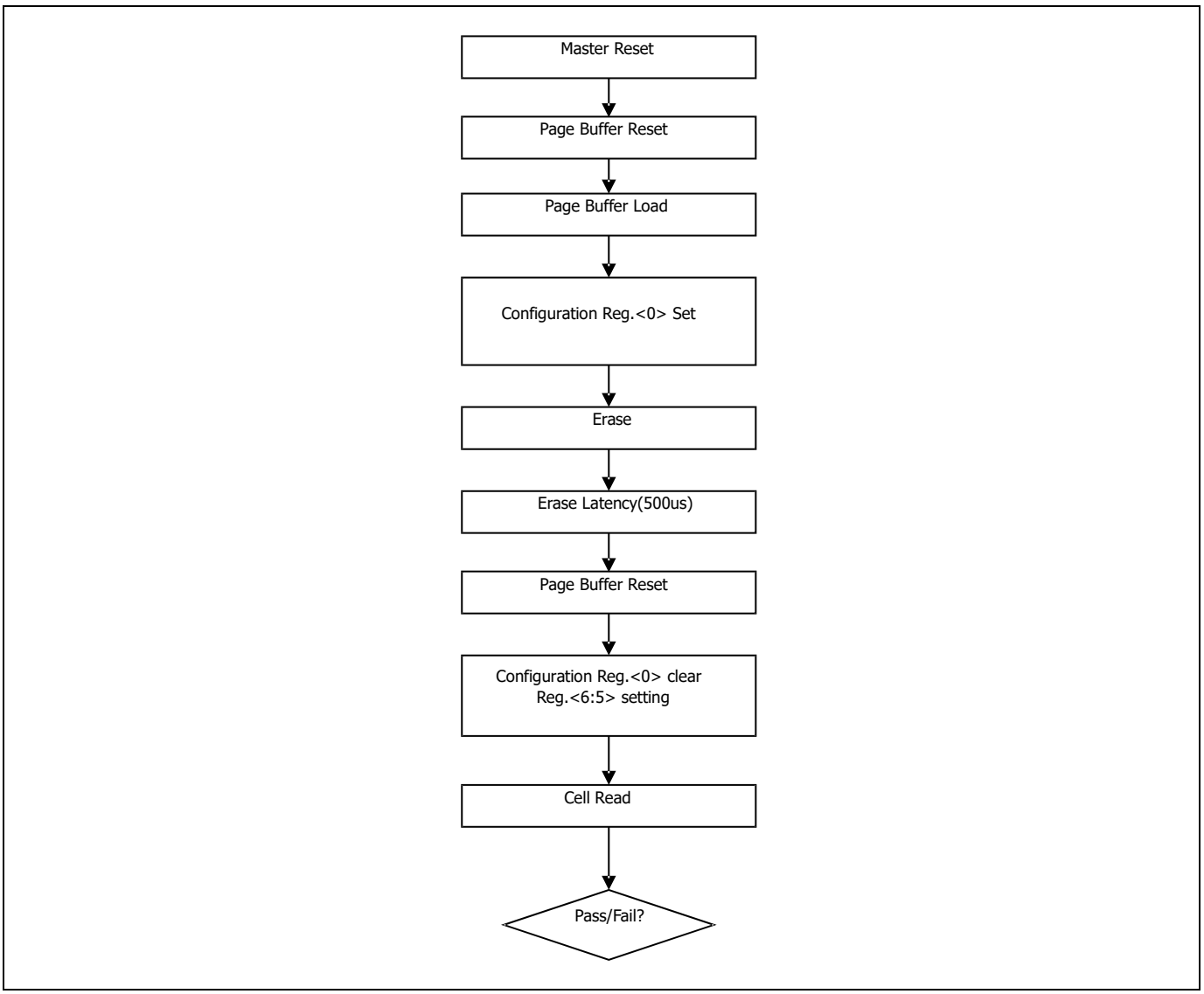


Figure 8.4 The sequence of bulk erase of Flash memory

15.4.1.1 Flash Read

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Read data from Flash.

15.4.1.2 Enable program mode

Step 1. Enter OCD(=ISP) mode.[1]

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Enter program/erase mode sequence.[2]

(1) Write 0xAA to 0xF555.

(2) Write 0x55 to 0xFAAA.

(3) Write 0xA5 to 0xF555.

[1] Refer to how to enter ISP mode..

[2] Command sequence to activate Flash write/erase mode. It is composed of sequentially writing data of Flash memory.

15.4.1.3 Flash write mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:1000_1001
Step 4. Write data to page buffer.(Address automatically increases by twin.)
Step 5. Set write mode. FEMR:1010_0001
Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx
Step 7. Set FETCR.
Step 8. Start program. FECR:0000_1011
Step 9. Insert one NOP operation
Step 10. Read FESR until PEVBSY is 1.
Step 11. Repeat step2 to step 8 until all pages are written.

15.4.1.4 Flash page erase mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:1000_1001
Step 4. Write 'h00 to page buffer. (Data value is not important.)
Step 5. Set erase mode. FEMR:1001_0001
Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx
Step 7. Set FETCR.
Step 8. Start erase. FECR:0000_1011
Step 9. Insert one NOP operation
Step 10. Read FESR until PEVBSY is 1.
Step 11. Repeat step2 to step 8 until all pages are erased.

15.4.1.5 Flash bulk erase mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:1000_1001
Step 4. Write 'h00 to page buffer. (Data value is not important.)
Step 5. Set erase mode. FEMR:1001_0001.
(Only main cell area is erased. For bulk erase including OTP area, select OTP area.(set FEMR to 1000_1101.)
Step 6. Set FETCR

Step 7. Start bulk erase. FECR:1000_1011

Step 8. Insert one NOP operation

Step 9. Read FESR until PEVBSY is 1.

15.4.1.6 Flash OTP area read mode

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Select OTP area. FEMR:1000_0101

Step 5. Read data from Flash.

15.4.1.7 Flash OTP area write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write data to page buffer.(Address automatically increases by twin.)

Step 5. Set write mode and select OTP area. FEMR:1010_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

15.4.1.8 Flash OTP area erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode and select OTP area. FEMR:1001_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

15.4.1.9 Flash program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:1010_0011

Step 3. Read data from Flash.

15.4.1.10 OTP program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:1010_0111

Step 3. Read data from Flash.

15.4.1.11 Flash erase verify mode

Step 1. Enable program mode.

Step 2. Set erase verify mode. FEMR:1001_0011

Step 3. Read data from Flash.

15.4.1.12 Flash page buffer read

Step 1. Enable program mode.

Step 2. Select page buffer. FEMR:1000_1001

Step 3. Read data from Flash.

15.4.2 Summary of Flash Program/Erase Mode

Operation mode		Description
FLASH	Flash read	Read cell by byte.
	Flash write	Write cell by bytes or page.
	Flash page erase	Erase cell by page.
	Flash bulk erase	Erase the whole cells.
	Flash program verify	Read cell in verify mode after programming.
	Flash erase verify	Read cell in verify mode after erase.
	Flash page buffer load	Load data to page buffer.

Table 15-3. Operation Mode

15.5 Parallel Mode

15.5.1 Overview

Parallel program mode transfers address and data by byte. 3-byte address can be entered by one from the lease significant byte of address. If only LSB is changed, only one byte can be transferred. And if the second byte is changed, the first and second byte can be transferred. Upper 4-bit of the most significant byte selects memory to be accessed. Table 15.4 shows memory type to be accessible by parallel mode. Address auto-increment is supported when read or write data without address.

P3[2] (RESETB)
P3[5] (nRD)
P0[6] (nWR)
P3[0] (nALE)
P3[3] (PDATA[[7])
P1[7:5] (PDATA[6:4])
P1[3:0] (PDATA[3:0]

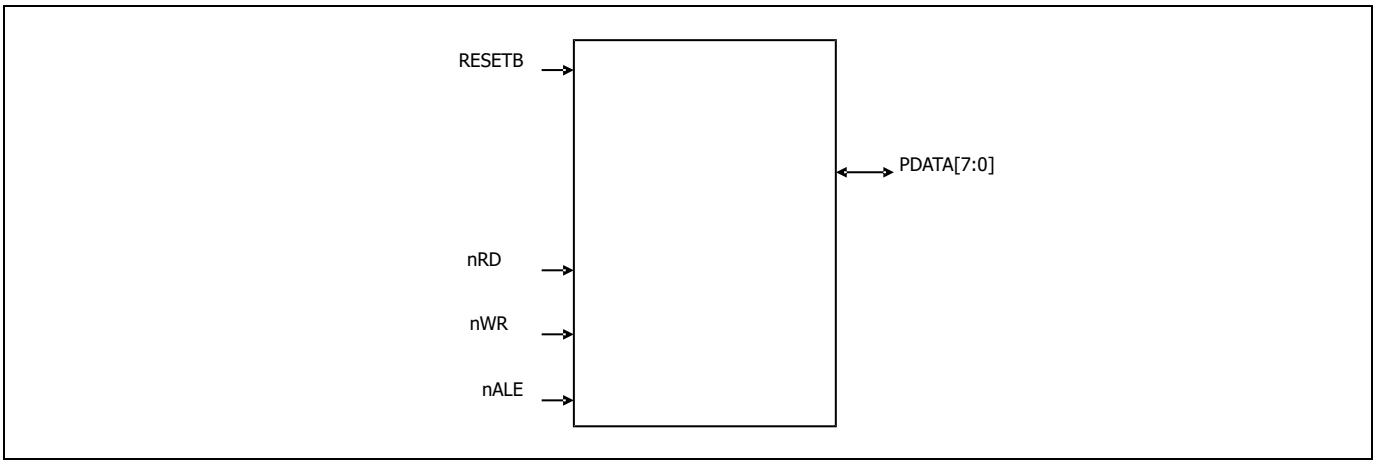


Figure 8.5 Pin diagram for parallel programming

ADDRH[7:4]				Memory Type
0	0	0	0	Program Memory
0	0	0	1	External Memory
0	0	1	0	SFR

Table 15-4. The selection of memory type by ADDRH[7:4]

15.5.2 Parallel Mode instruction format

Instruction	Signal	Instruction Sequence													
n-byte data read with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	H	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data write with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data read with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	H	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	L	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data write with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data read with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	H	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	
n-byte data write with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	

Table 1.1 Parallel mode instruction format

15.5.3 Parallel Mode timing diagram

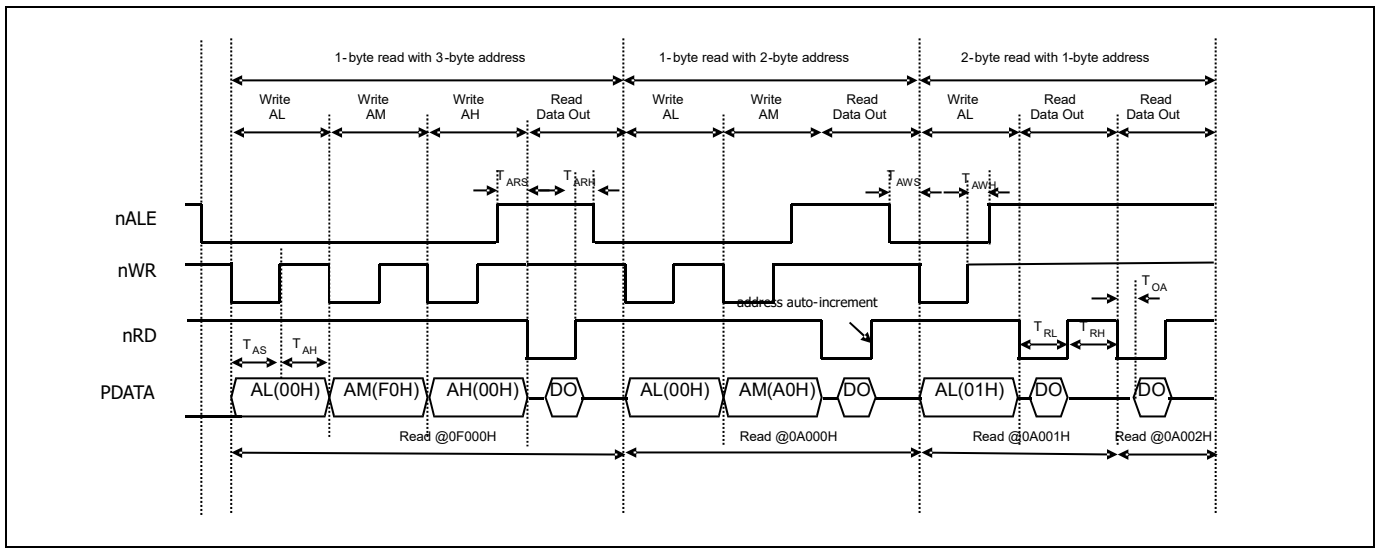


Figure 8.6 Parallel Byte Read Timing of Program Memory

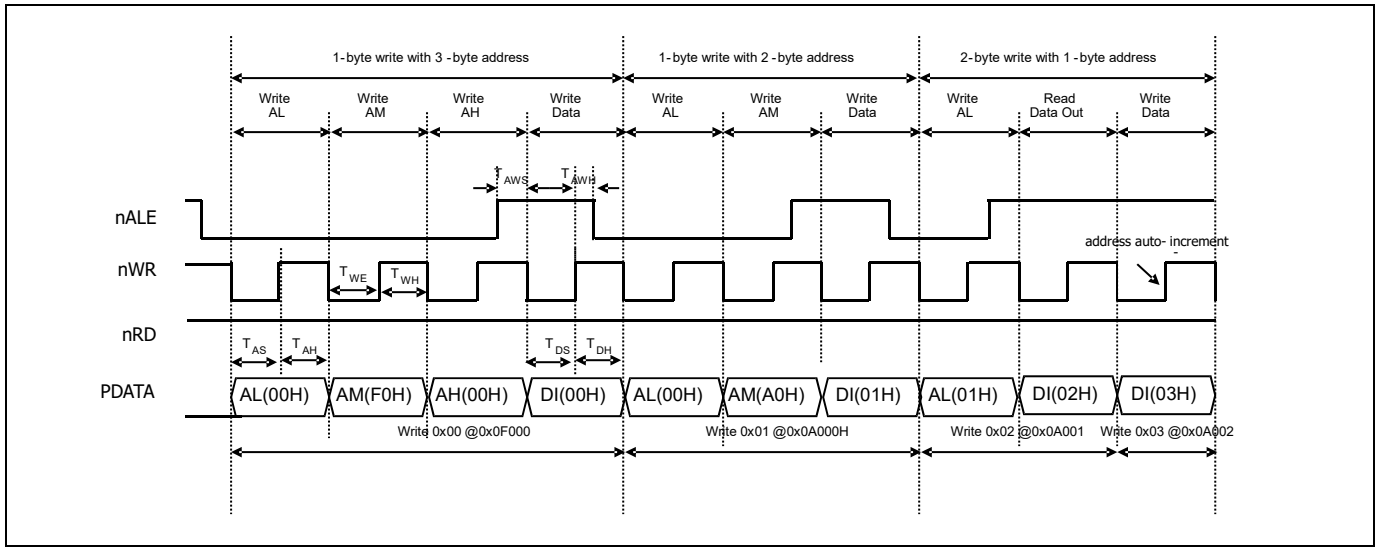


Figure 8.7 Parallel Byte Write Timing of Program Memory

15.6 Mode entrance method of ISP and byte-parallel mode

15.6.1 Mode entrance method for ISP

TARGET MODE	DSDA	DSCL	DSDA
OCD(ISP)	‘hC	‘hC	‘hC

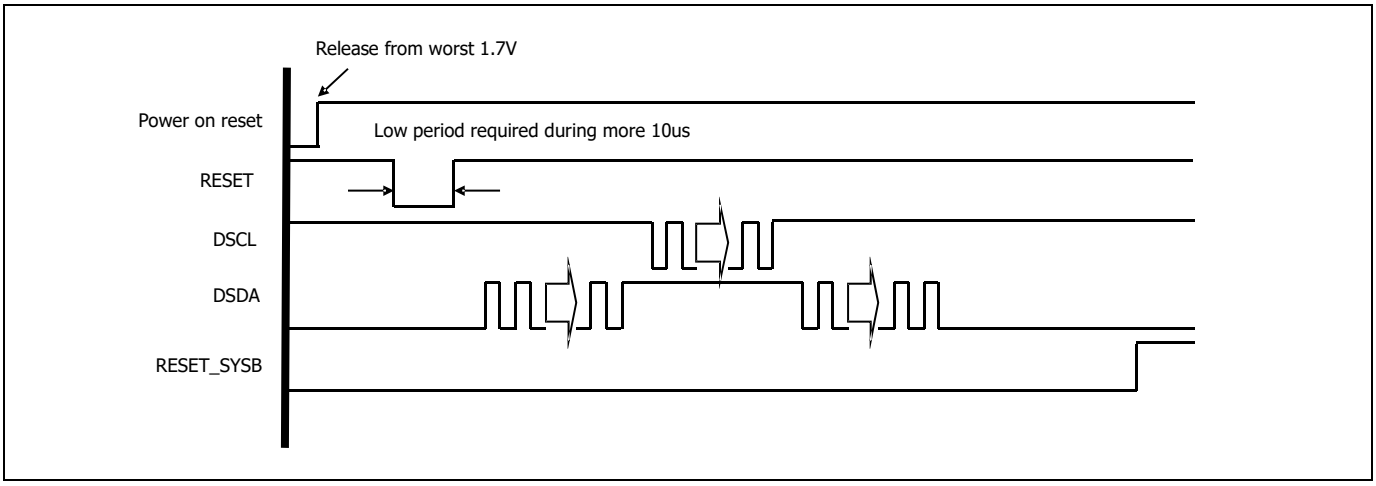


Figure 8.8 ISP mode

15.6.2 Mode entrance of Byte-parallel

TARGET MODE	P0[3:0]	P0[3:0]	P0[3:0]
Byte-Parallel Mode	4‘h5	4‘hA	4‘h5

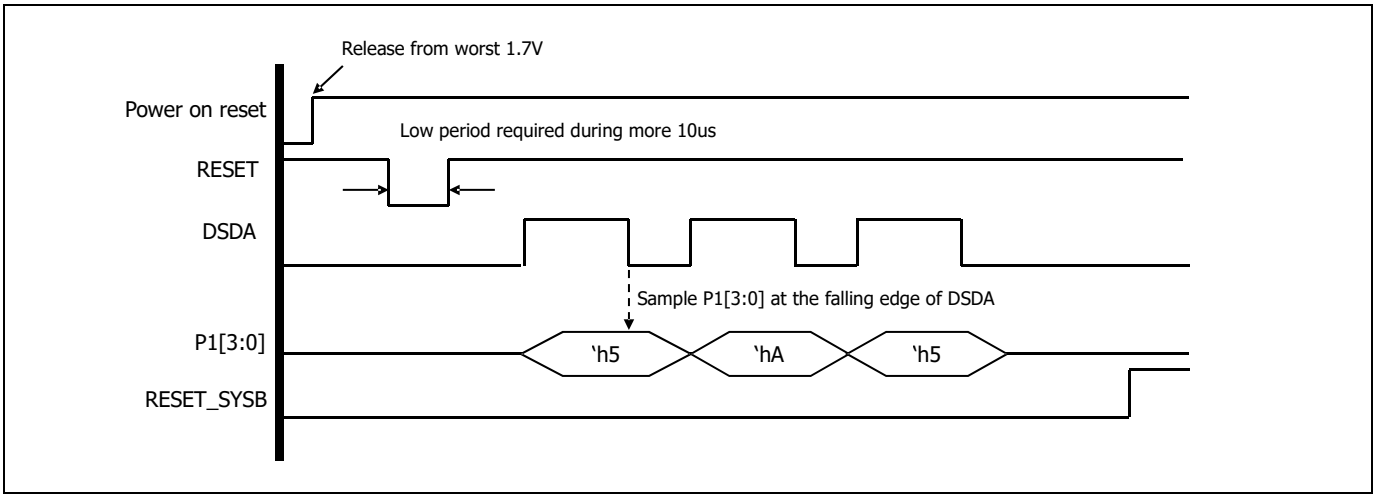


Figure 8.9 Byte-parallel mode

15.7 Security

MC97F6108A provides Lock bits which can be left unprogrammed (“0”) or can be programmed (“1”) to obtain the additional features listed in Table 15.5. The Lock bits can be erased to “0” with only the bulk erase command and a value of more than 0x80 at FETCR.

LOCK MODE	USER MODE								ISP/PMODE							
	FLASH				OTP				FLASH				OTP			
LOCKF	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	O	O	O	X	X	X	X	X	O	O	O	O	O	O	O	O
1	O	O	O	X	X	X	X	X	X	X	X	O	O	X	X	O

Table 15-5. Security policy using lock-bits

- LOCKF: Lock bit of Flash memory
- R: Read
- W: Write
- PE: Page erase
- BE: Bulk Erase
- O: Operation is possible.
- X: Operation is impossible.

16 Configure option

16.1 Configure option Control Register

FUSE_CFG0 (Pseudo-Configure Data)

7	6	5	4	3	2	1	0
BSIZE[1:0]		RSTEN	-	-	-	LOCKB	LOCKF
R	R	R	-	-	-	R	R

Initial value : 00H

BSIZE[1:0]	Enable Specific Area for Write Protection Bit	
	00	0000H~1EFFH (default)
	01	0100H~1DFFH
	10	0100H~1BFFH
	11	0100H~17FFH
LOCKB	Select Specific Area for Write Protection Bit	
	0	Hard LOCK Disable
	1	Hard LOCK Enable
RSTEN	Select RESETB pin	
	0	Disable RESETB pin (default)
	1	Enable RESETB pin
LOCKF	Code Read Protection Bit	
	0	LOCK Disable
	1	LOCK Enable

17 APPENDIX

A. Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.

Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following table.

1 machine cycle comprises 2 system clock cycles.

ARITHMETIC					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
ADD A,Rn	Add register to A	A=A+Rn	1	1	28-2F
ADD A,direct	Add direct byte to A	A=A+direct	2	1	25
ADD A,@Ri	Add indirect memory to A	A=A+@Ri	1	1	26-27
ADD A,#data	Add immediate to A	A=A+data	2	1	24
ADDC A,Rn	Add register to A with carry	A=A+Rn+C	1	1	38-3F
ADDC A,direct	Add direct byte to A with carry	A=A+direct+C	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	A=A+@Ri+C	1	1	36-37
ADDC A,#data	Add immediate to A with carry	A=A+data+C	2	1	34
SUBB A,Rn	Subtract register from A with borrow	A=A-Rn-C	1	1	98-9F
SUBB A,direct	Subtract direct byte from A with borrow	A=A-direct-C	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	A=A-@Ri-C	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	A=A-data-C	2	1	94
INC A	Increment A	A=A+1	1	1	04
INC Rn	Increment register	Rn=Rn+1	1	1	08-0F
INC direct	Increment direct byte	direct=direct+1	2	1	05
INC @Ri	Increment indirect memory	@Ri=@Ri+1	1	1	06-07
DEC A	Decrement A	A=A-1	1	1	14
DEC Rn	Decrement register	Rn=Rn-1	1	1	18-1F
DEC direct	Decrement direct byte	Direct=direct-1	2	1	15
DEC @Ri	Decrement indirect memory	@Ri=@Ri-1	1	1	16-17
INC DPTR	Increment data pointer	DPTR=DPTR+1	1	2	A3
MUL AB	Multiply A by B (the high byte remains in the B register)	{B,A}=A*B	1	4	A4
DIV AB	Divide A by B (A=quotient, B=remainder)	{A,B}=A/B	1	4	84
DA A	Decimal Adjust A	A={AH+6,AL+6}	1	1	D4

LOGICAL					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
ANL A,Rn	AND register to A	A=A&Rn	1	1	58-5F
ANL A,direct	AND direct byte to A	A=A&direct	2	1	55
ANL A,@Ri	AND indirect memory to A	A=A&@Ri	1	1	56-57
ANL A,#data	AND immediate to A	A=A&data	2	1	54
ANL direct,A	AND A to direct byte	direct=direct&A	2	1	52
ANL direct,#data	AND immediate to direct byte	direct=direct&data	3	2	53
ORL A,Rn	OR register to A	A=A|Rn	1	1	48-4F
ORL A,direct	OR direct byte to A	A=A|direct	2	1	45
ORL A,@Ri	OR indirect memory to A	A=A|@Ri	1	1	46-47
ORL A,#data	OR immediate to A	A=A|data	2	1	44
ORL direct,A	OR A to direct byte	direct=direct|A	2	1	42
ORL direct,#data	OR immediate to direct byte	direct=direct|data	3	2	43
XRL A,Rn	Exclusive-OR register to A	A=A^Rn	1	1	68-6F
XRL A,direct	Exclusive-OR direct byte to A	A=A^direct	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	A=A^@Ri	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	A=A^data	2	1	64
XRL direct,A	Exclusive-OR A to direct byte	direct=direct^A	2	1	62
XRL direct,#data	Exclusive-OR immediate to direct byte	direct=direct^data	3	2	63
CLR A	Clear A	A=#00H	1	1	E4
CPL A	Complement A	A=~A	1	1	F4
SWAP A	Swap Nibbles of A	A={AL,AH}	1	1	C4
RL A	Rotate A left(bit7→bit0)	A=A<<1	1	1	23
RLC A	Rotate A left through carry (bit7→C, C→bit0)	A=C,A<<	1	1	33
RR A	Rotate A right(bit0→bit7)	A=A>>1	1	1	03
RRC A	Rotate A right through carry (C→bit7, bit0→C)	A=C,A>>1	1	1	13

DATA TRANSFER					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
MOV A,Rn	Move register to A	A=Rn	1	1	E8-EF
MOV A,direct	Move direct byte to A	A=direct	2	1	E5
MOV A,@Ri	Move indirect memory to A	A=@Ri	1	1	E6-E7
MOV A,#data	Move immediate to A	A=data	2	1	74
MOV Rn,A	Move A to register	Rn=A	1	1	F8-FF
MOV Rn,direct	Move direct byte to register	Rn=direct	2	2	A8-AF
MOV Rn,#data	Move immediate to register	Rn=data	2	1	78-7F
MOV direct,A	Move A to direct byte	direct=A	2	1	F5
MOV direct,Rn	Move register to direct byte	direct=Rn	2	2	88-8F
MOV direct1,direct2	Move direct byte to direct byte	direct1=direct2	3	2	85
MOV direct,@Ri	Move indirect memory to direct byte	direct=@Ri	2	2	86-87
MOV direct,#data	Move immediate to direct byte	direct=data	3	2	75
MOV @Ri,A	Move A to indirect memory	@Ri=A	1	1	F6-F7
MOV @Ri,direct	Move direct byte to indirect memory	@Ri=direct	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	@Ri=data	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	DPTR=data16	3	3	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	A=@A+DPTR	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	A=@A+PC	1	2	83
MOVX A,@Ri	Move external data(A8) to A	A=@Ri	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	A=@DPTR	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	@Ri=A	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	@DPTR=A	1	2	F0
PUSH direct	Push direct byte onto stack	SP=SP+1,SP=direct	2	2	C0
POP direct	Pop direct byte from stack	Direct=SP,SP=SP-1	2	2	D0
XCH A,Rn	Exchange A and register	A<->Rn	1	1	C8-CF
XCH A,direct	Exchange A and direct byte	A<->direct	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	A<->@Ri	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	{AH,@RiL}<->{@RiH,AL}	1	1	D6-D7

BOOLEAN					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
CLR C	Clear carry	C=0	1	1	C3
CLR bit	Clear direct bit	bit=0	2	1	C2
SETB C	Set carry	C=1	1	1	D3
SETB bit	Set direct bit	bit=1	2	1	D2
CPL C	Complement carry	C=/C	1	1	B3
CPL bit	Complement direct bit	bit=/bit	2	1	B2
ANL C,bit	AND direct bit to carry	C=C&bit	2	2	82
ANL C,/bit	AND direct bit inverse to carry	C=C&/bit	2	2	B0
ORL C,bit	OR direct bit to carry	C=C|bit	2	2	72
ORL C,/bit	OR direct bit inverse to carry	C=C|/bit	2	2	A0
MOV C,bit	Move direct bit to carry	C=bit	2	1	A2
MOV bit,C	Move carry to direct bit	bit=C	2	2	92

BRANCHING					
Mnemonic	**Description**		**Bytes**	**Cycles**	**Opcode**
ACALL addr 11	Absolute jump to subroutine	SP=SP+1,SP=PCL SP=SP+1,SP=PCH PC=addr11	2	2	11→F1
LCALL addr 16	Long jump to subroutine	SP=SP+1,SP=PCL SP=SP+1,SP=PCH PC=addr16	3	2	12
RET	Return from subroutine	PCH=SP,SP=SP-1 PCL=SP,SP=SP-1 PC={PCH, PCL}	1	2	22
RETI	Return from interrupt	PCH=SP,SP=SP-1 PCL=SP,SP=SP-1 PC={PCH, PCL}	1	2	32
AJMP addr 11	Absolute jump unconditional	PC=addr11	2	2	01→E1
LJMP addr 16	Long jump unconditional	PC=addr16	3	2	02
SJMP rel	Short jump (relative address)	PC=PC+rel	2	2	80
JC rel	Jump on carry = 1	if(C=1),PC=PC+rel	2	2	40
JNC rel	Jump on carry = 0	if(C≠1),PC=PC+rel	2	2	50
JB bit,rel	Jump on direct bit = 1	if(bit=1),PC=PC+rel	3	2	20
JNB bit,rel	Jump on direct bit = 0	if(bit≠1),PC=PC+rel	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	if(bit=1),PC=PC+rel bit=0	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	PC=@A+DPTR	1	2	73
JZ rel	Jump on accumulator = 0	if(A=00h),PC=PC+rel	2	2	60
JNZ rel	Jump on accumulator ≠ 0	if(A≠00h),PC=PC+rel	2	2	70
CJNE A,direct,rel	Compare A,direct jne relative	if(A≠direct),PC=PC+rel	3	2	B5
CJNE A,#data,rel	Compare A,immediate jne relative	if(A≠data),PC=PC+rel	3	2	B4
CJNE Rn,#data,rel	Compare register, immediate jne relative	if(Rn≠data),PC=PC+rel	3	2	B8-BF
CJNE @Ri,#data,rel	Compare indirect, immediate jne relative	if(@Ri≠data),PC=PC+rel	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	Rn=Rn-1 if(Rn≠00h),PC=PC+rel	2	2	D8-DF
DJNZ direct,rel	Decrement direct byte, jnz relative	direct=direct-1 if(direct≠00h),PC=PC+rel	3	2	D5

MISCELLANEOUS					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
NOP	No operation	-	1	1	00

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])					
Mnemonic	**Description**	**Operation**	**Bytes**	**Cycles**	**Opcode**
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory		1	2	A5
TRAP	Software break command		1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex op-codes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

B. Package relation

	MC97F6108AD(G)	MC97F6108AM(B)
Pin count	20	16
Max I/O	18	14
Difference (removed functions on standard MC97F6108A)	-	P0[0], P1[0], P1[1], P1[2]-
	-	RSTB
	-	T2, EC0
	-	AN6
	-	DSCL1, DSDA1

Note)

When using 16-pin products, it is recommanded that configure internal pull-up to the floating pin in order to prevent current consumption.

Table of contents