HMS87C5216

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER FOR UR(Universal Remocon) & WIRELESS KEYBOARD

1. OVERVIEW

1.1 Description

The HMS87C5216 is an advanced CMOS 8-bit microcontroller with 16K bytes of ROM. The device is one of GMS800 family. The MagnaChip Semicon HMS87C5216 is a powerful microcontroller which provides a highly flexible and cost effective solution to many UR & Keyboard applications. The HMS87C5216 provides the following standard features: 16K bytes of ROM, 320 bytes of RAM, 8-bit timer/counter, on-chip oscillator,clock circuitry and RC wake up function. 4 chanel ADC, In addition, the HMS87C5216 Series supports power saving modes to reduce power consumption

Device name	ROM Size	EPROM Size	RAM Size	Operatind Voltage	Package
HMS87C5216	-	16K byte	320bytes	2.0 ~ 5.5V	28 SOP 40 PDIP 44 PLCC 44 QFP

1.2 Features

- **Instruction Cycle Time:**
  - 1us at 4MHz
- **Programmable I/O pins**

	28 PIN	40 PIN	44 PIN
INPUT	2	2	2
OUTPUT	2	2	2
I/O	22	34	38

- **Operating Voltage**
  - 2.0 ~ 5.5 V @ 4MHz
- **Timer**
  - Timer / Counter .... 16Bit * 1ch
  .... 16Bit * 2ch
  - Basic Interval Timer ...... 8Bit * 1ch
  - Watch Dog Timer .... 6Bit * 1ch
- **8 Interrupt sources**
  *** Nested Interrupt control is available.**
  - External input: 2
  - Keyscan input
  - Basic Interval Timer
  - Watchdog timer
  - Timer : 3
- **Power On Reset**
- **Power saving Operation Modes**
  - STOP
  - SLEEP
- **Low Voltage Detection Circuit**
- **Watch Dog Timer Auto Start (During 1second after Power on Reset)**
- **4 CHANEL ADC**
- **RC TIMER WAKE UP**

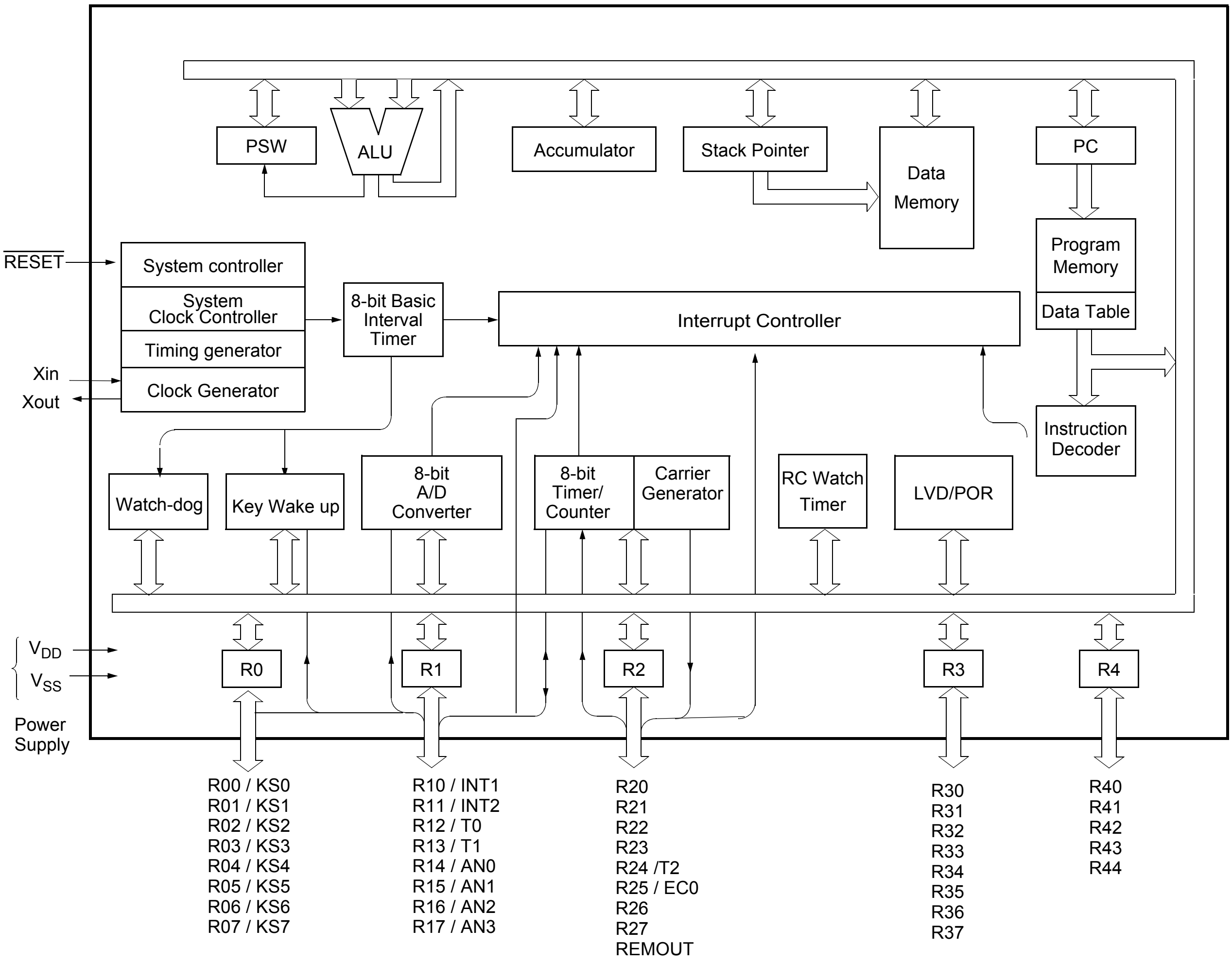
SEP. 2004 Ver 1.01

1.3 Development Tools

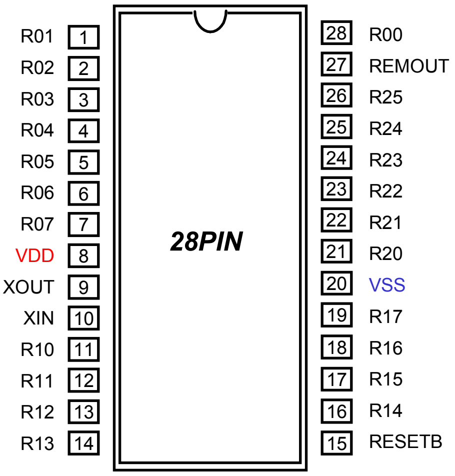
The HMS87C5216 and HMS87C5216 are supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Dr^{TM}.

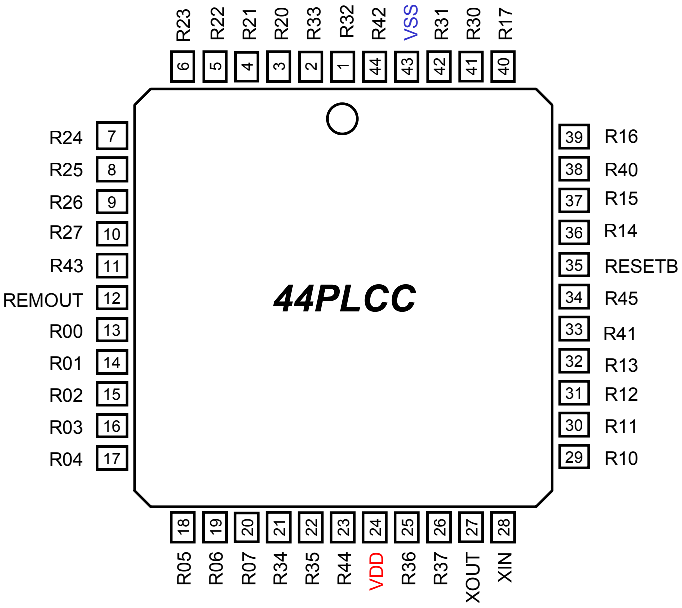
In Circuit Emulators	CHOICE-Dr.
Assembler	HME Macro Assembler
OTP Writer	Single Writer : Sigma
	4-Gang Writer : Dr.Gang
OTP Devices	HMS87C5216

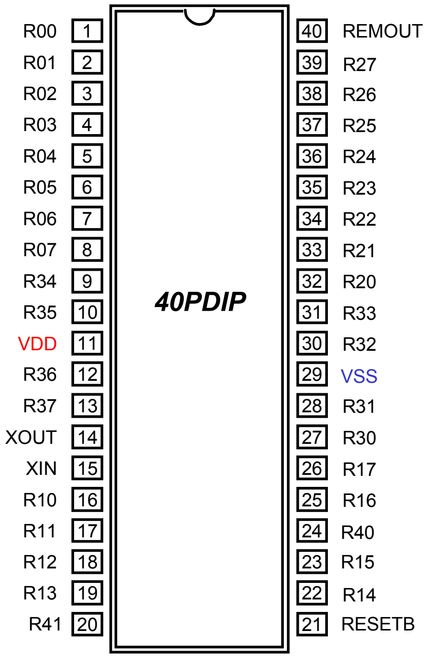
2. BLOCK DIAGRAM

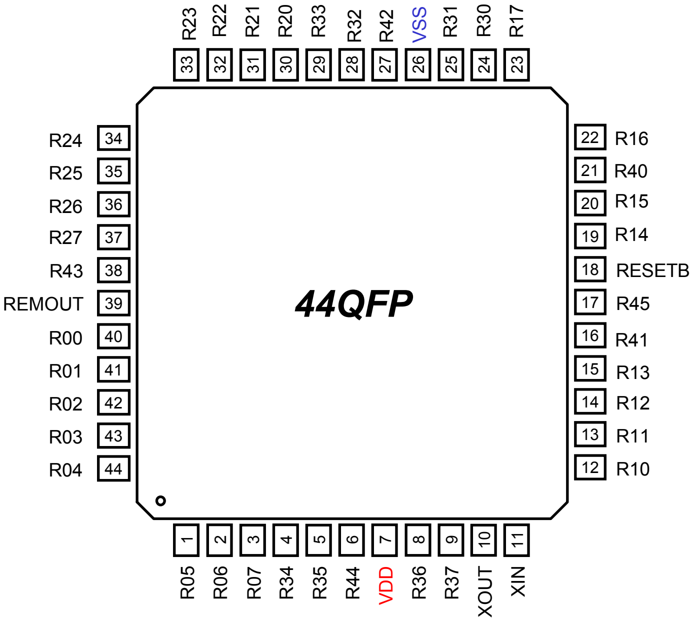


3. PIN ASSIGNMENT

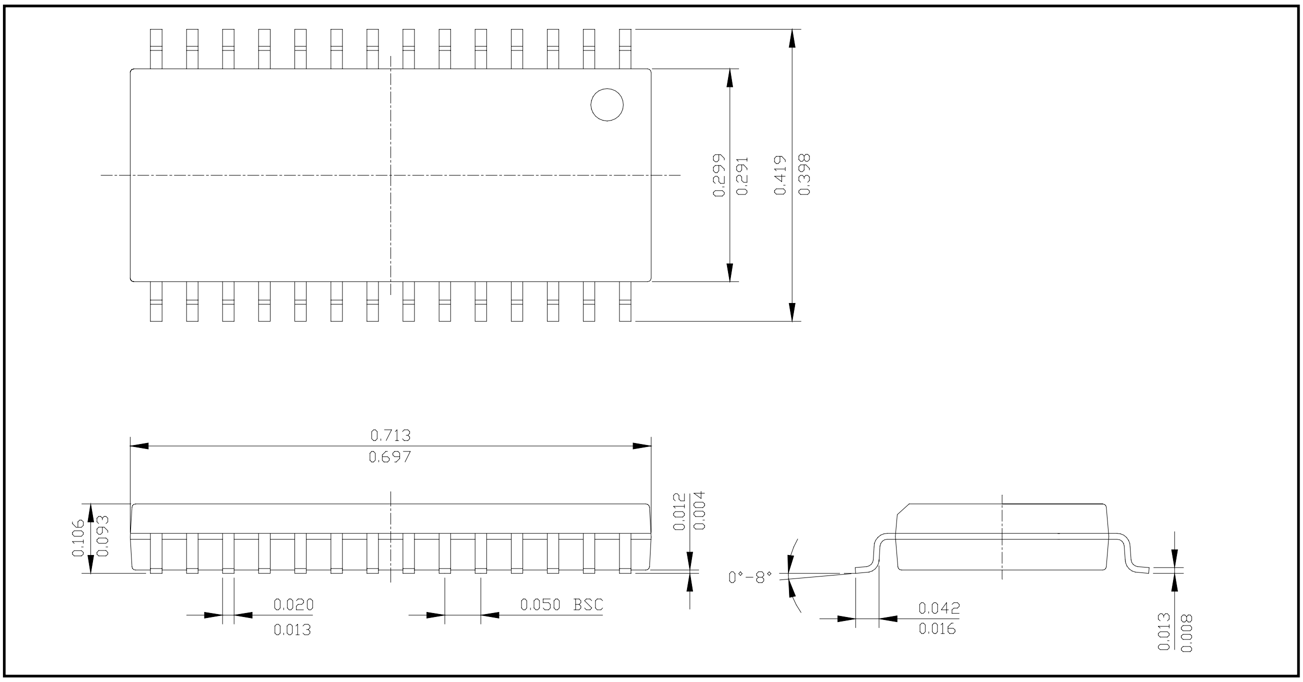


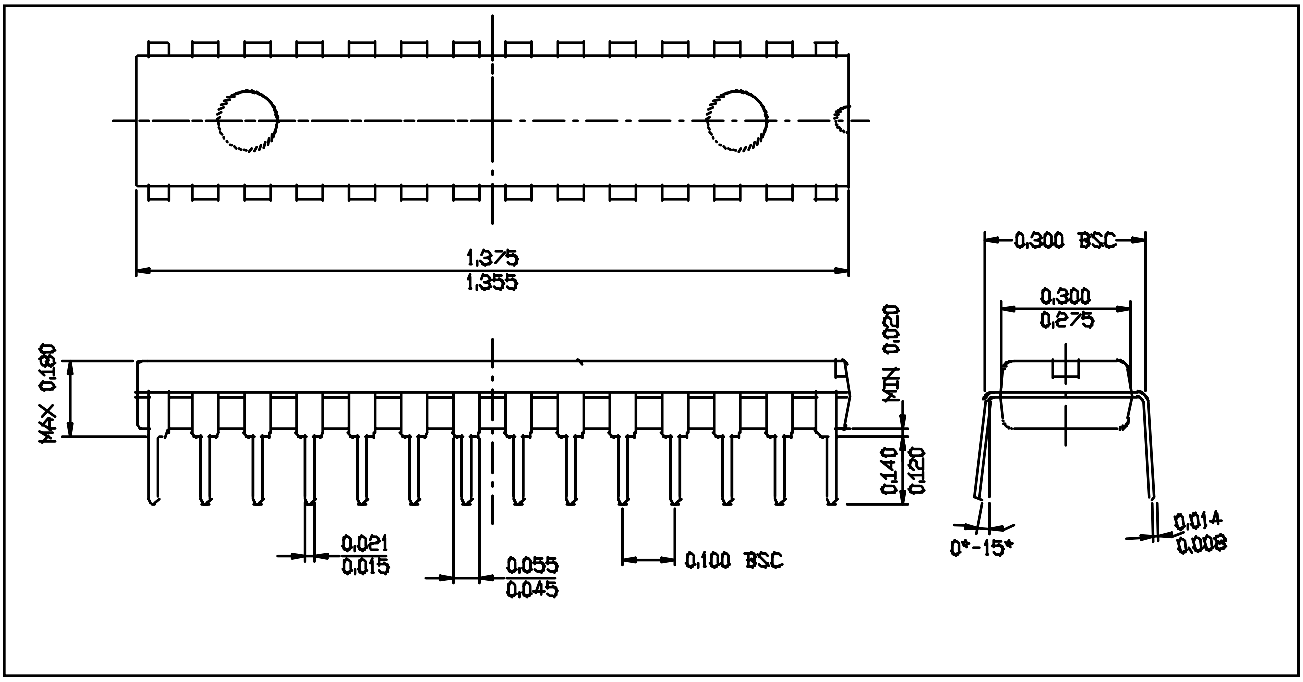


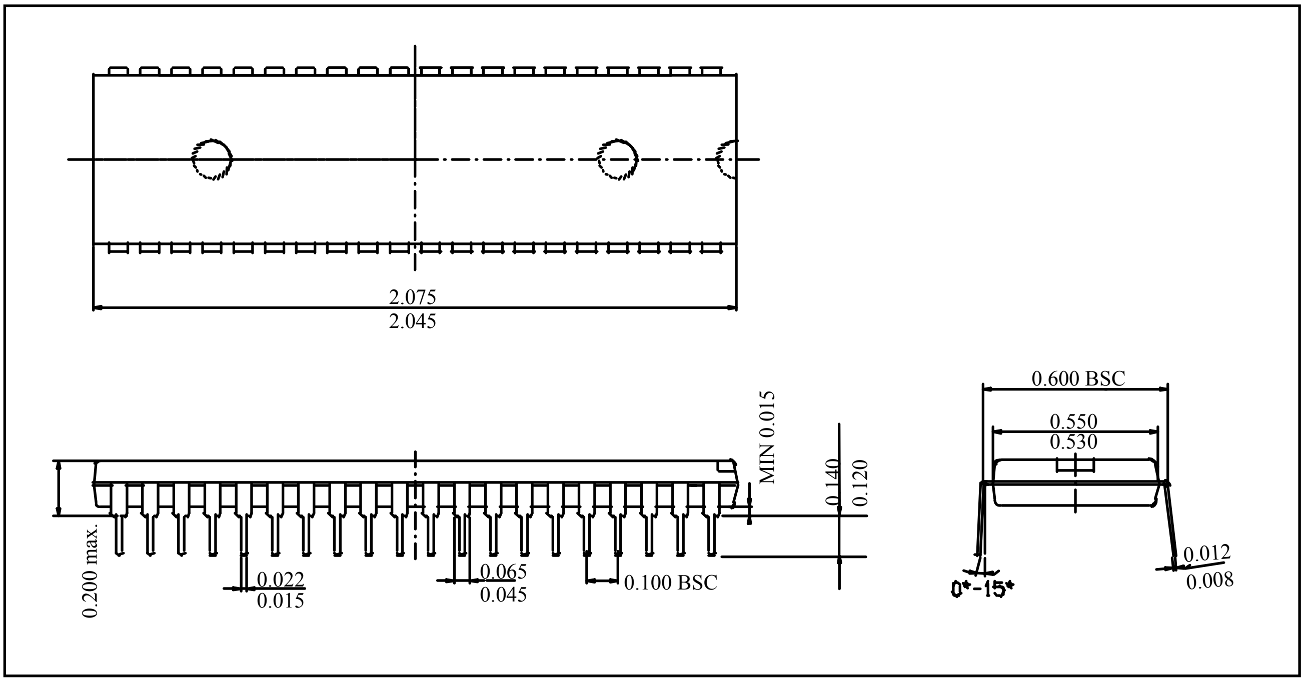


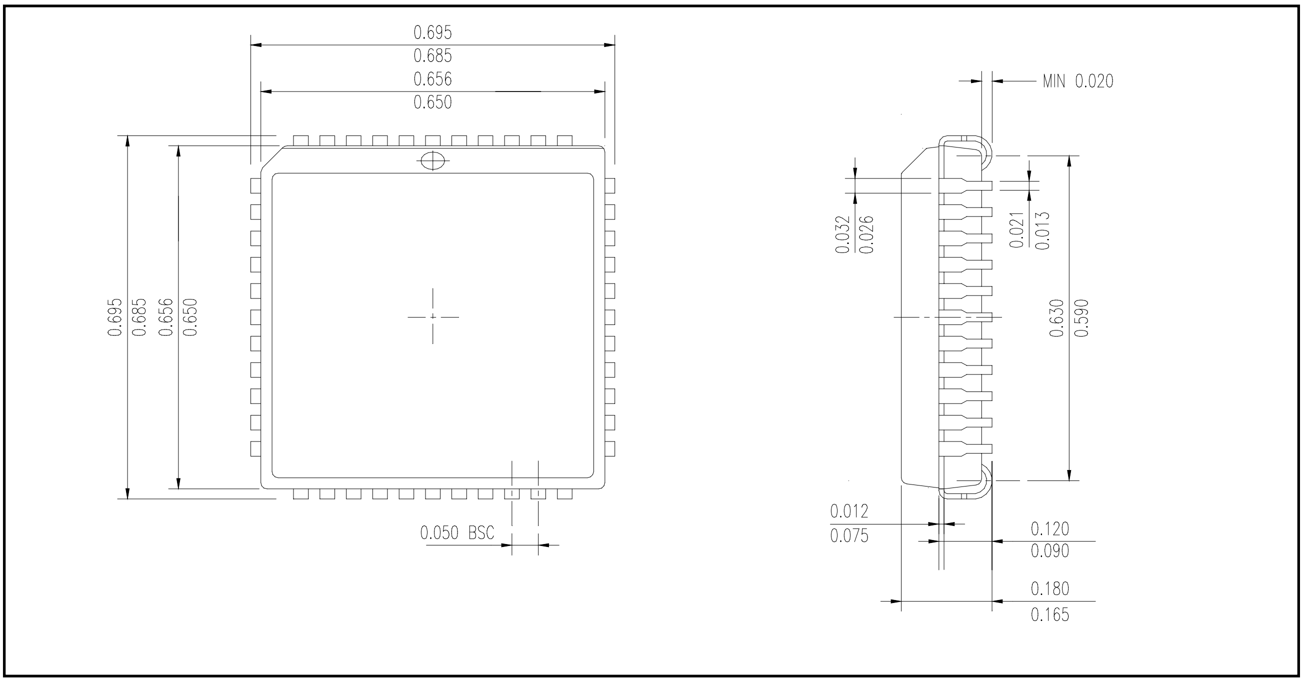


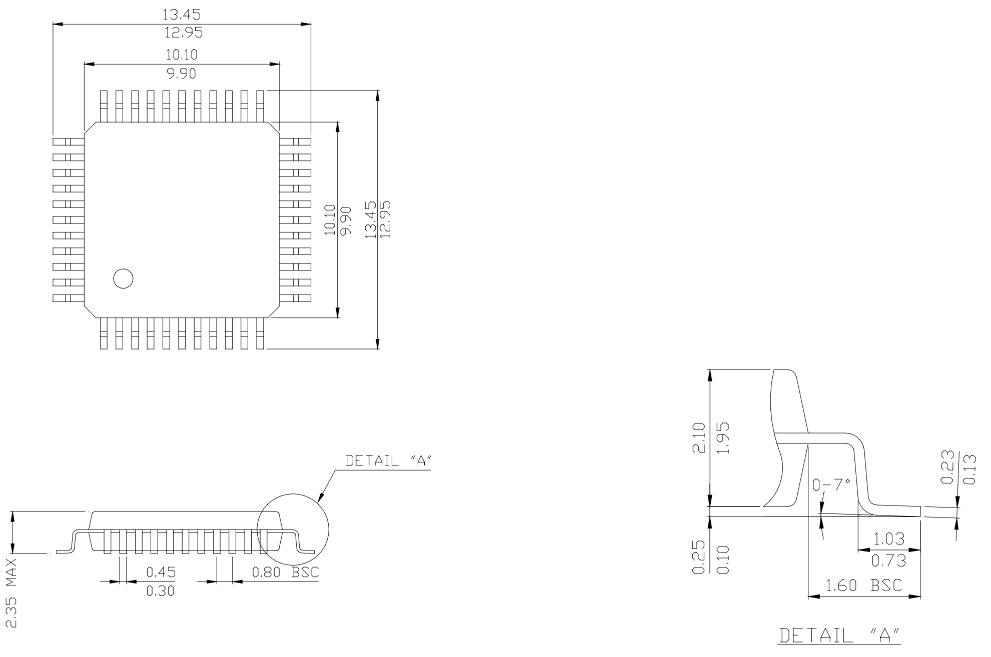
4. PIN DIAGRAM






2.075
2.045
0.200 max.
0.022
0.015
0.065
0.045
0.100 BSC
MIN 0.015
0.140
0.120
0.600 BSC
0.550
0.530
0.012
0.008
0°-15°


0.695
0.685
0.656
0.650
0.695
0.685
0.656
0.650
0.050 BSC
MIN 0.020
0.032
0.026
0.021
0.013
0.630
0.590
0.012
0.075
0.120
0.090
0.180
0.165


13.45
12.95
10.10
9.90
10.10
9.90
13.45
12.95
DETAIL "A"
2.35 MAX
0.45
0.30
0.80 BSC
2.10
1.95
0-7°
0.23
0.13
1.03
0.73
0.25
0.10
1.60 BSC
DETAIL "A"

5. PIN FUNCTION

V_{DD}: Supply voltage.

V_{SS}: Circuit ground.

$\overline{\text{RESET}}$: Reset the MCU.

X_{IN}: Input to the inverting oscillator amplifier and input to the internal main clock operating circuit.

X_{OUT}: Output from the inverting oscillator amplifier.

R00~R07: R0 is an 8-bit CMOS bidirectional I/O port. R0 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

R10~R17: R1 is an 8-bit CMOS bidirectional I/O port. R1 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

In addition, R1 serves the functions of the various following special features.

Port pin	Alternate function
R10	INT1 (External Interrupt input 1)
R11	INT2 (External Interrupt input 2)
R12	T0 (Timer / Counter inpit 0)
R13	T1 (Timer / Counter inpit 1)
R14	AN0 (ADC input 0)
R15	AN1 (ADC input 1)
R16	AN2 (ADC input 2)
R17	AN3 (ADC input 3)

R20~R22, **R30~R37** : R2 & R3 is a 8-bit CMOS bidirectional I/O port. Each pins 1 or 0 written to the their Port Direction Register can be used as outputs or inputs.

In addition, R2 serves the functions of the various following special features.

Port pin	Alternate function
R24	T2 (Timer / Counter inpit 2)
R25	/EC (Event Counter input)

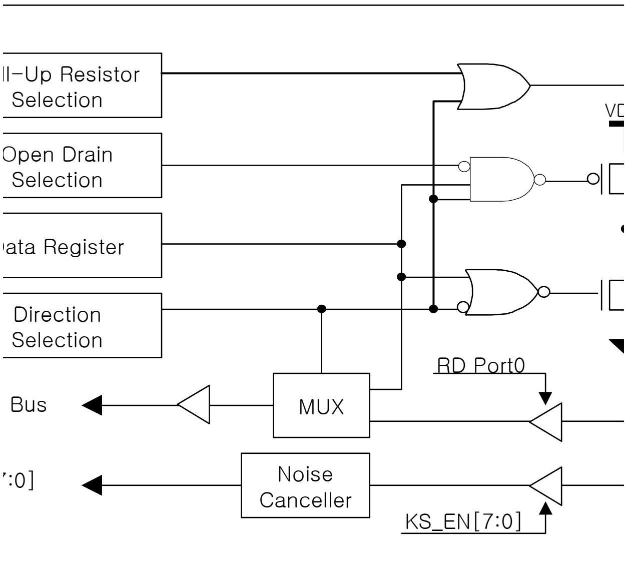
R40~R43 : R4 is 1-bit CMOS bidirectional I/O port. This pin 1 or 0 written to the its Port Direction Register can be used as outputs or inputs.

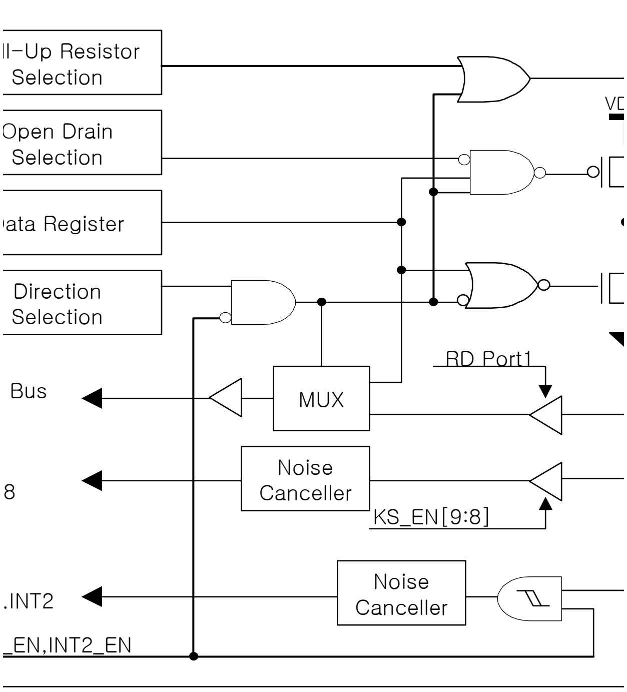
6.

PIN Name	I/O	#	Description	@Reset	@STOP
XIN XOUT	I,O	2	Main Clock Input,Output	Oscillation	L, L
R00/KS0 R01/KS1 R02/KS2 R03/KS3 R04/KS4 R05/KS5 R06/KS6 R07/KS7	I/O	8	▶Each bit of the port can be individually configured as an input or an output by user software ▶Push-pull output ▶CMOS input with pull-up resistor (can be programmable) ▶Programmable Key Scan Input or Open drain output ▶Pull-ups are automatically disabled at output mode	Input	State of before STOP
R10/KS8/INT1 R11/KS9/INT2 R12/KS10/T0 R13/KS11/T1 R14/KS12/AN0 R15/KS13/AN1 R16/KS14/AN2 R17/KS15/AN3	I/O	8	▶Each bit of the port can be individually configured as an input or an output by user software ▶Push-pull output ▶CMOS input with pull-up resistor (can be programmable) ▶Programmable Key Scan Input or Open drain output ▶Direct Driving of LED (N-TR) ▶Pull-ups are automatically disabled at output mode ▶R1[7:4] is High Driving Capability ▶R1[7:4] is Schmitt Trigger Input.	Input	State of before STOP
R2[3:0] R24/T2 R25/EC0 R2[7:6] R3[7:0] R4[5:0]	I/O	22	▶Each bit of the port can be individually configured as an input or an output by user software ▶Push-pull output ▶CMOS input with pull-up resistor (can be programmable) ▶Programmable Open drain output ▶Direct Driving of LED (N-TR) ▶Pull-ups are automatically disabled at output mode	Input	State of before STOP
RESETB	I	1	Resetb Pin	L	H
REMOUT	O	1	Remocon Output	L	L
VDD	-	1	Power Supply	VDD	VDD
VSS	-	1	Ground	VSS	VSS

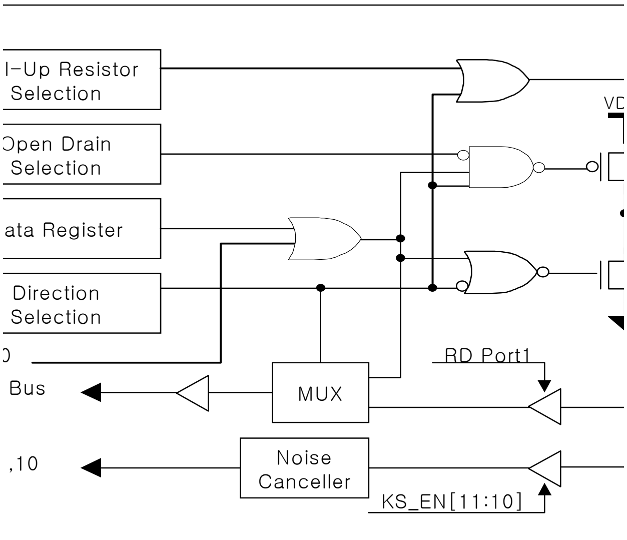
7. PORT STRUCTURES

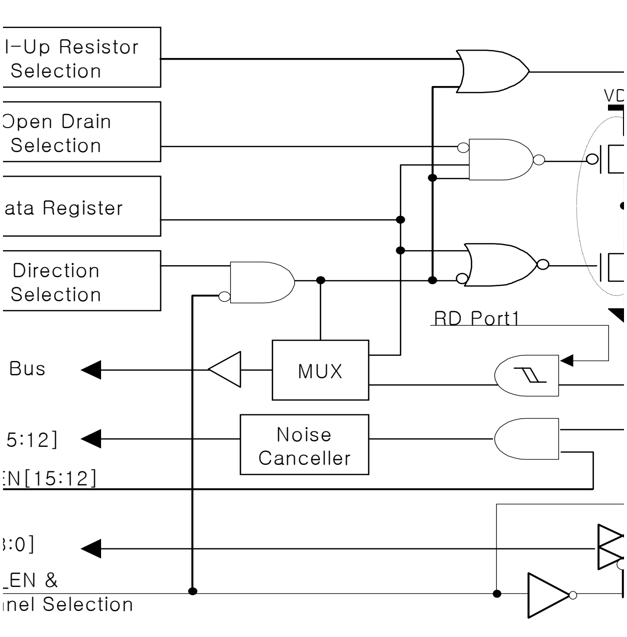
- **$\overline{\text{RESET}}$**

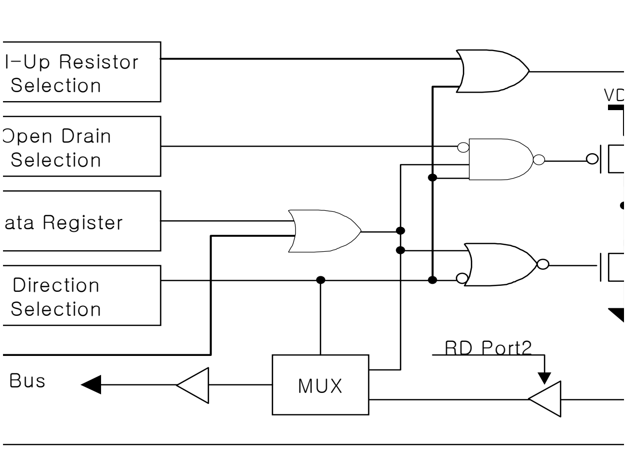




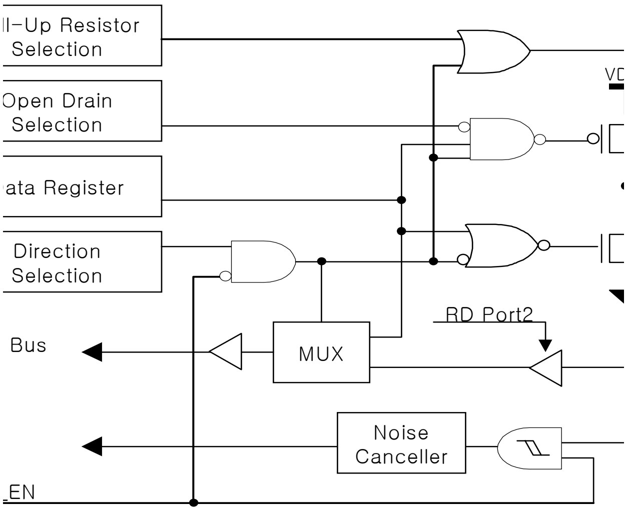
- **Xin, Xout**

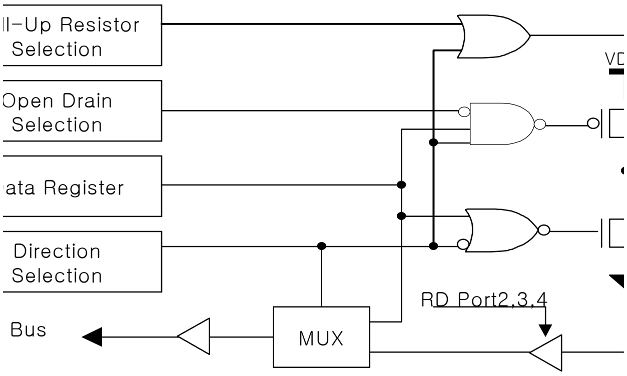

l-Up Resistor Selection
Open Drain Selection
ata Register
Direction Selection
)
Bus
,10
VD
MUX
RD Port1
Noise Canceller
KS_EN[11:10]

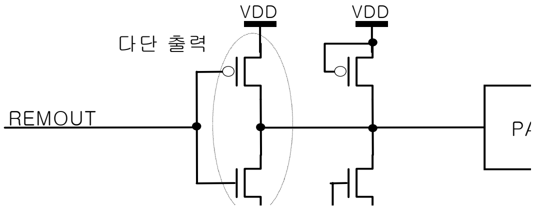

l-Up Resistor Selection
Open Drain Selection
ata Register
Direction Selection
Bus
5:12]
EN[15:12]
:0]
_EN &
nel Selection
VD
MUX
RD Port1
Noise Canceller


l-Up Resistor Selection
Open Drain Selection
ata Register
Direction Selection
Bus
VD
MUX
RD Port2

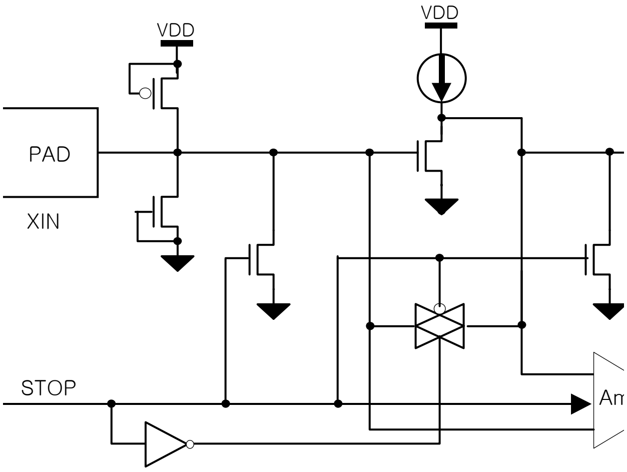
- **RA0/EC0**







- **RA1/AN1 ~ RA7/AN7**



8. ELECTRICAL CHARACTERISTICS (HMS87C5216/GMS81C1408)

8.1 Absolute Maximum Ratings

Supply voltage...-0.3 to +7.0 V

Storage Temperature ....-40 to +125 °C

aximum current out of V_{SS} pin .... TBD mA

Maximum current into V_{DD} pin .... TBD mA

Maximum current sunk by (I_{OL} per I/O Pin) .... TBD mA

Maximum output current sourced by (I_{OH} per I/O Pin) .... TBD mA

Maximum current (ΣI_{OL}) .... TBDmA

Maximum current (ΣI_{OH})... TBDmA

***Note:** Stresses above 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 above 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.*

8.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Specifications		Unit
			Min.	Max.	
Supply Voltage	V_{DD}	f_{XIN}=4MHz	2.0	5.5	V
Operating Frequency	f_{XIN}	V_{DD}=2.0~5.5V	1	4	MHz
Operating Temperature	T_{OPR}	V_{DD}=2.0~5.5V	-20	85	°C

8.3 A/D Converter Characteristics

(T_A=25° C, V_{SS}=0V, V_{DD}=3.072V @f_{XIN} =4MHz)

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Analog Input Voltage Range	V_{AIN}	-	V_{SS}-0.3	-	V_{DD}+0.3	V
Current Following Between AVdd and AVss	IAVdd	-	-	-	200	uA
Overall Accuracy	N_{ACC}	-	-	±1.0	±2.0	LSB
Non-Linearity Error	N_{NLE}	-	-	±1.0	±2.0	LSB
Differential Non-Linearity Error	N_{DNLE}	-	-	±1.0	±2.0	LSB
Zero Offset Error	N_{ZOE}		-	±0.5	±1.5	LSB
Full Scale Error	N_{FSE}		-	±0.25	±0.5	LSB
Gain Error	N_{NLE}		-	±1.0	±1.5	LSB
Conversion Time	T_{CONV}	f_{XIN}=4MHz	-	-	30	uS

8.4 DC Electrical Characteristics

(T_A=-20~85° C for HMS87C5216/1408 or T_A=-40~85° C for HMS87C5216E/1408E, V_{DD}=2.2~5.5V, V_{SS}=0V),

Parameter	Symbol	Pin	Condition	Specifications			Unit
				Min.	Typ.	Max.	
Input High Voltage	V_{IH1}	X_{IN}, $\overline{RESET}$		0.8 V_{DD}	-	V_{DD}	V

Parameter	Symbol	Pin	Condition	Specifications			Unit
				Min.	Typ.	Max.	
Input High Voltage	V_{IH1}	RESET,XIN,INT1,INT2,EC0,R1<7:4>		$0.8V_{DD}$	-	V_{DD}	V
	V_{IH2}	R0,R1,R2,R3,R4		$0.7V_{DD}$	-	V_{DD}	V
Input Low Voltage	V_{IL1}	RESET,XIN,INT1,INT2,EC0,R1<7:4>		0	-	$0.2V_{DD}$	V
	V_{IL2}	R0,R1,R2,R3,R4		0	-	$0.3V_{DD}$	V
Input High Leakage Current	I_{IH}	R0,R1,R2,R3,R4 RESETB	V_{IH}=VDD	-	-	1.0	µA
Input Low Leakage Current	I_{IL}	R0,R1,R2,R3,R4	V_{IL}=0V	-	-	-1.0	µA
Output High Voltage	V_{OH1}	R0,R1<3:0>,R2,R3,R4	Ioh1=-0.8mA,VDD=3V	VDD-0.4	-	-	V
	V_{OH2}	R1<7:0>,	Ioh2=-2.0mA,VDD=3V	VDD-0.4	-	-	V
	V_{OH3}	XOUT	Ioh3=-50uA,VDD=3V	VDD-0.5	-	-	V
Output Low Voltage	V_{OL1}	R0,R1<3:0>,R2,R3,R4	I_{OL}=5mA,V_{DD}=3V	-	-	0.8	V
	V_{OL2}	XOUT	I_{OL}=50uA,V_{DD}=3V	-	-	0.5	V
Output High Leakage Current	I_{IOHL}	R0,R1,R2,R3,R4	V_{OH}=VDD	-	-	1.0	µA
Output Low Leakage Current	I_{IOLL}	R0,R1,R2,R3,R4	V_{OL}=0V	-	-	-1.0	µA
Output High Current	I_{OH}	REMOUT	VDD=3V,VOH=2.0V	-20	-	-5	mA
Output Low Current	I_{OL}	REMOUT	VDD=3V,VOL=1.0V	-0.5	-	3	mA
Input Pull-up	I_P	R0,R1,R2,R3,R4 RESETB	V_{DD}=3V	50	100	200	κ
Hysteresis	$\| V_T \|$	Hysteresis Input[1]	V_{DD}=5V	0.5	-	-	V
Feed Back Resistor	RF!	Main OSC Feedback Resistor	V_{DD}=3.0V, f_{XIN}=4MHz	0.2	-	1.0	
Supply Currnet	I_{DD}	Active Mode	V_{DD}=4.0V	-	4.0	10	mA
			V_{DD}=2.0V	-	2.4	6	mA
	I_{sleep}	Sleep Mode	V_{DD}=4.0V	-	2.0	3.0	mA
			V_{DD}=2.0V	-	1.0	2.0	mA
	I_{stop}	Stop Mode,Osc Stop	V_{DD}=4.0V	-	5.0	30	µA
			V_{DD}=2.0V	-	3.0	25	µA

8.5 AC Characteristics

(T_A=-20~85° C for HMS87C5216/1408 or T_A=-40~85° C for HMS87C5216E/1408E, V_{DD}=5V±10%, V_{SS}=0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Operating Frequency	f_{MCP}	X_{IN}	1	-	4	MHz
Systemp Clock Cycle Time	t_{SYS}	-	0.5	-	2.0	uS
Oscillation Stabilizing Time(4MHz)	$t_{MST!}$	X_{IN}, X_{OUT}	-	-	20	mS
External Clock “H” or “L” Pulse Width	t_{CPW}	X_{IN}	80			nS
External Clock Transition Time	t_{RCP},t_{FCP}	X_{IN}	-	-	20	nS
Interrupt Input Pulse Width	t_{IW}	INT1,INT2	2	-		t_{SYS}
RESETB Input Pulse “L” Width	t_{RST}	RESETB	8	-	-	t_{SYS}
Event Couter Input “H” or “L” Pulse Width	t_{TCW}	ECo	2	-	-	t_{SYS}
Event Couter Transition Time	$t_{REC,}t_{FEC}$	ECo	0	-	20	nS

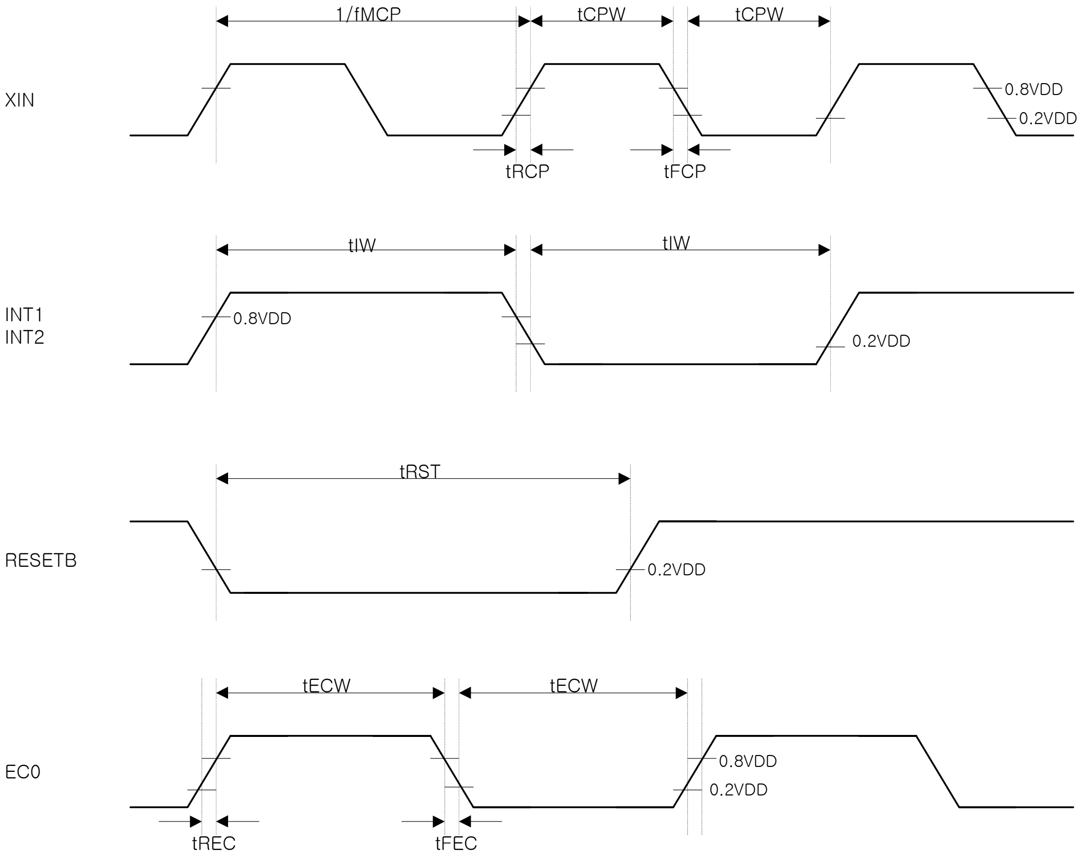


Figure 8-1 Timing Chart

9. MEMORY ORGANIZATION

The HMS87C5216 have separate address spaces for Program memory and Data Memory. Program memory can only be read, not written to. It can be up to 16K bytes of Program memory. Data memory can be read and written to up to 320 bytes including the stack area.

9.1 Registers

This device has six registers that are the Program Counter (PC), a Accumulator (A), two index registers (X, Y), the Stack Pointer (SP), and the Program Status Word (PSW). The Program Counter consists of 16-bit register.

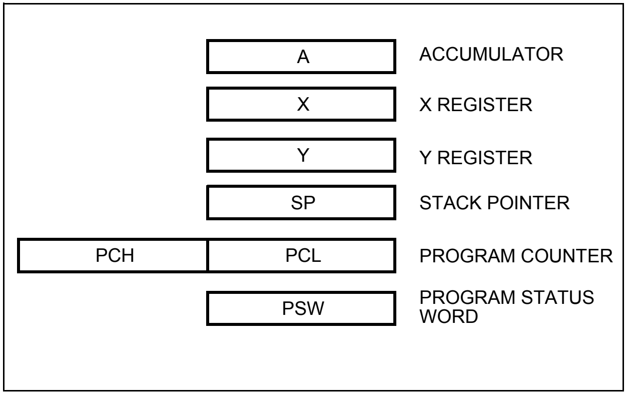
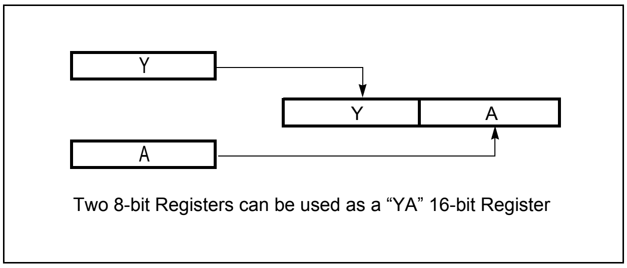


Figure 9-1 Configuration of Registers

Accumulator: The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional judgement, etc.

The Accumulator can be used as a 16-bit register with Y Register as shown below.



Two 8-bit Registers can be used as a “YA” 16-bit Register

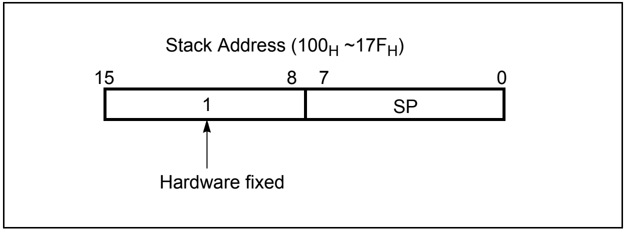
Figure 9-2 Configuration of YA 16-bit Register

X, Y Registers: In the addressing mode which uses these index registers, the register contents are added to the specified address, which becomes the actual address. These modes are extremely effective for referencing subroutine tables and memory tables. The index registers also have increment, decrement, comparison and data transfer functions, and they can be used as simple accumulators.

Stack Pointer: The Stack Pointer is an 8-bit register used for occurrence interrupts and calling out subroutines. Stack Pointer identifies the location in the stack to be accessed (save or restore).

Generally, SP is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost.

The stack can be located at any position within 100_H to $17F_H$ of the internal data memory. The SP is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of “$17F_H$” is used.



Note: *The Stack Pointer must be initialized by software because its value is undefined after RESET.*
Example: To initialize the SP

```
        LDX       #07FH
        TXSP                        ; SP ←7FH
```

Program Counter: The Program Counter is a 16-bit wide which consists of two 8-bit registers, PCH and PCL. This counter indicates the address of the next instruction to be executed. In reset state, the program counter has reset routine address (PC_H:$0FF_H$, PC_L:$0FE_H$).

Program Status Word: The Program Status Word (PSW) contains several bits that reflect the current state of the CPU. The PSW is described in Figure 9-3. It contains the Negative flag, the Overflow flag, the Break flag the Half Carry (for BCD operation), the Interrupt enable flag, the Zero flag, and the Carry flag.

[Carry flag C]

This flag stores any carry or borrow from the ALU of CPU after an arithmetic operation and is also changed by the Shift Instruction or Rotate Instruction.

[Zero flag Z]

This flag is set when the result of an arithmetic operation or data transfer is “0” and is cleared by any other result.

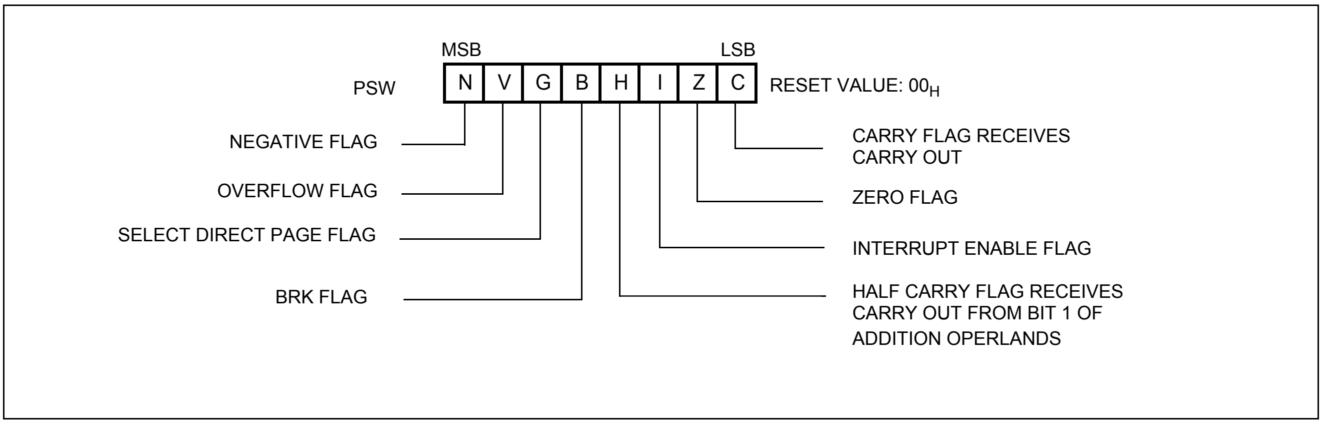


Figure 9-3 PSW (Program Status Word) Register

[Interrupt disable flag I]

This flag enables/disables all interrupts except interrupt caused by Reset or software BRK instruction. All interrupts are disabled when cleared to “0”. This flag immediately becomes “0” when an interrupt is served. It is set by the EI instruction and cleared by the DI instruction.

[Half carry flag H]

After operation, this is set when there is a carry from bit 3 of ALU or there is no borrow from bit 4 of ALU. This bit can not be set or cleared except CLRV instruction with Overflow flag (V).

[Break flag B]

This flag is set by software BRK instruction to distinguish BRK from TCALL instruction with the same vector address.

[Overflow flag V]

This flag is set to “1” when an overflow occurs as the result of an arithmetic operation involving signs. An overflow occurs when the result of an addition or subtraction exceeds +127(7F$_H$) or -128(80$_H$). The CLRV instruction clears the overflow flag. There is no set instruction. When the BIT instruction is executed, bit 6 of memory is copied to this flag.

[Negative flag N]

This flag is set to match the sign bit (bit 7) status of the result of a data or arithmetic operation. When the BIT instruction is executed, bit 7 of memory is copied to this flag.

[Direct page flag G]

This flag assigns RAM page for direct addressing mode. In the direct addressing mode, addressing area is from zero page 00 to FF when this flag is 0. If it is set to 1, addressing area is 1 page. It is set by instruction and cleared by CLRG.

9.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but these devices have 16K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

Figure 9-4, shows a map of Program Memory. After reset, the CPU begins execution from reset vector which is stored in address $FFFE_H$ and $FFFF_H$ as shown in Figure 9-5.

As shown in Figure 9-4, each area is assigned a fixed location in Program Memory. Program Memory area contains the user program.

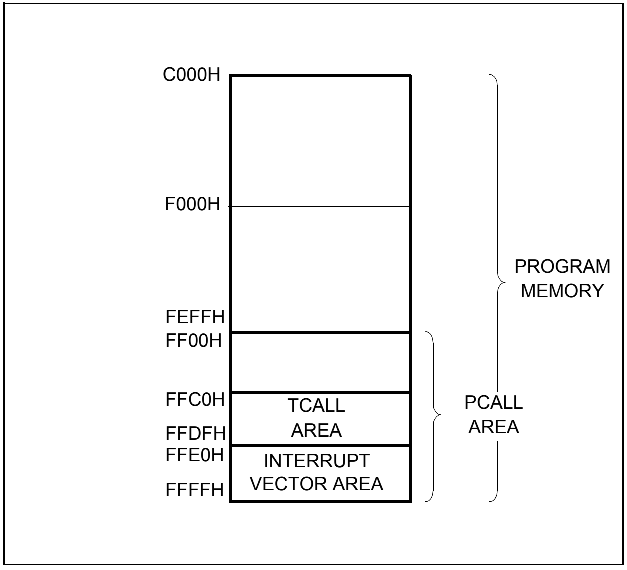


Figure 9-4 Program Memory Map

Page Call (PCALL) area contains subroutine program to reduce program byte length by using 2 bytes PCALL instead of 3 bytes CALL instruction. If it is frequently called, it is more useful to save program byte length.

Table Call (TCALL) causes the CPU to jump to each TCALL address, where it commences the execution of the service routine. The Table Call service area spaces 2-byte for every TCALL: $0FFC0_H$ for TCALL15, $0FFC2_H$ for TCALL14, etc., as shown in Figure 9-6.

Example: Usage of TCALL

```
LDA      #5
         TCALL  0FH          ;1BYTE INSTRUCTION
         :                   ;INSTEAD OF 3 BYTES
         :                   ;NORMAL CALL
;
;TABLE CALL ROUTINE
;
FUNC_A:  LDA    LRG0
         RET
;
FUNC_B:  LDA    LRG1       ②  ①
         RET
;
;TABLE CALL ADD. AREA
;
         ORG    0FFC0H          ;TCALL ADDRESS AREA
         DW     FUNC_A
         DW     FUNC_B
```

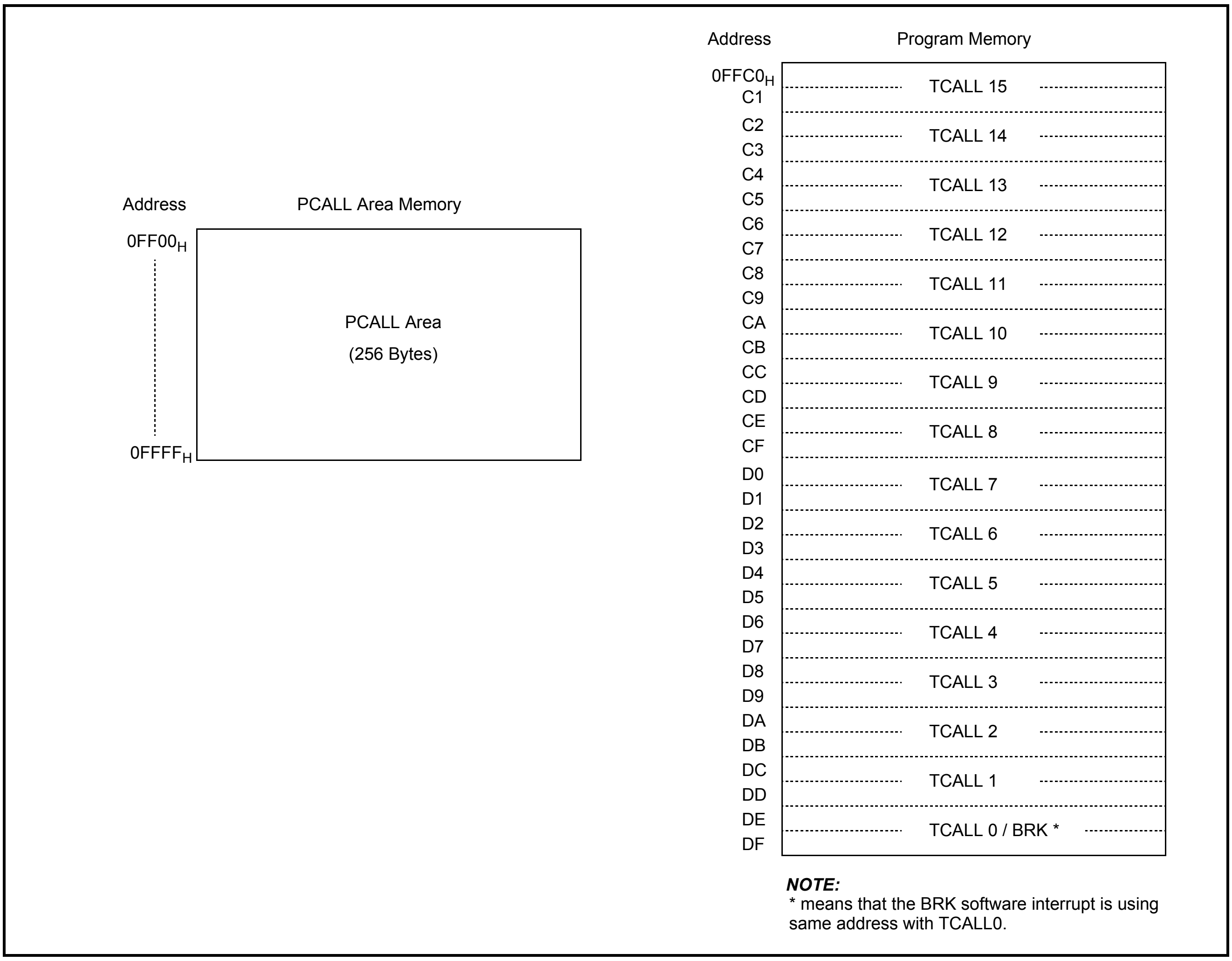
The interrupt causes the CPU to jump to specific location, where it commences the execution of the service routine. The External interrupt 0, for example, is assigned to location $0FFFA_H$. The interrupt service locations spaces 2-byte interval: $0FFF8_H$ and $0FFF9_H$ for External Interrupt 1, $0FFFA_H$ and $0FFFB_H$ for External Interrupt 0, etc.

As for the area from $0FF00_H$ to $0FFFF_H$, if any area of them is not going to be used, its service location is available as general purpose Program Memory.

Address	Vector Area Memory
$0FFE0_H$	-
E2	-
E4	ADC Interrupt Vector Area
E6	RC WT Interrupt Vector Area
E8	BIT Interrupt Vector Area
EA	WDT Interrupt Vector Area
EC	-
EE	Timer/Counter 2 Interrupt Vector Area
F0	Timer/Counter 1 Interrupt Vector Area
F2	Timer/Counter 0 Interrupt Vector Area
F4	-
F6	EXT2 Interrupt Vector Area
F8	EXT1 Interrupt Vector Area
FA	KEY SCAN Interrupt Vector Area
FC	-
FE	RESET Vector Area

NOTE:
"-" means reserved area.

Figure 9-5 Interrupt Vector Area

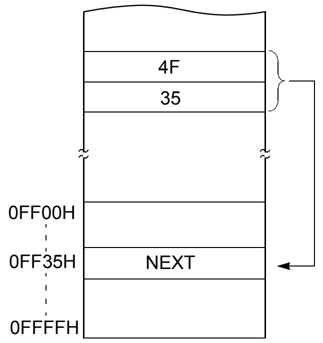


NOTE:
* means that the BRK software interrupt is using same address with TCALL0.

Figure 9-6 PCALL and TCALL Memory Area

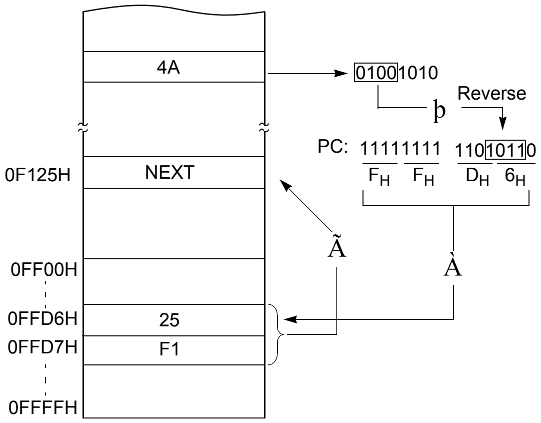
PCALL→rel

```
4F35     PCALL  35H
```



TCALL→n

```
4A       TCALL  4
```



Example: The usage software example of Vector address and the initialize part.

```
        ORG    0FFE0H

        DW     NOT_USED; (0FFEO)
        DW     NOT_USED; (0FFE2)
        DW     ADC_INT; (0FFE4) A/D Interface
        DW     RC_WT_INT; (0FFE6) RC WAKE UP Timer
        DW     BIT_INT; (0FFE8) BIT Timer
        DW     WDT_INT; (0FFEA) WDT
        DW     NOT_USED; (0FFEC)
        DW     TMR2_INT; (0FFEE) Timer-2
        DW     TMR1_INT; (0FFF0) Timer-1
        DW     TMR0_INT; (0FFF2) Timer-0
        DW     NOT_USED; (0FFF4)
        DW     EXT2_INT; (0FFF6) External2
        DW     EXT1_INT; (0FFF8) External1
        DW     KEY_SCAN; (0FFFA) Key Scan
        DW     NOT_USED; (0FFFC)
        DW     RESET; (0FFFE) Reset


        ORG    0F000H

;********************************************
;           MAIN     PROGRAM  *
;*******************************************
;
RESET:  DI      ;Disable All Interrupts
        LDX    #0
RAM_CLR:LDA    #0;RAM Clear(!0000H->!00BFH)
        STA    {X}+
        CMPX   #0C0H
        BNE    RAM_CLR
;
        LDX    #07FH;Stack Pointer Initialize
        TXSP
;
        CALL   INITIAL;
;
        LDM    R1, #0;Normal Port A
        LDM    R1DD,#1000_0010B;Normal Port Direction
        LDM    R2, #0;Normal Port B
        LDM    R2DD,#1000_0010B;Normal Port Direction
        :
        :
        :
        :
```

9.3 Data Memory

Figure 9-7 shows the internal Data Memory space available. Data Memory is divided into two groups, a user RAM (including Stack) and control registers.

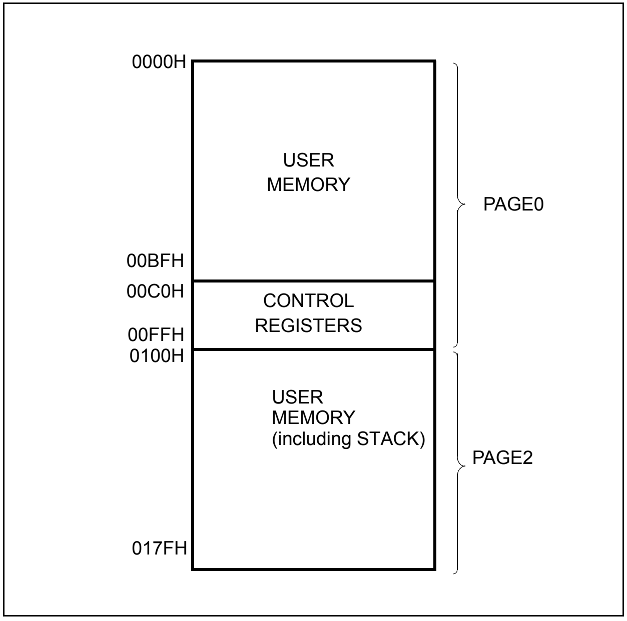


Figure 9-7 Data Memory Map

User Memory

The HMS87C5216 has 330 × 8 bits for the user memory (RAM).

Control Registers

The control registers are used by the CPU and Peripheral function blocks for controlling the desired operation of the device. Therefore these registers contain control and status bits for the interrupt system, the timer/ counters, analog to digital converters and I/O ports. The control registers are in address range of 0C0$_H$ to 0FF$_H$.

Note that unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

More detailed informations of each register are explained in each peripheral section.

***Note:** Write only registers can not be accessed by bit manipulation instruction. Do not use read-modify-write instruction. Use byte manipulation instruction.*

Example; To write at CKCTLR

```
        LDM    CKCTLR,#09H ;Divide ratio ÷16
```

***Note:** Several names are given at same address. Refer to-*

Address	Symbol	R/W	RESET Value	Addressing mode
Address	**Symbol**	**R/W**	**RESET Value**	**Addressing mode**
0C0H	R0	R/W	Undefined	byte, bit[1]
0C1H	R0DR	W	0000_0000	byte[2]
0C2H	R1	R/W	Undefined	byte, bit
0C3H	R1DR	W	0000_0000	byte
0C4H	R2	R/W	Undefined	byte, bit
0C5H	R2DR	W	0000_0000	byte
0C6H	TMR1	W	0000_0000	byte
0C7H	CKCTLR	W	--11_0111	byte
0C7H	BITR	R	0000_0000	byte
0C8H	WDTR	W	-000_1111	byte
0C9H	PSR	W	--00_0000	byte
0CAH	RCWTR	W	----_1000	byte,bit
0CBH	IESR	W	--00_00--	byte,bit
0CCH	IENL	R/W	-000_-0--	byte,bit
0CDH	IRQL	R/W	-000_-0--	byte,bit
0CEH	IENH	R/W	000-_000-	byte,bit
0CFH	IRQH	R/W	000-_000-	byte,bot
0D0H	TM0	R/W	0000_0000	byte, bit
0D1H	TM1	R/W	0000_-000	byte, bit
0D2H	TM2	R/W	---0_0000	byte, bit
0D3H	T0HMD	W	Undefined	byte
0D4H	T0HLD	W	Undefined	byte
0D5H	T0MC	R	0000_0000	byte
0D5H	T0LMD	W	Undefined	byte
0D6H	T0LC	R	0000_0000	byte
0D6H	T0LLD	W	Undefined	byte
0D7H	T1HD	W	Undefined	byte
0D8H	T1C	R	0000_0000	byte
0D8H	T1LD	W	Undefined	byte
0D9H	T2C	R	0000_0000	byte
0D9H	T2D	W	Undefined	byte
0DAH	TM01	R/W	0000_0000	byte
0DCH	KSR0	W	00000_000	byte
0DDH	KSR1	W	0000_0000	byte
0DEH	R10D	W	0000_0000	byte
0DFH	R2OD	W	0000_0000	byte,bit
0E0H	R3OD	W	0000_0000	byte
0E1H	R4OD	W	--00_0000	byte
0E4H	R0OD	W	0000_0000	byte
0E5H	R3	R/W	Undefined	byte,bit
0E6H	R3DR	W	0000_0000	byte
0E7H	R4	R/W	Undefined	byte,bit
0E8H	R4DR	W	--00_0000	byte
0EEH	TMR2	R	0000_0000	byte
0EFH	LVDR	R	---_-00-	byte

Table 9-1 Control Registers

0F0H	SMR	W	----_---0	byte
0F4H	ADMR	R/W	-000_0001	byte, bit
0F5H	ADDR	R	Undefined	byte
0F6H	KRL0	W	0000_0000	byte
0F7H	KRL1	W	0000_0000	byte
0F8H	R0PU	W	0000_0000	byte
0F9H	R1PU	W	0000_0000	byte
0FAH	R2PU	W	0000_0000	byte
0FBH	R3PU	W	0000_0000	byte
0FCH	R4PU	W	--00_0000	byte

Table 9-1 Control Registers

1. “byte, bit” means that register can be addressed by not only bit but byte manipulation instruction.
2. “byte” means that register can be addressed by only byte manipulation instruction. On the other hand, do not use any read-modify-write instruction such as bit manipulation for clearing bit.

below table.

Stack Area

The stack provides the area where the return address is saved before a jump is performed during the processing routine at the execution of a subroutine call instruction or the acceptance of an interrupt.

When returning from the processing routine, executing the subroutine return instruction [RET] restores the contents of the program counter from the stack; executing the interrupt return instruction [RETI] restores the contents of the program counter and flags.

The save/restore locations in the stack are determined by the stack pointed (SP). The SP is automatically decreased after the saving, and increased before the restoring. This means the value of the SP indicates the stack location number for the next save.

9.4 Addressing Mode

The HMS87C5216 and GMS81C1408 uses six addressing modes;

- **Register addressing**
- **Immediate addressing**
- **Direct page addressing**
- **Absolute addressing**
- **Indexed addressing**
- **Register-indirect addressing**

(1) Register Addressing

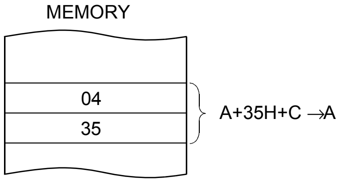
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing →#imm

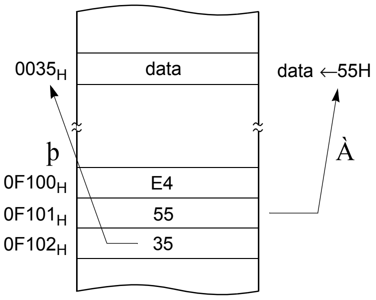
In this mode, second byte (operand) is accessed as a data immediately.

Example:

```
0435     ADC    #35H
```



```
E45535   LDM    35H,#55H
```

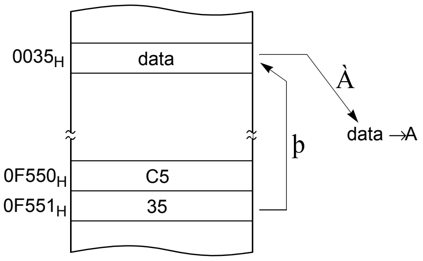


(3) Direct Page Addressing →dp

In this mode, a address is specified within direct page.

Example;

```
C535     LDA    35H          ;A ←RAM[35H]
```



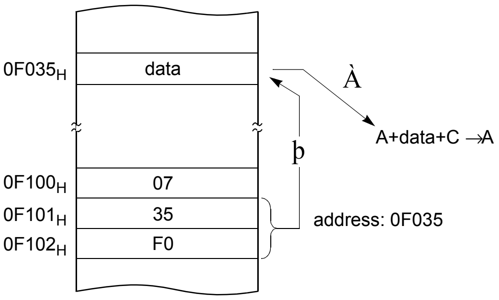
(4) Absolute Addressing →!abs

Absolute addressing sets corresponding memory data to Data, i.e. second byte(Operand I) of command becomes lower level address and third byte (Operand II) becomes upper level address. With 3 bytes command, it is possible to access to whole memory area.

ADC, AND, CMP, CMPX, CMPY, EOR, LDA, LDX, LDY, OR, SBC, STA, STX, STY

Example;

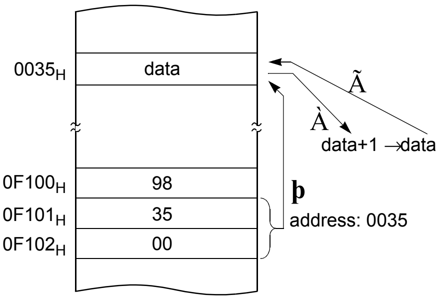
```
0735F0   ADC    !0F035H      ;A ←ROM[0F035H]
```



The operation within data memory (RAM)
ASL, BIT, DEC, INC, LSR, ROL, ROR

Example; Addressing accesses the address 0135_H.

```
983500   INC    !0035H      ;A ←RAM[035H]
```



(5) Indexed Addressing

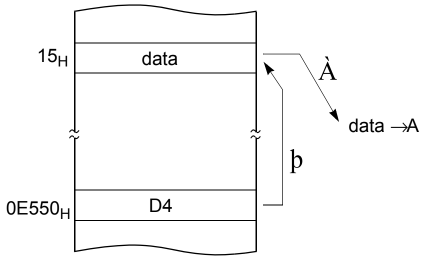
X indexed direct page (no offset) →{X}

In this mode, a address is specified by the X register.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA, XMA

Example; $X=15_H$

```
D4       LDA    {X}         ;ACC←RAM[X].
```



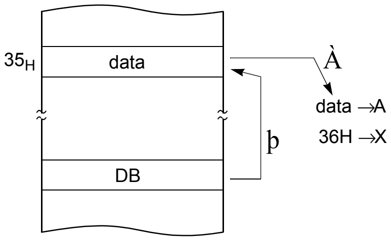
X indexed direct page, auto increment→{X}+

In this mode, a address is specified within direct page by the X register and the content of X is increased by 1.

LDA, STA

Example; $X=35_H$

```
DB       LDA    {X}+
```



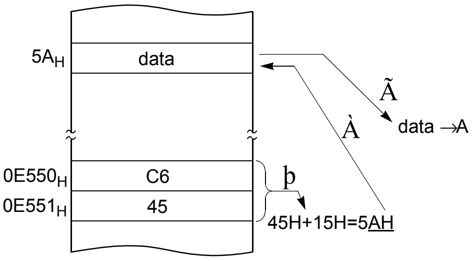
X indexed direct page (8 bit offset) →dp+X

This address value is the second byte (Operand) of command plus the data of X-register. And it assigns the memory in Direct page.

ADC, AND, CMP, EOR, LDA, LDY, OR, SBC, STA STY, XMA, ASL, DEC, INC, LSR, ROL, ROR

Example; $X=015_H$

```
C645     LDA    45H+X
```



Y indexed direct page (8 bit offset) →dp+Y

This address value is the second byte (Operand) of command plus the data of Y-register, which assigns Memory in Direct page.

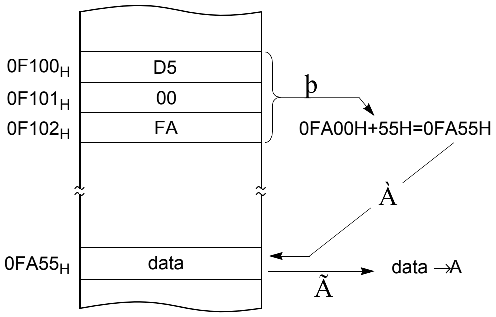
This is same with above (2). Use Y register instead of X.

Y indexed absolute →abs+Y

Sets the value of 16-bit absolute address plus Y-register data as Memory. This addressing mode can specify memory in whole area.

Example; Y=55_H

```
D500FA   LDA    !0FA00H+Y
```



(6) Indirect Addressing

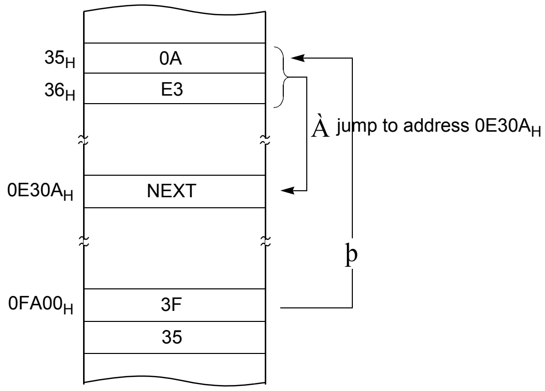
Direct page indirect →[dp]

Assigns data address to use for accomplishing command which sets memory data(or pair memory) by Operand.
Also index can be used with Index register X,Y.

JMP, CALL

Example;

```
3F35     JMP    [35H]
```



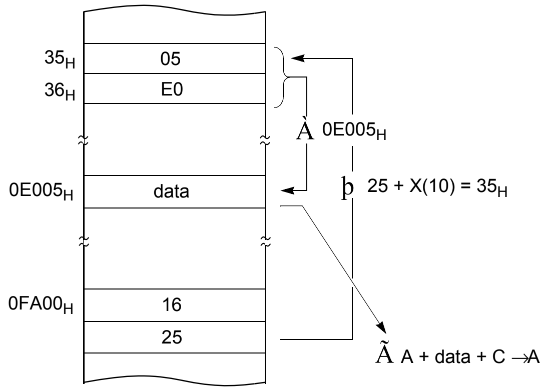
X indexed indirect →[dp+X]

Processes memory data as Data, assigned by 16-bit pair memory which is determined by pair data [dp+X+1][dp+X] Operand plus X-register data in Direct page.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example; X=10_H

```
1625     ADC    [25H+X]
```



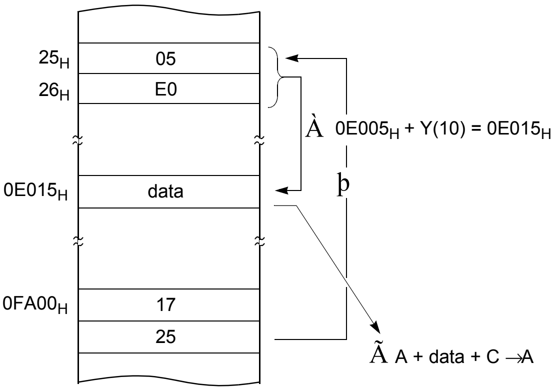
Y indexed indirect →[dp]+Y

Processes memory data as Data, assigned by the data [dp+1][dp] of 16-bit pair memory paired by Operand in Direct page plus Y-register data.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example; Y=10_H

```
1725     ADC    [25H]+Y
```



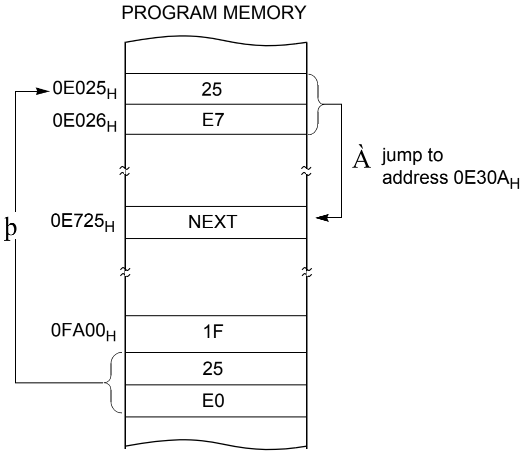
Absolute indirect →[!abs]

The program jumps to address specified by 16-bit absolute address.

JMP

Example;

```
1F25E0   JMP    [!0C025H]
```



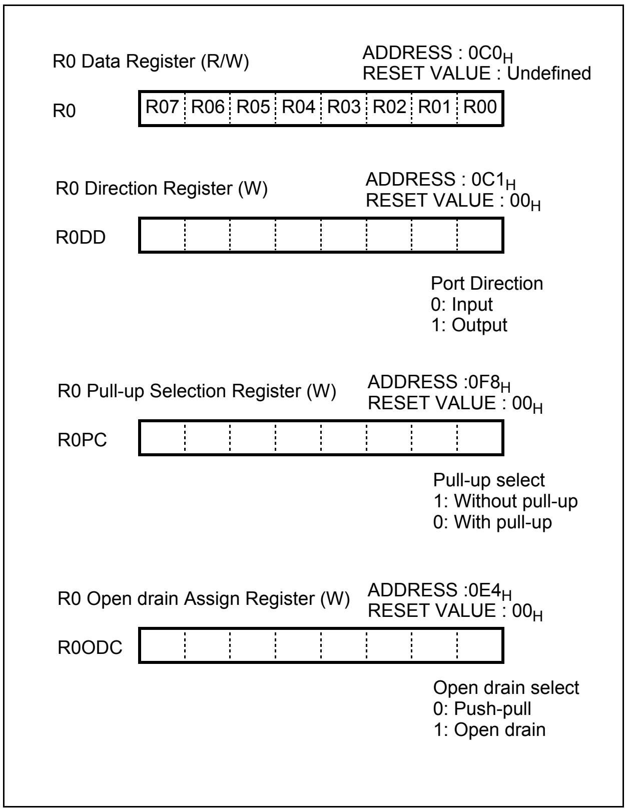
10. I/O PORTS

The GMS87C5216 has 38 I/O ports which are PORT0(8 I/O), PORT1 (8 I/O), PORT2 (8 I/O), PORT3 (8 I/O), PORT4 (6 I/O). Pull-up resistor of each port can be selectable by program. Each port contains data direction register which controls I/O and data register which stores port data

10.1 R0 Ports

R0 is an 8-bit CMOS bidirectional I/O port (address $0C0_H$). Each I/O pin can independently used as an input or an output through the R0DD register (address $0C1_H$).

R0 has internal pull-ups that is independently connected or disconnected by R0PC. The control registers for R0 are shown below.



(1) R0 I/O Data Direction Register (R0DD)

R0 I/O Data Direction Register (R0DD) is 8-bit register, and can assign input state or output state to each bit. If R0DD is ``1``, port R0 is in the output state, and if ``0``, it is in the input state. R0DD is write-only register. Since R0DD is initialized as ``00 h`` in reset state, the whole port R0 becomes input state.

(2) R0 Data Register (R0)

R0 data register (R0) is 8-bit register to store data of port R0. When set as the output state by R0DD, and data is written in R0, data is outputted into R0 pin. When set as the input state, input state of pin is read. The initial value of R0 is unknown in reset state.

(3) R0 Open drain Assign Register (R0ODC)

R0 Open Drain Assign Register (R0ODC) is 8bit register, and can assign R0 port as open drain output port each bit, if corresponding port is selected as output. If R0ODC is selected as ``1``, port R0 is open drain output, and if selected as ``0``, it is push-pull output. R0ODC is write-only register and initialized as ``00 h`` in reset state.

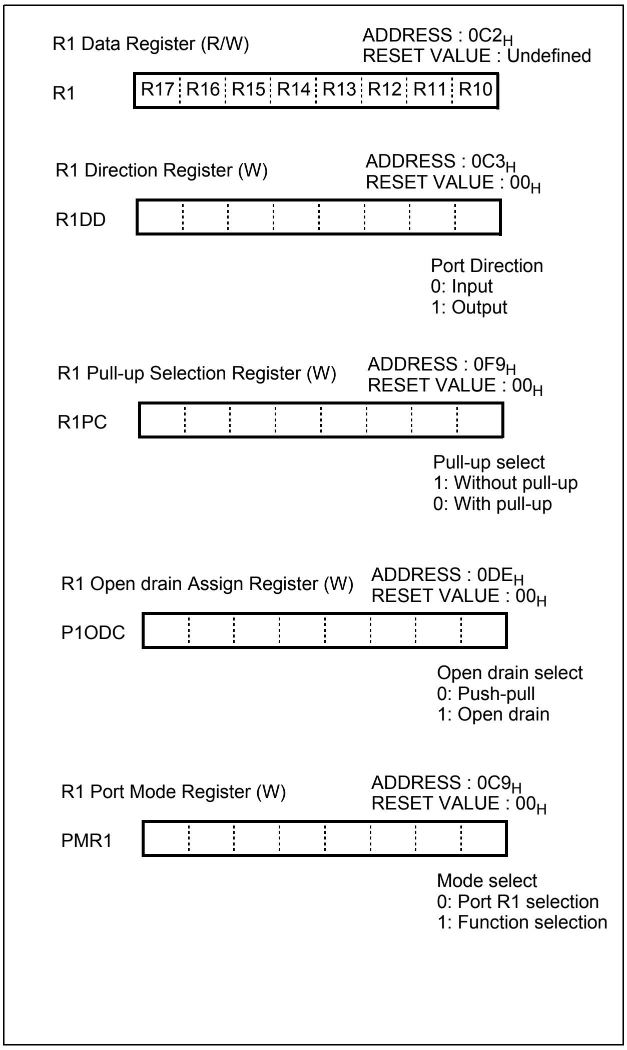
(4) R0 Pull-up Resistor Control Register (R0PC)

R0 pull-up resistor control register (R0PC) is 8-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If R0PC is selected as ``1``, pull-up ia disabled and if selected as ``0``, it is enabled. R0PC is write-only register and initialized as ``00 h`` in reset state. The pull-up is automatically disabled, if corresponding port is selected as output.

10.2 R1 Ports

R1 is an 8-bit CMOS bidirectional I/O port (address $0C2_H$). Each I/O pin can independently used as an input or an output through the R1DD register (address $0C3_H$).

R1 has internal pull-ups that is independently connected or disconnected by register R1PC. The control registers for R1 are shown below.



(1) R1 I/O Data Direction Register (R1DD)

R1 I/O Data Direction Register (R1DD) is 8-bit register, and can assign input state or output state to each bit. If R1DD is ``1``, port R1 is in the output state, and if ``0``, it is in the input state. R1DD is write-only register. Since R1DD is initialized as ``00 h`` in reset state, the whole port R1 becomes input state.

(2) R1 Data Register (R1)

R1 data register (R1) is 8-bit register to store data of port R1. When set as the output state by R1DD, and data is written in R1, data is outputted into R1 pin. When set as the input state, input state of pin is read. The initial value of R1 is unknown in reset state.

(3) R1 Mode Register (PMR1)

R1 Port Mode Register (PMR1) is 8-bit register, and can assign the selection mode for each bit. When set as ``0``, corresponding bit of PMR1 acts as port R1 selection mode, and when set as ``1``, it becomes function selection mode.

PMR1 is write-only register and initialized as ``00 h`` in reset state. Therefore, becomes Port selection mode. Port R1 can be I/O port by manipulating each R1DD bit, if corresponding PMR1 bit is selected as ``0``.

Pin Name	PMR1	Selection Mode	Remarks
-			
-			
EC0	0	R25(I/O)	-
	1	EC0(I)	EVENT COUNT0
T2	0	R24(I/O)	-
	1	T2(O)	TIMER2
T1	0	R13 (I/O)	-
	1	T1(O)	TIMER1
T0	0	R12 (I/O)	-
	1	T0(O)	TIMER0
INT2	0	R11 (I/O)	-
	1	INT2(I)	EXT INT2
INT1	0	R10(I/O)	-
	1	INT1(I)	EXT INT1

Table 10-1 Selection mode of PMR1

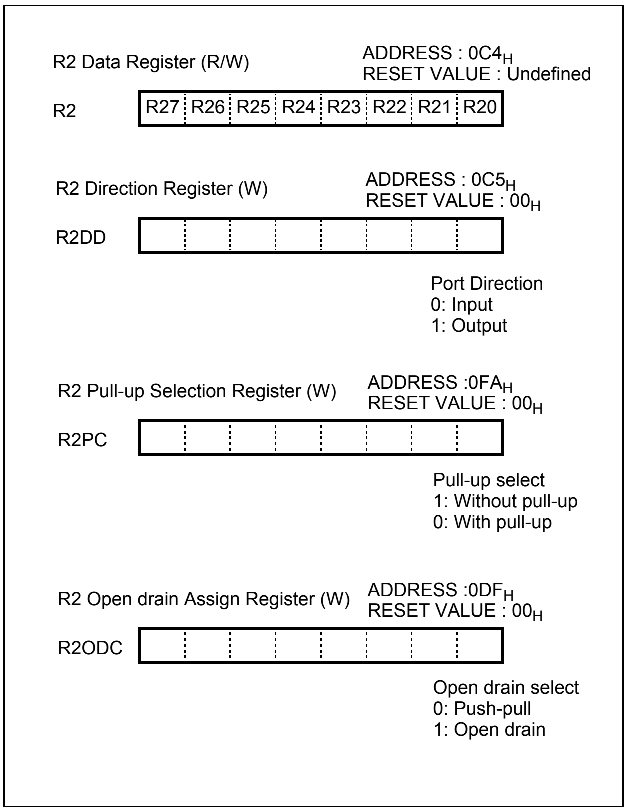
(4) R1 Pull-up Resistor Control Register (R1PC)

R1 pull-up resistor control register (R1PC) is 8-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If R1PC is selected as ``1``, pull-up ia disabled and if selected as ``0``, it is enabled. R1PC is write-only register and initialized as ``00 h`` in reset state. The pull-up is automatically disabled, if corresponding port is selected as output.

10.3 R2 Port

R2 is an 8-bit CMOS bidirectional I/O port (address 0C4$_H$). Each I/O pin can independently used as an input or an output through the R2DD register (address 0C5$_H$).

R2 has internal pujll-ups that is independently connected or disconnected by R2PC (address 0FA$_H$). The control registers for R2 are shown as below.



(1) R2 I/O Data Direction Register (R2DD)

R2 I/O Data Direction Register (R2DD) is 8-bit register, and can assign input state or output state to each bit. If R2DD is ``1``, port R2 is in the output state, and if ``0``, it is in the input state. R2DD is write-only register. Since R2DD is initialized as ``00 h`` in reset state, the whole port R2 becomes input state.

(2) R2 Data Register (R2)

R2 data register (R2) is 8-bit register to store data of port R2. When set as the output state by R2DD, and data is written in R2, data is outputted into R2 pin. When set as the input state, input state of pin is read. The initial value of R2 is unknown in reset state.

(3) R2 Open drain Assign Register (R2ODC)

R2 Open Drain Assign Register (R2ODC) is 8bit register, and can assign R2 port as open drain output port each bit, if corresponding port is selected as output. If R2ODC is selected as ``1``, port R2 is open drain output, and if selected as ``0``, it is push-pull output. R2ODC is write-only register and initialized as ``00 h`` in reset state.

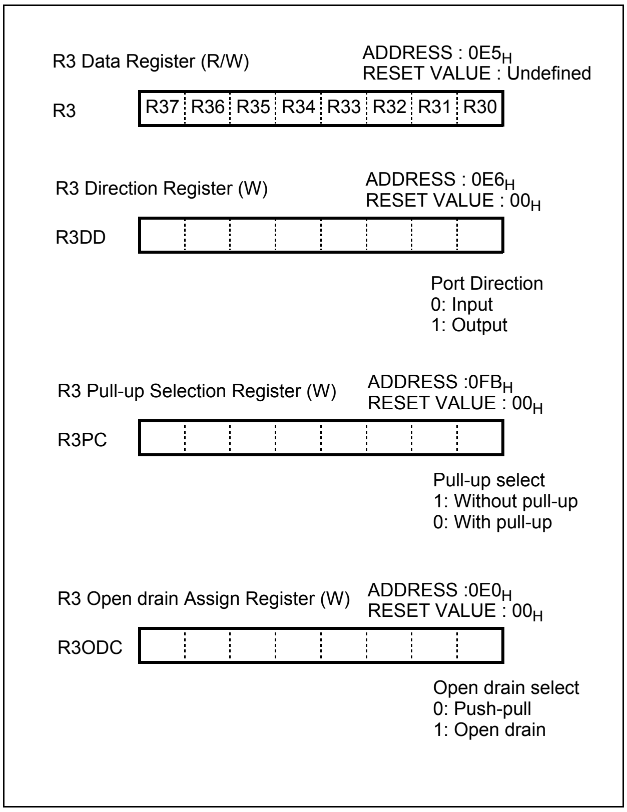
(4) R2 Pull-up Resistor Control Register (R2PC)

R2 pull-up resistor control register (R2PC) is 8-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If R2PC is selected as ``1``, pull-up ia disabled and if selected as ``0``, it is enabled. R2PC is write-only register and initialized as ``00 h`` in reset state. The pull-up is automatically disabled, if corresponding port is selected as output.

R3 Port

R3 is an 8-bit CMOS bidirectional I/O port (address $0E5_H$). Each I/O pin can independently used as an input or an output through the R3DD register (address $0E6_H$).

R3 has internal pull-ups that is independently connected or disconnected by R3PC (address $0FB_H$). The control registers for R3 are shown as below.



(1) R3 I/O Data Direction Register (R3DD)

R3 I/O Data Direction Register (R3DD) is 8-bit register, and can assign input state or output state to each bit. If R3DD is ``1``, port R3 is in the output state, and if ``0``, it is in the input state. R3DD is write-only register. Since R3DD is initialized as ``00 h`` in reset state, the whole port R3 becomes input state.

(2) R3 Data Register (R3)

R3 data register (R3) is 8-bit register to store data of port R3. When set as the output state by R3DD, and data is written in R3, data is outputted into R3 pin. When set as the input state, input state of pin is read. The initial value of R3 is unknown in reset state.

(3) R3 Open drain Assign Register (R3ODC)

R3 Open Drain Assign Register (R3ODC) is 8bit register, and can assign R3 port as open drain output port each bit, if corresponding port is selected as output. If R3ODC is selected as ``1``, port R3 is open drain output, and if selected as ``0``, it is push-pull output. R3ODC is write-only register and initialized as ``00 h`` in reset state.

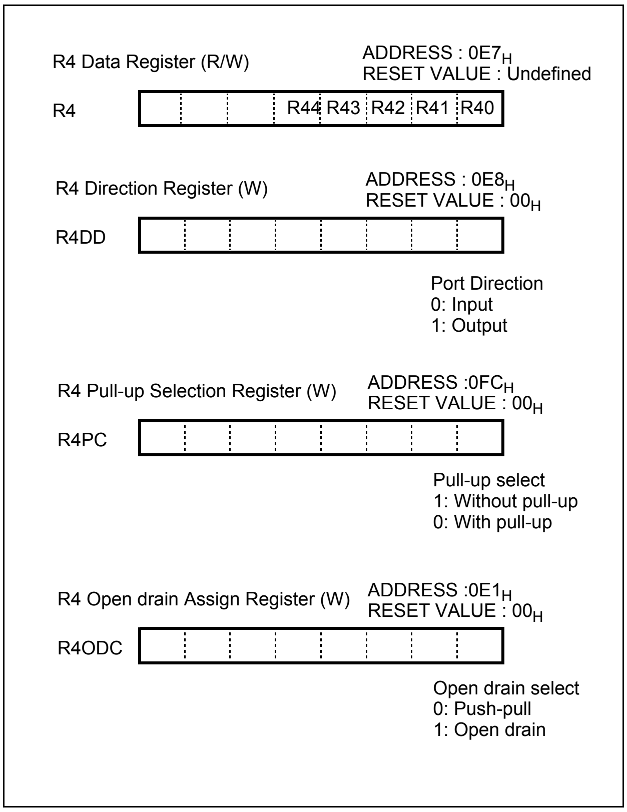
(4) R3 Pull-up Resistor Control Register (R3PC)

R3 pull-up resistor control register (R3PC) is 8-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If R3PC is selected as ``1``, pull-up ia disabled and if selected as ``0``, it is enabled. R3PC is write-only register and initialized as ``00 h`` in reset state. The pull-up is automatically disabled, if corresponding port is selected as output.

R4 Port

R4 is an 1-bit CMOS bidirectional I/O port (address $0E7_H$). Each I/O pin can independently used as an input or an output through the R4DD register (address $0E8_H$).

R3 has internal pull-ups that is independently connected or disconnected by R4PC (address $0FC_H$). The control registers for R4 are shown as below.



R4 Data Register (R/W) ADDRESS : $0E7_H$ RESET VALUE : Undefined

R4 | | | | R44 | R43 | R42 | R41 | R40

R4 Direction Register (W) ADDRESS : $0E8_H$ RESET VALUE : 00_H

R4DD

Port Direction
0: Input
1: Output

R4 Pull-up Selection Register (W) ADDRESS :$0FC_H$ RESET VALUE : 00_H

R4PC

Pull-up select
1: Without pull-up
0: With pull-up

R4 Open drain Assign Register (W) ADDRESS :$0E1_H$ RESET VALUE : 00_H

R4ODC

Open drain select
0: Push-pull
1: Open drain

(1) R4 I/O Data Direction Register (R4DD)

R4 I/O Data Direction Register (R4DD) is 1-bit register, and can assign input state or output state to each bit. If R4DD is ``1``, port R4 is in the output state, and if ``0``, it is in the input state. R4DD is write-only register. Since R4DD is initialized as ``00 h`` in reset state, the whole port R4 becomes input state.

(2) R4 Data Register (R4)

R4 data register (R4) is 1-bit register to store data of port R4. When set as the output state by R4DD, and data is written in R4, data is outputted into R4 pin. When set as the input state, input state of pin is read. The initial value of R4 is unknown in reset state.

(3) R4 Open drain Assign Register (R4ODC)

R4 Open Drain Assign Register (R4ODC) is 1-bit register, and can assign R4 port as open drain output port each bit, if corresponding port is selected as output. If R4ODC is selected as ``1``, port R4 is open drain output, and if selected as ``0``, it is push-pull output. R4ODC is write-only register and initialized as ``00 h`` in reset state.

(4) R4 Pull-up Resistor Control Register (R4PC)

R4 pull-up resistor control register (R4PC) is 1-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If R4PC is selected as ``1``, pull-up ia disabled and if selected as ``0``, it is enabled. R4PC is write-only register and initialized as ``00 h`` in reset state. The pull-up is automatically disabled, if corresponding port is selected as output.

11. CLOCK GENERATOR

Clock generating circuit consists of Clock Pulse Generator (C.P.G), Prescaler, Basic Interval Timer (B.I.T) and Watch Dog Timer. The clock applied to the Xin pin divided by two is used as the internal system clock.

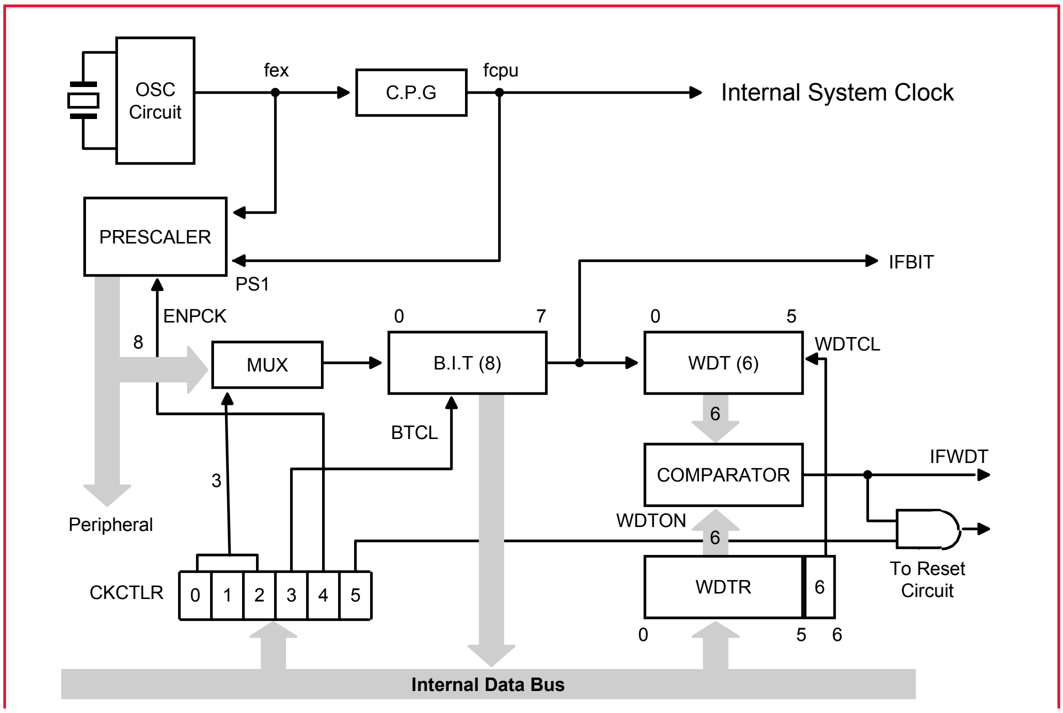


Figure 11-1 Block Diagram of Clock Generator

Prescaler consists of 12-bit binary counter. The clock supplied from oscillation circuit is input to prescaler (fex). The divided output from each bit of prescaler is provided to peripheral hardware.

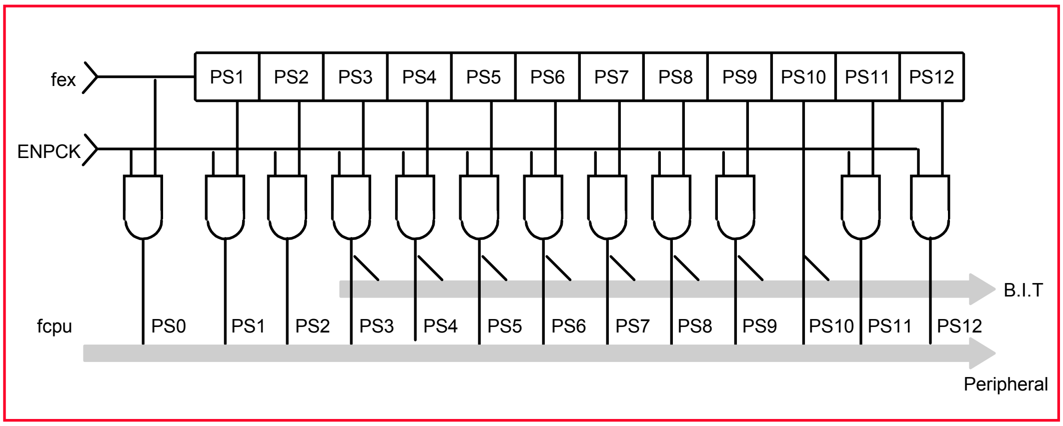

fex
ENPCK
PS1 PS2 PS3 PS4 PS5 PS6 PS7 PS8 PS9 PS10 PS11 PS12
B.I.T
fcpu
PS0 PS1 PS2 PS3 PS4 PS5 PS6 PS7 PS8 PS9 PS10 PS11 PS12
Peripheral

Figure 11-2 Block diagram of Prescaler

fex (MHz)	4 MHz		2 MHz	
	frequency	period	frequency	period
ps 0	4 MHz	250 ns	2 MHz	500 ns
ps 1	2 MHz	500 ns	1 MHz	1 us
ps 2	1 MHz	1 us	500 KHz	2 us
ps 3	500 KHz	2 us	250 KHz	4 us
ps 4	250 KHz	4 us	125 KHz	8 us
ps 5	125 KHz	8 us	62.5 KHz	16 us
ps 6	62.5 KHz	16 us	31.25 KHz	32 us
ps 7	31.25 KHz	32 us	15.63 KHz	64 us
ps 8	15.63 KHz	64 us	7.183 KHz	128 us
ps 9	7.183 KHz	128 us	3.906 KHz	256 us
ps 10	3.906 KHz	256 us	1.953 KHz	512 us
ps 11	1.953 KHz	512 us	0.976 KHz	1024 us
ps 12	0.976 KHz	1024 us	0.488 KHz	2048 us

Table 11-1 ps output period

lock to peripheral hardware can be stopped by bit4 (ENPCK) of CKCTLR Register. ENPCK is set to ``1`` in reset state.

12. Timer

12.1 Basic Interval Timer

The GMS81C5016/24/32 has one 8-bit Basic Interval Timer that is free-run and can not stop. Block diagram is shown in Figure 12-1.

The Basic Interval Timer generates the time base for key scanning, watchdog timer counting, and etc. It also provides a Basic interval timer interrupt (IFBIT). As the count overflow from FF_H to 00_H, this overflow causes the interrupt to be generated.

-8bit binary counter

-Use the bit output of prescaler as input to secure the oscillation stabilization time after power-on

-Secures the oscillation stabilization time in standby mode (stop mode) release

-Contents of B.I.T can be read

-Provides the clock for watch dog timer.

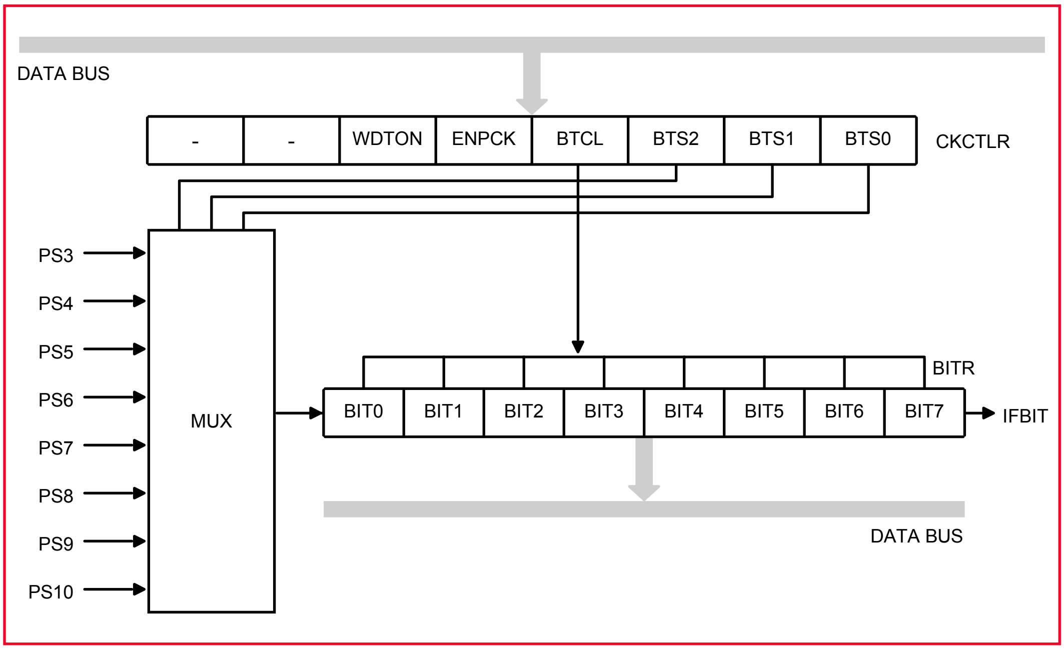
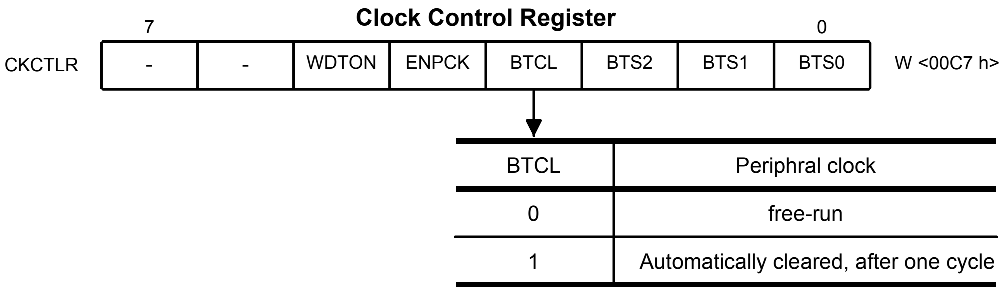


Figure 12-1 Block Diagram of Basic Interval Timer

(1) Control of B.I.T

The Basic Interval Timer is controlled by the clock control register (CKCTLR) shown in Figure 12-2. If bit3(BTCL) of CKCTLR is set to ``1``, B.I.T is cleared, and then, after one machine cycle, BTCL becomes ``0``, and B.I.T starts counting. BTCL is set to ``0`` in reset state.



Clock Control Register

CKCTLR	7							0	
	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7 h>

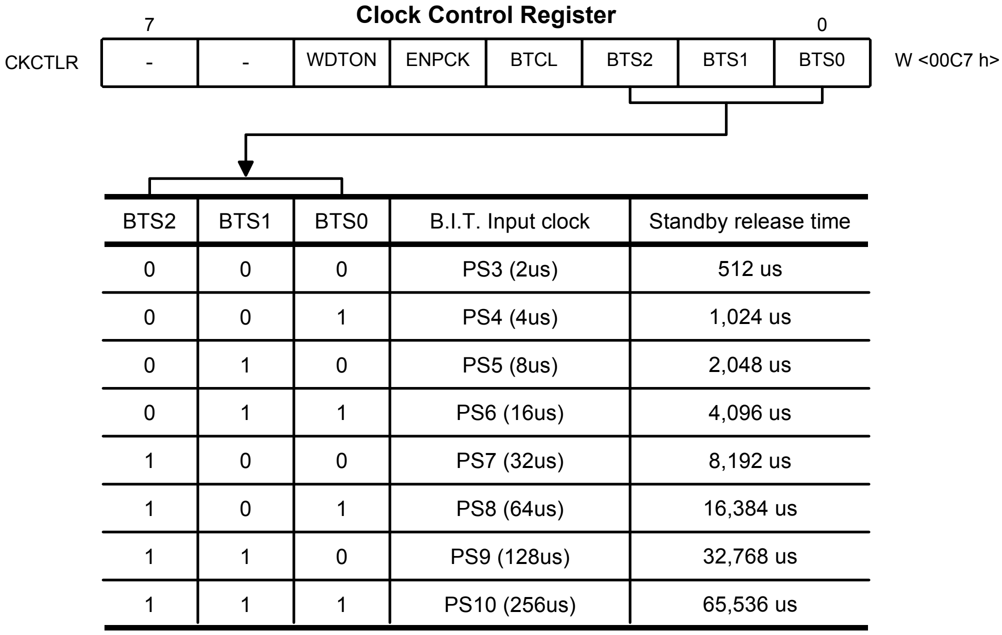
BTCL	Periphral clock
0	free-run
1	Automatically cleared, after one cycle

Figure 12-2 BTCL mode of B.I.T

(2) Input clock selection of B.I.T

The input clock of B.I.T can be selected from the prescaler within a range of 2us to 256us by clock input selection bits (BTS2~BTS0). (at fex = 4MHz). In reset state, or power on reset, BTS2=``1``, BTS1=``1``, BTS0=``1`` to secure the longest oscillation stabilization time. B.I.T can generate the wide range of basic interval time interrupt request (IFBIT) by selecting prescaler output. Interrupt interval can be selected to kinds of interval time as shown in

Figure 12-3.



Clock Control Register

CKCTLR	7							0	
	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7 h>

BTS2	BTS1	BTS0	B.I.T. Input clock	Standby release time
0	0	0	PS3 (2us)	512 us
0	0	1	PS4 (4us)	1,024 us
0	1	0	PS5 (8us)	2,048 us
0	1	1	PS6 (16us)	4,096 us
1	0	0	PS7 (32us)	8,192 us
1	0	1	PS8 (64us)	16,384 us
1	1	0	PS9 (128us)	32,768 us
1	1	1	PS10 (256us)	65,536 us

Figure 12-3 Basic Interval Timer Interrupt Time

(3) Reading Basic Interval Timer

By reading of the Basic Interval Timer Register (BITR), we can read counter value of B.I.T. Because B.I.T can be cleared or read, the spending time up to maximum 65.5ms can be available. B.I.T is read-only register. If B.I.T register is written, then CKCTLR register with same address is written.

Basic Interval Timer Register

	7							0	
BITR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R <00C7 h>

12.2 Timer0, Timer1, Timer2

(1) Timer Operation Mode

Timer consists of 16bit binary counter Timer0 (T0), 8bit binary Timer1 (T1), Timer2 (T2), Timer Data Register, Timer Mode Register (TM01, TM0, TM1, TM2) and control circuit. Timer Data Register Consists of Timer0 High-MSB Data Register (T0HMD), Timer0 High-LSB Data Register (T0HLD), Timer0 Low-MSB Data Register (T0LMD), Timer0 Low-LSB Data Register (T0LLD), Timer1 High Data Register (T1HD), Timer1 Low Data Register (T1LD), Timer2 Data Register (T2DR). Any of the PS0 ~ PS5, PS11 and external event input EC can be selected as clock source for T0. Any of the PS0 ~ PS3, PS7 ~ PS10 can be selected as clock T1. Any of the PS5 ~ PS12 can be selected as clock source for T2.

* Relevant Port Mode Register (PMR1 : 00C9 h) value should be assigned for event counter,

Timer0	**- 16-bit Interval Timer** **- 16-bit Event Counter** **- 16-bit Input Capture** **- 16-bit rectangular-wave output**	**- Single/Modulo-N Mode** **- Timer Output Initial Value Setting** **- Timer0~Timer1 combination Logic Output** **- One Interrupt Generating Every 2nd Counter Overflow**
Timer1	**- 8-bit Interval Timer** **- 8-bit rectangular-wave output**	
Timer2	**- 8-bit Interval Timer** **- 8-bit rectangular-wave output** **- Modulo-N Mode**	

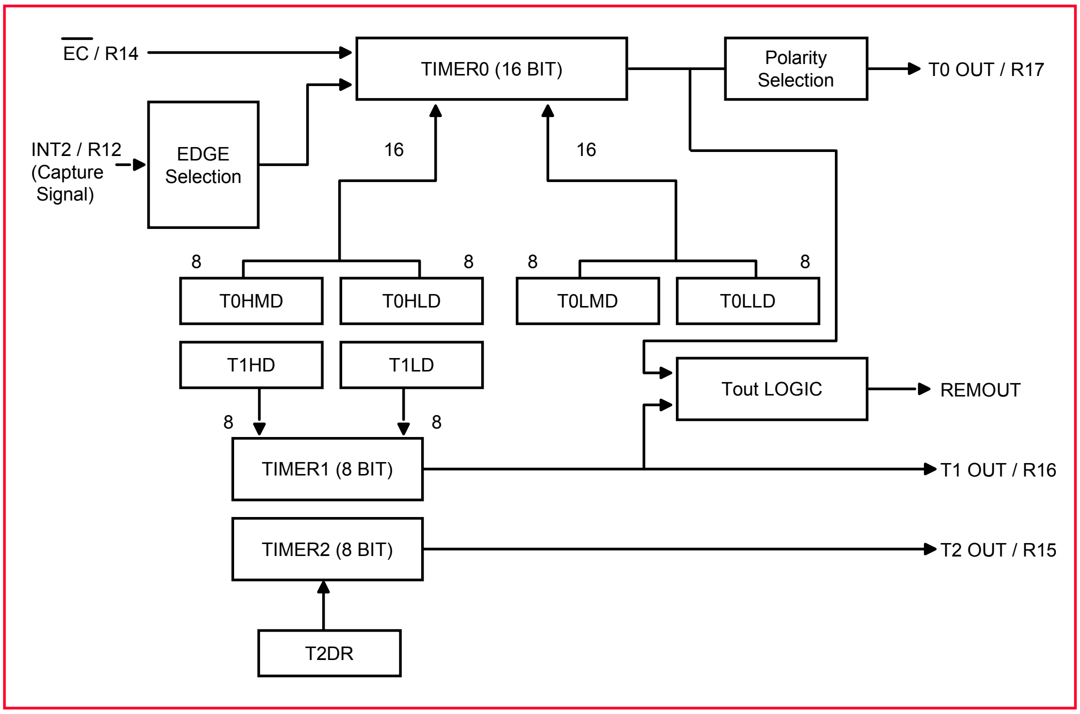


Figure 12-4 Timer / Counter Block diagram

(2) Function of Timer & Counter

fex = 4MHz

16bit Timer (T0)		8bit Timer (T1)		8bit Timer (T2)	
Resolution (CK)	Max. Count	Resolution (CK)	Max. Count	Resolution (CK)	Max. Count
PS0 (0.25 us)	16,384 us	PS0 (0.25 us)	64 us	PS5 (8 us)	2.048 us
PS1 (0. 5 us)	32,768 us	PS1 (0.5 us)	128 us	PS6 (16 us)	4,096 us
PS2 (1 us)	65,536 us	PS2 (1 us)	256 us	PS7 (32 us)	8,192 us
PS3 (2 us)	131,072 us	PS3 (2 us)	512us	PS8 (64 us)	16,384 us
PS4 (4 us)	262,144 us	PS7 (32 us)	8,192 us	PS9 (128 us)	32,768 us
PS5 (8 us)	524,288 us	PS8 (64 us)	16,384 us	PS10 (256 us)	65,536 us
PS11 (512 us)	33,554,432 us	PS9 (128 us)	32,768 us	PS11 (512 us)	131,072 us
EC	-	PS10 (256 us)	65,536 us	PS12 (1,024 us)	262,144 us

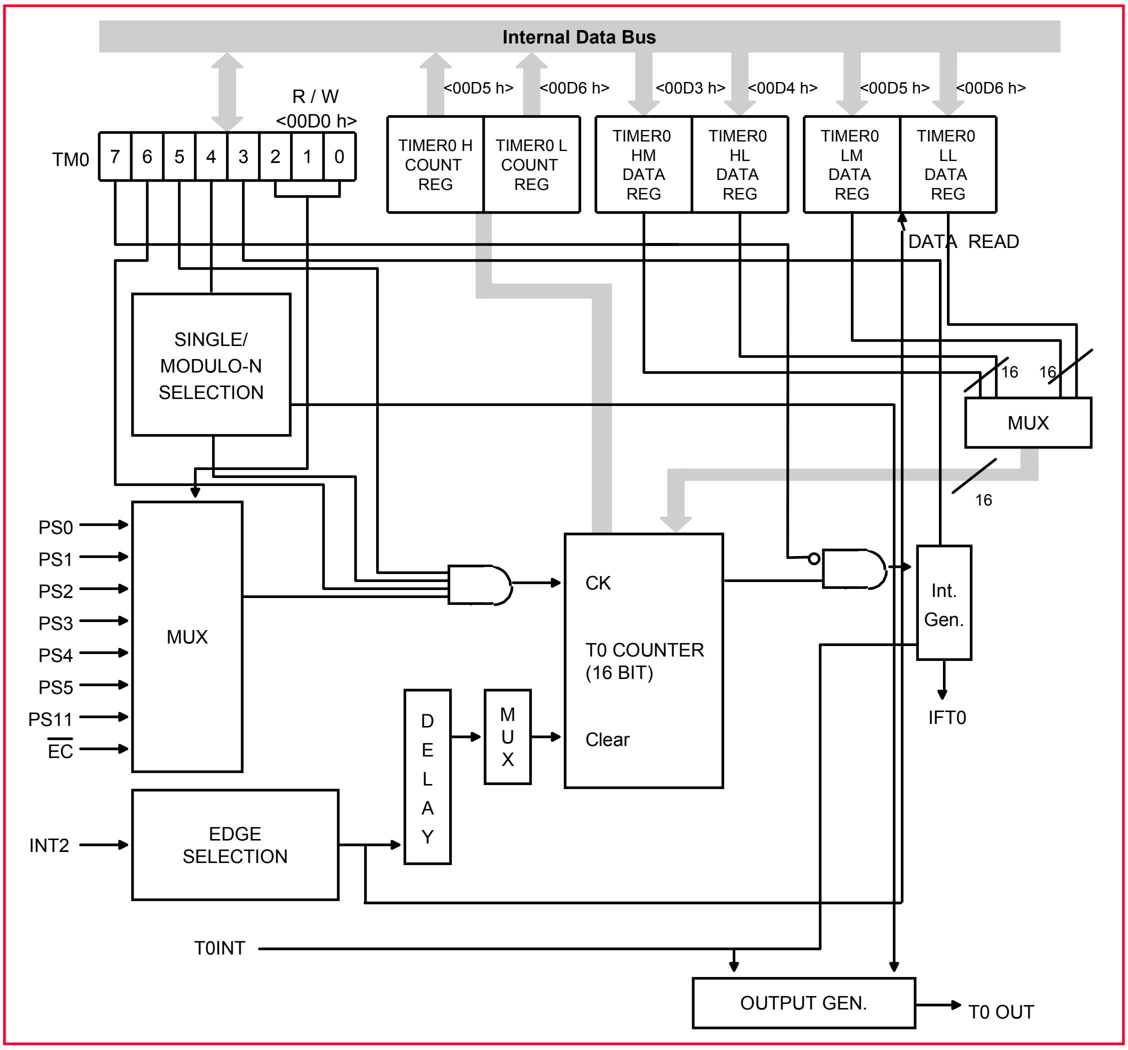


Figure 12-5 Block Diagram of Timer0

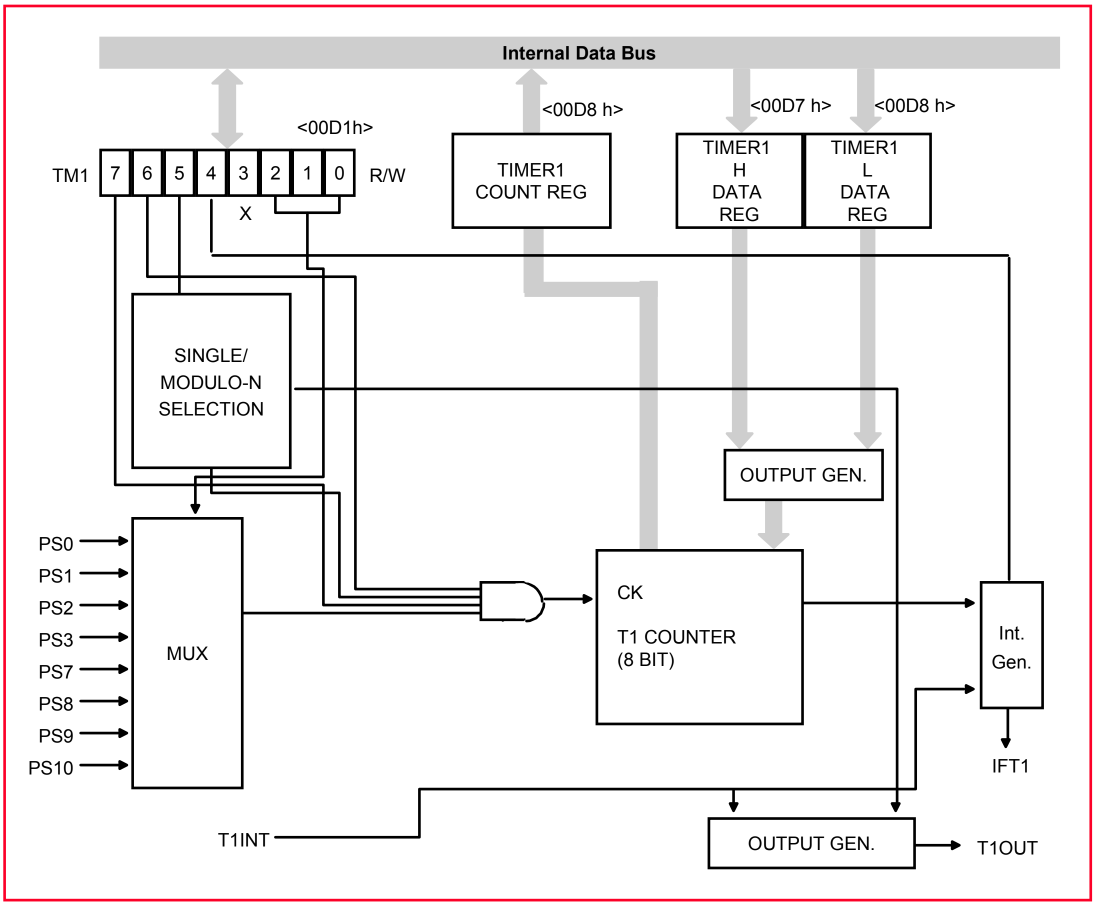


Figure 12-6 Block Diagram of Timer1

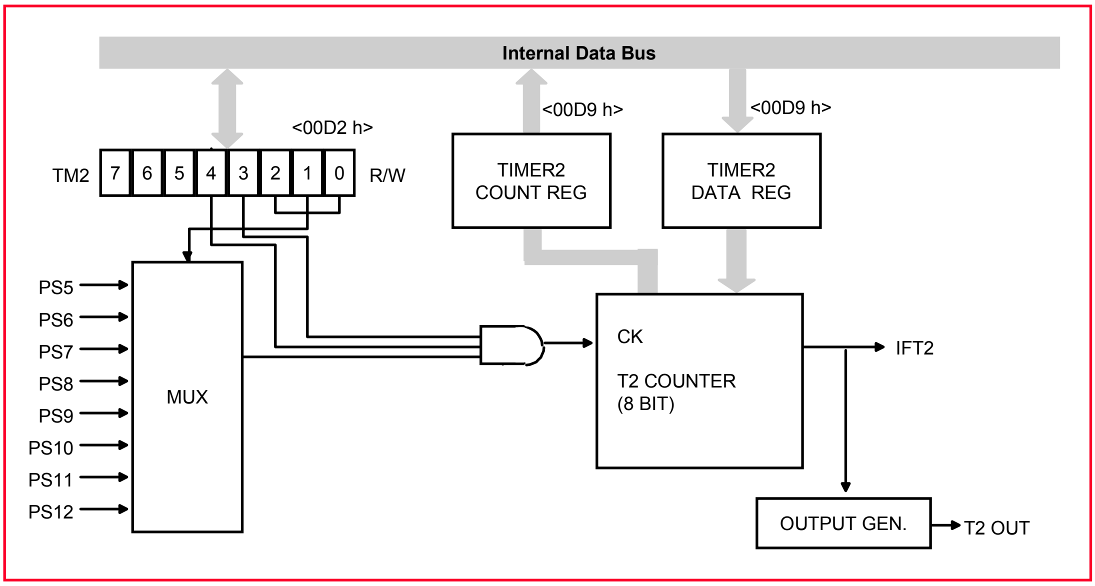
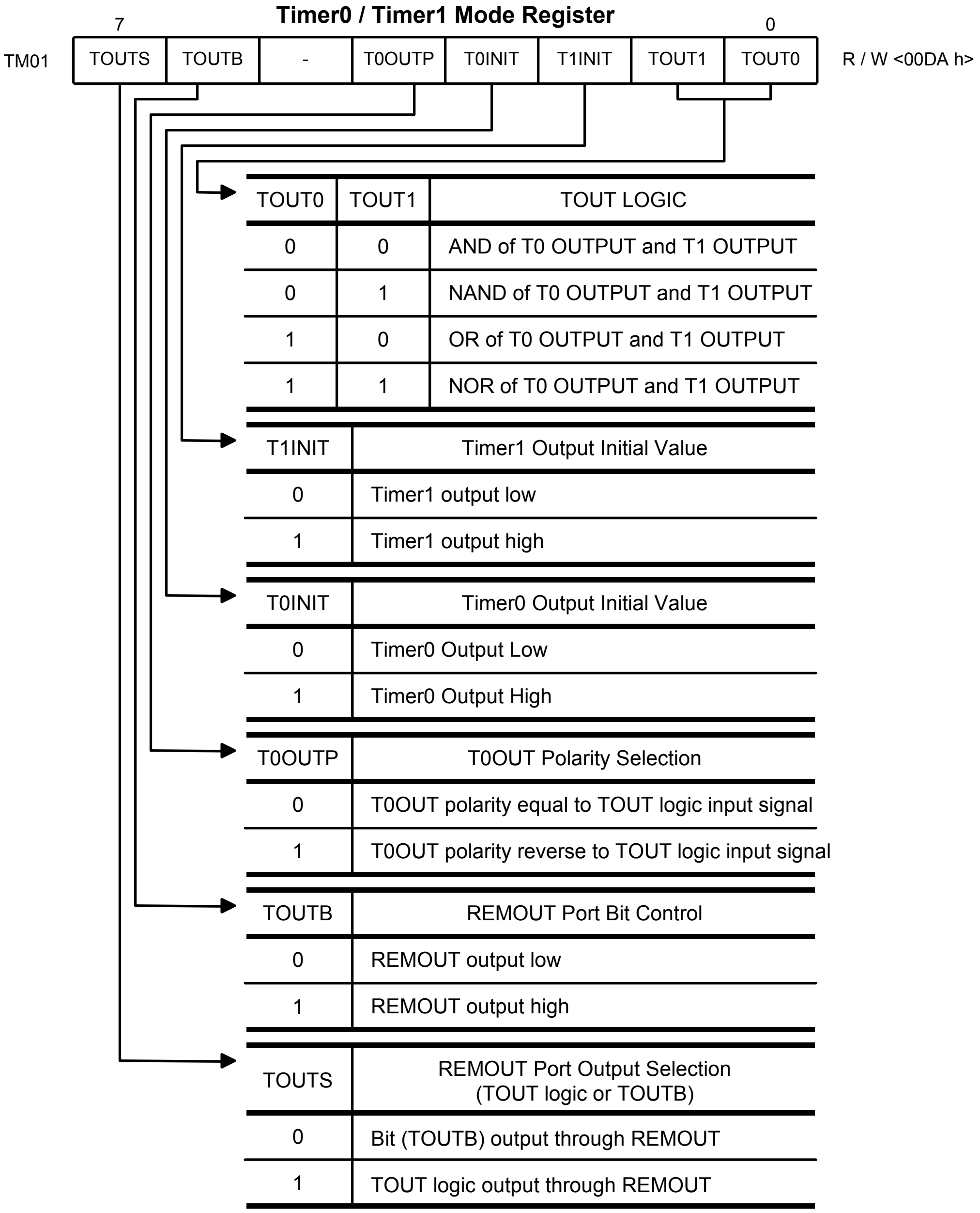


Figure 12-7 Block Diagram of Timer2



Timer0 / Timer1 Mode Register

	7							0	
TM01	TOUTS	TOUTB	-	T0OUTP	T0INIT	T1INIT	TOUT1	TOUT0	R / W <00DA h>

TOUT0	TOUT1	TOUT LOGIC
0	0	AND of T0 OUTPUT and T1 OUTPUT
0	1	NAND of T0 OUTPUT and T1 OUTPUT
1	0	OR of T0 OUTPUT and T1 OUTPUT
1	1	NOR of T0 OUTPUT and T1 OUTPUT

T1INIT	Timer1 Output Initial Value
0	Timer1 output low
1	Timer1 output high

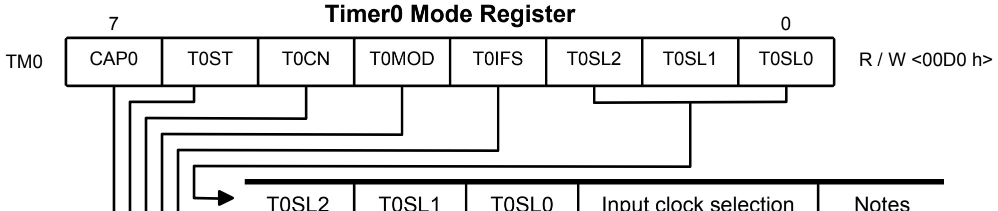
T0INIT	Timer0 Output Initial Value
0	Timer0 Output Low
1	Timer0 Output High

T0OUTP	T0OUT Polarity Selection
0	T0OUT polarity equal to TOUT logic input signal
1	T0OUT polarity reverse to TOUT logic input signal

TOUTB	REMOUT Port Bit Control
0	REMOUT output low
1	REMOUT output high

TOUTS	REMOUT Port Output Selection (TOUT logic or TOUTB)
0	Bit (TOUTB) output through REMOUT
1	TOUT logic output through REMOUT

Figure 12-8 Timer0 / Timer1 Mode Register



T0SL2	T0SL1	T0SL0	Input clock selection	Notes
0	0	0	PS0 (250ns)	
0	0	1	PS1 (500ns)	*
0	1	0	PS2 (1us)	
0	1	1	PS3 (2us)	
1	0	0	PS4 (4us)	
1	0	1	PS5 (8us)	
1	1	0	PS11 (512us)	Event
1	1	1	$\overline{\text{EC}}$	Counter

T0IFS	Timer0 Interrupt Selection
0	Interrupt every counter overflow
1	Interrupt every 2nd counter overflow

T0MOD	Timer0 Single/Modulo-N Selection
0	Modulo-N
1	Single

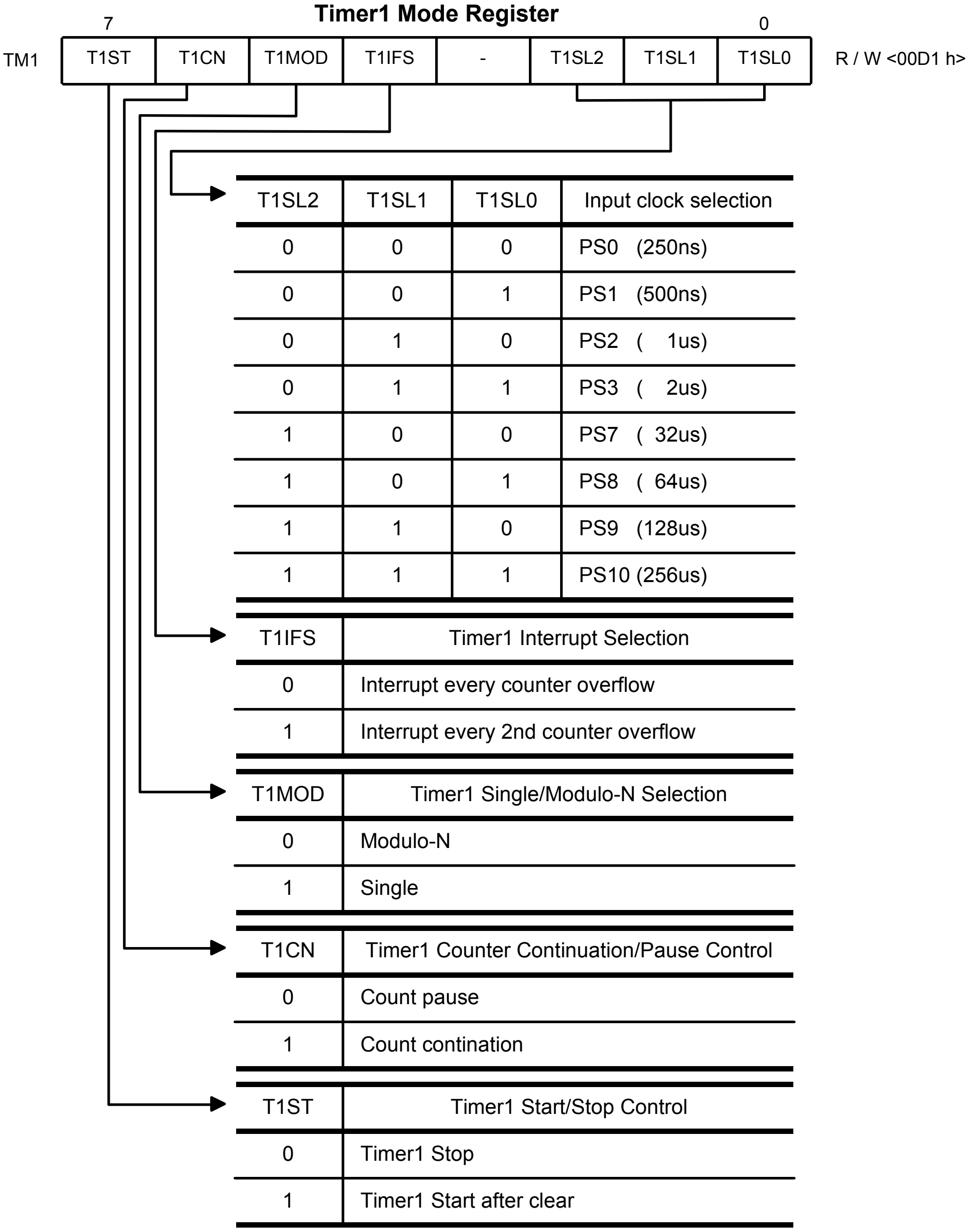
T0CN	Timer0 Counter Continuation/Pause Control
0	Count pause
1	Count contination

T0ST	Timer0 Start/Stop Control
0	Timer0 Stop
1	Timer Start after clear

CAP0	Timer0 Interrupt Selection
0	Timer/Counter
1	Input capture *

* PS1 : not supporting input capture.

Figure 12-9 Timer0 Mode Register



Timer1 Mode Register

	7							0	
TM1	T1ST	T1CN	T1MOD	T1IFS	-	T1SL2	T1SL1	T1SL0	R / W <00D1 h>

T1SL2	T1SL1	T1SL0	Input clock selection
0	0	0	PS0 (250ns)
0	0	1	PS1 (500ns)
0	1	0	PS2 (1us)
0	1	1	PS3 (2us)
1	0	0	PS7 (32us)
1	0	1	PS8 (64us)
1	1	0	PS9 (128us)
1	1	1	PS10 (256us)

T1IFS	Timer1 Interrupt Selection
0	Interrupt every counter overflow
1	Interrupt every 2nd counter overflow

T1MOD	Timer1 Single/Modulo-N Selection
0	Modulo-N
1	Single

T1CN	Timer1 Counter Continuation/Pause Control
0	Count pause
1	Count contination

T1ST	Timer1 Start/Stop Control
0	Timer1 Stop
1	Timer1 Start after clear

Figure 12-10 Timer1 Mode Register

Timer2 Mode Register

	7							0	
TM2	-	-	-	T2ST	T2CN	T2SL2	T2SL1	T2SL0	R / W <00D2 h>

T2SL2	T2SL1	T2SL0	Input clock selection
0	0	0	PS5 (8us)
0	0	1	PS6 (16us)
0	1	0	PS7 (32us)
0	1	1	PS8 (64us)
1	0	0	PS9 (128us)
1	0	1	PS10 (256us)
1	1	0	PS11 (512us)
1	1	1	PS12 (1024us)

T2CN	Timer2 Counter Continuation/Pause Control
0	Count pause
1	Count contination

T2ST	Timer2 Start/Stop Control
0	Timer2 Stop
1	Timer2 Start after clear

Figure 12-11 Timer2 Mode Register

External Interrupt Signal Edge Selection Register

	7							0	
IEDS	-	-	IED2H	IED2L	IED1H	IED1L	-	-	W <00CB h>

IED*H	IED*L	INT*
0	0	-
0	1	Falling Edge Selection
1	0	Rising Edge Selection
1	1	Both Edge Selection

Figure 12-12 External Interrupt Signal Edge Selection Register

(3) Timer0, Timer1

TIMER0 and TIMER1 have an up-counter. When value of the up-counter reaches the content of Timer Data Register (TDR), the up-counter is cleared to ``00 h``, and interrupt (IFT0, IFT1) is occured at the next clock.

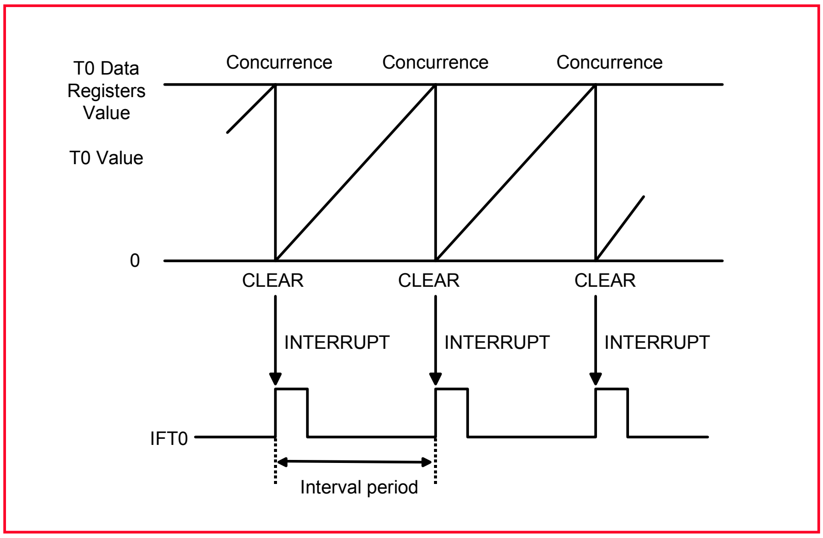


Figure 12-13 Operatiion of Timer0

For Timer0, the internal clock (PS) and the external clock (EC) can be selected as counter clock. But Timer1 and Timer2 use only internal clock. As internal clock. Timer0 can be used as internal-timer which period is determined by Timer Data Register (TDR). Chosen as external clock, Timer0 executes as event-counter. The counter execution of Timer0 and Timer1 is controlled by T0CN, T0ST, CAP0, T1CN, T1ST, of Timer Mode Register TM0 and TM1. T0CN, T1CN are used to stop and start Timer0 and Timer1 without clearing the counter. T0ST, T1ST is used to clear the counter. For clearing and starting the counter, T0ST or T1ST should be temporarily set to ``0`` and then set to ``1``. T0CN, T1CN, T0ST and T1ST should be set ``1``, when Timer counting-up. Controlling of CAP0 enables Timer0 as input capture. By programming of CAP0 to ``1``, the period of signal from INT2 can be measured and then, event counter value for INT2 can be read. During counting-up, value of counter can be read.

Timer execution is stopped by the reset signal (RESET = ``L``)

Note: *In the process of reading 16-bit Timer Data, first read the upper 8-bit data. Then read the lower 8-bit data, and read the upper 8-bit data again. If the earlier read upper 8-bit data are matched with the later read upper 8-bit data, read 16-bit data are correct. If not, caution should be taken in the selection of upper 8-bit data.*

(Example)

1) Upper 8-bit Read 0A 0A

2) Lower 8-bit Read FF 01

3) Upper 8-bit Read 0B 0B

====================

- -

0AFF 0B01

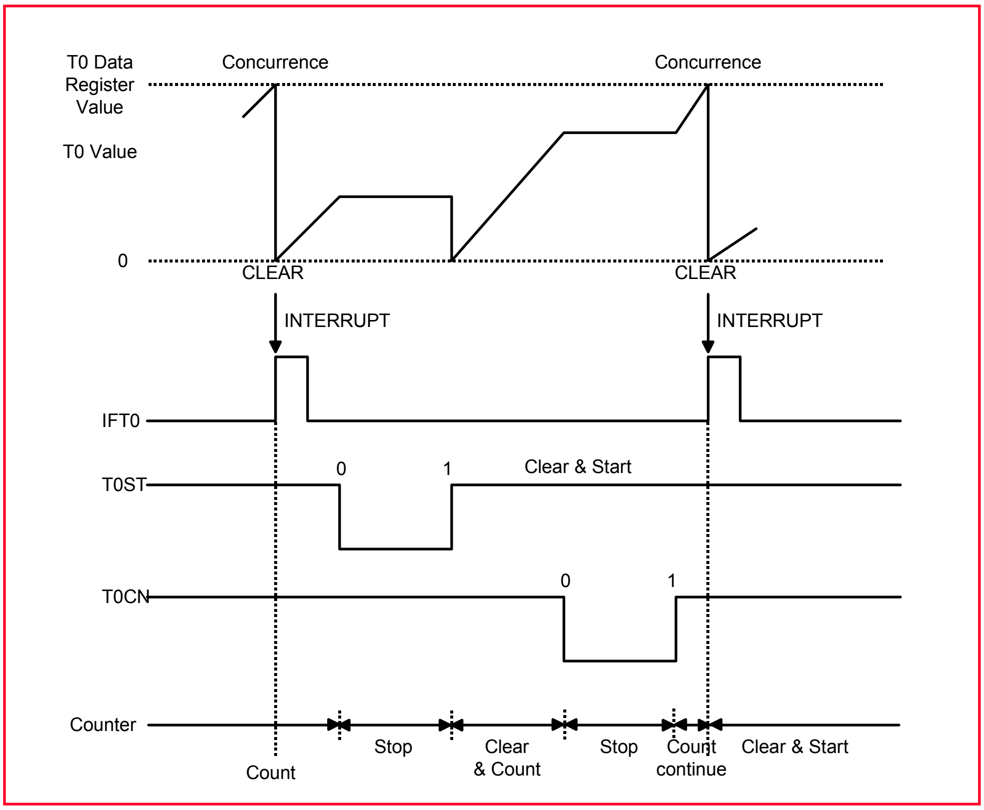


Figure 12-14 Start/Stop operation of Timer0

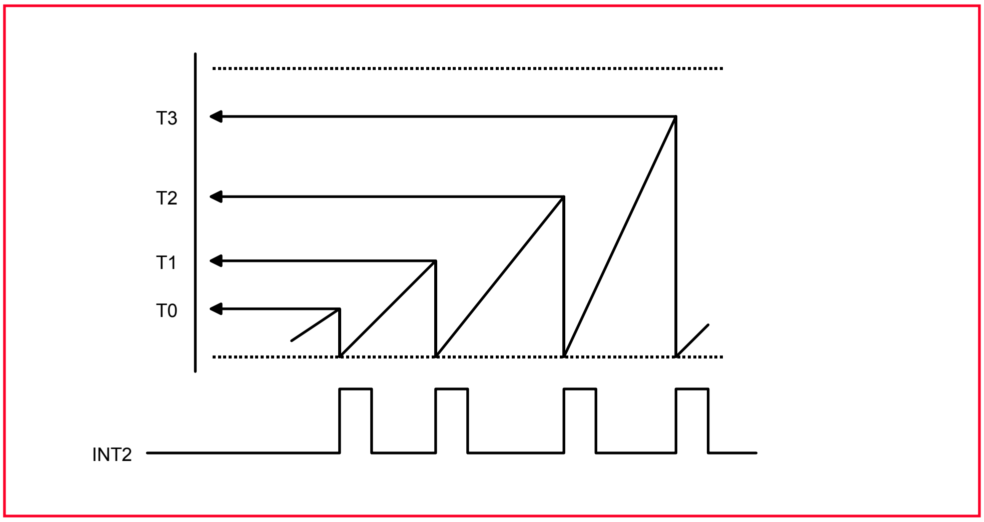


Figure 12-15 Input capture operation of Timer0

* Single/Modulo-N Mode

Timer0 (Timer1) can select initial (T0INIT, T1INIT of TM01) output level of Timer Output port. If initial level is ``L``, Low-Data Register value of Timer Data Register is transferred to comparator and T0OUT (T1OUT) is to be ``Low``, if initial level is High? High -Data Register is transferred and to be ``High``. Single Mode can be set by Mode Select bit (T0MOD, T1MOD) of

Timer Mode Register (TM0, TM1) to ``1`` When used as Single Mode, Timer counts up and compares with value of Data Register. If the result is same, Time Out interrupt occurs and level of Timer Output port toggle, then counter stops as reset state. When used as Modulo-N Mode, T0MOD (T1MOD) should be set ``0``. Counter counts up until the value of Data Register and occurs Time-out interrupt. The level of Timer Output port toggle and repeats process of

counting the value which is selected in Data Register. During Modulo-N Mode, If interrupt select bit (T0IFS, T1IFS) of Mode Register is ``0``, Interrupt occurs on every Time-out. If it is ``1``, Interrupt occurs every second time-out.

Note: *(*note. Timer Output is toggled whenever time out happen)*

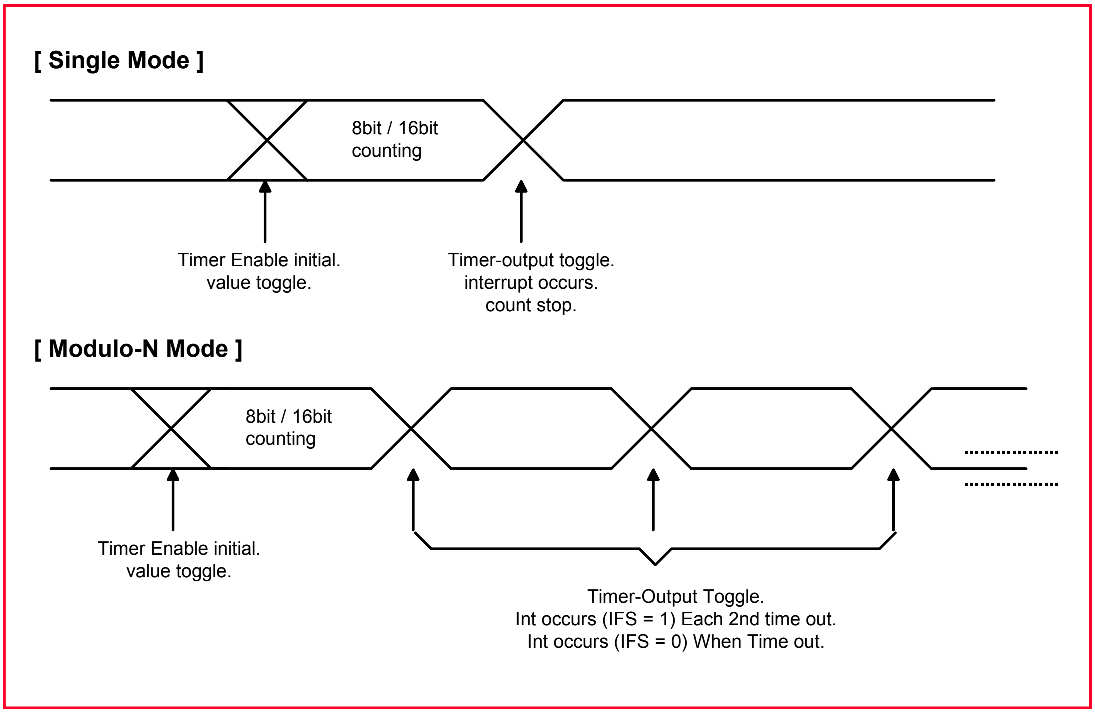

[Single Mode]
8bit / 16bit counting
Timer Enable initial. value toggle.
Timer-output toggle. interrupt occurs. count stop.
[Modulo-N Mode]
8bit / 16bit counting
Timer Enable initial. value toggle.
Timer-Output Toggle.
Int occurs (IFS = 1) Each 2nd time out.
Int occurs (IFS = 0) When Time out.

Figure 12-16 Operation Diagram for Single/Modulo-N Mode

(4) Timer 2

Timer2 operates as a up-counter. The content of T2DR are compared with the contents of up-counter. If a match is found. Timer2 interrupt (IFT2) is generated and the up-counter is cleared to ``00 h``. Therefore, Timer2 executes as a interval timer. Interrupt period is determined by the count source clock for the Timer2 and content of T2DR. When T2ST is set to ``1``, count value of Timer 2 is cleared and starts counting-up. For clearing and starting the Timer2. T2ST have to set to ``1`` after set to ``0``. In order to write a value directly into the T2DR, T2ST should be set to ``0``. Count value of Timer2 can be read at any time.

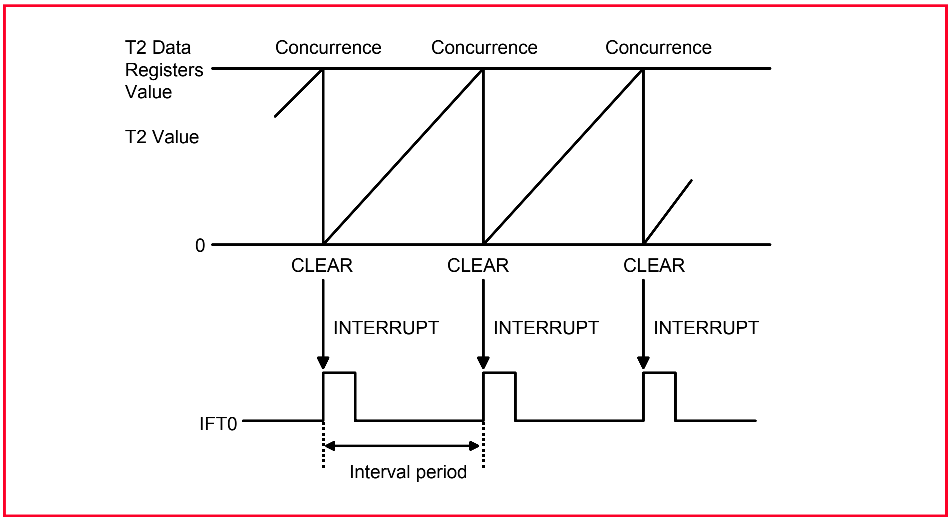


Figure 12-17 Operation of Timer2

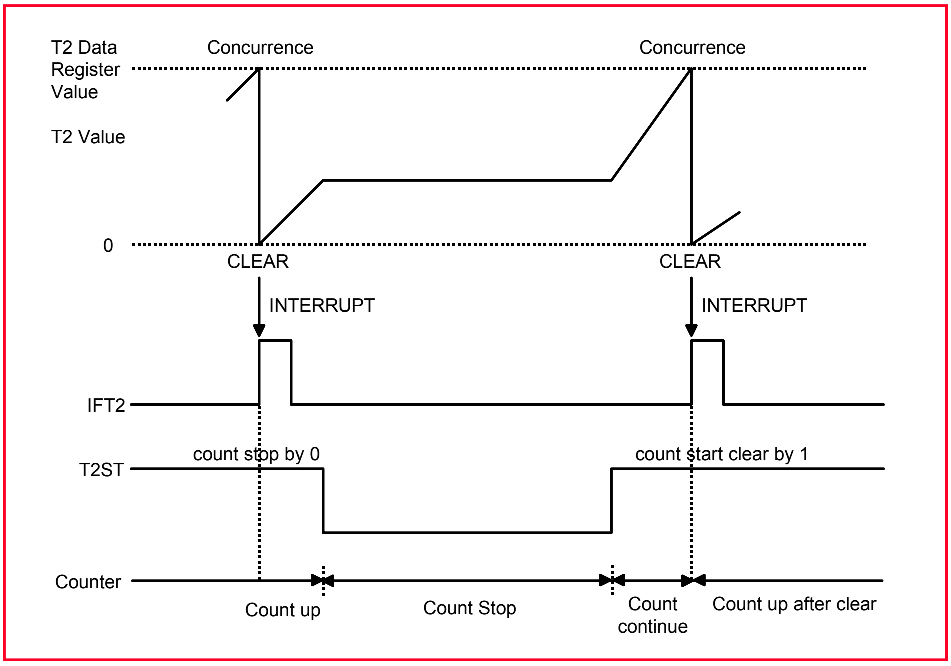


Figure 12-18 Start/Stop of Timer2

13. INTERRUPTS

The GMS81C5016/24/32 interrupt circuits consist of Interrupt Mode Register (MOD), Interrupt enable register (IENH, IENL), Interrupt request flags of IRQH, IRQL, Priority circuit and Master enable flag ("I" flag of PSW). 8 interrupt sources are provided. The configuration of interrupt circuit is shown in Figure 13-1.

The GMS81C5016/24/32 contains 8 interrupt sources; 3 externals and 5 internals. Nested interrupt services with priority control is also possible. Software interrupt is non-maskable interrupt, the others are all maskable interrupts.

- 8 interrupt source (2Ext, 3Timer, BIT, WDT and Key Scan)
- 8 interrupt vector
- Nested interrupt control is possible
- Programmable interrupt mode
- Hardware accept mode
- Software selection accept mode
- Read and write of interrupt request flag are possible.
- In interrupt accept, request flag is automatically cleared.

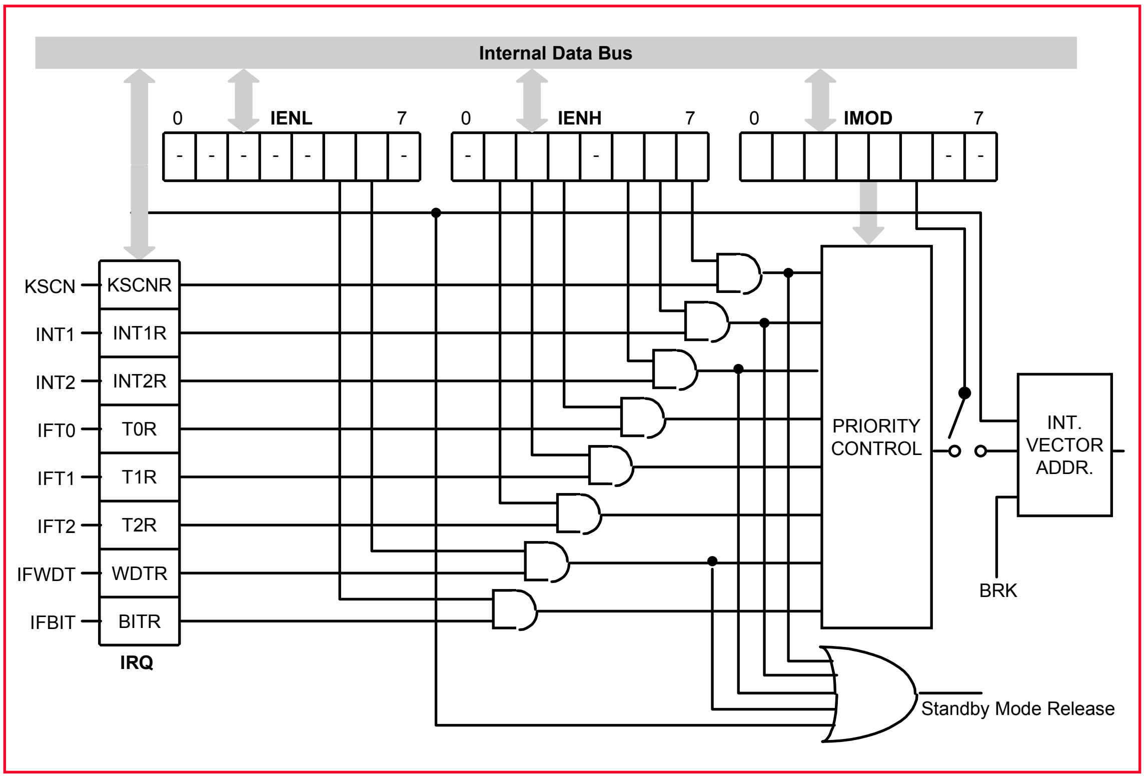


Figure 13-1 Block Diagram of Interrupt

13.1 Interrupt priority and sources.

Each interrupt vector is independent and has its own priority. Software interrupt (BRK) is also available. Interrupt source classification is shown in Table 13-1.

	Mask	Priority	Interrupt Source	INT Vector High	INT Vector Low
Hardware Interrupt	non-maskable	-	RST ($\overline{\text{RESET}}$ pin)	FFFF	FFFE
	maskable	0	KSCNR (Key Scan)	FFFB	FFFA
		1	INT1R (External Interrupt1)	FFF9	FFF8
		2	INT2R (External Interrupt2)	FFF7	FFF6
		3	T0R (Timer0)	FFF3	FFF2
		4	T1R (Timer1)	FFF1	FFF0
		5	T2R (Timer2)	FFEF	FFEE
		6	WDTR (Watctdog Timer)	FFE9	FFE8
		7	BITR (Basic Interval Timer)	FFE7	FFE6
	-	-	BRK instruction	FFDF	FFDE

Table 13-1 Interrupt Priority & Source

13.2 INTERRUPT CONTROL REGISTER

I flag of PSW is a interrupt mask enable flag. When I flag = ``0``, all interrupts become disable. When I flag = ``1``, interrupts can be selectively enabled and disabled by contents of corresponding Interrupt Enable Register. When interrupt is occured, interrupt request flag is set, and Interrupt request is detected at the edge of interrupt signal. The accepted interrupt request flag is automatically cleared during interrupt cycle process. The interrupt request flag maintains ``1`` until the interrupt is accepted or is cleared in program. In reset state, interrupt request flag register (IRQH, IRQL) is cleared to ``0``. It is possible to read the state of interrupt register and to mainpulate the contents of register and to generate interrupt. (Refer to software interrupt).

IENL	-	WDTR	BITE	-	-	-	-	-	R/W <00CCh>
IENH	KSCNE	INT1E	INT2E	-	T0E	T1E	T2E	-	R/W <00CEh>
IRQL	-	WDTR	BITE	-	-	-	-	-	R/W <00CDh>
IRQH	KSCNE	INT1R	INT2R	-	T0R	T1R	T2R	-	R/W <00CFh>

IENL : INTERRUPT ENABLE REGISTER LOW

IENH : INTERRUPT ENABLE REGISTER HIGH

IRQL : INTERRUPT REQUEST REGISTER LOW

IRQH : INTERRUPT REQUEST REGISTER HIGH

13.3 INTERRUPT ACCEPT MODE

The interrupt priority order is determined by bit (IM1, IM0) of IMOD register.

Interrupt Mode Register

	7							0	
IMOD	-	-	IM1	IM0	IP3	IP2	IP1	IP0	R/W <00CA h>

Assigning by interrupt accept mode bit

IM1	IM0	Priority
0	0	fixed by hardware
0	1	changeable by IP3~ IP0
1	*	Interrupt is inhibited

(1) Selection of Interrupt by IP3-IP0

The condition allow for accepting interrupt is set state of the interrupt mask enable flag and the interrupt enable bit must be ``1``. In Reset state, these IP3 - IP0 registers become all ``0``.

IP3	IP2	IP1	IP0	Selection Interrupt
0	0	0	1	KSCNR (Key Scan)
0	0	1	0	INT1R (External interrupt 1)
0	0	1	1	INT2R (External interrupt 2)
0	1	0	0	Reserved
0	1	0	1	T0R (Timer 0)
0	1	1	0	T1R (Timer 1)
0	1	1	1	T2R (Timer 2)
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	WDTR (Watch Dog Timer)
1	0	1	1	BITR (Basic Interval Timer)
1	1	0	0	Reserved

Table 13-1 Interrupt Selection by IP3 - IP0

(2) Interrupt Timing

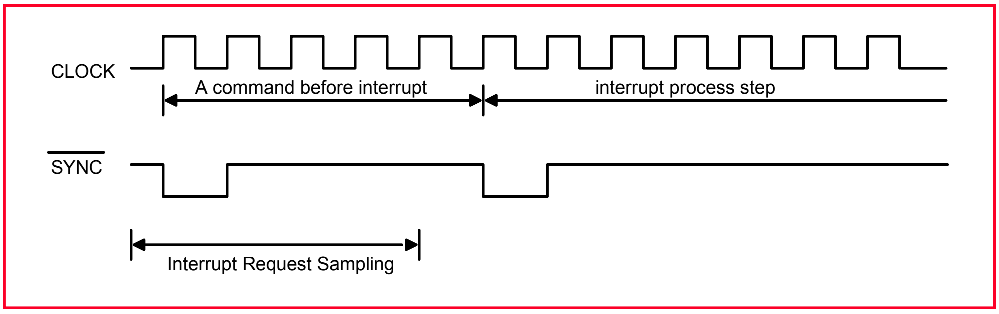


Figure 13-2 Interrupt Enable Accept Timing

*Interrupt Request sampling time

-Maximum 12 machine cycle (When execute DIV instruction)

-Minimum 0 machine cycle

*Interrupt preprocess step is 8 machine cycle

*Interrupt overhead

-Maximum 1 + 12 + 8 = 21 machine cycle

-Minimum 1 + 0 + 8 = 9 machine cycle

(3) The valid timing after executing Interrupt control instructions

I flag is valid just after executing of EI/DI on the contrary. Interrupt Enable register is valid one instruction after controlling interrupt Enable Register.

13.4 INTERRUPT PROCESSING SEQUENCE

When an interrupt is accepted, the on-going process is stopped and the interrupt service routine is executed. After the interrupt service routine is completed it is necessary to restore everything to the state before the interrupt occured.As soon as an interrupt is accepted, the content of the program counter and PSW are savedin the stack area. At the same time, the content of the vector address corresponding to the accepted interrupt, which is in the interrupt vector table, enters into the program counter and interrupt service is executed. In order to execute the interrupt service routine, it is necessary to write the jump addresses in the vector table (FFE0 h ~ FFFF h) corresponding to each interrupt

* Interrupt Processing Step

1) Store upper byte of Program Counter, SP <= SP

2) Store lower byte of Program Counter, SP <= SP - 1

3) Store Program Status Word, SP <= SP - 2

4) After resetting of I-flag, clear accepted Interrupt Request Flag. (Set B-flag for BRK Instruction)

5) Call Interrupt service routine

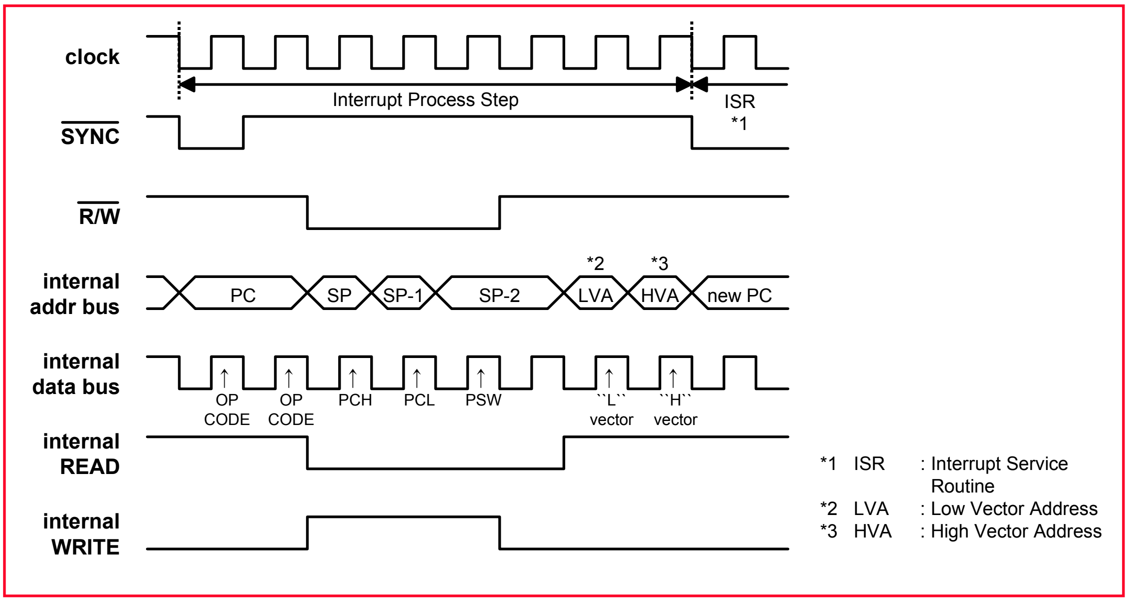
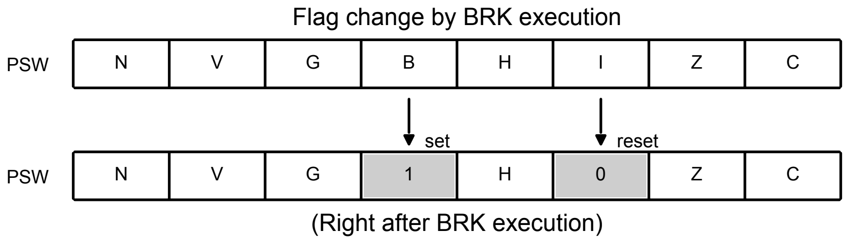


Figure 13-3 Interrupt Procesing Step Timing

13.5 SOFTWARE INTERRUPT (Interrupt by Break (BRK) Instruction)

Software interrupt is available just by writing ``Break(BRK)`` instruction. The values of PC and PSW is stacked by BRK instruction and then B flag of PSW is set and I flag is reset.



Interrupt vector of BRK instruction is shared by vector of Table Call (TCALL0). When both instruction of BRK and TCALL0 are used, as shown in Figure 13-4each processing routine is judged by contents of B flag. There is no instruction to reset directly B flag.

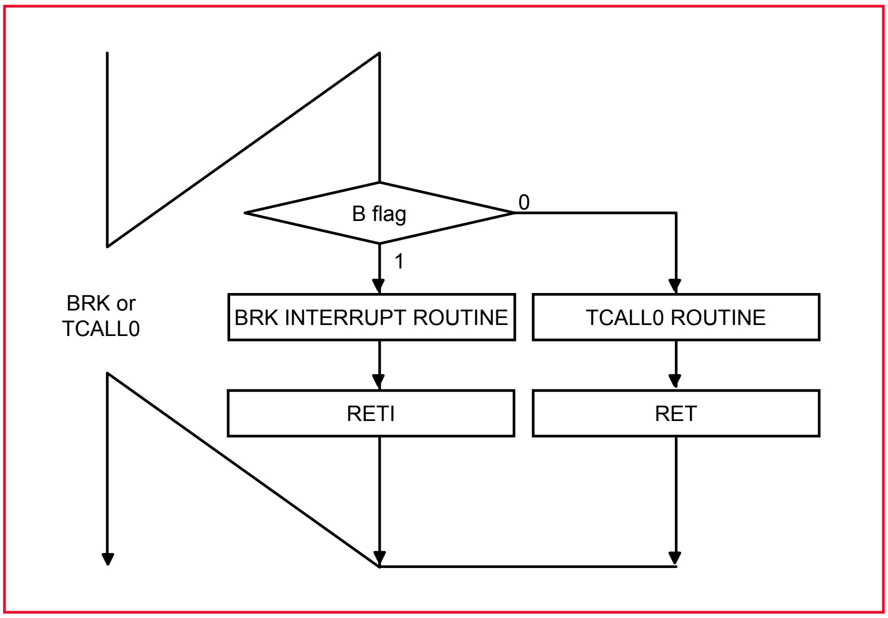


Figure 13-4 Execution of BRK or TCALL0

13.6 MULTIPLE INTERRUPT

If there is an interrupt, Interrupt Mask Enable Flag is automatically cleared before entering the Interrupt Service Routine. After then, no interrupt is accepted. If EI instruction is executed, interrupt mask enable bit becomes ``1``, and each enable bit can accept interrupt request. When two or more interrupts are generated simultaneously, the highest priority interrupt set by Interrupt Mode Register is accepted.

13.7 Key Scan Input Processing

(1) Standby Mode Release Register (SMRR)

Key Scan Interrupt is generated by detecting low or high Input from each Input pin (R0, R1) is one of the sources which release standby (SLEEP, STOP) mode. Key Scan ports are all 16bit which are controlled by Standby Mode Release Register (SMRR0, SMRR1). Key Input is considered as Interrupt, therefore, KSCNE bit of IEHN should be set for correct interrupt executing, SLEEP mode and STOP mode, the rest of executing is the same as that of external Interrupt. Each SMRR Register bit is allowed for each port (for Bit= ``0``, no Key Input, for Bit= ``1``, Key Input available). At reset, SMRR becomes ``00 h``. So, there is no Key Input source.

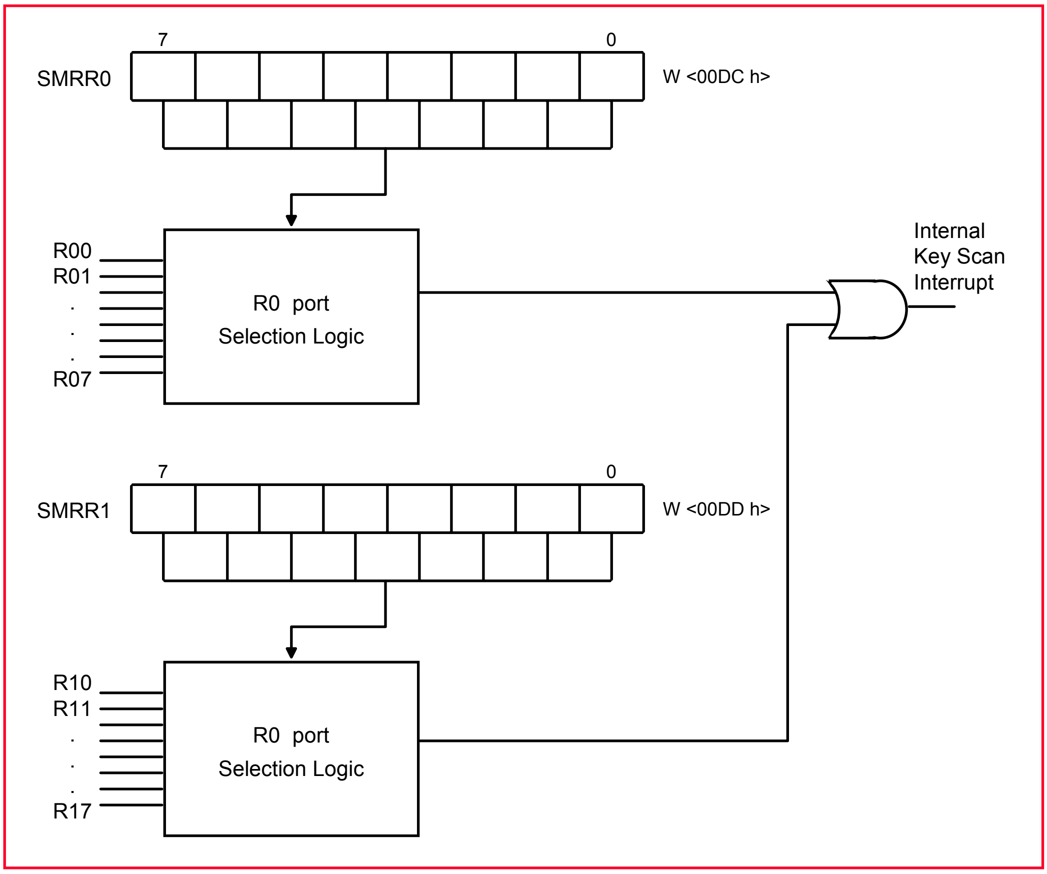


Figure 13-5 Key Scan Block

SMRR0 Register

	7							0	
SMRR0	KR07	KR06	KR05	KR04	KR03	KR02	KR01	KR00	W <00DC h>

SMRR1 Register

	7							0	
SMRR1	KR17	KR16	KR15	KR14	KR13	KR12	KR11	KR10	W <00DD h>

SMRR0		SMRR1		Key Input Selection
KR07	0	KR17	0	no select
	1		1	select
KR06	0	KR16	0	no select
	1		1	select
KR05	0	KR15	0	no select
	1		1	select
KR04	0	KR14	0	no select
	1		1	select
KR03	0	KR13	0	no select
	1		1	select
KR02	0	KR12	0	no select
	1		1	select
KR01	0	KR11	0	no select
	1		1	select
KR00	0	KR10	0	no select
	1		1	select

(2) Standby Release Level Control Register (SRLC)

Standby release level control register (SRLC) can select the key scan input level ``L`` or ``H`` for standby release by each bit pin (R0, R1). Standby release level control register (SRLC) is write-only register and initialized as ``00 h`` in reset state.

SRLC0 Register

	7							0	
SRLC0	KLR07	KLR06	KLR05	KLR04	KLR03	KLR02	KLR01	KLR00	W <00F6 h>

SRLC1 Register

	7							0	
SRLC1	KLR17	KLR16	KLR15	KLR14	KLR13	KLR12	KLR11	KLR10	W <00F7 h>

SRLC0		SRLC1		Key Input Level
KLR07	0	KLR17	0	Low
	1		1	High
KLR06	0	KLR16	0	Low
	1		1	High
KLR05	0	KLR15	0	Low
	1		1	High
KLR04	0	KLR14	0	Low
	1		1	High
KLR03	0	KLR13	0	Low
	1		1	High
KLR02	0	KLR12	0	Low
	1		1	High
KLR01	0	KLR11	0	Low
	1		1	High
KLR00	0	KLR10	0	Low
	1		1	High

14. WATCH DOG TIMER

Watch Dog Timer (WDT) consists of 6-bit binary counter, 6-bit comparator, and Watch Dog Timer Register (WDTR).

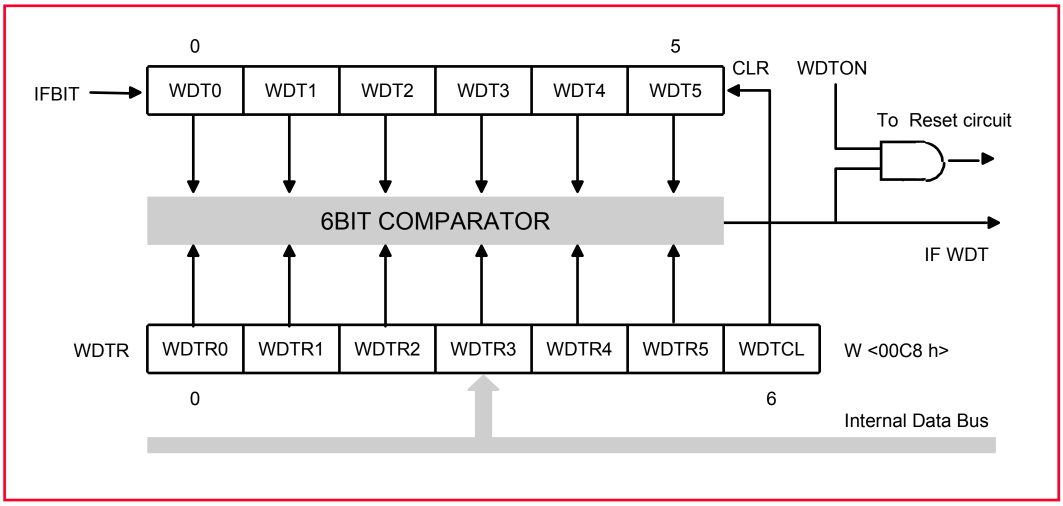


Figure 14-1 Block diagram of Watch Dog Timer

14.1 Control of WDT

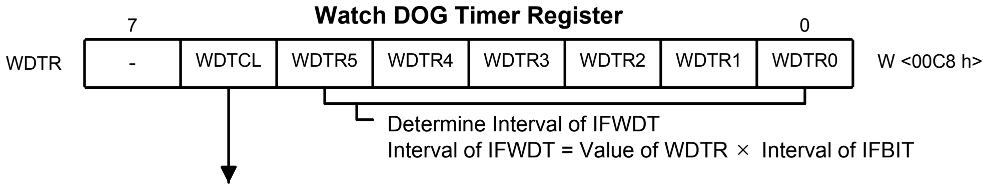
Watch Dog Timer can be used 6-bit general Timer or specific Watch dog timer by setting bit5 (WDTON) of Clock Control Register (CKCTLR).

Clock Control Register

	7							0	
CKCTLR	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7 h>

WDTON	Watch Dog Timer Function Control
0	6-bit Timer
1	Watch Dog Timer

By assigning bit6(WDTCL) of WDTR, 6-bit counter can be cleared.


Watch DOG Timer Register
7
0
WDTR
-
WDTCL
WDTR5
WDTR4
WDTR3
WDTR2
WDTR1
WDTR0
W <00C8 h>
Determine Interval of IFWDT
Interval of IFWDT = Value of WDTR × Interval of IFBIT

WDTCL	Watch Dog Timer Operation
0	free-run
1	Automatically cleared, after one machine cycle

14.2 WDT Interrupt Interval

WDT Interrupt (IFWDT) interval is determined by the interrupt IFBIT interval of Basic Interval Timer and the value of WDT Register.

-Interval of IFWDT = (IFBIT interval) * (WDTR value)

-Interval of IFWDT : 512 us * 1 = 512 us (MIN>)

-65,536us * 63 = 4,128,768 us (MAX>)

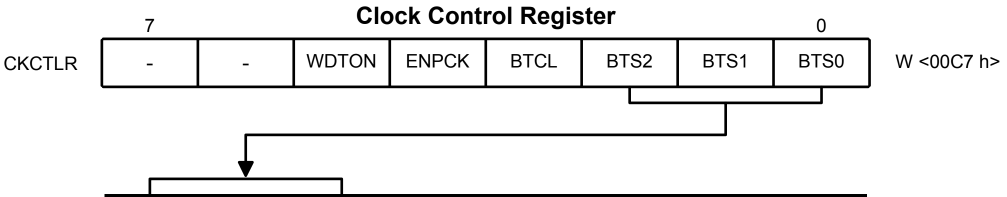
As IFBIT (Basic Interval Timer Interrupt Request) is used for input clock of WDT, Input clock cycle is possible from 512 us to 65,536 us by BTS. (at fex = 4MHz)

*At Hardware reset time ,WDT starts automatically. Therefore, the user must select the CKCTLR, WDTR before WDT overflow.

-Reset WDTR value = 0F h,15

-interval of WDT = 65,536 * 15 = 983040 us

(about 1second)



BTS2	BTS1	BTS0	WDT Input clock	Max. Interval of WDT Output (*note1)
0	0	0	512 us	32,756 us
0	0	1	1,024 us	64,512 us
0	1	0	2,048 us	129,024 us
0	1	1	4,096 us	258,048 us
1	0	0	8,192 us	516,096 us
1	0	1	16,384 us	1,032,192 us
1	1	0	32,768 us	2,064,384 us
1	1	1	65,536 us	4,128,768 us

***Note:** When WDTR Register value is 63 (3F h) (Caution) : Do not use ``0`` for WDTR Register value.*

Device come into the reset state by WDT

15. STANDBY FUNCTION

To save power consumption, there is STOP modes. In this modes, the execution of program stops.

15.1 Sleep Mode

SLEEP mode can be entered by setting the bit of SLEEP mode register (SLPM). In the mode, CPU clock stops but oscillator keeps running. B.I.T and a part of peripheral hardware execute, but prescalerís output which provide clock to peripherals can be stopped by program. (Except, PS10 canít stopped.) In SLEEP mode, more consuming power can be saved by not using other peripheral hardware except for B.I.T. By setting ENPCK (peripheral clock control bit) of CKCTLR (clock control register) to ``0``, peripheral hardware halted, and SLEEP mode is entered. To release SLEEP mode by BITR (basic interval timer interrupt), bit10 of prescaler should be selected as B.I.T input clock before entering SLEEP mode. ``NOP`` instruction should be follows setting of SLEEP mode for rising precharge time of data bus line.

(ex) setting of SLEEP mode : set the bit of SLEEP
; mode register (SLPM)
NOP : NOP instruction

SLEEP MODE CONTROL Register

	7							0	
SLPM	-	-	-	-	-	-	-	SLPM0	W <00F0 h>

SLPM0	condition
0	sleep mode release
1	sleep mode

Colck Control Register

	7							0	
CKCTLR	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C8 h>

ENCPK	Peripheral Clock
0	stopped
1	provided

15.2 STOP MODE

STOP mode can be entered by STOP instruction during program. In STOP mode, oscillator is stopped to make all clocks stop, which leads to less power consumption. All registers and RAM data are preserved. ``NOP`` instruction should be follows STOP instruction for rising precharge time of Data Bus line.

(ex) STOP : STOP instruction execution
NOP : NOP instruction

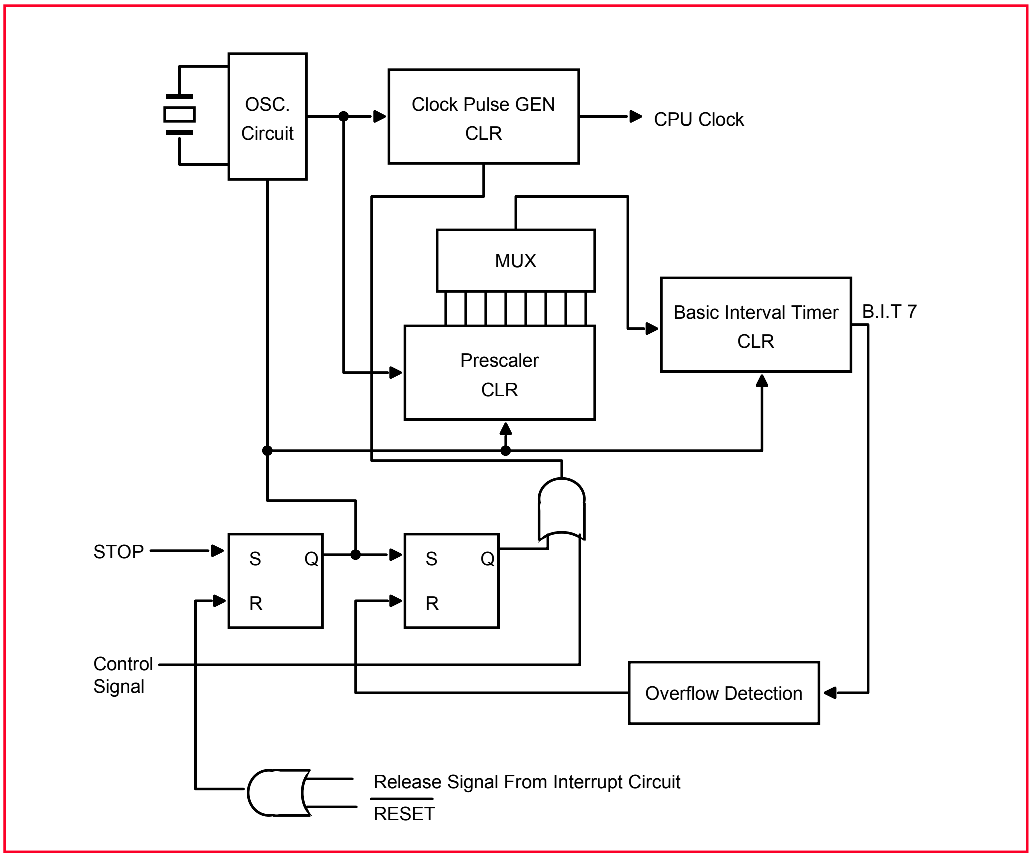


Figure 15-1 Block Diagram of Standby Circuit

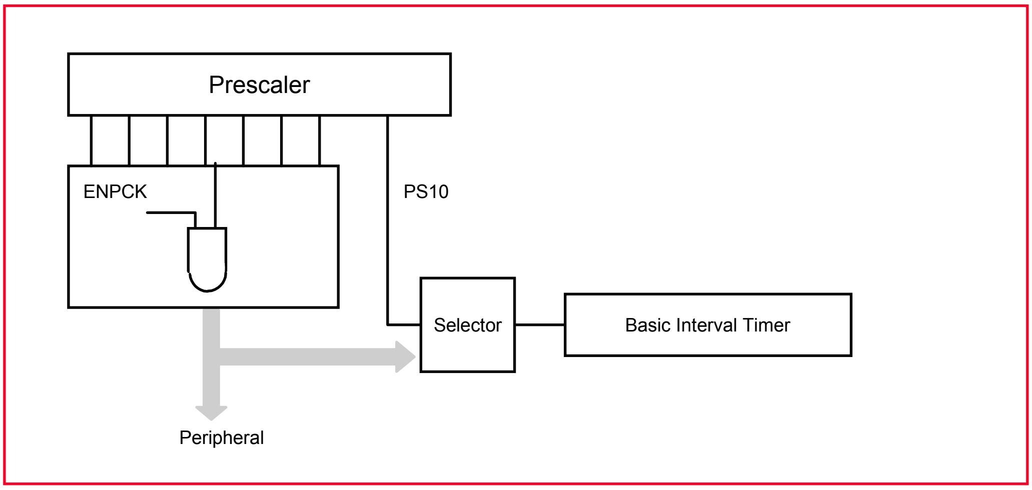


Figure 15-2 ENPCK and Basic Interval Timer Clock

15.3 STANDBY MODE RELEASE

Release of STANDBY mode is executed by RESET input and Interrupt signal. Register value is defined when Reset. When there is a release signal of STOP mode (Interrupt, RESET input), the instruction execution starts after stabilization oscillation time is set by value of BTS2 ~ BTS0 and set ENPCK to ``1``.

Release Signal	SLEEP	STOP
$\overline{\text{RESET}}$	O	O
KSCN (key input)	O	O
INT1 , INT2	O	O
B.I.T	O	X

Table 15-1 Standby Mode Register

Release Factor	Release Method
$\overline{\text{RESET}}$	By $\overline{\text{RESET}}$ Pin = Low level, Standby mode is release and system is initialized
KSCN (key input)	Standby mode is released by low input of selected pin by key scan Input (SMRR0, SMRR1) In case of interrupt mask enable flag = ``0``, program executes just after standby instruction, if flag = ``1``, enters each interrupt service routine.
INT1 INT2	When external interrupt (INT1, INT2) enable flag is ``1``, standby mode is released at the rising edge of each terminal. When Standby mode is released at interrupt. Mask Enable flag = ``0``, program executes from the next instruction of standby instruction. When ``1``, enters each interrupt service routine.
Basic Interval Timer (IFBIT)	When B.I.T is executed only by bit10 of prescaler (PS10), SLEEP mode can be release. Interrupt release SLEEP mode, when BIT interrupt enable flag is ``1``. When standby mode is released at interrupt. Mask enable flag = ``0``, program executes from the next instruction of SLEEP instruction. When ``1``, enters each interrupt service routine.

Table 15-2 Standby Mode Release

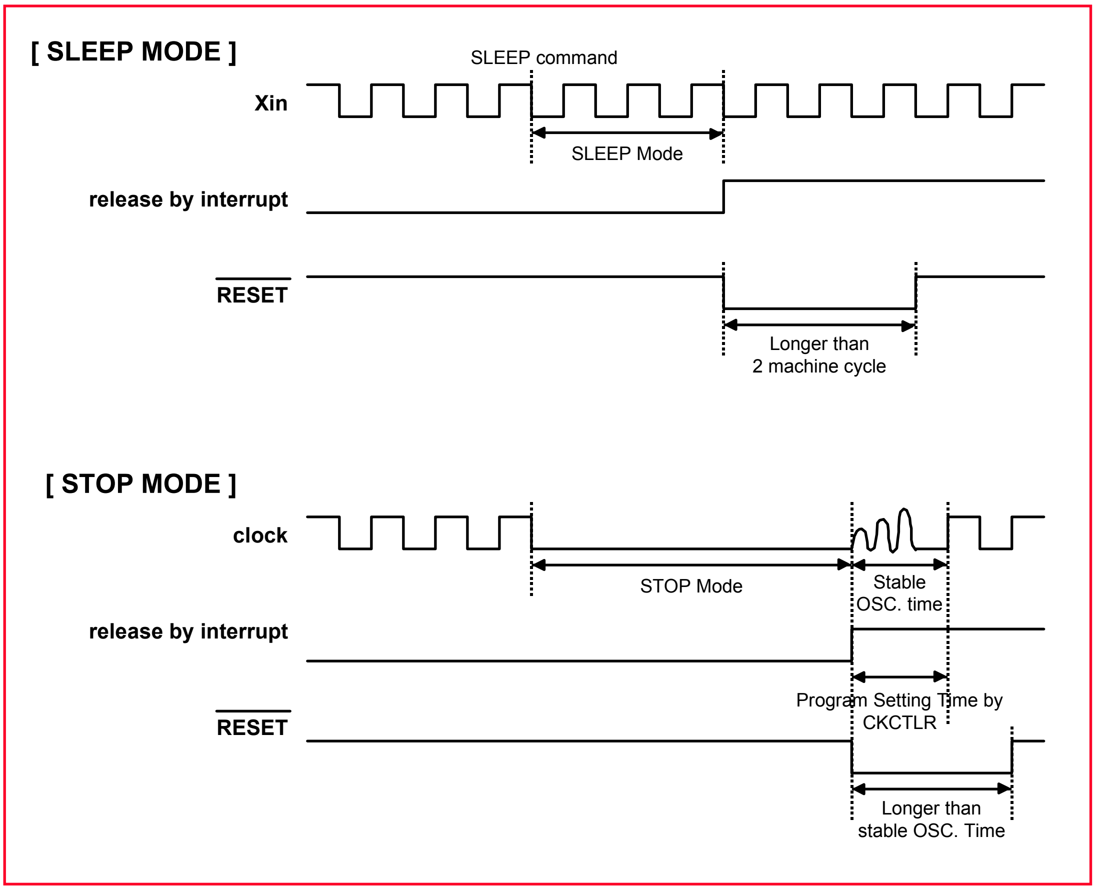


Figure 15-3 Release Timing of Standby Mode

15.4 RELEASE OPERATION OF STANDBY MODE

After standby mode is released, the operation begins according to content of related interrupt register just before standby mode start (Figure 15-4)

(1) Interrupt Enable Flag(I) of PSW = ``0``

Release by only interrupt which interrupt enable flag = ``1``, and starts to execute from next to standby instruction (SLEEP or STOP).

(2) Interrupt Enable Flag(I) of PSW = ``1``

Released by only interrupt which each interrupt enable flag = ``1``, and jump to the relevant interrupt service routine.

***Note:** When STOP instruction is used, B.I.T should guarantee the stabilization oscillation time. Thus, just before entering STOP mode, clock of bit10 (PS10) of prescaler is selected or peripheral hardware clock control bit (ENPCK) to ``1``, Therefore the clock necessary for stabilization oscillation time should be input into B.I.T. otherwise, standby mode is released by reset signal. In case of interrupt request flag and interrupt enable flag are both ``1``, standby mode is not entered.*

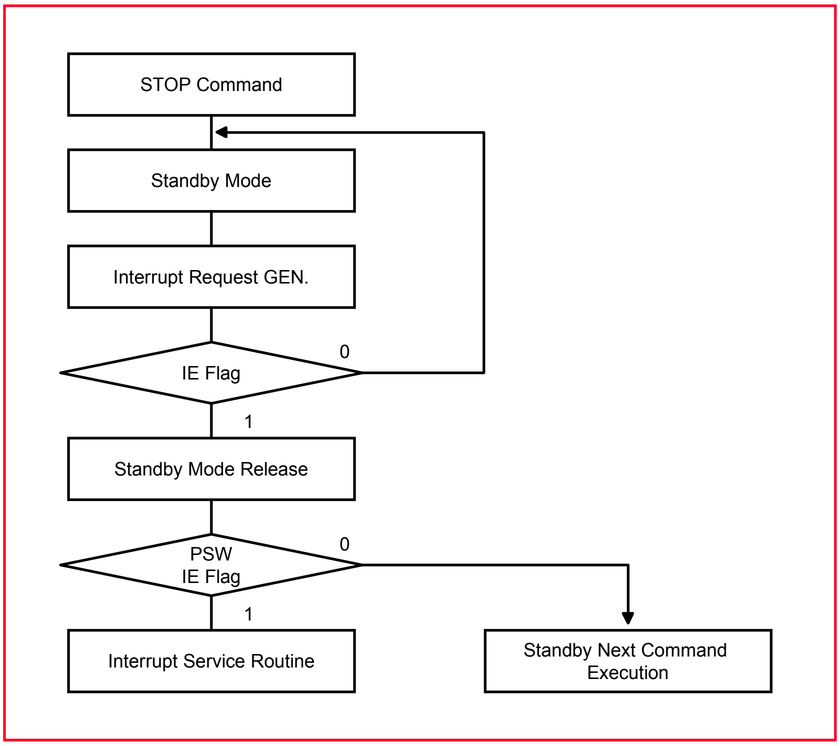


Figure 15-4 Standby Mode Release Flow

Internal circuit	SLEEP mode	STOP mode
Oscillator	Active	Stop
Internal CPU clock	Stop	Stop
Register	Retained	Retained
RAM	Retained	Retained
I/O port	Retained	Retained
Prescaler	Active	Retained
Basic Interval Timer	PS10 selected : Active Others : Stop	Stop
Watch Dog Timer	Stop	Stop
Timer	Stop	Stop
Address Bus, Data Bus	Retained	Retained

Table 15-1 Operation State in Standby Mode

16. OSCILLATION CIRCUIT

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Fig. 4.2-(a) shows circuit diagrams using a crystal (or ceramic) oscillator. As shown in the diagram, oscillation circuits can be constructed by connecting a oscillator between Xout and Xin. Clock from oscillation circuit makes CPU clock via clock pulse generator, and then enters prescaler to make peripheral hardware clock. Alternately, the oscillator may be driven from an external source as shown is Fig. 4.2.-(b). In the Standby (STOP) mode, oscillatiion stop, Xout state goes to ``HIigh``, Xin state goes to ``Low``, and built-in feed back resistor is disabled.

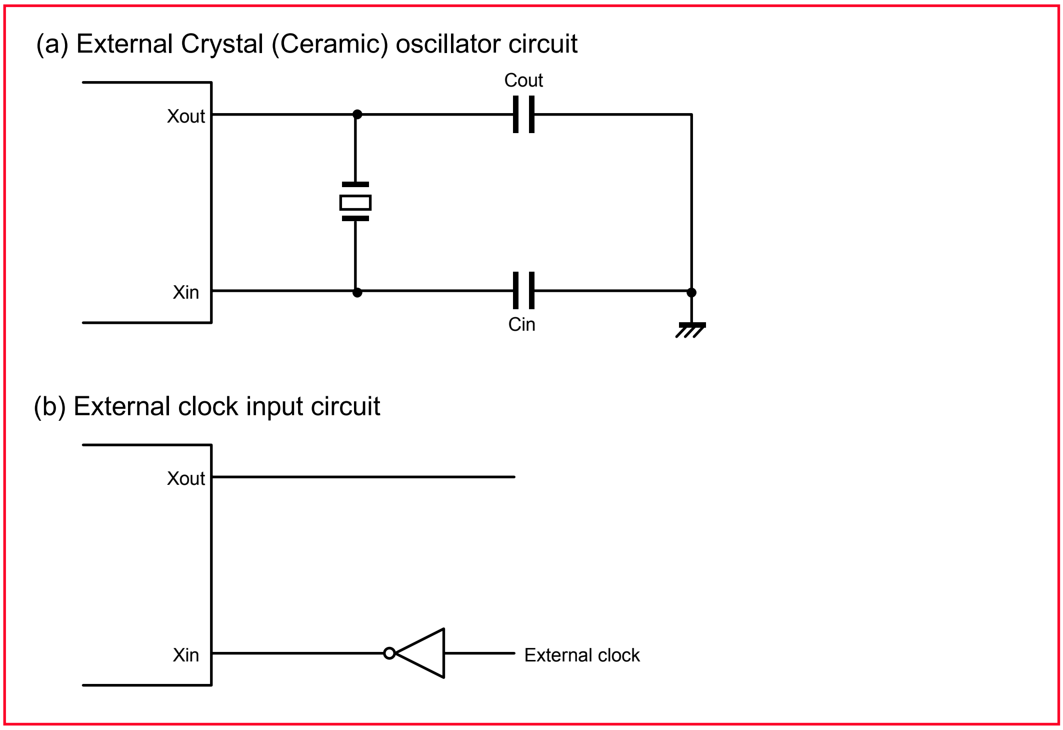

(a) External Crystal (Ceramic) oscillator circuit
Cout
Xout
Xin
Cin
(b) External clock input circuit
Xout
Xin
External clock

Figure 16-1 Oscillator configurations

* Recommendable resonator

Frequency	Resonator Maker	Part Name	Load Capacitor	Operating Voltage
4.0 MHz	CQ	ZTA4.00MG	Cin=Cout=30pF	2.2 ~ 4.0V
	TDK	FCR4.0MC5	Cin=Cout=open	2.2 ~ 4.0V
	TDK	FCR4.0M5	Cin=Cout=33pF	2.2 ~ 4.0V
	TDK	CCR4.0MC3		2.2 ~ 4.0V

* MC type is building in load capacitior.CCR type is chip type.

17. RESET FUNCTION

17.1 EXTERNAL RESET

The RESET pin should be held at low for at least 2machine cycles with the power supply voltage within the operating voltage range and must be connected 0.1uF capacitor for stable system initialization. The RESET pin contains a Schmitt trigger with an internal pull-up resistor.

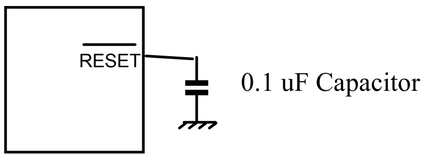


Figure 17-1

17.2 POWER ON RESET

Power On Reset circuit automatically detects the rise of power voltage (the rising time should be within 50ms) the power voltage reaches a certain level, RESET terminal is maintained at °»L°» Level until a crystal ceramic oscillator oscillates stably. After power applies and starting of oscillation, this reset state is maintained for about oscillation cycle of 219 (about 65.5ms : at 4MHz).The execution of built-in Power On Reset circuit is as follows :

(1) Latch the pulse from Power On Detection Pulse Generator circuit, and reset Prescaler, B.I.T and B.I.T Overflow detection circuit.

(2) Once B.I.T Overflow detection circuit is reset. Then, Prescaler starts to count.

(3) Prescaler output is inputted into B.I.T and PS10 of Prescaler output is automatically selected. If overflow of B.I.T is detected, Overflow detection circuit is set.

(4) Reset circuit generates maximum period of reset pulse from Prescaler and B.I.T.

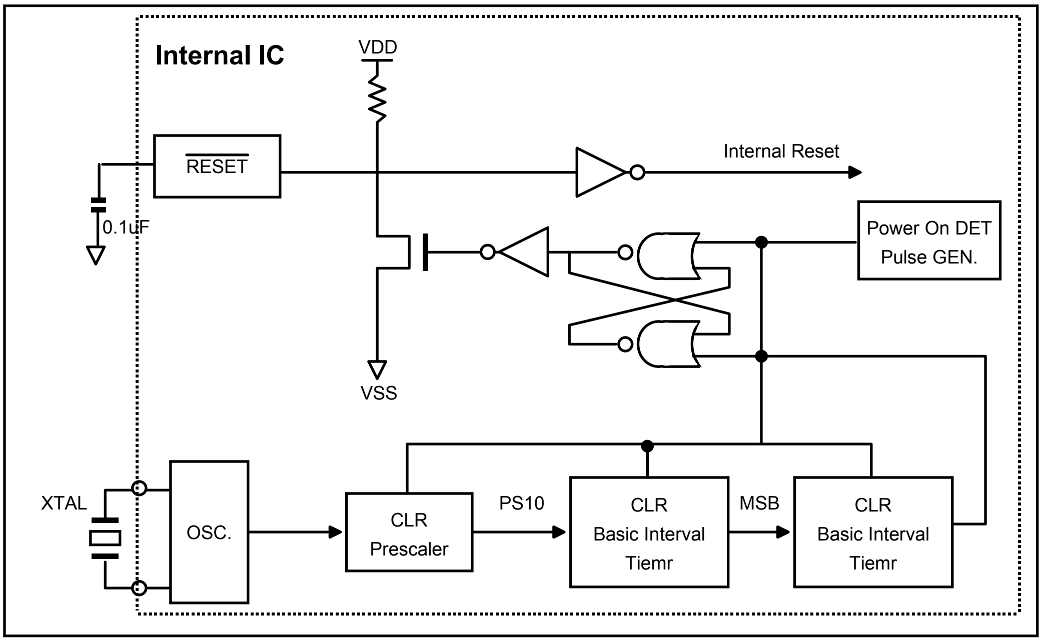


Figure 17-2 Block Diagram of Power On Reset Circuit

Note: *Notice ; When Power On Reset, oscillator stabilization time doesn`t include OSC. Start time.*

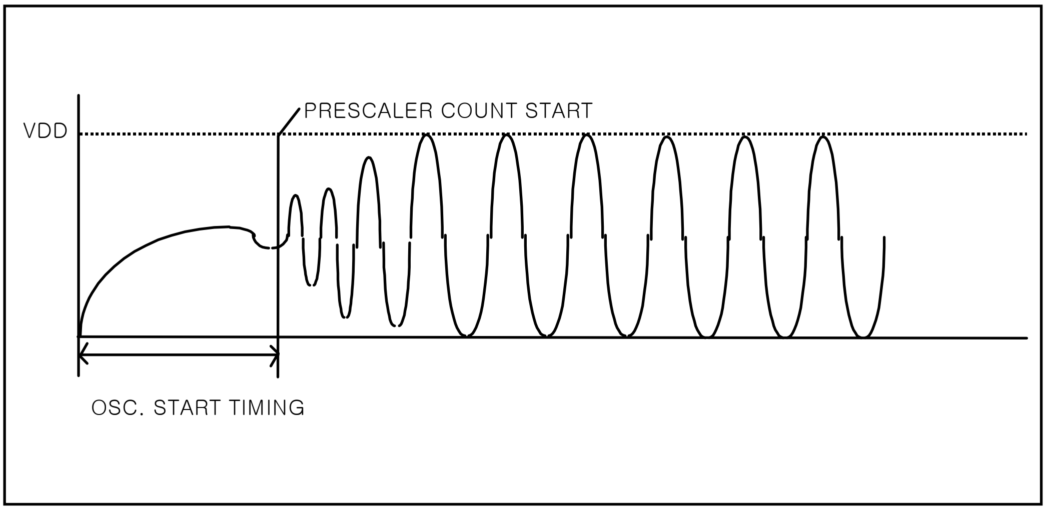


Figure 17-3 Oscillator stabilization diagram

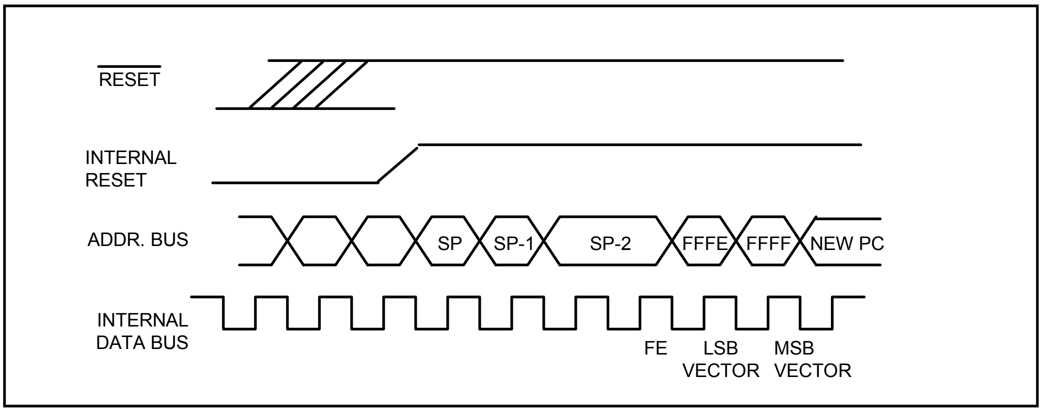


Figure 17-4 Reset Timing by Diagram

17.3 Low Voltage Detection Mode

(1) Low voltage detection condition

An on board voltage comparator checks that VDD is at the required level to ensure correct operation of the device. If VDD is below a certain level, Low voltage detector forces the device into low voltage detection mode.

(2) Low Voltage Detection Mode

There is no power consumption except stop current, stop mode release function is disabled. All I/O port is configured as input mode and Data memory is retained until voltage through external capacitor is worn out. In this mode, all port can be selected with Pull-up resistor by Mask option. If there is no information on the Mask option sheet ,the default pull up option (all port connect to pull-up resistor) is selected.

(3) Release of Low Voltage Detection Mode

Reset signal result from new battery(normally 3V) wakes the low voltage detection mode and come into normal reset state. It depends on user whether to execute RAM clear routine or not.

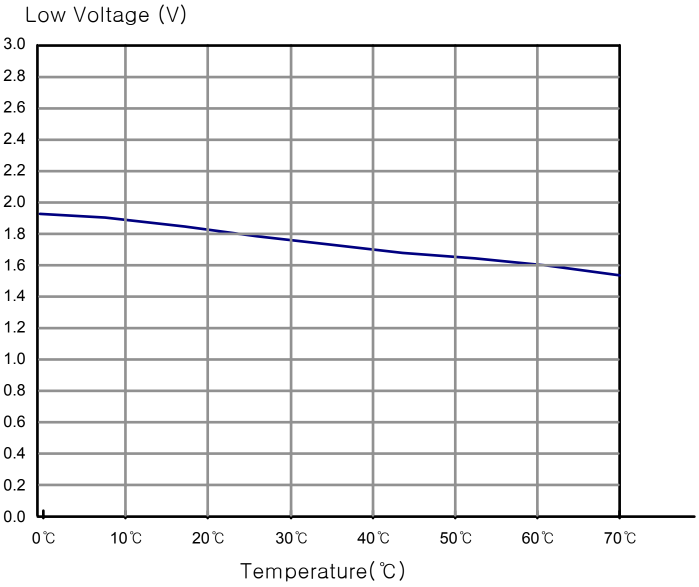
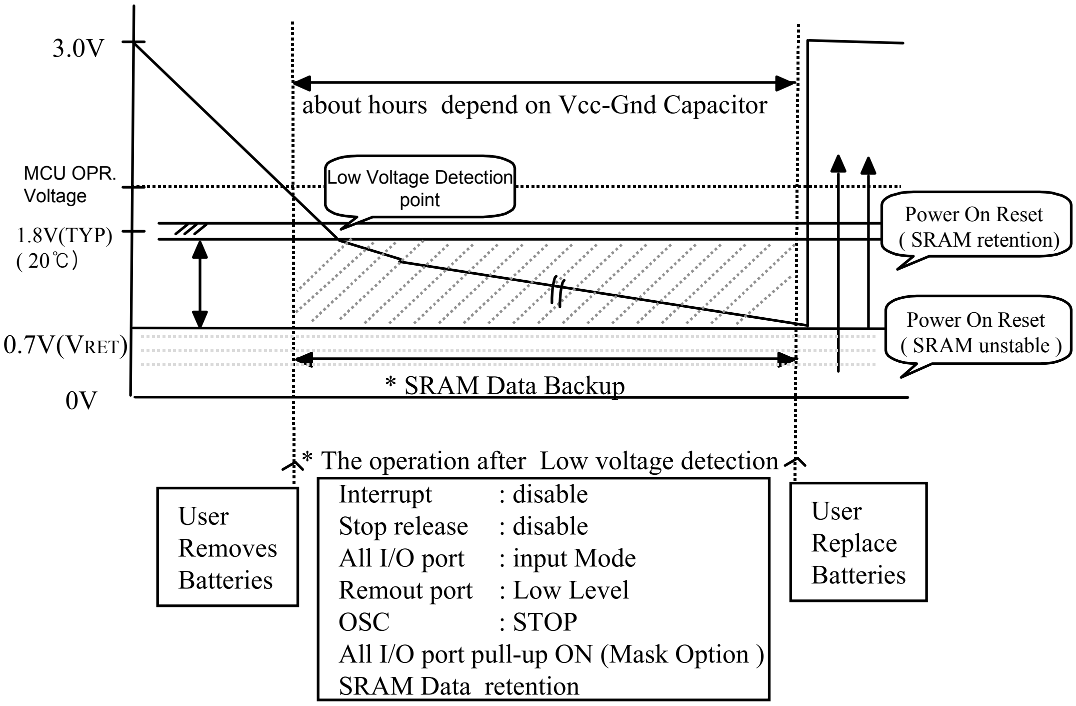


Figure 17-5 Low Voltage vs Temperature

(4) SRAM BACK-UP after Low Voltage Detection.


3.0V
about hours depend on Vcc-Gnd Capacitor
MCU OPR. Voltage
Low Voltage Detection point
1.8V(TYP) (20℃)
Power On Reset (SRAM retention)
Power On Reset (SRAM unstable)
0.7V(VRET)
* SRAM Data Backup
0V
* The operation after Low voltage detection
User Removes Batteries
User Replace Batteries

Interrupt : disable
Stop release : disable
All I/O port : input Mode
Remout port : Low Level
OSC : STOP
All I/O port pull-up ON (Mask Option)
SRAM Data retention

Figure 17-6 Low Voltage Detection and Protection

(5) S/W flow chart example after Reset using SRAM Back-up

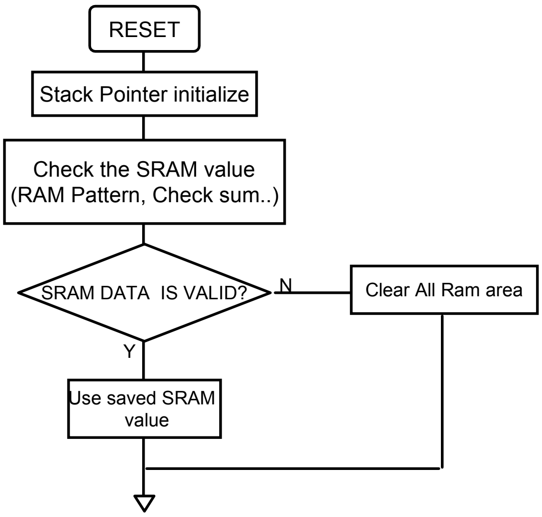


Figure 17-7 S/W Flow Chart Example for SRAM Back-up

17.4 Low Voltage Indicator Register (LVIR)

Low Voltage Indication Register (LVIR) is read only Register. It is useful to display the consumption of Batteries. If VDD power level is below a cirtain level which is higher than low voltage detection level (refer to Figure 17-6) , The bit of LVIR register could be set according to the VDD level sequentially. The VDD dection levels for Indication are two , that is , Bit1 and Bit0 of LVIR Register. The detection level of Bit0 is higer than Bit1.

bit	7	6	5	4	3	2	1	0	
LVIR	-	-	-	-	-	-	LVIR1	LVIR0	<00EF h>
initial value	-	-	-	-	-	-	0	0	
R / W	-	-	-	-	-	-	R	R	

18. CLOCK GENERATOR

Clock generating circuit consists of Clock Pulse Generator (C.P.G), Prescaler, Basic Interval Timer (B.I.T) and Watch Dog Timer. The clock applied to the Xin pin divided by two is used as the internal system clock.

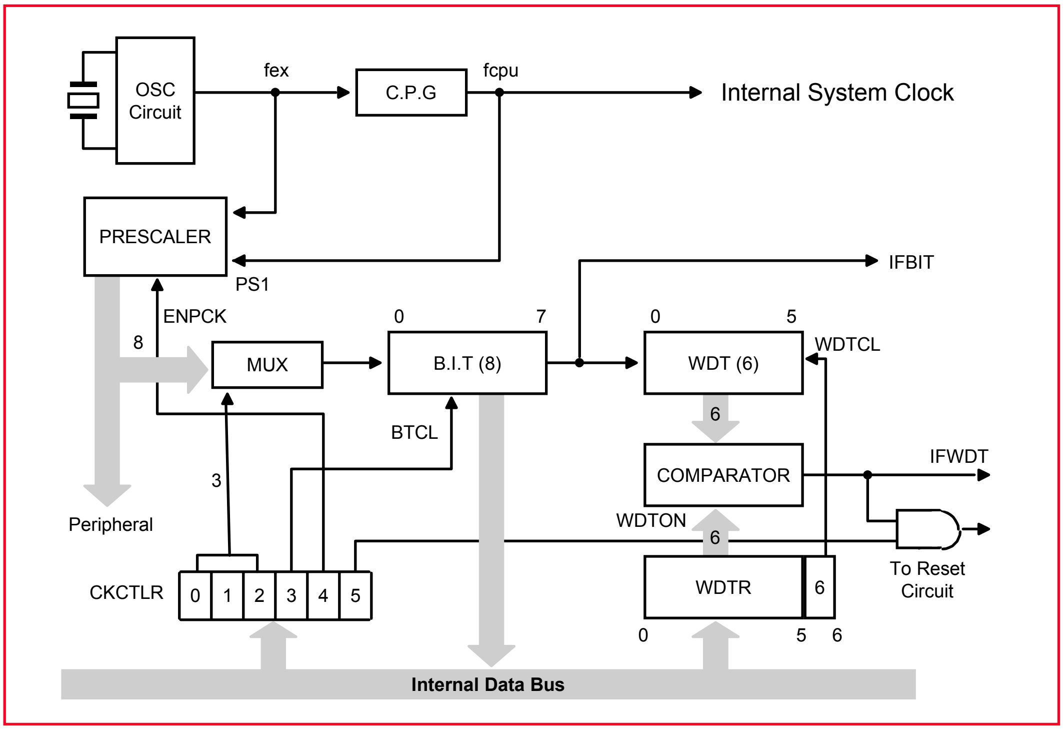


Figure 18-1 Block Diagram of Clock Generator

Prescaler consists of 12-bit binary counter. The clock supplied from oscillation circuit is input to prescaler (fex). The divided output from each bit of prescaler is provided to peripheral hardware.

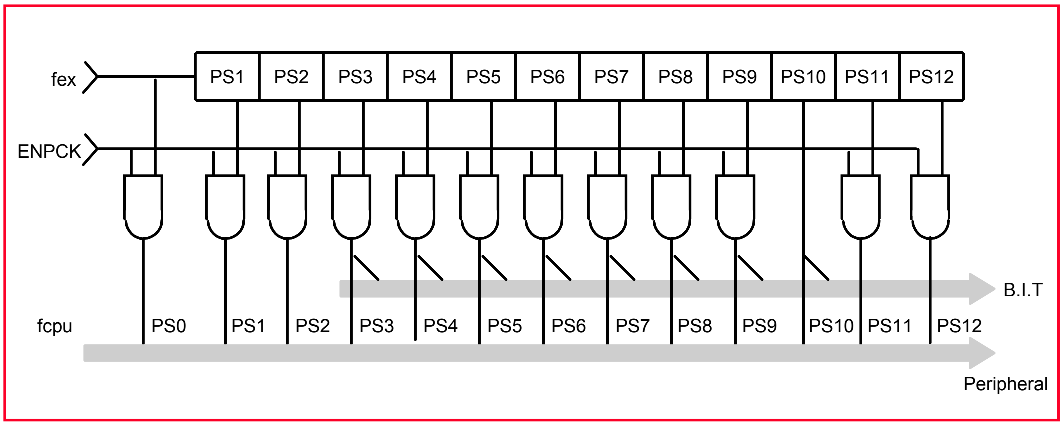


Figure 18-2 Block diagram of Prescaler

fex (MHz)	4 MHz		2 MHz	
	frequency	period	frequency	period
ps 0	4 MHz	250 ns	2 MHz	500 ns
ps 1	2 MHz	500 ns	1 MHz	1 us
ps 2	1 MHz	1 us	500 KHz	2 us
ps 3	500 KHz	2 us	250 KHz	4 us
ps 4	250 KHz	4 us	125 KHz	8 us
ps 5	125 KHz	8 us	62.5 KHz	16 us
ps 6	62.5 KHz	16 us	31.25 KHz	32 us
ps 7	31.25 KHz	32 us	15.63 KHz	64 us
ps 8	15.63 KHz	64 us	7.183 KHz	128 us
ps 9	7.183 KHz	128 us	3.906 KHz	256 us
ps 10	3.906 KHz	256 us	1.953 KHz	512 us
ps 11	1.953 KHz	512 us	0.976 KHz	1024 us
ps 12	0.976 KHz	1024 us	0.488 KHz	2048 us

Table 18-1 ps output perio Basic Interval Timer

The HMS87C5216 and GMS81C1408 has one 8-bit Basic Interval Timer that is free-run, can not stop. Block diagram is shown in Figure 18-3.The 8-bit Basic interval timer register (BITR) is increased every internal count pulse which is divided by prescaler. Since prescaler has divided ratio by 8 to 1024, the count rate is 1/8 to 1/1024 of the oscillator frequency. As the count overflows from FF_H to 00_H, this overflow causes to generate the Basic interval timer interrupt. The BITF is interrupt request flag of Basic interval timer.

When write “1” to bit BTCL of CKCTLR, BITR register is cleared to “0” and restart to count-up. The bit BTCL becomes “0” after one machine cycle by hardware.

If the STOP instruction executed after writing “1” to bit WAKEUP of CKCTLR, it goes into the wake-up timer mode. In this mode, all of the block is halted except the oscillator, prescaler (only fxin÷2048) and Timer0.

If the STOP instruction executed after writing “1” to bit RCWDT of CKCTLR, it goes into the internal RC oscillated watchdog timer mode. In this mode, all of the block is halted except the internal RC oscillator, Basic Interval Timer and Watchdog Timer. More detail informations are explained in Power Saving Function. The bit WDTON decides Watchdog Timer or the normal 7-bit timer

Note: *All control bits of Basic interval timer are in CKCTLR register which is located at same address of BITR (address EC_H). Address EC_H is read as BITR, written to CKCTLR. Therefore, the CKCTLR can not be accessed by bit manipulation instruction.*

.

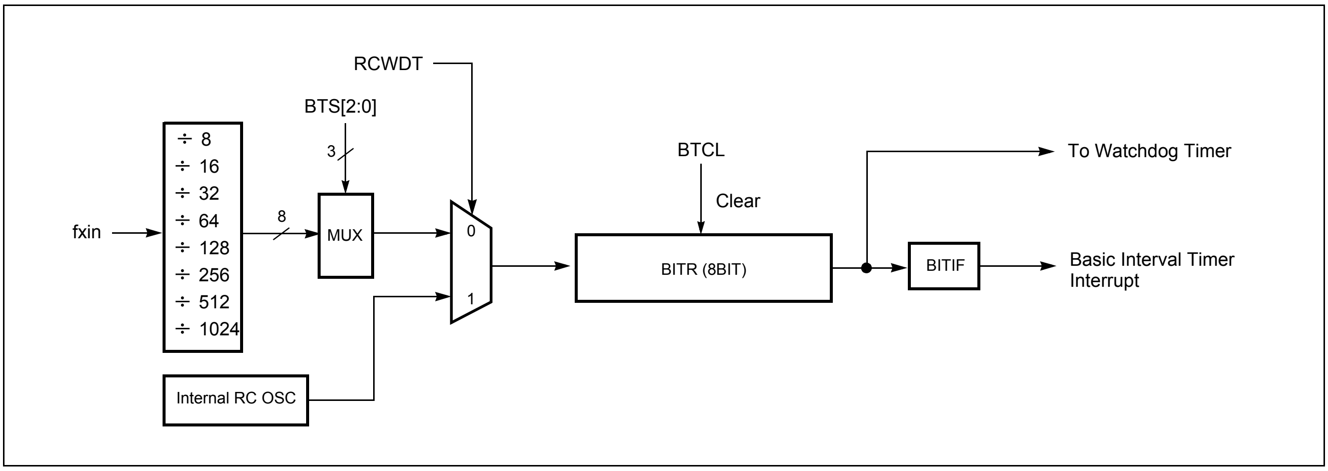


Figure 18-3 Block Diagram of Basic Interval Timer

Clock Control Register

CKCTLR	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0

ADDRESS : ECH
RESET VALUE : -0010111
Bit Manipulation Not Available

Symbol	Function Description
WAKEUP	1 : Enables Wake-up Timer 0 : Disables Wake-up Timer
RCWDT	1 : Enables Internal RC Watchdog Timer 0 : Disables Internal RC Watchdog Time
WDTON	1 : Enables Watchdog Timer 0 : Operates as a 7-bit Timer
BTCL	1 : BITR is cleared and BTCL becomes “0” automatically after one machine cycle, and BITR continue to count-up

Basic Interval Timer Clock Selection

000 : fxin ÷ 8
001 : fxin ÷ 16
010 : fxin ÷ 32
011 : fxin ÷ 64
100 : fxin ÷ 128
101 : fxin ÷ 256
110 : fxin ÷ 512
111 : fxin ÷ 1024

Figure 18-4 CKCTLR: Clock Control Register

19. ANALOG TO DIGITAL CONVERTER

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 8-bit digital value. The A/D module has eight analog inputs, which are multiplexed into one sample and hold. The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

The analog reference voltage is V_{DD}. The A/D module has two registers which are the control register ADMR and A/D result register ADDR. The ADMR register, shown in Figure 19-2, controls the operation of the A/D converter module. The port pins can be configure as analog inputs or digital I/O.

To use analog inputs, each port is assigned analog input port by setting the bit ANSEL[7:0] in RAFUNC register. And selected the corresponding channel to be converted by setting ADS[2:0].

The processing of conversion is start when the start bit ADST is set to “1”. After one cycle, it is cleared by hardware. The register ADCR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCR, the A/D conversion status bit ADSF is set to “1”, and the A/D interrupt flag ADIF is set. The block diagram of the A/D module is shown in Figure 19-1. The A/D status bit ADSF is set automatically when A/D conversion is completed, cleared when A/D conversion is in process. The conversion time takes maximum 30 uS (at fxin=4 MHz).

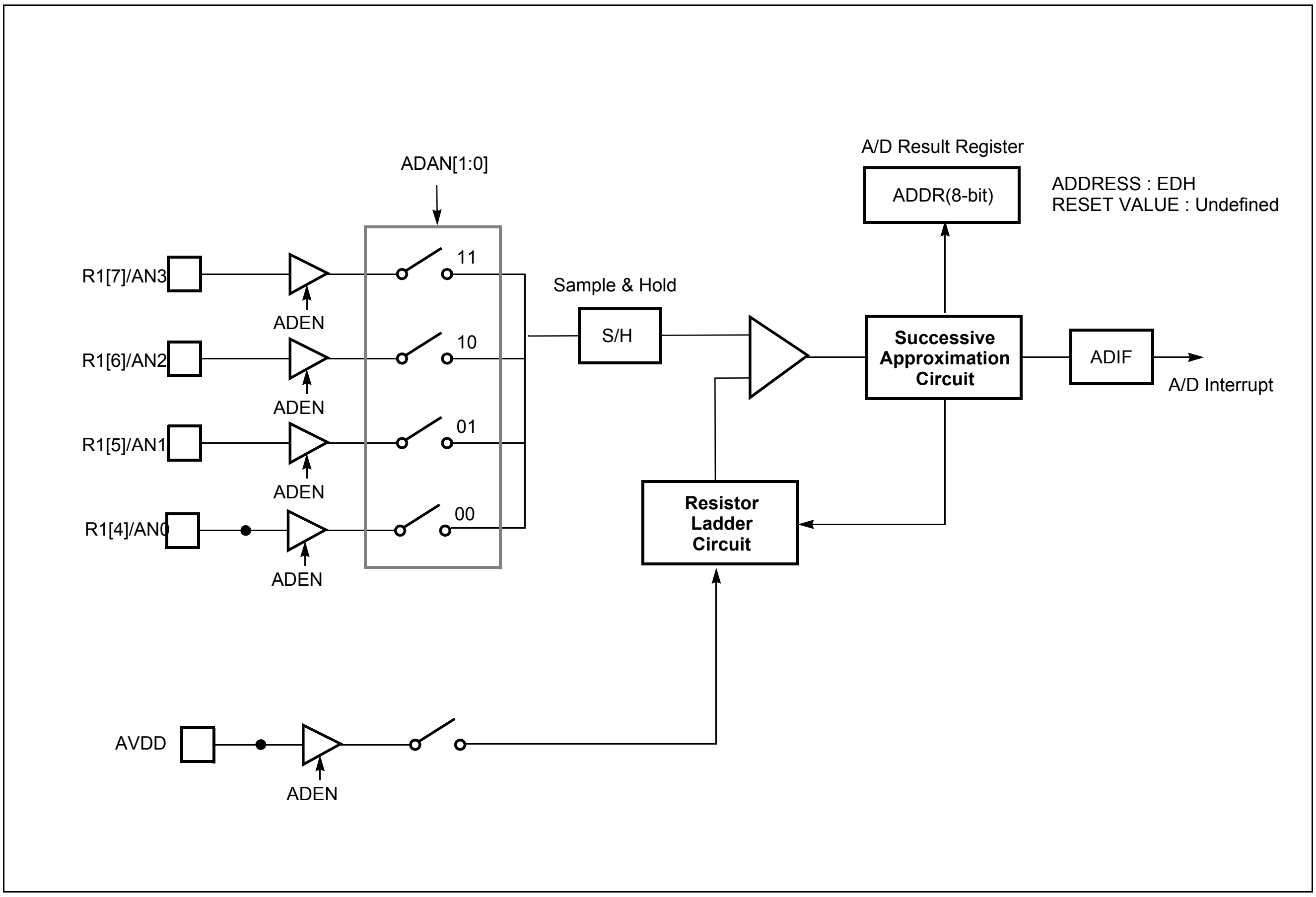


Figure 19-1 A/D Converter Block Diagram

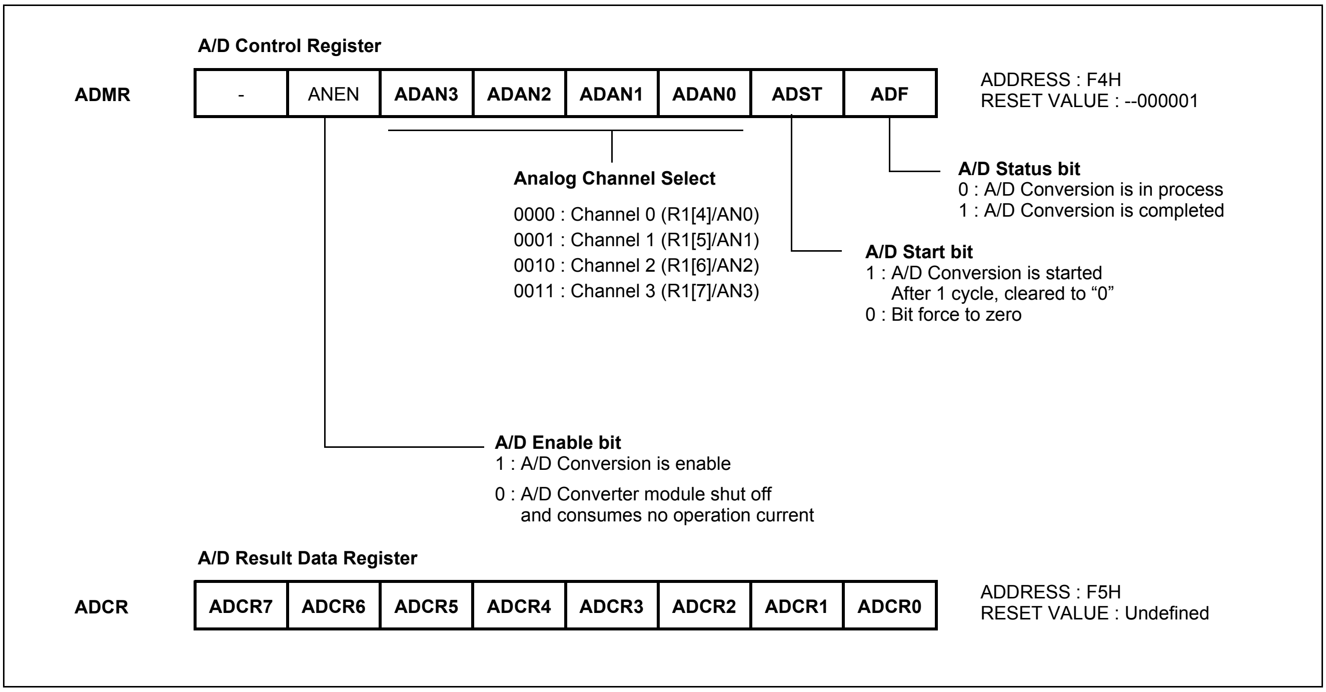

A/D Control Register
ADMR
-
ANEN
ADAN3
ADAN2
ADAN1
ADAN0
ADST
ADF
ADDRESS : F4H
RESET VALUE : --000001
Analog Channel Select
0000 : Channel 0 (R1[4]/AN0)
0001 : Channel 1 (R1[5]/AN1)
0010 : Channel 2 (R1[6]/AN2)
0011 : Channel 3 (R1[7]/AN3)
A/D Status bit
0 : A/D Conversion is in process
1 : A/D Conversion is completed
A/D Start bit
1 : A/D Conversion is started
After 1 cycle, cleared to "0"
0 : Bit force to zero
A/D Enable bit
1 : A/D Conversion is enable
0 : A/D Converter module shut off
and consumes no operation current
A/D Result Data Register
ADCR
ADCR7
ADCR6
ADCR5
ADCR4
ADCR3
ADCR2
ADCR1
ADCR0
ADDRESS : F5H
RESET VALUE : Undefined

Figure 19-2 A/D Converter Registers

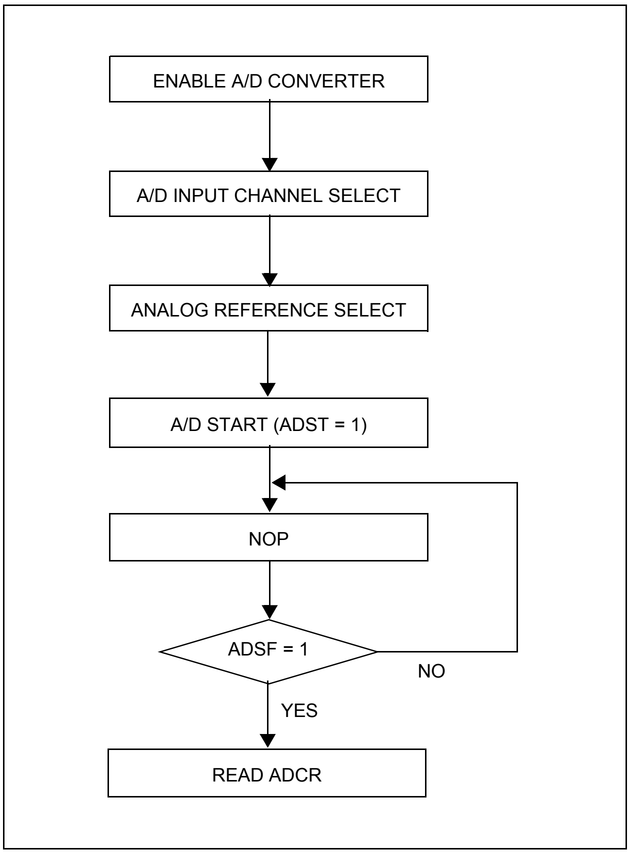


Figure 19-3 A/D Converter Operation Flow

A/D Converter Cautions

(1) Input range of AN0 to AN3

The input voltage of AN0 to AN3 should be within the specification range. In particular, if a voltage above VDD or below Vss is input (even if within the absolute maximum rating range), the conversion value for that channel can not be indeterminate. The conversion values of the other channels may also be affected.

(2) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins VDD and AN0 to AN3. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 19-4 in order to reduce noise.

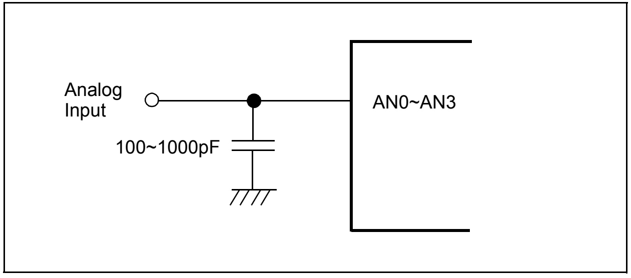


Figure 19-4 Analog Input Pin Connecting Capacitor

(3) Pins AN0/R1[4] and AN1/R1[5] to AN3/R1[7]

The analog input pins AN0 to AN3 also function as input/output port (PORT R1) pins. When A/D conversion is performed with any of pins AN0 to AN3 selected, be sure not to execute a PORT input instruction while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.