APRIL 2008 Rev. 3.2

8-BIT SINGLE CHIP MICROCOMPUTERS

GMS810 SERIES

USER`S MANUAL

- GMS81004
- GMS81008
- GMS81016
- GMS81024
- GMS81032

Revision 3.2

Published by
MCU Application Team in ABOV Semiconductor Co., Ltd.

Additional information of this manual may be served by ABOV Offices in Korea or Distributors and Representative listed at address directory.

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

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

Table of Contents

Chapter 5

Interrupt

Chapter 6

Standby Function

Chapter 7

Reset Function

Appendix

CHAPTER 1. OVERVIEW

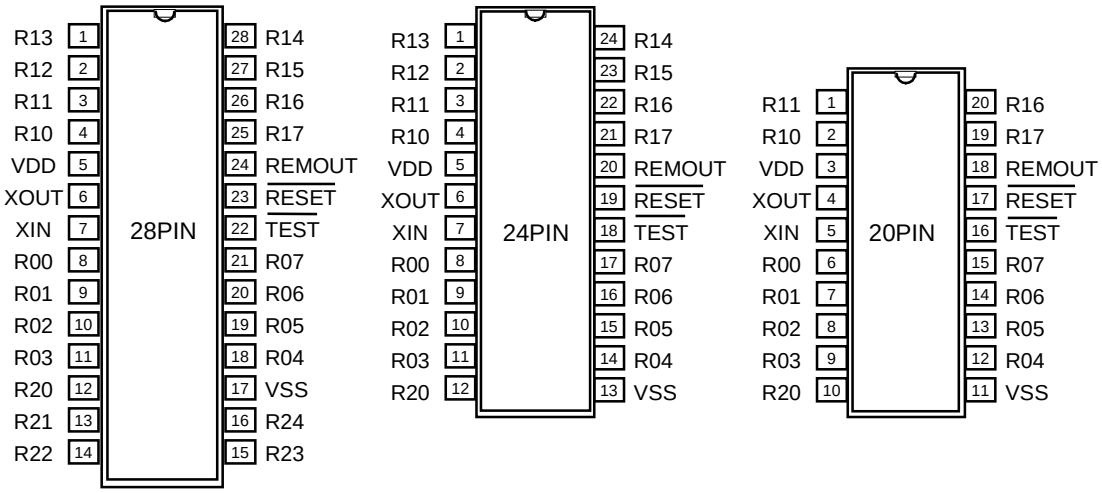
The GMS810 Series is the high speed and Low voltage operating 8-bit single chip microcomputers. This MCU contains G8MC core, ROM, RAM, input/output ports and five multi-function timer/counters.

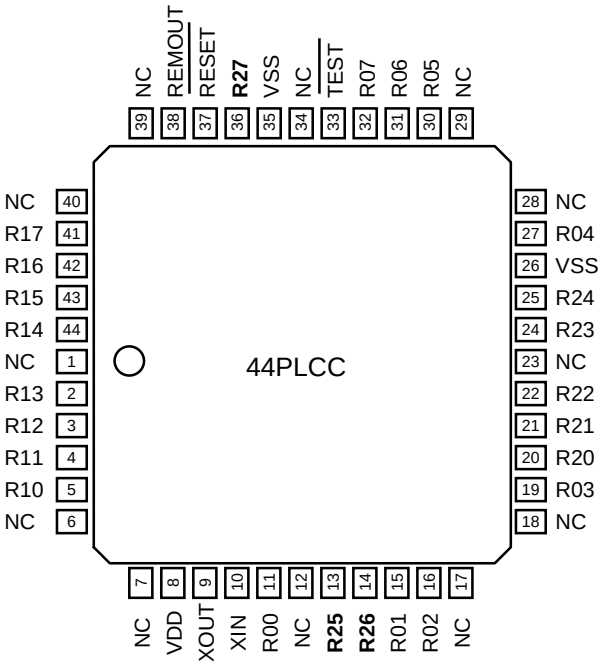
1.1 FEATURES & PIN ASSIGNMENTS (TOP VIEW)

- ROM size . . . . . 4,096 Bytes (GMS81004) , 8,192 Bytes (GMS81008)
  . . . . . 16,384 Bytes (GMS81016) ,24,576 Bytes(GMS81024)
  . . . . . 32,768 Bytes (GMS81032)
- RAM size . . . . . 448 Bytes
- Instruction Execution Time . . 1us @Xin=4MHz
- Timer
  - Timer/Counter . . . . . 8Bit * 2ch , 16Bit * 1ch
  - Basic Interval Time . . . 8Bit * 1ch
  - Watch Dog Timer . . . . 6Bit * 1ch
- Power On Reset
- Power Saving Operation Modes
  - STOP
- 8 Interrupt Sources
  - Nested Interrupt Control is Available
- Operating Voltage
  - 2.0~4.0V @2MHz
  - 2.2~4.0V @4MHz
- Low Voltage Detection Circuit
- Watch dog Timer Auto Start (During 1Second after Power on Reset)
- Package
  - 20SOP/20PDIP/24SOP/24Skinny DIP/28SOP/28Skinny DIP
  - 44PLCC
- I/O Port

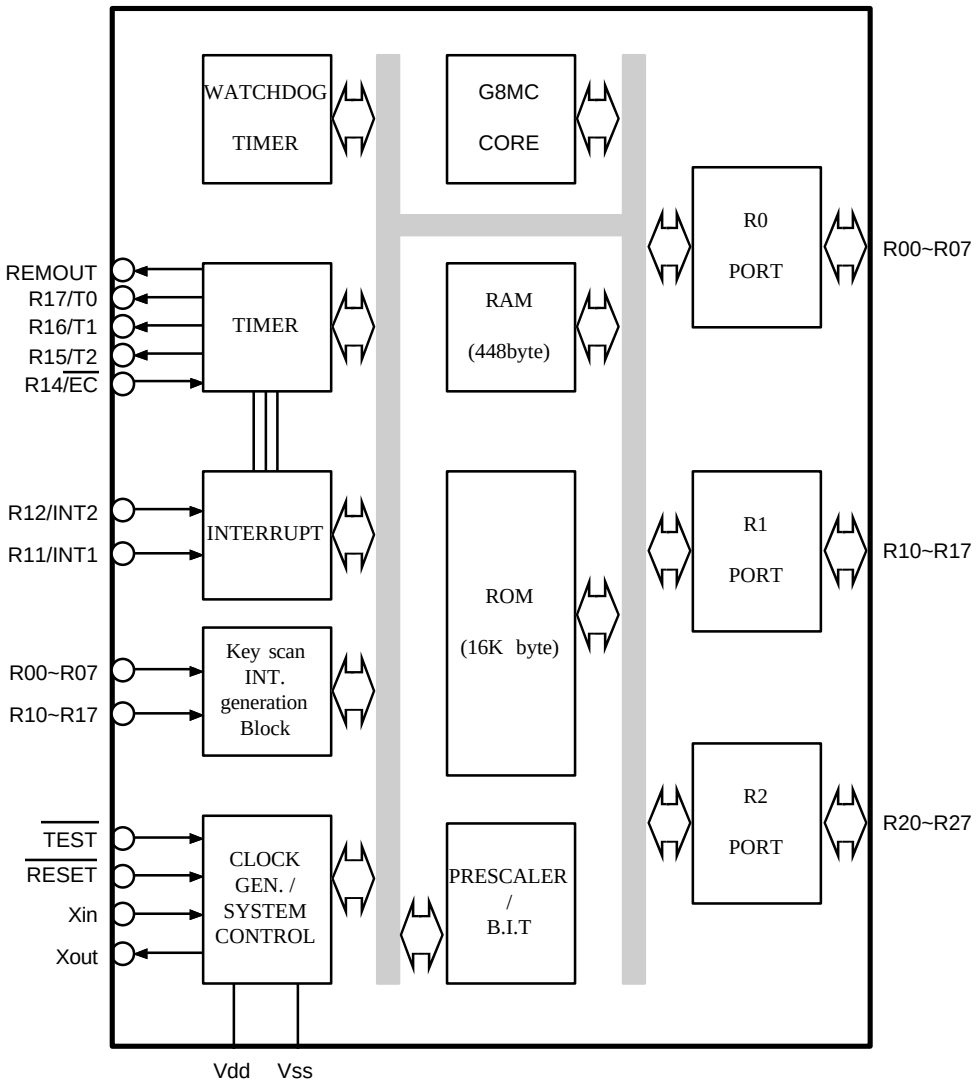
	20pin	24pin	28pin	44pin
input	3	3	3	3
output	2	2	2	2
I/O	13	17	21	24

PIN ASSIGNMENT



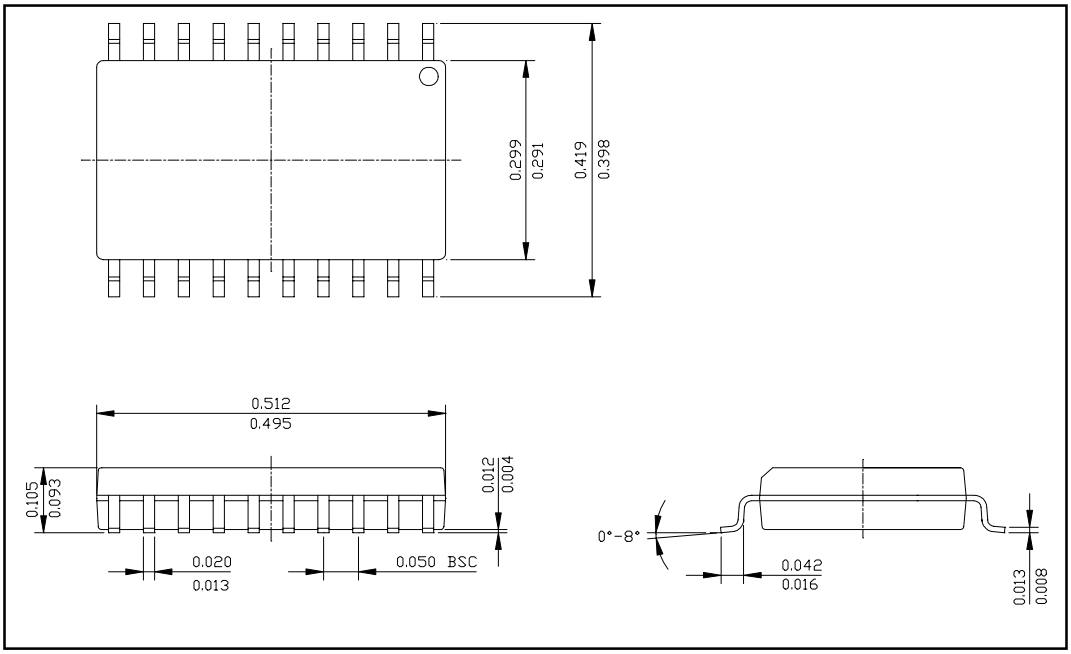


1.2 Block Diagram

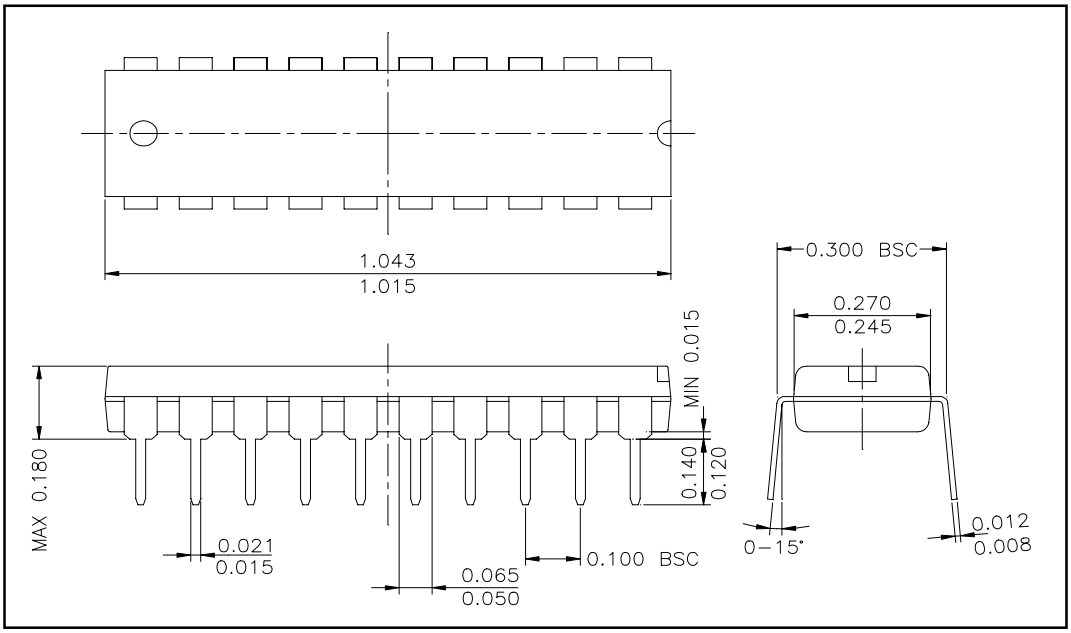


1.3 Package Dimension

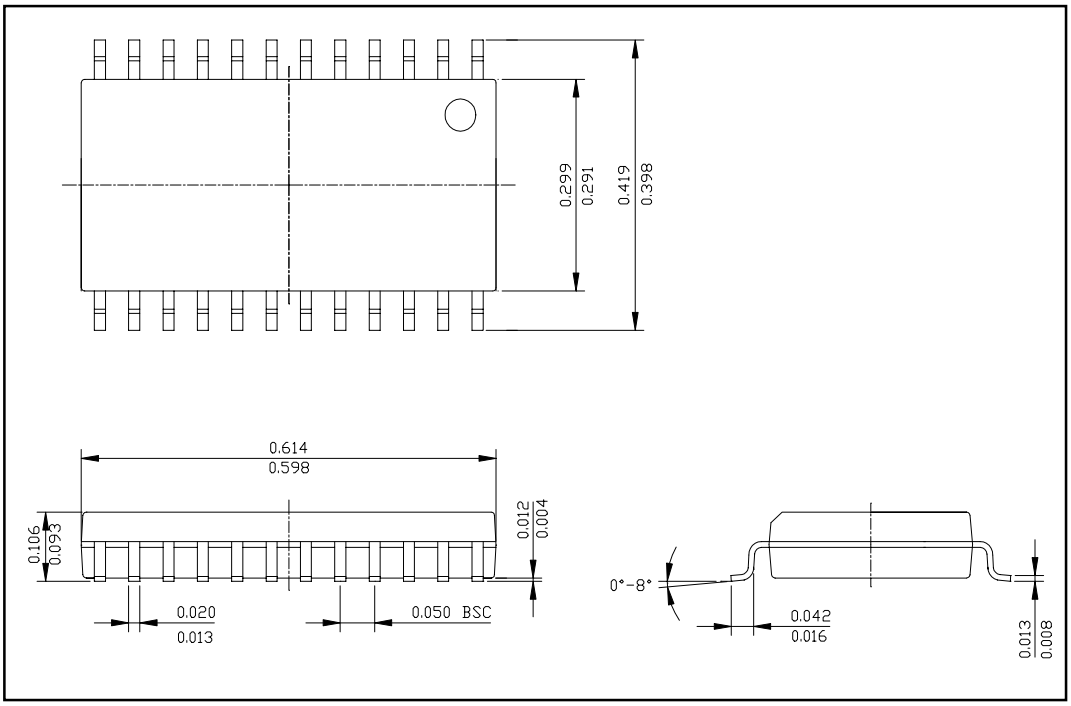
1.3.1 20SOP Pin Dimension(dimensions in inch)



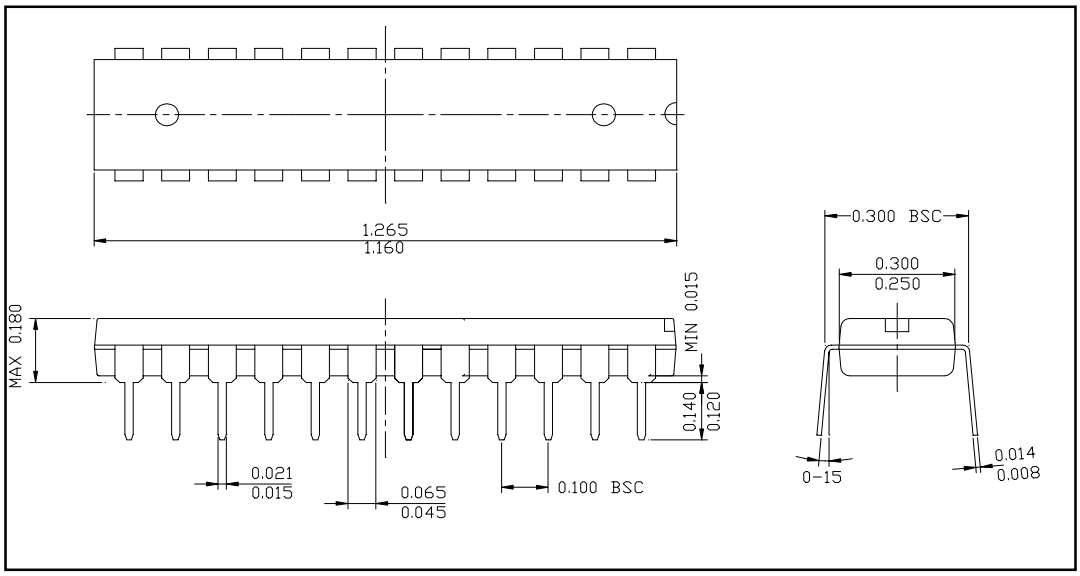
1.3.2 20PDIP Pin Dimension (dimensions in inch)



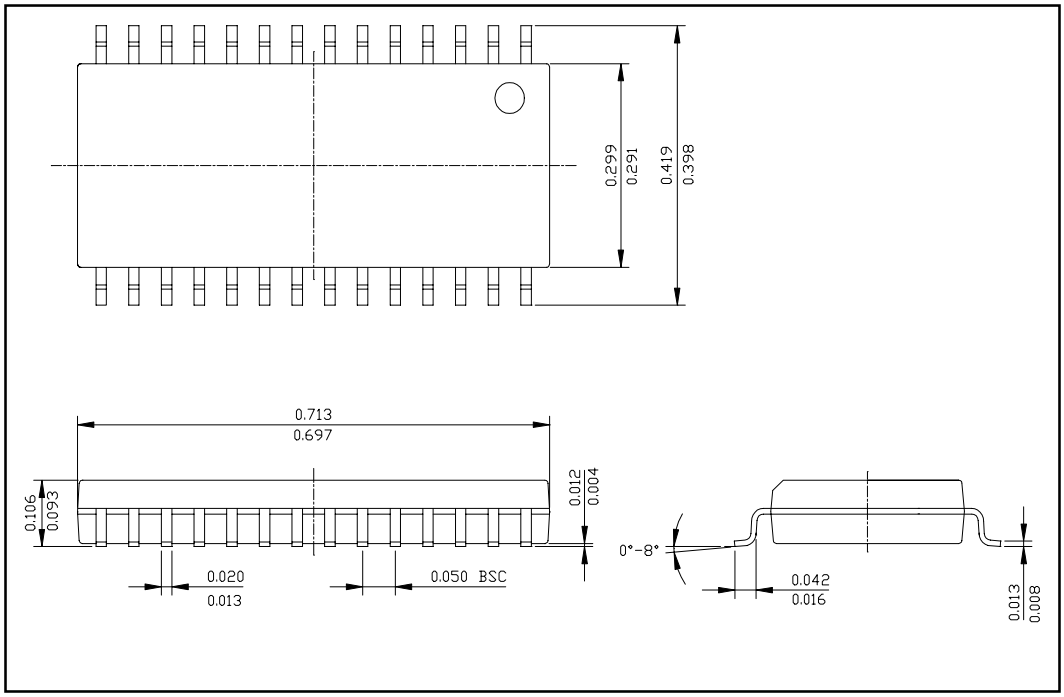
1.3.3 24SOP Pin Dimension (dimensions in inch)



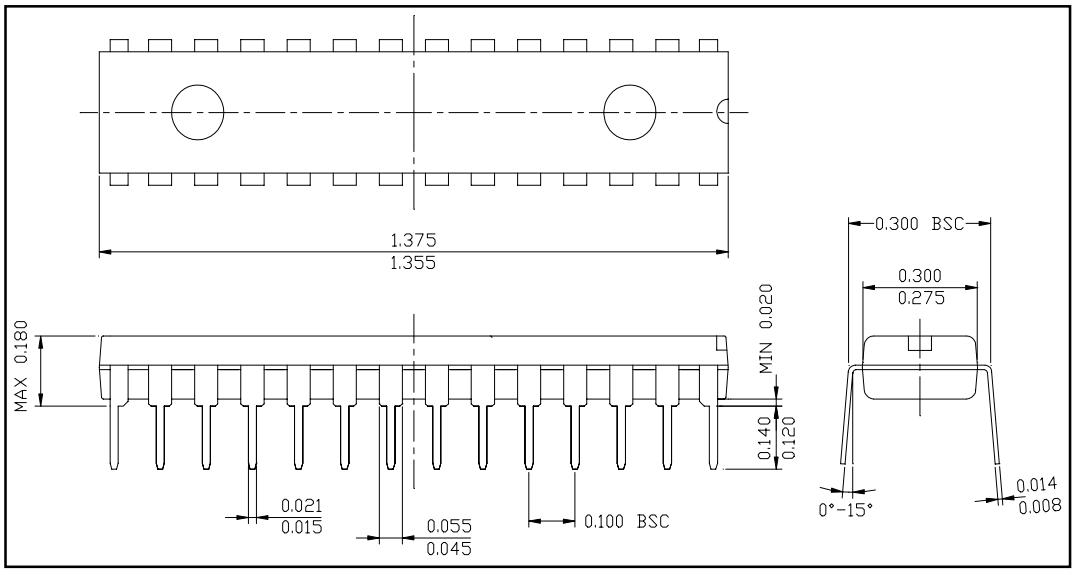
1.3.4 24skinnyDIP Pin Dimension (dimensions in inch)



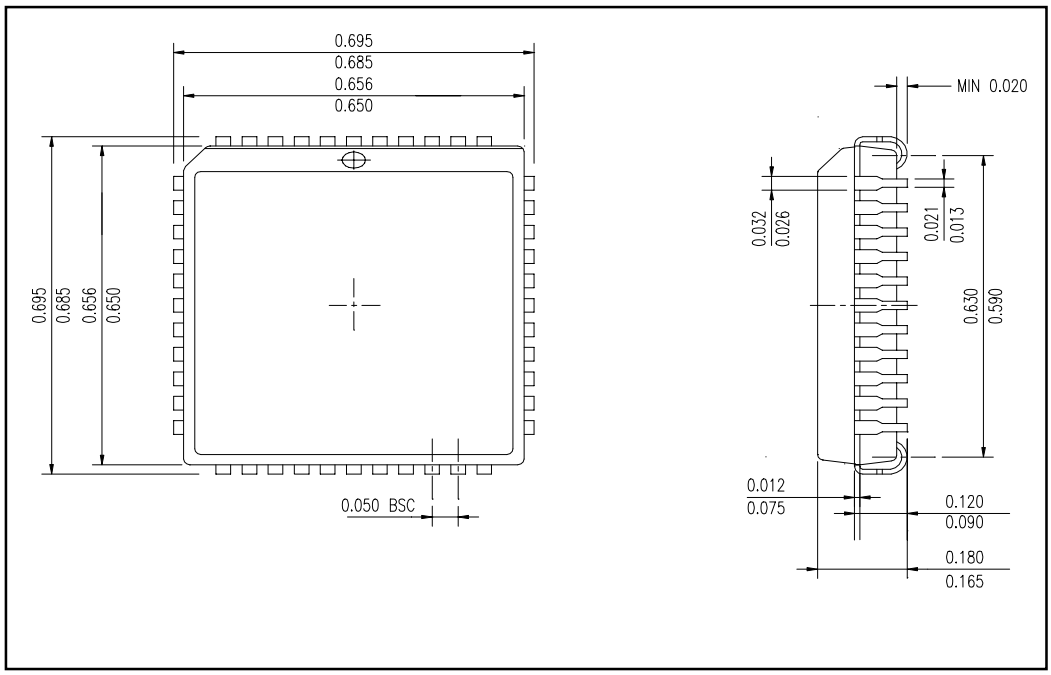
1.3.5 28SOP Pin Dimension (dimensions in inch)



1.3.6 28skinnyDIP Pin Dimension (dimensions in inch)



1.3.7 44PLCC Pin Dimension (dimensions in mm)

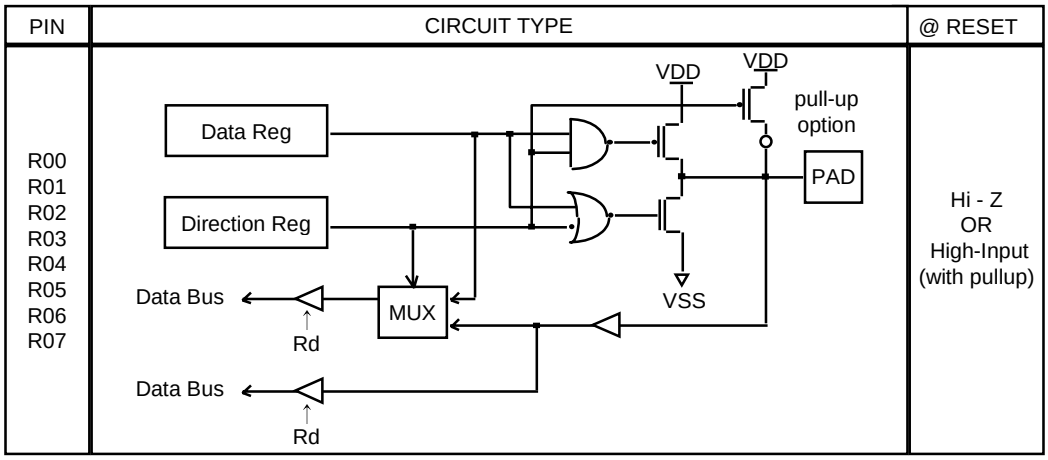


1.4 Pin Function

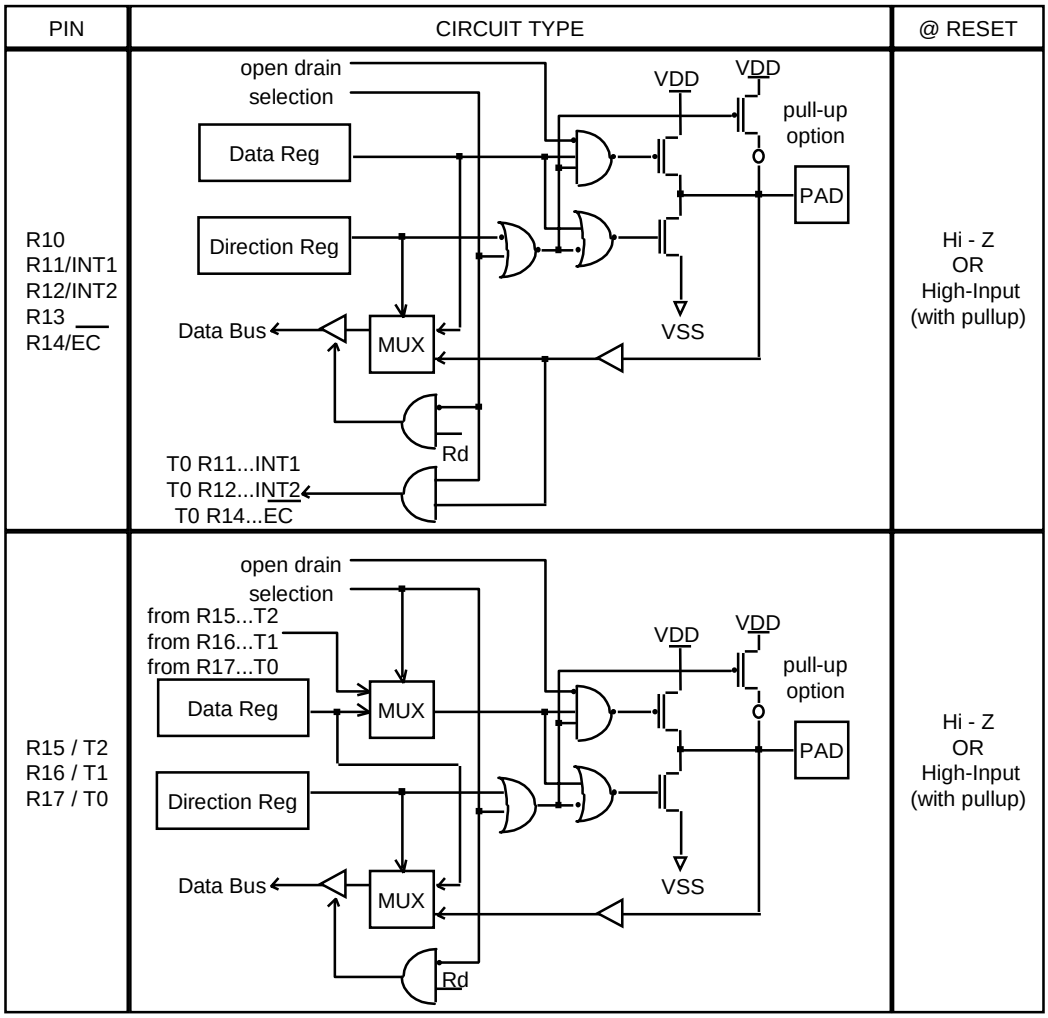
PIN NAME	INPUT/ OUTPUT	INPUT				Function	@ RESET	@ STOP
		20Pin	24Pin	28Pin	44Pin			
R00	I/O	6	8	8	11	- Each bit of the port can be individually configured as an input or an output by user software - Push-pull output - CMOS input with pull-up resistor (option) - Can be programmable as Key Scan Input - Pull-ups are automatically disabled at output mode	INPUT	State of before STOP
R01	I/O	7	9	9	15			
R02	I/O	8	10	10	16			
R03	I/O	9	11	11	19			
R04	I/O	12	14	18	27			
R05	I/O	13	15	19	30			
R06	I/O	14	16	20	31			
R07	I/O	15	17	21	32			
R10	I/O	2	4	4	5			
R11/INT1	I/O	1	3	3	4	- CMOS input with pull-up resistor (option) - Push-pull output - Can be programmable as Key Scan Input or Open drain output - Direct Driving of LED(N-TR) - Pull-ups are disabled at output mode	INPUT	State of before STOP
R12/INT2	I/O	-	2	2	3			
R13	I/O	-	1	1	2			
R14/EC	I/O	-	24	28	44			
R15/T2	I/O	-	23	27	43			
R16/T1	I/O	20	22	26	42			
R17/T0	I/O	19	21	25	41			
R20	I/O	10	12	12	20	- CMOS input with pull-up resistor (option) - Push-pull output - Direct Driving of LED(N-TR) - Pull-ups are disabled at output mode	INPUT	State of before STOP
R21	I/O	-	-	13	21			
R22	I/O	-	-	14	22			
R23	I/O	-	-	15	24			
R24	I/O	-	-	16	25			
R25	I/O	-	-	-	13			
R26	I/O	-	-	-	14			
R27	I/O	-	-	-	36			
XIN	I	5	7	7	10	- Oscillator Input - Oscillator Output		Low High
XOUT	O	4	6	6	9			
REMOUT	O	18	20	24	38	- High Current Output	`L` output	`L` Output
$\overline{\text{RESET}}$	I	17	19	23	37	- Includes pull-up resistor	`L` level	state of before STOP
$\overline{\text{TEST}}$	I	16	18	22	33	- Includes pull-up resistor		
VDD	P	3	5	5	8	- Positive power supply		
VSS	P	11	13	17	26	- Ground		
VSS	P	-	-	-	35			

1.5 Port Structure

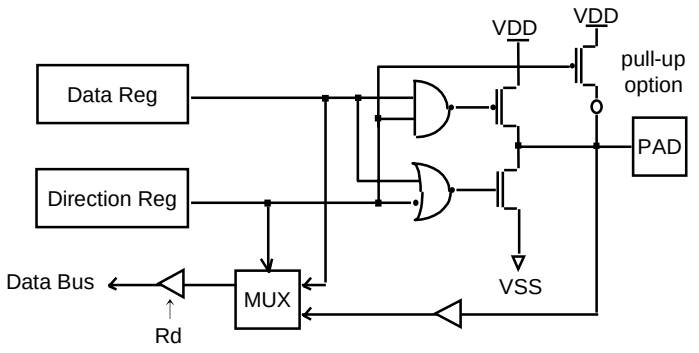
1.5.1 R0 PORT



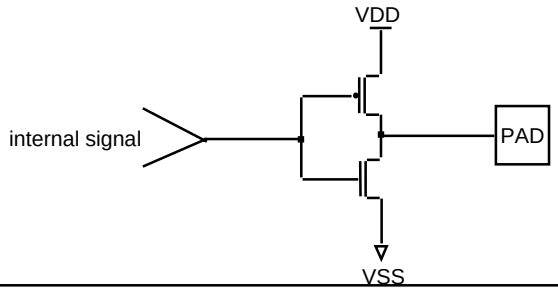
1.5.2 R1 PORT


PIN
CIRCUIT TYPE
@ RESET
R10
R11/INT1
R12/INT2
R13
R14/EC
open drain
selection
Data Reg
Direction Reg
Data Bus
MUX
VDD
VDD
pull-up
option
PAD
VSS
Rd
T0 R11...INT1
T0 R12...INT2
T0 R14...EC
Hi - Z
OR
High-Input
(with pullup)
R15 / T2
R16 / T1
R17 / T0
from R15...T2
from R16...T1
from R17...T0

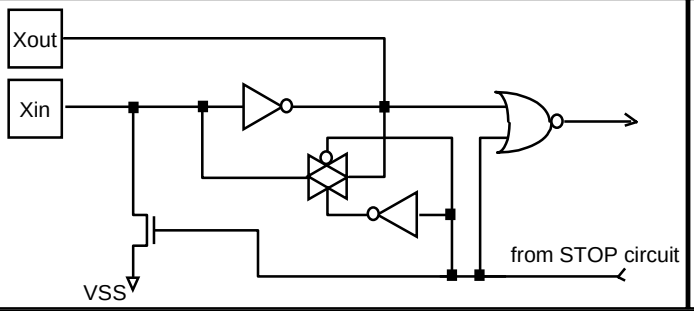
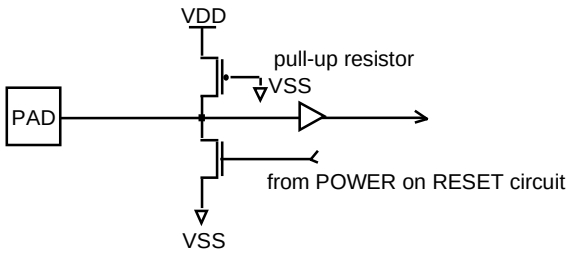
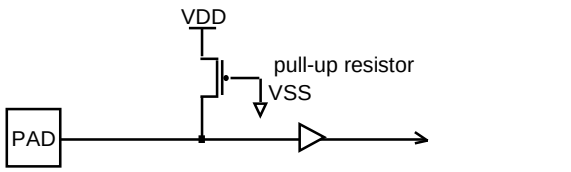
1.5.3 R2 PORT

PIN	CIRCUIT TYPE	@ RESET
R20 R21 R22 R23 R24 R25 R26 R27		Hi - Z OR High-Input (with pullup)

REMOUT PORT

PIN	CIRCUIT TYPE	@ RESET
REMOUT		Low level

1.5.4 Miscellaneous Ports

PIN	CIRCUIT TYPE	@ RESET
Xin Xout		oscillation
$\overline{\text{RESET}}$		Low level
$\overline{\text{TEST}}$		High level

1.6 Electrical Characteristics

1.6.1 Absolute Maximum Ratings (Ta = 25℃)

PARAMETER	SYMBOL	RATINGS	UNIT
Supply Voltage	VDD	-0.3 ~ +7.0	V
Input Voltage	VI	-0.3 ~ VDD + 0.3	V
Output Voltage	VO	-0.3 ~ VDD + 0.3	V
Operating Temperature	Topr	0 ~ 70	℃
Storage Temperature	Tstg	-65 ~ 150	℃
Power Dissipation	PD	700	mW

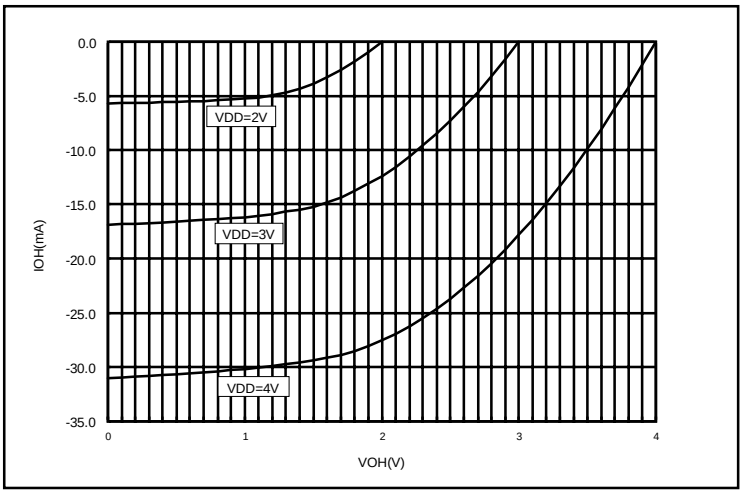
1.6.2 Recommended Operating Ranges

PARAMETER	SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Voltage	VDD1	fXin = 1MHz fXin = 2MHz	2.0		4.0	V
	VDD2	fXin = 4MHz	2.2		4.0	V
Oscillation Frequency	fXin		1.0	2.0	4.0	MHz
Operating Temperature	Topr		0		70	℃

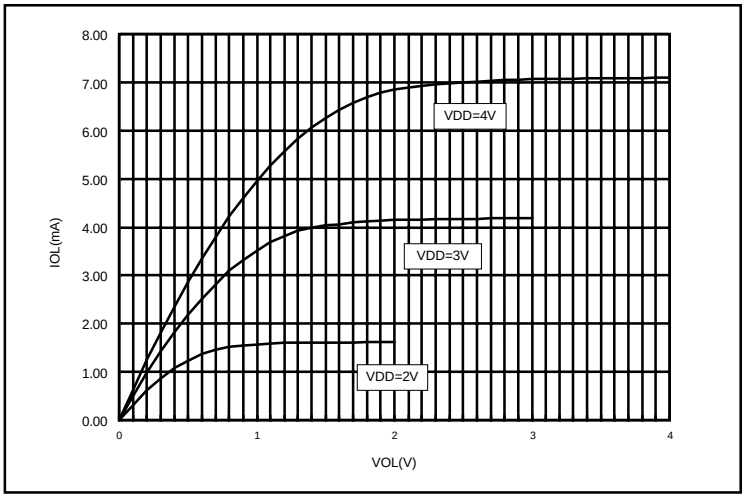
1.6.3 DC Characteristics (VDD = 2.0~4.0, Vss = 0V, Ta = 0℃ ~ 70℃)

Parameter	Symbol	Condition			Specification			Unit
					min	typ	max	
high level input voltage	$V_{IH}1$	R11, R12, R14, RESETB			0.8V_{DD}		V_{DD}	V
	$V_{IH}2$	R0, R1(Except R11,R12,R14) , R2			0.7V_{DD}		V_{DD}	V
low level input voltage	$V_{IL}1$	R11, R12, R14, RESETB			0		0.2V_{DD}	V
	$V_{IL}2$	R0, R1(Except R11,R12,R14) , R2			0		0.3V_{DD}	V
high level input leakage current	I_{IH}	R0,R1,R2,RESETB	$V_{IH}=V_{DD}$				1	uA
low level input leakage current	I_{IL}	R0,R1,R2,RESETB (without pull-up)	V_{IL}=0V				-1	uA
high level output voltage	$V_{OH}1$	R0	I_{OH}=-0.5mA		V_{DD}-0.4			V
	$V_{OH}2$	R1(ExceptR17),R2	I_{OH}=-1mA		V_{DD}-0.4			V
	$V_{OH}3$	R17	I_{OH}=-8mA		V_{DD}-0.9			V
	$V_{OH}5$	OSC	I_{OH}=-200uA		V_{DD}-0.9			V
low level output voltage	$V_{OL}1$	R0	I_{OL}=1mA				0.4	V
	$V_{OL}2$	R1, R2	I_{OL}=5mA				0.8	V
	$V_{OL}5$	OSC	I_{OL}=200uA				0.8	V
high level output leakage current	I_{OHL}	R0, R1, R2	$V_{OH}=V_{DD}$				1	uA
low level output leakage current	I_{OLL}	R0, R1, R2	V_{OL}=0V				-1	uA
high level output current	I_{OH}	REMOUT	V_{OH}=2V		-30	-12	-5	mA
low level output current	I_{OL}	REMOUT	V_{OL}=1V		0.5	-	3	mA
input pull-up current	I_P1	RESETB	V_{DD}=3V		15	30	60	uA
	I_P2	R0, R1, R2	V_{DD}=3V		10	20	40	uA
POWER SUPPLY CURRENT	I_{DD}	operating current	f_{XIN}=4MHz	V_{DD}=4V		4	10	mA
				V_{DD}=2.2V		2.4	6	mA
			f_{XIN}=2MHz	V_{DD}=4V		2.4	6	mA
				V_{DD}=2V		1.2	3	mA
	I_{STOP}	stop mode current	oscillator stop	V_{DD}=4V	---	3	10	uA
				V_{DD}=2V	---	2	8	uA
RAM retention supply voltage	VRET				0.7			V

● GMS810 Series REMOUT port I_{OH} Characteristics graph



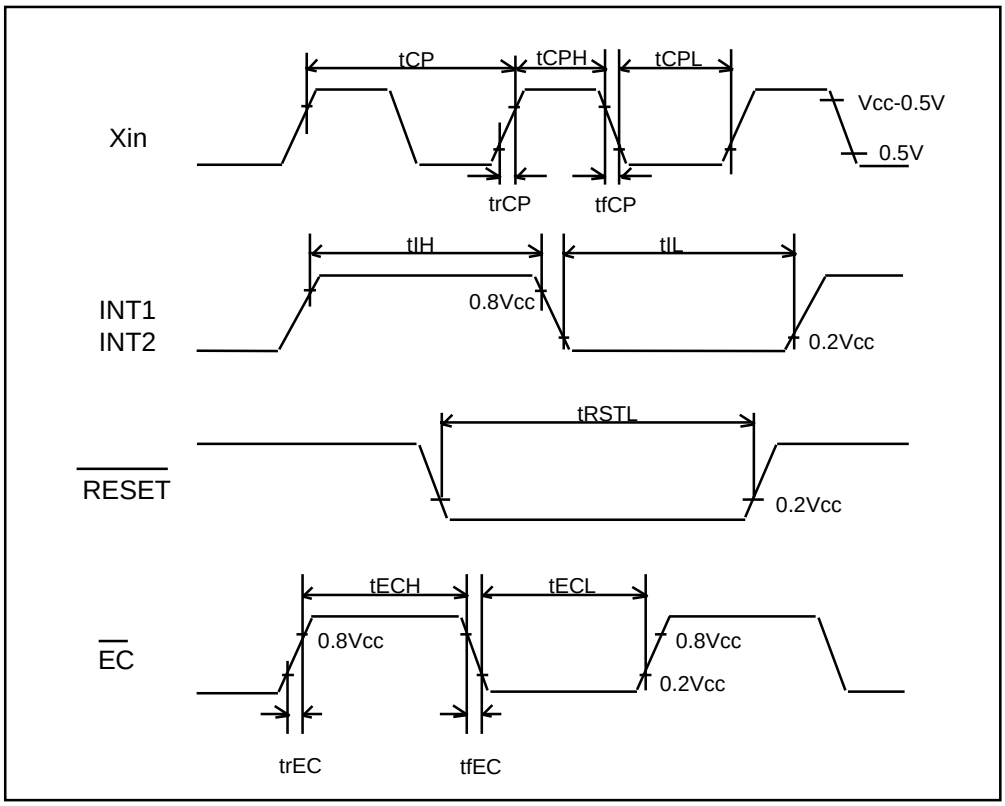
● GMS810 Series REMOUT port I_{OL} Characteristics graph



1.6.4 AC Characteristics (VDD = 2.0~4.0, Vss = 0V, Ta = 0℃ ~ 70℃)

No	Parameter	Symbol	Pin	Specification			Unit
				min	typ	max	
1	External clock input cycle time	tcp	Xin	250	500	1000	ns
2	System clock cycle time	tsys		500	1000	2000	ns
3	External clock pulse width High	tcpH	Xin	40			ns
4	External clock pulse width Low	tcpL	Xin	40			ns
5	External clock rising time	trcp	Xin			40	ns
6	External clock falling time	tfcp	Xin			40	ns
7	interrupt pulse width High	tIH	INT1~INT2	2			tsys
8	Interrupt pulse width Low	tIL	INT1~INT2	2			tsys
9	Reset input pulse width low	tRSTL	$\overline{\text{RESET}}$	8			tsys
10	Event counter input pulse width high	tECH	$\overline{\text{EC}}$	2			tsys
11	Event counter input pulse width low	tECL	$\overline{\text{EC}}$	2			tsys
12	Event counter input pulse rising time	trEC	$\overline{\text{EC}}$			40	ns
13	Event counter input pulse falling time	tfEC	$\overline{\text{EC}}$			40	ns

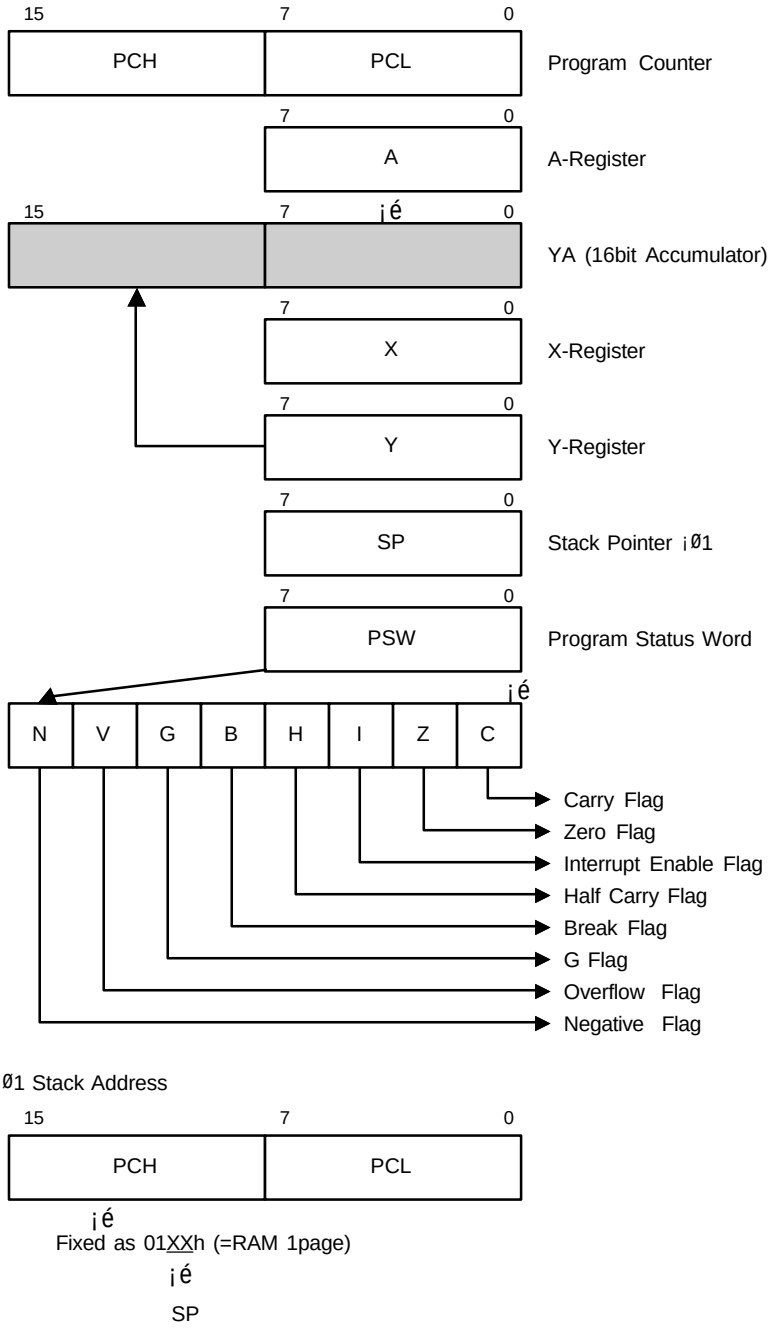
* Refer to Fig 1-1



* FIG-1 : Clock, INT, RESET. EC input timing

CHAPTER 2. FUNCTION DESCRIPTION

2.1 REGISTERS



2.1.1 A register

- 8bit Accumulator.
- In the case of 16-bit operation, compose the lower 8-bit of A, upper 8bit in Y (16-bit Accumulator)
- In the case of multiplication instruction, execute as a multiplier register. After multiplication operation, the lower 8-bit of the result enters. (Y*A ¡æ YA)
- In the case of division instruction, execute as the lower 8-bit of dividend. After division operation, quotient enters.

2.1.2 X register

- General-purpose 8-bit register
- In the case of index addressing mode within direct page(RAM area), execute as index register.
- In the case of division instruction, execute as register.

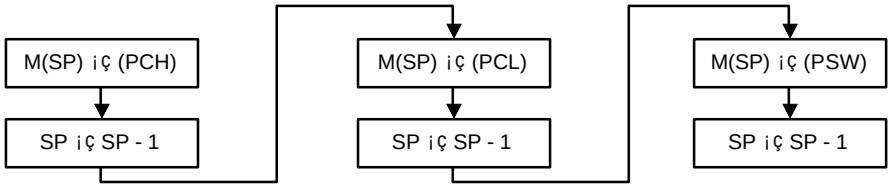
2.1.3 Y register

- General-purpose 8-bit register
- In the case of index addressing mode, execute as index register
- In the case of 16-bit operation instruction, execute as the upper 8-bit of YA (16-bit accumulator).
- In the case of multiplication instruction, execute as a multiplicand register. After multiplication operation, the upper 8-bit of the result enters.
- In the case of division instruction, execute as the upper 8-bit of dividend. After division operation, remains enters.
- Can be used as loop counter of conditional branch command. (e.g.DBNE Y, rel)

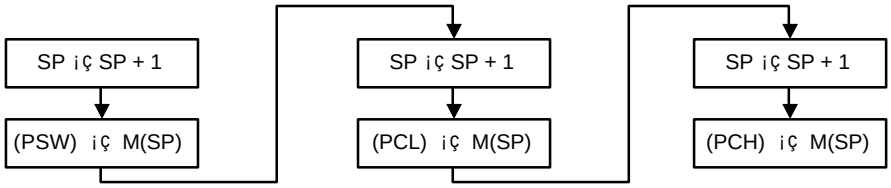
2.1.4 Stack Pointer

- In the cases of subroutine call, Interrupt and PUSH, POP, RETI, RET instruction, stack data on RAM or in the case of returning, assign the storage location having stacked data.
- Stack area is constrained within 1-page (00H-FFH). The SP is post-decremented when a subroutine call or a push instruction is executed, or when an interrupt is accepted; and the SP is pre-incremented when a return or a pop instruction is executed.
- SP should be initialized as follows
  ex) LDX #0FEH : 0FEH ¡æ X reg.
  TXSP : X reg. ¡æ SP
- The behaviors of stack pointer according to each instruction are the following.

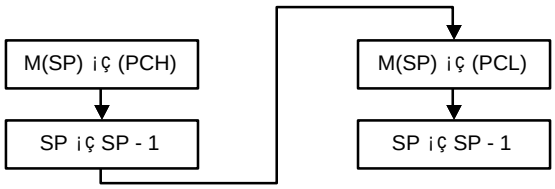
2.1.4.1 Interrupt



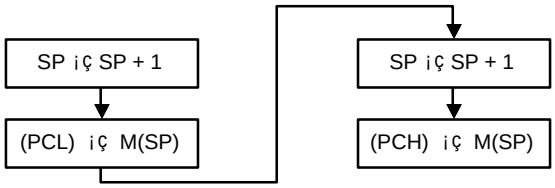
2.1.4.2 RETI(Return from interrupt)



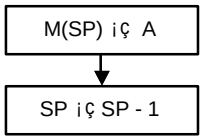
2.1.4.3 Subroutine call



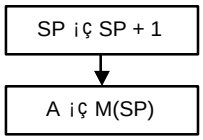
2.1.4.4 RET(Return from subroutine)



2.1.4.5 PUSH A(X, Y, PSW)



2.1.4.6 POP A(X, Y, PSW)



2.1.5 PC (Program Counter)

- Program counter is a 16-bit counter consisted of 8-bit register PCH and PCL.
- Addressing space is 64K bytes.

2.1.6 PSW (Program Status Word)

- PSW is an 8-bit register.
- Consisted of the flags showing the post state of operation and the flags determining the CPU operation, initialized as 00H in reset state.

2.1.7 Flag register.

2.1.7.1 Carry flag (C)

- After operation, set when there is a carry from bit7 of ALU or there is not a borrow.
- Set by SETC and clear by CLRC.
- Executable as 1-bit accumulator.
- Branch condition flag of BCS, BCC.

2.1.7.2 Zero flag (Z)

- After operation also including 16-bit operatiion, set if the result is ¡È0¡È
- Branch condition flag of BEQ, BNE.

2.1.7.3 Interrupt enable flag (I)

- Master enable flag of interrupt except for RST (reset).
- Set and cleared by EI, DI

2.1.7.4 Half carry flag (H)

- After operation, set when there is a carry from bit3 of ALU or there is not a borrow from bit4 of ALU.
- Can not be set by any instruction.
- Cleared by CLRV instruction like V flag.

2.1.7.5 Break flag (B)

- Set by BRK (S/W interrupt) instruction to distinguish BRK and TCALL instruction having the same vector address.

2.1.7.6 G flag (G)

- Set and cleared by SETG, CLRG instruction.
- Assign direct page (0-page, 1-page).
- Addressable directly to RAM 1-page by SETG. and to RAM 0-page by CLRG.

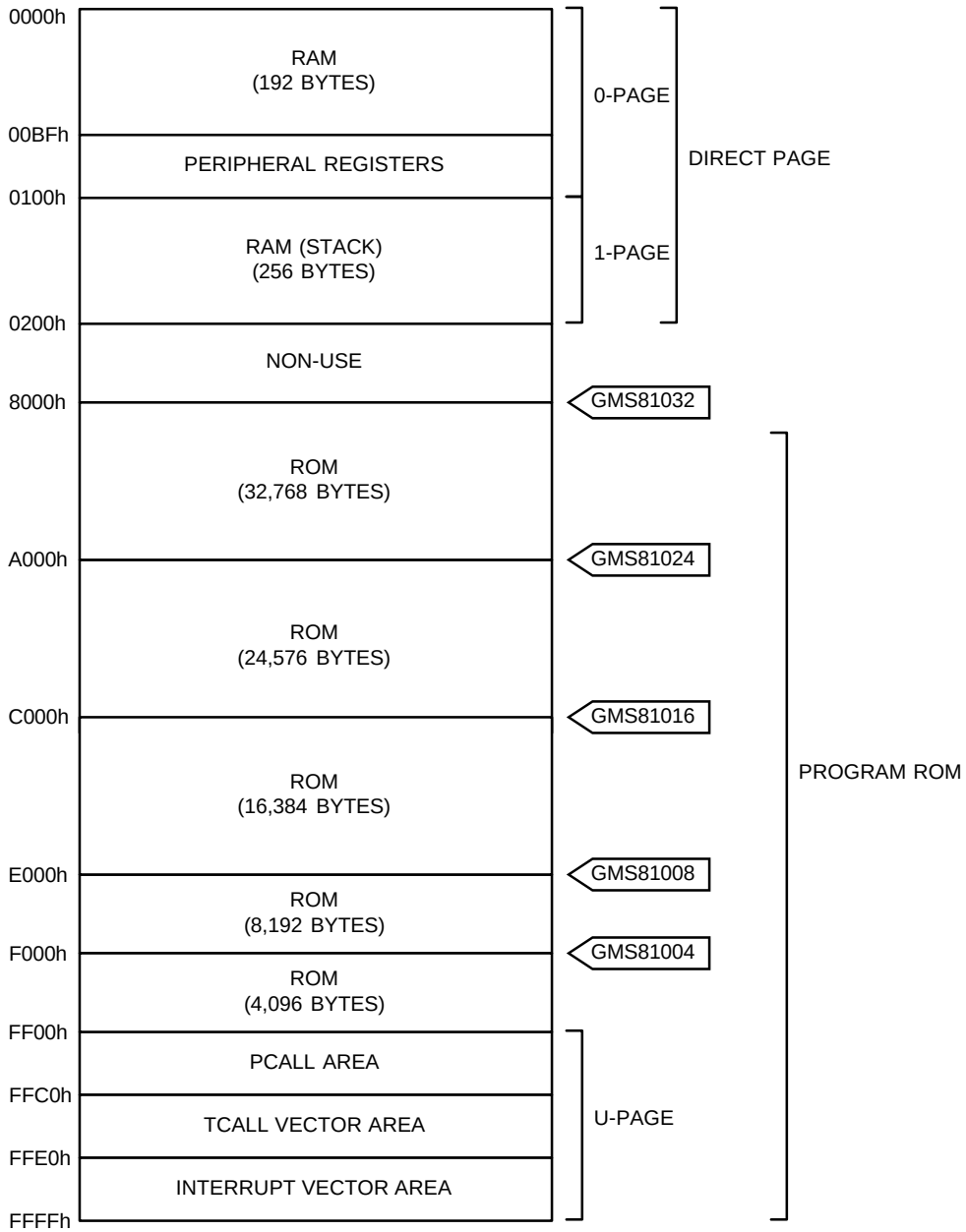
2.1.7.7 Overflow flag (V)

- After operation, set when overflow or underflow occurs.
- In the case of BIT instruction, bit6 memory location is transferred to V-flag.
- Cleared by CLRV instruction, but not set by any instruction.
- Branch condition flag of BVS, BVC.

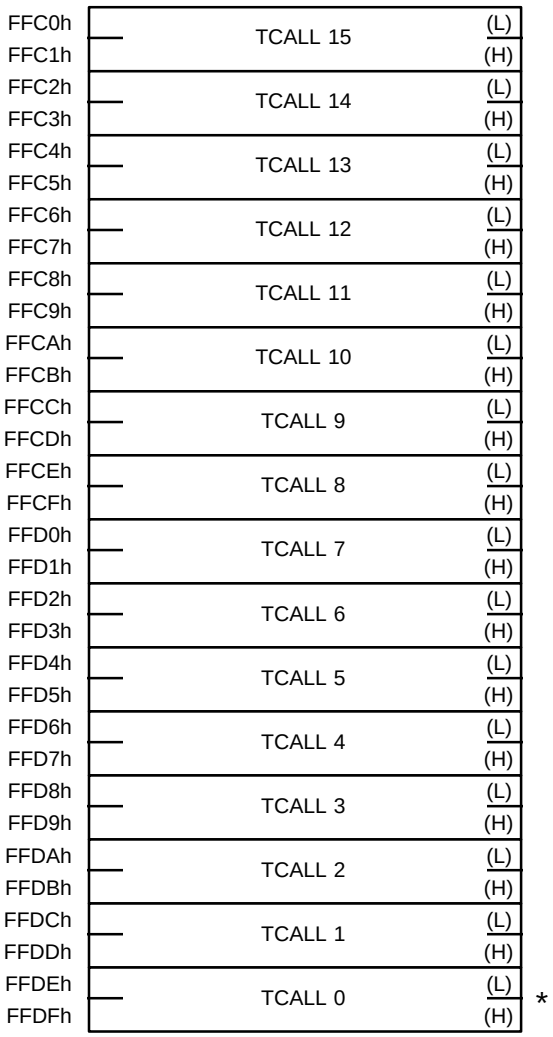
2.1.7.8 Negative flag (N)

- Set whenever the result of a data transfer or operation is negative (bit7 is set to ¡È1¡È).
- In the case of BIT instruction, bit7 of memory location is transferred to N-flag
- N-flag is not affected by CLR or SET instruction.
- Branch condition flag of BPL, BMI.

2.2 MEMORY MAP



2.3 TCALL VECTOR AREA



* This vector area is used in BRK command and TCALL0 command.

2.4 ZERO-PAGE PERIPHERAL REGISTERS

ADDRESS	FUNCTION REGISTERS	R/W	SYMBOL	RESET VALUE							
				7	6	5	4	3	2	1	0
00C0H	PORT R0 DATA REG.	R/W	R0	Undefined							
00C1H	PORT R0 DATA DIRECTION REG.	W	R0DD	00							
00C2H	PORT R1 DATA REG.	R/W	R1	Undefined							
00C3H	PORT R1 DATA DIRECTION REG.	W	R1DD	00							
00C4H	PORT R2 DATA REG.	R/W	R2	Undefined							
00C5H	PORT R2 DATA DIRECTION REG.	W	R2DD	00							
00C6H	Reserved										
00C7H	CLOCK CONTROL REG.	W	CKCTLR	-	-	1	1	0	1	1	1
	BASIC INTERVAL REG.	R	BITR	Undefined							
00C8H	WATCH DOG TIMER REG.	W	WDTR	-	0	0	0	1	1	1	1
00C9H	PORT R1 MODE REG.	W	PMR1	00							
00CAH	INT. MODE REG.	R/W	IMOD	-	-	0	0	0	0	0	0
00CBH	EXT. INT. EDGE SELECTION	W	IEDS	00							
00CCH	INT. ENABLE REG. HIGH	R/W	IENL	-	0	0	-	-	-	-	-
00CDH	INT. REQUEST FLAG REG. LOW	R/W	IRQL	-	0	0	-	-	-	-	-
00CEH	INT. ENABLE REG. HIGH	R/W	IENH	0	0	0	-	0	0	0	-
00CFH	INT. REQUEST FLAG REG. HIGH	R/W	IRQH	0	0	0	-	0	0	0	-
00D0H	TIMER 0 (16bit) MODE REG.	R/W	TM0	00							
00D1H	TIMER 1 (8bit) MODE REG.	R/W	TM1	00							
00D2H	TIMER 2 (8bit) MODE REG.	R/W	TM2	00							
00D3H	TIMER 0 HIGH-MSB DATA REG.	W	T0HMD	Undefined							
00D4H	TIMER 0 HIGH-LSB DATA REG.	W	T0HLD	Undefined							
00D5H	TIMER0 LOW-MSB DATA REG.	W	T0LMD	Undefined							
	TIMER0 LOW-MSB COUNT REG.	R		Undefined							
00D6H	TIMER0 LOW-LSB DATA REG.	W	T0LLD	Undefined							
	TIMER0 LOW-LSB COUNT REG.	R		Undefined							
00D7H	TIMER 1 HIGH DATA REG.	W	T1HD	Undefined							
00D8H	TIMER1 LOW DATA REG.	W	T1LD	Undefined							
	TIMER1 LOW COUNT REG.	R		Undefined							
00D9H	TIMER2 DATA REG.	W	T2DR	Undefined							
	TIMER2 COUNT REG.	R		Undefined							
00DAH	TIMER 0/ TIMER1 MODE REG.	R/W	TM01	00							
00DBH	Reserved										
00DCH	STANDBY MODE RELEASE REG0	W	SMRR0	00							
00DDH	STANDBY MODE RELEASE REG1	W	SMRR1	00							
00DEH	PORT R1 OPEN DRAIN ASSIGN REG.	W	R1ODC	00							

- ; Not used

* Caution : Write only register can not be accessed by bit manipulation instruction.

: Do not access the Reserved registers .

CHAPTER 3. I/O PORTS

The GMS810series has 21 I/O ports which are PORT0(8 I/O), PORT1 (8 I/O) and PORT2 (8 I/O). Each port contains data direction register which controls I/O and data register which stores port data.

3.1 PORT R0

3.1.1 PORT R0 Registers

REGISTER	SYMBOL	R/W	RESET VALUE	ADDRESS
R0 I/O Data Direction Register	R0DD	W	00H	00C1H
R0 Data Register	R0	R/W	Undefined	00C0H

Table 3.1 Port R0 Registers

3.1.2 I/O Data Direction Register (R0DD)

bit	7	6	5	4	3	2	1	0	
R0DD	R0DD7	R0DD6	R0DD5	R0DD4	R0DD3	R0DD2	R0DD1	R0DD0	<00C1H>
initial value	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	

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 ¡È1¡È, port R0 is in the output state, and if ¡È0¡È, it is in the input state. R0DD is write-only register. Since R0DD is initialized as ¡È00H¡È in reset state, the whole port R0 becomes input state.

3.1.1 Data Register(R0)

bit	7	6	5	4	3	2	1	0	
R0	R07	R06	R05	R04	R03	R02	R01	R00	<00C0H>
initial value	X	X	X	X	X	X	X	X	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PORT0 data register (R0) is 8-bit register to store data of port R0.
When setted 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.

3.2 PORT R1

PIN NAME	PORT SELECTION	FUNCTION SELECTION
R10	R10(I/O)	
R11/INT1	R11(I/O)	INT1 (INPUT)
R12/INT2	R12(I/O)	INT2 (INPUT)
R13	R13(I/O)	
R14/$\overline{EC}$	R14(I/O)	$\overline{EC}$ (INPUT)
R15/T2	R15(I/O)	T2 (OUTPUT)
R16/T1	R16(I/O)	T1 (OUTPUT)
R17/T0	R17(I/O)	T0 (OUTPUT)

Fig 3.1 Pin Function of port R1

3.2.1 PORT R1 Register

REGISTER	SYMBOL	R/W	RESET VALUE	ADDRESS
R1 I/O Data Direction Register	R1DD	W	00H	00C3H
R1 Data Register	R1	R/W	Undefined	00C2H
R1 Port Mode Register	PMR1	W	00H	00C9H
R1 Port Open drain Assign Register	R10DC	W	00H	00CEH

Table 3.1 Port R1 Registers

3.2.2 I/O Data Direction Register (R1DD)

bit	7	6	5	4	3	2	1	0	
R1DD	R1DD7	R1DD6	R1DD5	R1DD4	R1DD3	R1DD2	R1DD1	R1DD0	<00C3H>
initial value	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	

R1 Data Direction Register(R1DD) is 8-bit register, and can assign input state or output state to each bit. If R1DD is ¡È1¡È, port R1 is in the output state, and if ¡È0¡È, it is in the input state. R1DD is write-only register. Since R1DD is initialized as ¡È00H¡È in reset state, the whole port R1 becomes input state.

3.2.3 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 output into R1 pin.
The initial value of R1 is unknown in reset state.

bit	7	6	5	4	3	2	1	0	
R1	R17	R16	R15	R14	R13	R12	R11	R10	<00C2H>
initial value	X	X	X	X	X	X	X	X	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

3.2.4 Port R1 Open drain Assign Register (R1ODC)

bit	7	6	5	4	3	2	1	0	
R1ODC	R17OD	R16OD	R15OD	R14OD	R13OD	R12OD	R11OD	R10OD	<00DEH>
initial value	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	

Port R1 Open Drain Assign Register(R1ODC) is 8bit register, and can assign R1 port as open drain output port each bit, if corresponding port is selected as output. If R1ODC is selected as ¡È1¡È, port R1 is open drain output, and if selected as¡È0¡È, it is push-pull output. R1ODC is write-only register and initialized as ¡È00H¡È in reset state.

3.2.5 Port R1 Mode Register (PMR1)

bit	7	6	5	4	3	2	1	0	
PMR1	T0S	T1S	T2S	ECS	-	INT2S	INT1S	-	<00C9H>
initial value	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	

R1 Port Mode Register(PMR1) is 8-bit register, and can assign the selection mode for each bit. When set as¡È0¡È, corresponding bit of PMR1 acts as port R1 selection mode, and when set as¡È1¡È, it becomes function selection mode.

BIT NAME	PMR1	Selection Mode	Remarks
T0S	0	R17 Sel(I/O)	-
	1	T0 Sel (Output)	Output Port of Timer0
T1S	0	R16 Sel (I/O)	-
	1	T1 Sel (Output)	Output Port of Timer1
T2S	0	R15 Sel (I/O)	-
	1	T2 Sel (Output)	Output Port of Timer2
ECS	0	R14 Sel (I/O)	-
	1	$\overline{EC}$ Sel (Input)	Input Port of Timer0 Event Input
-	0		
	1		
INT2S	0	R12 sel (I/O)	-
	1	INT2 Sel (Input)	Input Port of Timer0 Input capture
INT1S	0	R11 Sel (I/O)	-
	1	INT1 Sel (Input)	-
-			

Table 3.3 Selection Mode of PMR1

PMR1 is write-only register and initialized as ¡È 00H¡È 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 ¡È0¡È.

3.3 PORT R2

3.3.1 PORT R2 Registers

REGISTERS	SYMBOL	R/W	RESET VALUE	ADDRESS
R2 I/O Data Direction Register	R2DD	W	00H	00C5H
R2 Data Register	R2	R/W	Undefined	00C4H

Table 3.3 Port R2 Registers

3.3.2 I/O Data Direction Register (R2DD)

bit	7	6	5	4	3	2	1	0	
R2DD	R2DD7	R2DD6	R2DD5	R2DD4	R2DD3	R2DD2	R2DD1	R2DD0	<00C5H>
initial value	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	

R2 Data Direction Register(R2DD) is 8-bit register, and can assign input state or output state or output state to each bit. If R2DD is ¡È1¡È, port R2 is in the output state, and if ¡È0¡È, it is in the input state.
R2DD is write-only register. Since R2DD is initialized as ¡È00H¡È in reset state, the whole port R2 becomes input state.

3.3.3 Data Register (R2)

bit	7	6	5	4	3	2	1	0	
R2	R27	R26	R25	R24	R23	R22	R21	R20	<00C4H>
initial value	X	X	X	X	X	X	X	X	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

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

CHAPTER 4. PERIPHERAL HARDWARE

4.1 CLOCK GENERATING CIRCUIT

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.

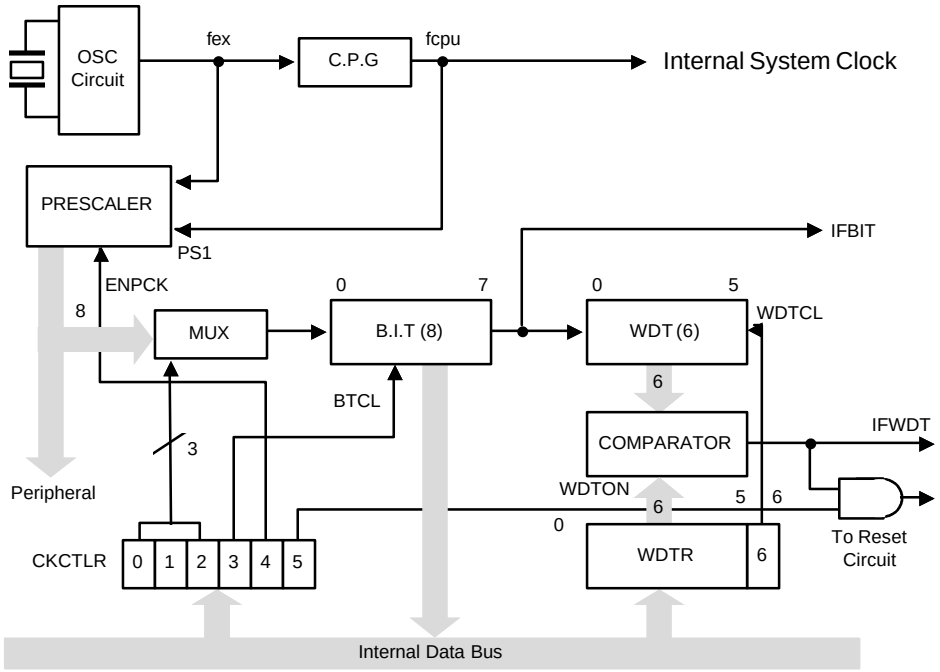
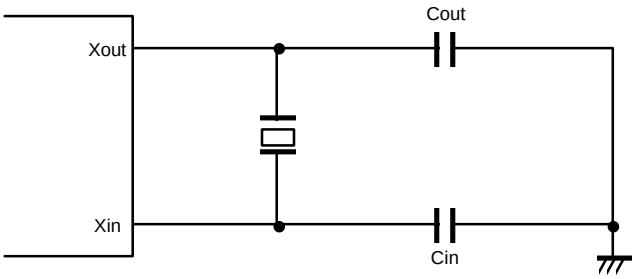


Fig. 4.1 Block diagram of clock generating circuit

4.1.1 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 ¡ÈHIGH¡È, Xin state goes to ¡ÈLOW¡È, and built-in feed back resistor is disabled.

(a) External Crystal (Ceramic) oscillator circuit



(b) External clock input circuit

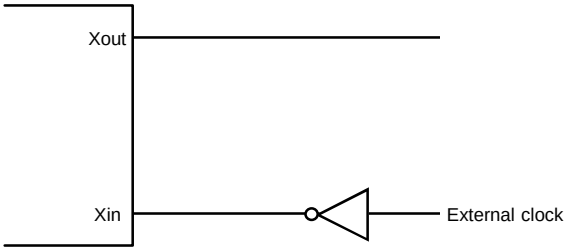


Fig. 4.2 Oscillator configurations

***. Recommendable resonator**

Frequency	Resonator Maker	Part Name	Load Capacitor	Operating Voltage
4.0MHz	CQ	ZTA4.00MG	Cin=Cout=30pF	2.2 ~ 4.0V
	KYOCERA	KBR- 4.0MKC	Cin=Cout=open	2.2 ~ 4.0V
	KYOCERA	KBR- 4.0MSB	Cin=Cout=33pF	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	Cin=Cout=open	2.2 ~ 4.0V

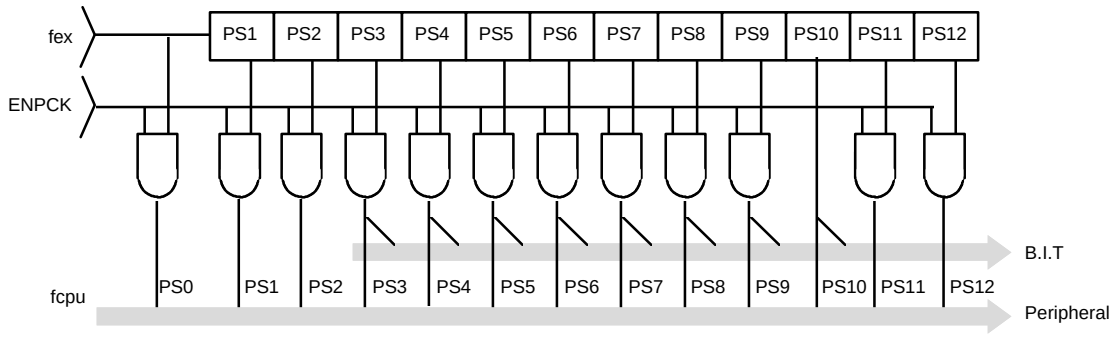
¡Ø MC type is building in load capacitior.CCR type is chip type.

4.1.2 Prescaler

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.

4.1.3 Peripheral hardware clock control

Clock to peripheral hardware can be stopped by bit4 (ENPCK) of CKCTLR Register. ENPCK is set to ¡È1¡È in reset state.



fex(MHz)		PS0	PS1	PS2	PS3	PS4	PS5	PS6	PS7	PS8	PS9	PS10	PS11	PS12
4	Freq	4M	2M	1M	500K	250K	125K	62.5K	31.25K	15.63K	7.183K	3.906K	1.953K	0.976K
	Period(s)	250n	500n	1u	2u	4u	8u	16u	32u	64u	128u	256u	512u	1024u
2	Freq	2M	1M	500k	250K	125K	62.5K	31.25K	15.63K	7.183K	3.906K	1.953K	0.976K	0.488K
	Period(s)	500n	1u	2u	4u	8u	16u	32u	64u	128u	256u	512u	1024u	2048u

Fig. 4.3 Block diagram of Prescaler

Clock Control Register

	7							0	
CKCTLR	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7H>

ENPCK	Peripheral Clock
0	Stopped
1	Provided

4.1.4 Basic Interval Timer (B.I.T)

- 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.

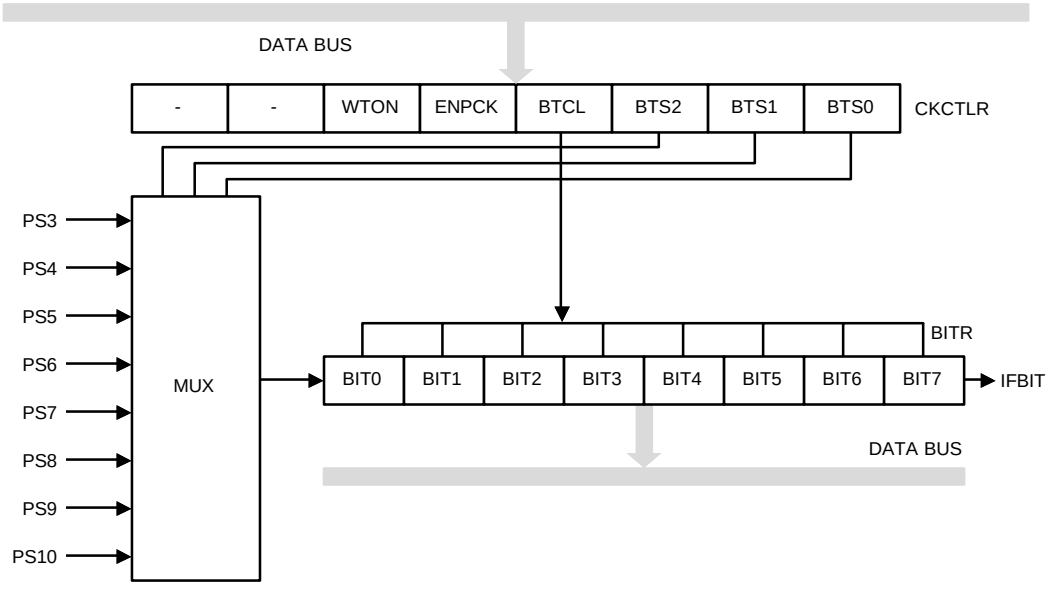


Fig. 4.4 Block diagram of Basic Interval Timer

4.1.4.1 Control of B.I.T

If bit3(BTCL) of CKCTLR is set to ¡È1¡È, B.I.T is cleared, and then, after one machine cycle, BTCL becomes ¡È0¡È, and B.I.T starts counting. BTCL is set to ¡È0¡È in reset state.

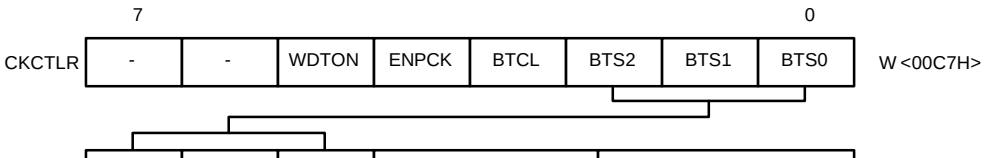
Clock Control Register

	7							0	
CKCTLR	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7H>

BTCL	B.I.T Operation
0	free-run
1	Automatically cleared, after one cycle

4.1.4.2 Input Clock Selection of Basic Interval Timer

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 8 kinds of interval time as shown in Table. 4.1.



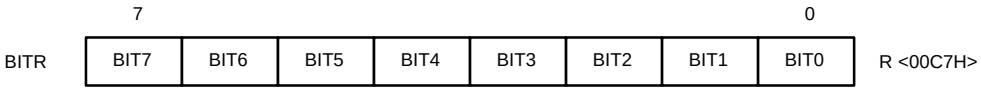
BTS2	BTS1	BTS0	B.I.T. Input clock	Standby release time
0	0	0	PS3 (2*us*)	512 *us*
0	0	1	PS4 (4*us*)	1,024 *us*
0	1	0	PS5 (8*us*)	2,048 *us*
0	1	1	PS6 (16*us*)	4,096 *us*
1	0	0	PS7 (32*us*)	8,192 *us*
1	0	1	PS8 (64*us*)	16,384 *us*
1	1	0	PS9 (128*us*)	32,768 *us*
1	1	1	PS10 (256*us*)	65,536 *us*

Table 4.1 Standby release time according to BTS

4.1.4.3 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



4.1.5 Watch Dog Timer

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

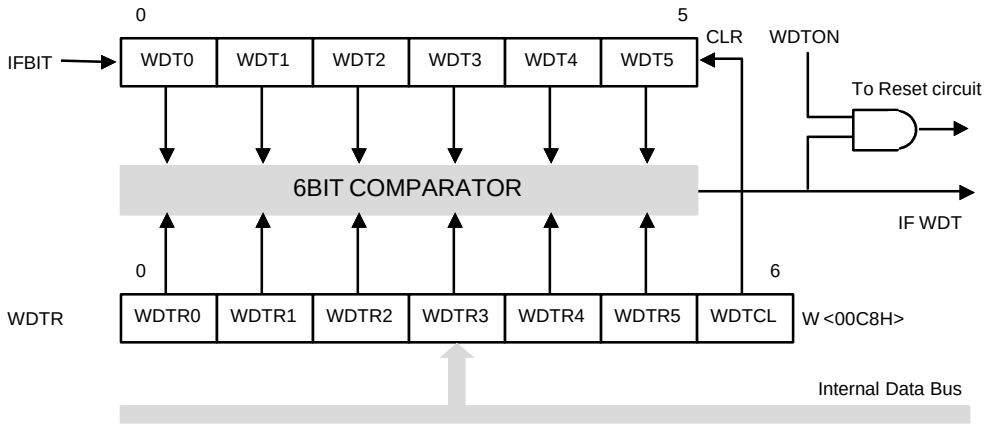


Fig. 4.5 Block diagram of Watch Dog Timer

4.1.5.1 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).

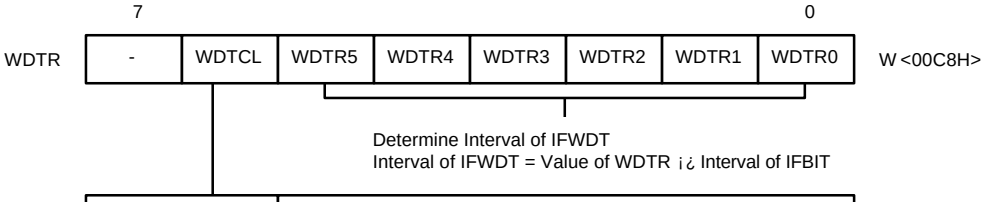
Clock Control Register

	7							0	
CKCTLR	-	-	WDTON	ENPCK	BTCL	BTS2	BTS1	BTS0	W <00C7H>

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



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

(Caution) : after WDTCL = 1, timer maximum error is one cycle of IFBIT.

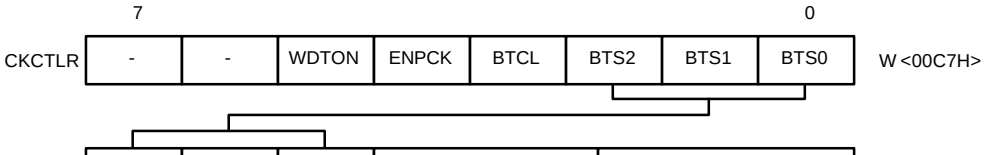
4.1.5.2 WDT Interrupt Interval

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 : 512us ¡¿ 1 = 512us (MIN>)
: 65,536us ¡¿ 63 = 4,128,768us (MAX>)

As IFBIT (Basic Interval Timer Interrupt Request) is used for input clock of WDT, Input clock cycle is possible from 512us to 65,536us 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 = 0Fh,15
interval of WDT = 65,536 ¡¿ 15 = 983040 uS (about 1second))



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

*note1) When WDTR Register value is 63(3FH)

Caution : Do not use ¡È0¡È for WDTR Register value.
Device come into the reset state by WDT

4.2 TIMER

4.2.1 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 $\overline{EC}$ 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.

<table>
<tr><td>Timer0</td><td>- 16-bit Interval Timer
- 16-bit Event Counter
- 16-bit Input Capture
- 16-bit rectangular-wave output</td><td rowspan="2">- Single/Modulo-N Mode
- Timer Output Initial Value Setting
- Timer0~Timer1 combination Logic Output
- One Interrupt Generating Every 2nd Counter Overflow</td></tr>
<tr><td>Timer1</td><td>- 8-bit Interval Timer
-8-bit rectangular-wave output</td></tr>
<tr><td>Timer2</td><td colspan="2">- 8-bit Interval Timer
-8-bit rectangular-wave output
- Modulo-N Mode</td></tr>
</table>

*Relevant Port Mode Register (PMR1 : 00C9H) value should be assigned for event counter, rectangular-wave output and input capture mode.

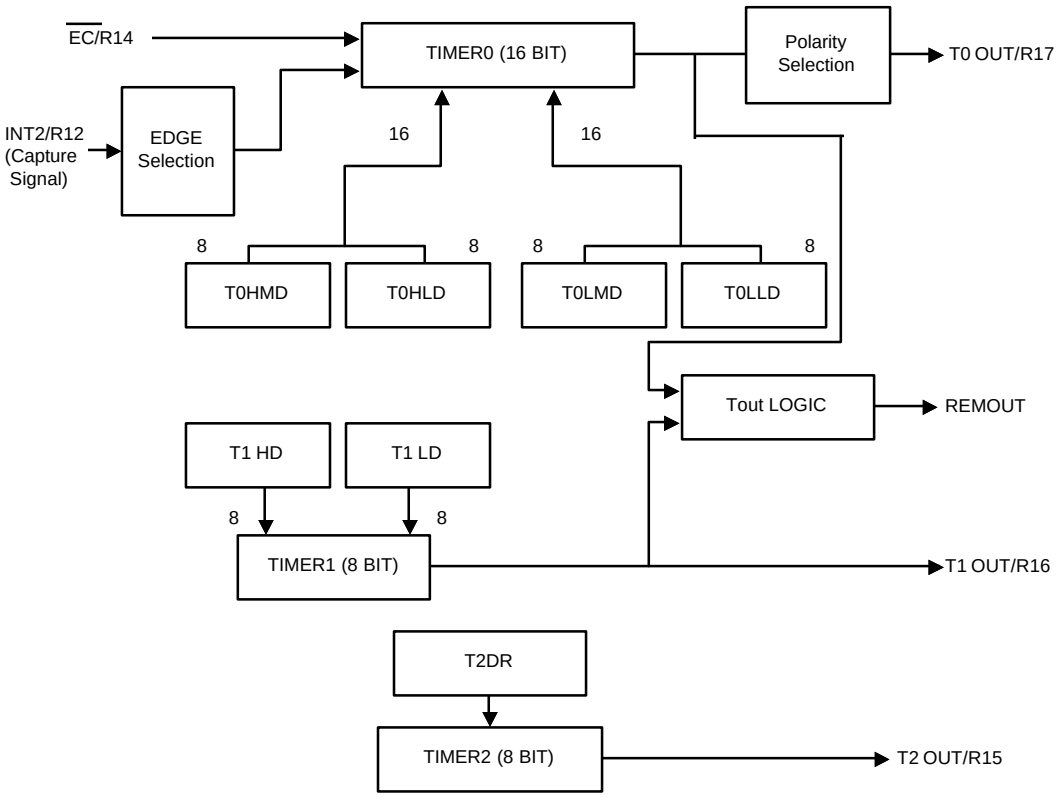


Fig. 4.6 Timer/Counter Block diagram

4.2.2 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 (0.25us)	16,384us	PS0 (0,.25us)	64us	PS5 (8us)	2.048us
PS1 (0. 5us)	32,768us	PS1 (0,5us)	128us	PS6 (16us)	4,096us
PS2 (1us)	65,536us	PS2 (1us)	256us	PS7 (32us)	8,192us
PS3 (2us)	131,072us	PS3 (2us)	512us	PS8 (64us)	16,384us
PS4 (4us)	262,144us	PS7 (32us)	8,192us	PS9 (128us)	32,768us
PS5 (8us)	524,288us	PS8 (64us)	16,384us	PS10 (256us)	65,536us
PS11 (512us)	33,554,432us	PS9 (128us)	32,768us	PS11 (512us)	131,072us
EC	-	PS10 (256us)	65,536us	PS12 (1,024us)	262,144us

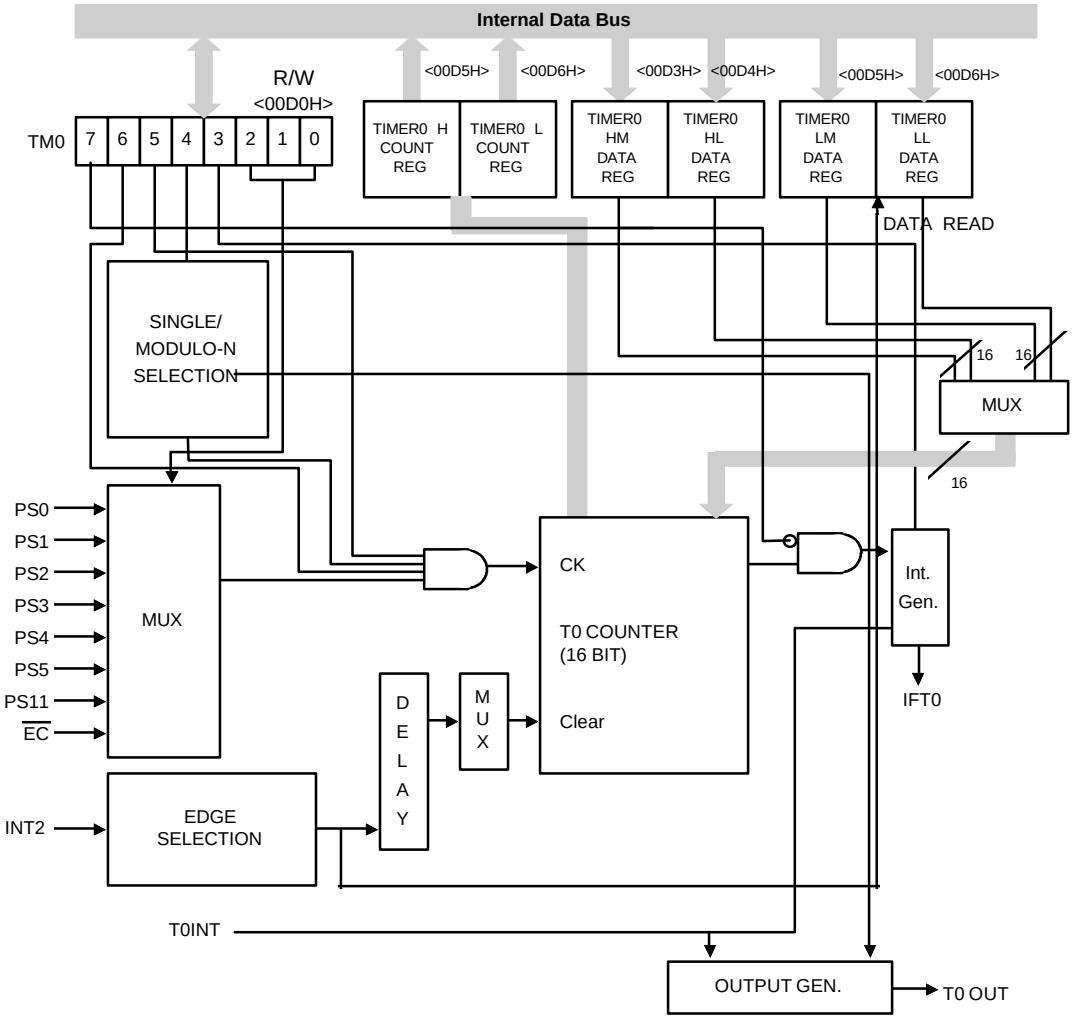


Fig. 4.7 Block Diagram of Timer0

Timer0 mode Register

	7							0	
TM0	CAP0	T0ST	T0CN	T0MOD	T0IFS	T0SL2	T0SL1	T0SL0	R/W <00D0H>

T0SL2	T0SL1	T0SL0	Input Clock Sel.	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	$\overline{EC}$	Counter

T0IFS	Timer0 Interrupt Sel.
0	Interrupt Every Counter Overflow
1	Interrupt Every 2nd Counter Overflow

T0MOD	Timer0 Single / Modulo-N Sel.
0	Modulo-N
1	Single Mode

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

T0ST	Timer0 Start/Stop control
0	Timer0 Stop
1	Timer0 Start after Clear

CAP0	Timer0 Interrupt Sel.
0	Timer/Counter
1	Input Capture*

*PS1 : not supporting input capture.

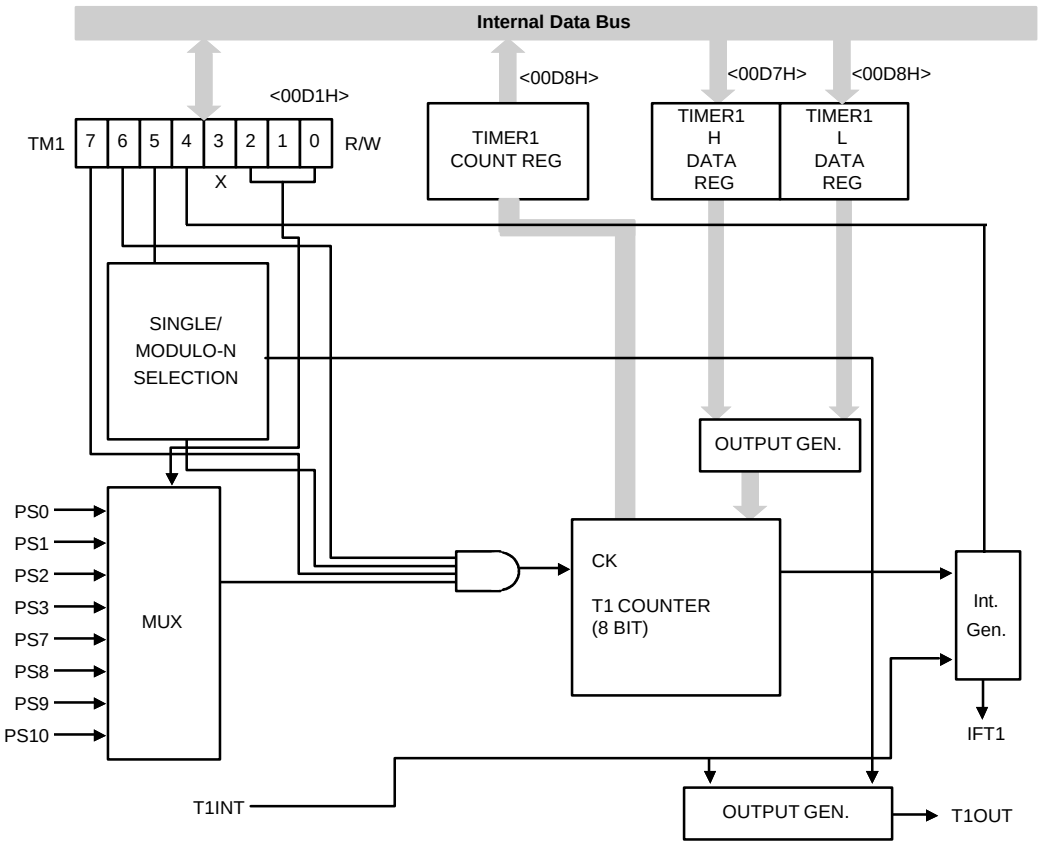
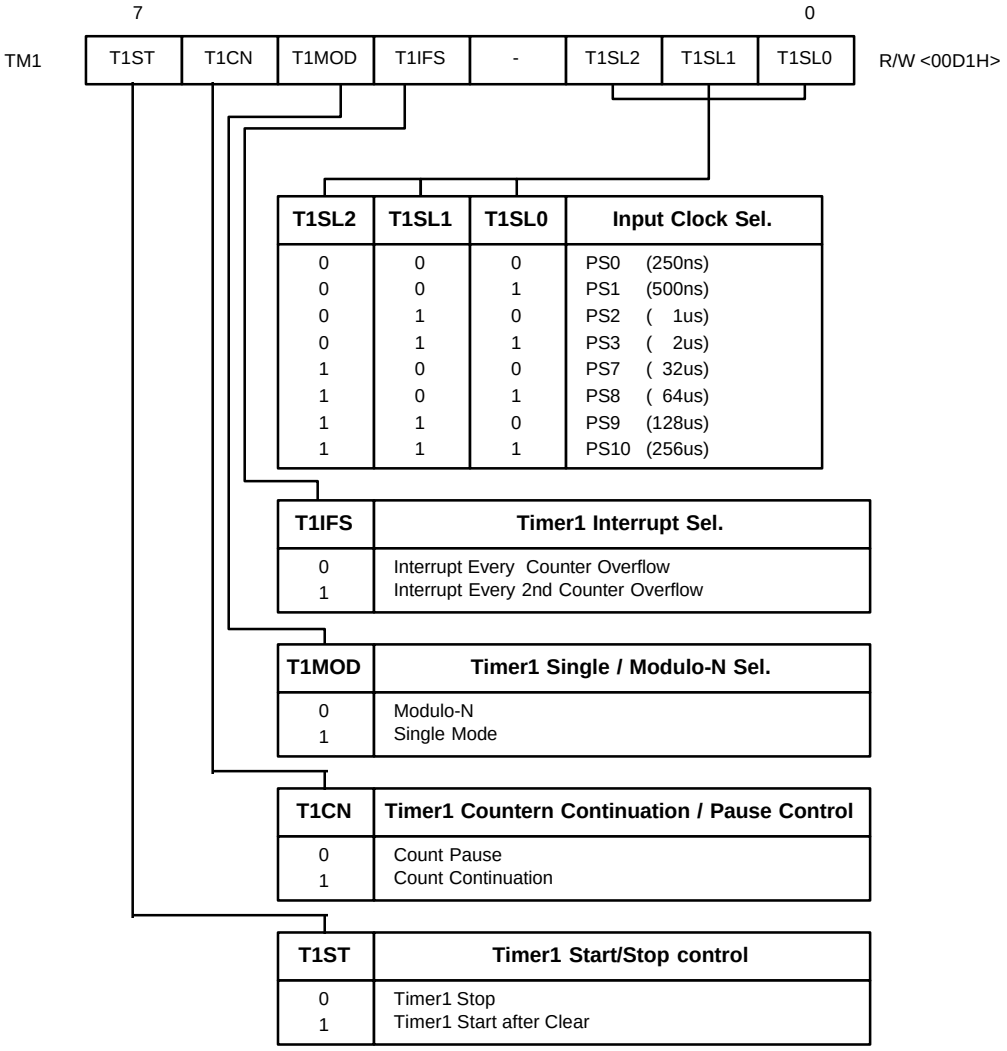


Fig. 4.8 Block Diagram of Timer1

Timer1 mode Register



T1SL2	T1SL1	T1SL0	Input Clock Sel.
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)

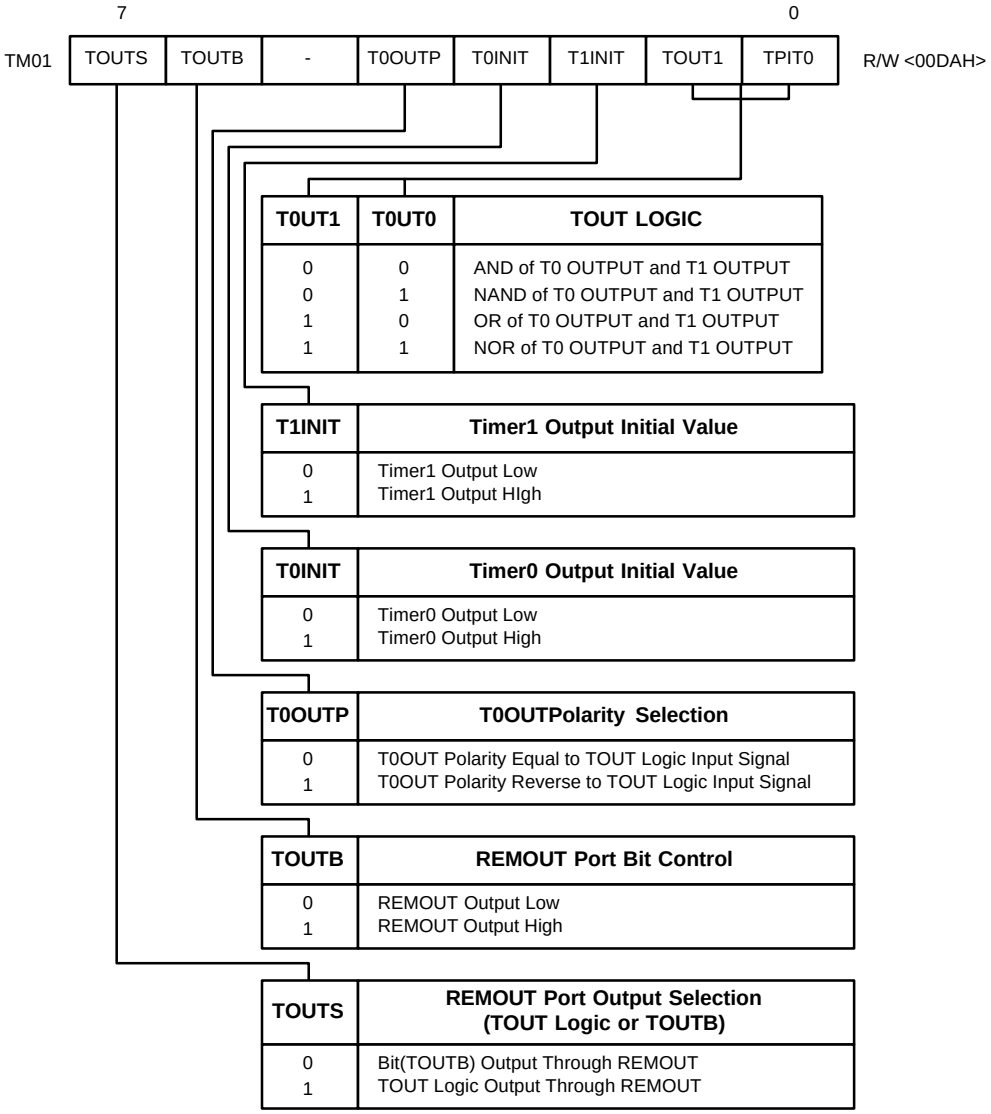
T1IFS	Timer1 Interrupt Sel.
0	Interrupt Every Counter Overflow
1	Interrupt Every 2nd Counter Overflow

T1MOD	Timer1 Single / Modulo-N Sel.
0	Modulo-N
1	Single Mode

T1CN	Timer1 Countern Continuation / Pause Control
0	Count Pause
1	Count Continuation

T1ST	Timer1 Start/Stop control
0	Timer1 Stop
1	Timer1 Start after Clear

Timer0/Timer1 mode Register



T0UT1	T0UT0	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	T0OUTPolarity 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

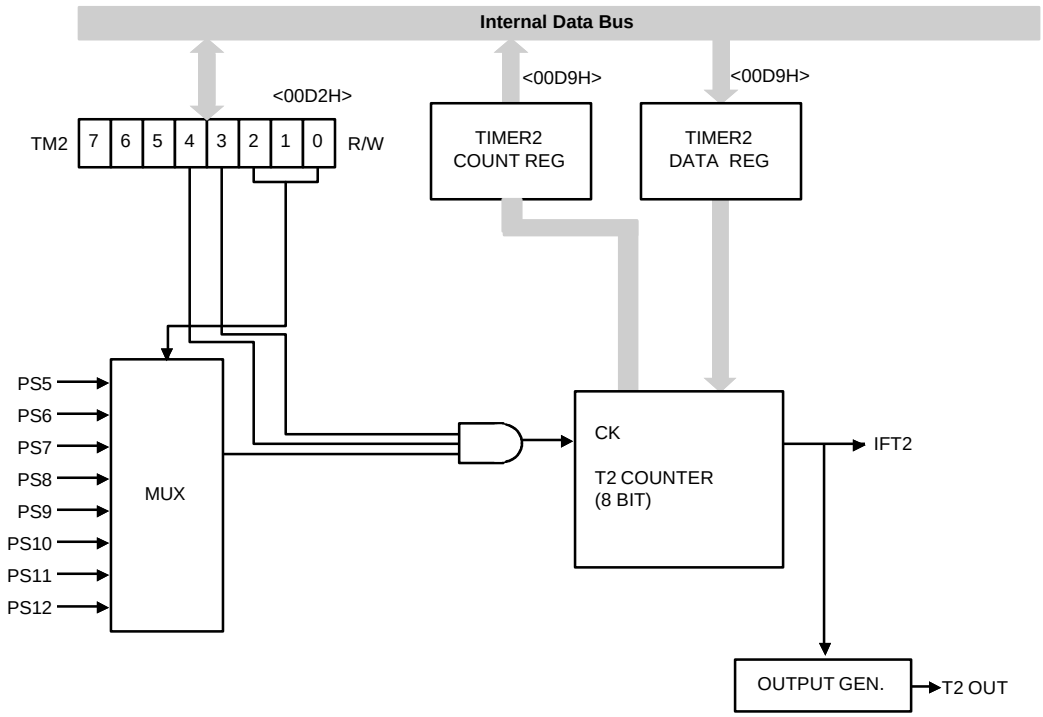
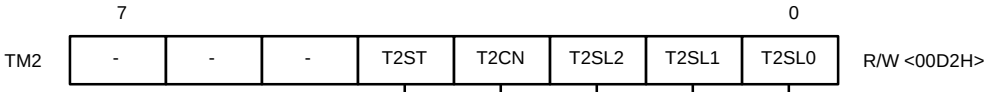


Fig. 4.9 Block Diagram of Timer2

Timer2 mode Register



T2SL2	T2SL1	T2SL0	Input Clock Sel.
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 (1,024us)

T2cn	Timer2 Counter Continuation / Pause Control
0	Count Pause
1	Count Continuation

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

PORT mode Register1

	7							0	
PMR1	T0S	T1S	T2S	ECS	-	INT2S	INT1S	-	W <00C9H>

PMR1		PORT Sel.	Remarks
T0S	0	R17 (I/O)	-
	1	T0 (Output)	Output Port of Timer0
T1S	0	R16 (I/O)	-
	1	T1 (Output)	Output Port of Timer1
T2S	0	R15 (I/O)	-
	1	T2 (Output)	Output Port of Timer2
ECS	0	R14 (I/O)	-
	1	$\overline{EC}$ (Input)	Input Port of Timer0 Event
-	-	-	-
	-	-	-
INT2S	0	R12 (I/O)	-
	1	INT2 (Input)	Input Port of Timer0 Input Capture
INT1S	0	R11 (I/O)	-
	1	INT1 (Input)	-
-	-	-	-
	-	-	-

External Interrupt Signal Edge Selectin Register

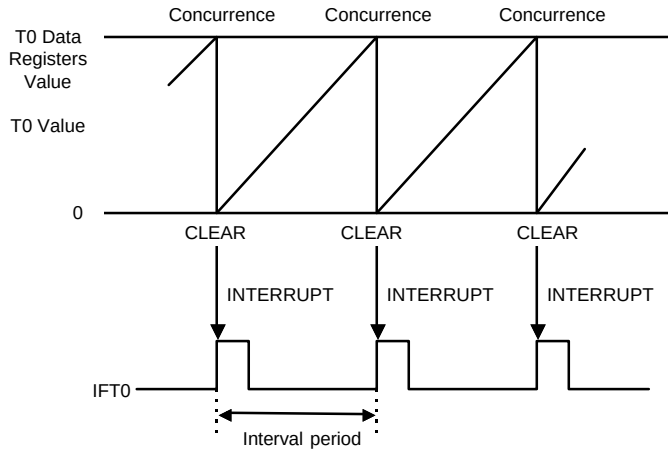
	7							0	
IEDS	-	-	IED2H	IED2L	IED1H	IED1L	-	-	W <00CBH>

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

4.2.3 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 ¡È00H¡È, and interrupt(IFT0, IFT1) is occured at the next clock

Fig. 4. 10 Operatiion of Timer0



For Timer0, the internal clock(PS) and the external clock($\overline{EC}$) 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 ¡È0¡È and then set to ¡È1¡È. T0CN, T1CN, T0ST and T1ST should be set ¡È1¡È, when Timer counting-up. Controlling of CAP0 enables Timer0 as input capture. By programming of CAP0 to ¡È1¡È, the period of signal from INT2 can be measured and then, event counter value for INT2 can be read.

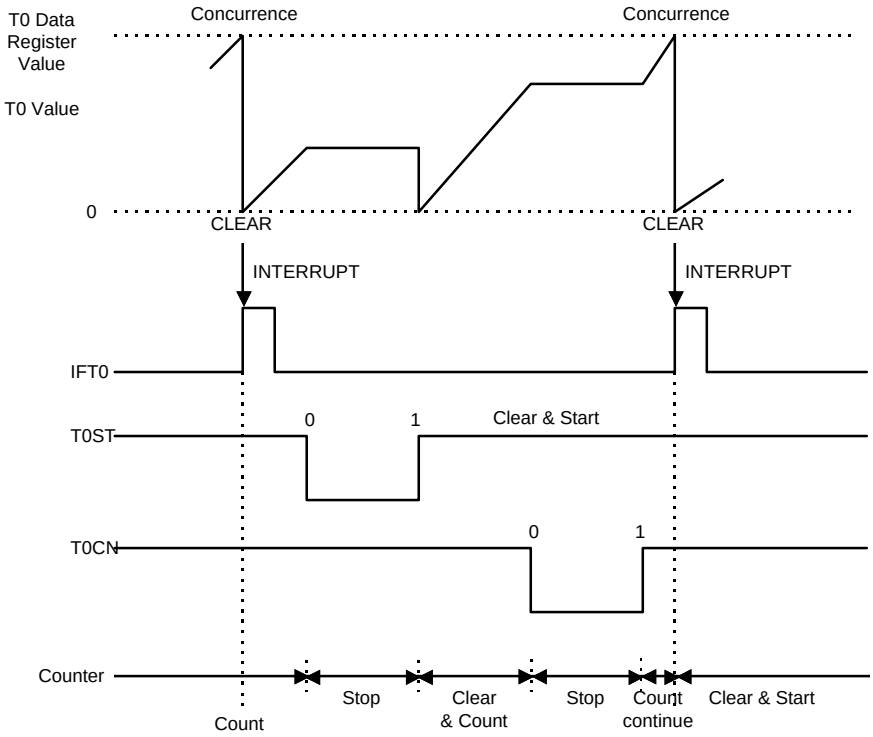


Fig. 4. 11. Start/Stop operation of Timer0

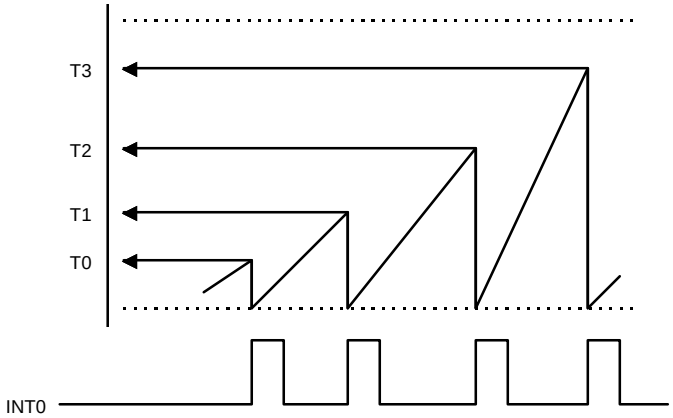


Fig. 4. 12. Input capture operation of Timer0

During counting-up, value of counter can be read. Timer execution is stopped by the reset signal (RESET = ¡ÈL¡È)
(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

¡é ¡é
0AFF 0B01

4.2.3.1 Single/Modulo-N Mode

Timer0 (Timer1) can select initial (T0INIT, T1INIT of TM0, TM1) output level of Timer Output port. If initial level is ¡ÈL¡È, Low-Data Register value of Timer Data Register is transferred to comparator and T0OUT(T1OUT) is to be ¡ÈLow¡È, if initial level is ¡ÈHigh¡È, High -Data Register is transferred and to be ¡ÈHigh¡È. Single Mode can be set by Mode Select bit(T0MOD, T1MOD) of Timer Mode Register (TM0, TM1) to ¡È1¡È 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 ¡È0¡È. 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 ¡È0¡È, Interrupt occurs on every Time-out. If it is ¡È1¡È, Interrupt occurs every second time-out.
(*note. Timer Output is toggled whenever time out happen)

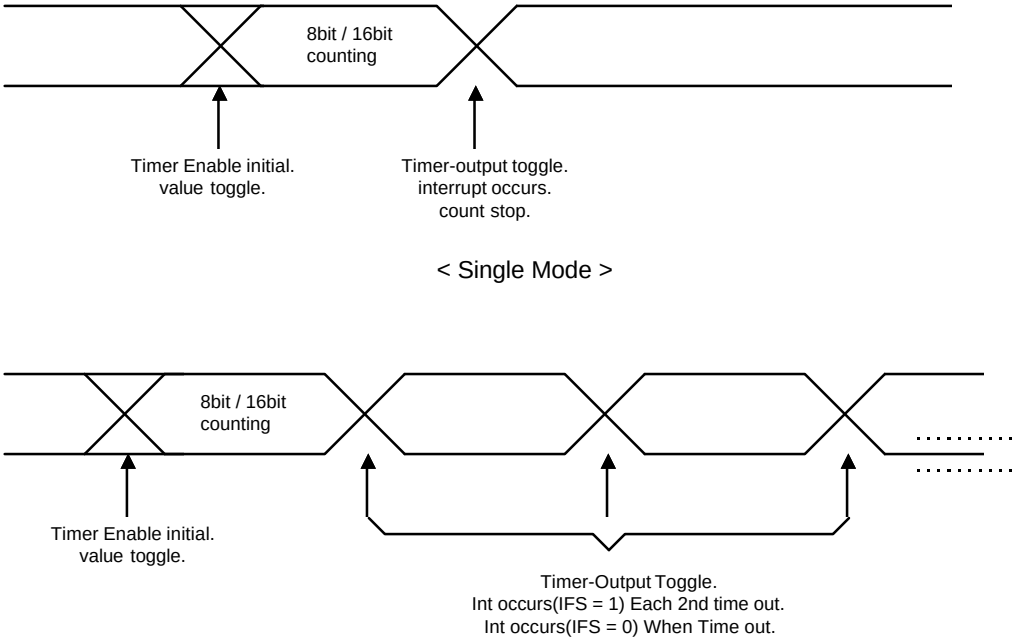


Fig. 4. 13 Operation Diagram for Single/Modulo-N Mode

4.2.4 Timer2

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 ¡È00H¡È. 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-cup. For clearing and starting the Timer2. T2ST have to set to ¡È1¡È after set to ¡È0¡È. In order to write a value directly into the T2DR, T2ST should be set to ¡È0¡È. Count value of Timer2 can be read at any time.

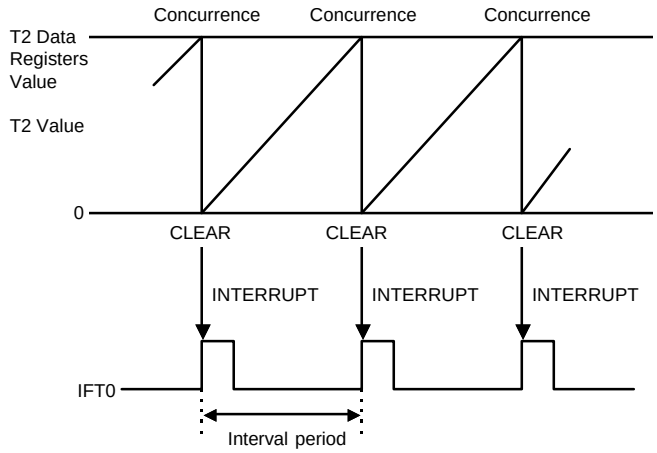


Fig. 4. 14 Operation of Timer2

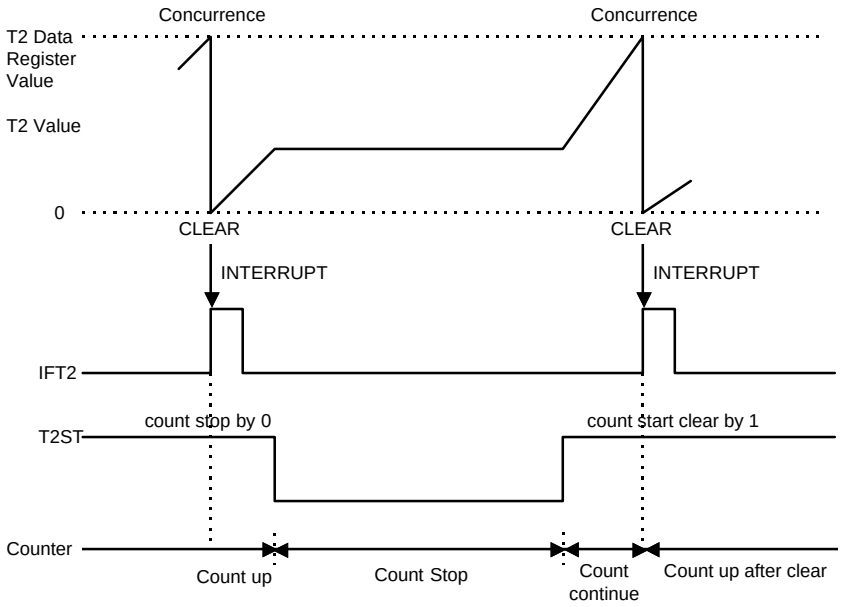


Fig. 4. 15. Start/Stop of Timer2

CHAPTER 5. INTERRUPT

The GMS810 Series 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.

Interrupt hardware consists of Interrupt Mode Register(MOD), Interrupt Enable Register High (IENH), Interrupt Enable Register Low(IENL), Interrupt Request Register High(IRQH), Interrupt Request Register Low(IRQL) and priority circuit. Interrupt function block diagram is shown in Fig. 5.1

5.1 INTERRUPT SOURCE

Each interrupt vector is independent and has its own priority. Software interrupt(BRK) is also available. Interrupt source classification is shown in Table 5.1

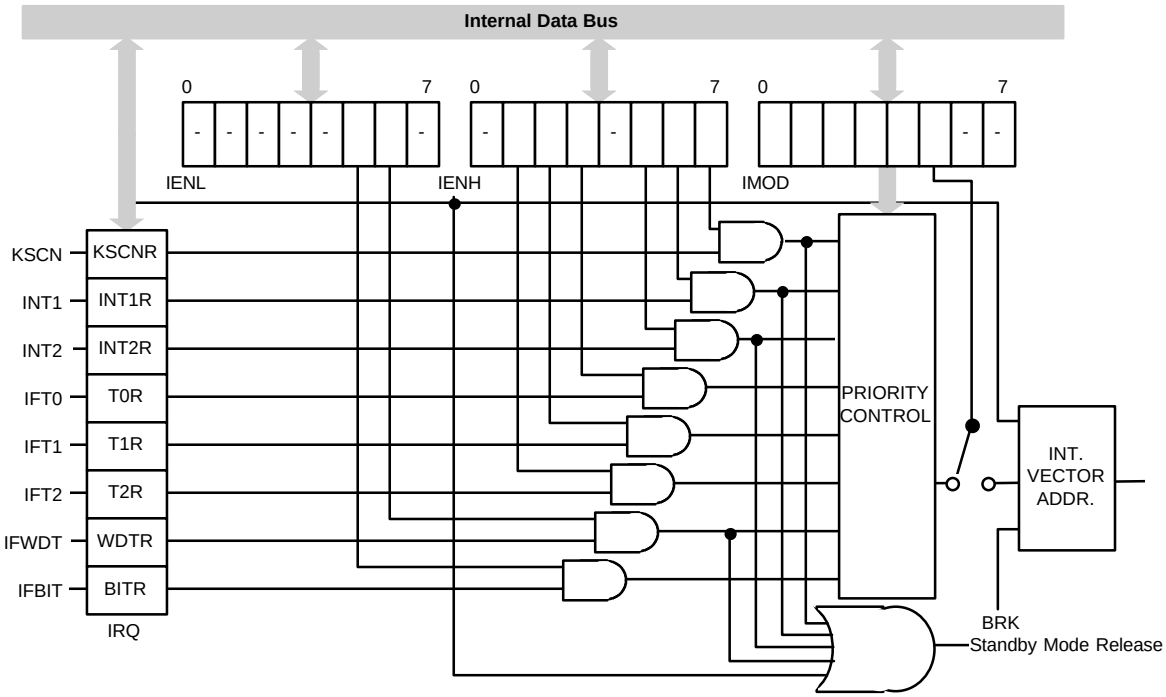


Fig. 5.1 Interrupt Source

	Mask	Priority	Interrupt Source	INT Vector H	INT Vector L
Hardware Interrupt	Non-maskable	-	RST ($\overline{\text{RESET}}$ PIN)	FFFF	FFFE
	Maskable	0	KSCNR (Key Scan)	FFFB	FFFA
		1	INT1R(External Interrupt 1)	FFF9	FFF8
		2	INT2R(External Interrupt 2)	FFF7	FFF6
		3	T0R (Timer0)	FFF3	FFF2
		4	T1R (Timer1)	FFF1	FFF0
		5	T2R (Timer2)	FFEF	FFEE
		6	WDTR (Watch Dog Timer)	FFE9	FFE8
		7	BITR (Basic Interval Timer)	FFE7	FFE6
Software Interrupt	**-**	**-**	**BRK Instruction**	**FFDF**	**FFDE**

Table 5.1 Interrupt Source

5.2 INTERRUPT CONTROL REGISTER

I flag of PSW is a interrupt mask enable flag. When I flag = ¡È0¡È, all interrupts become disable. When I flag = ¡È1¡È, 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 ¡È1¡È until the interrupt is accepted or is cleared in program.
In reset state, interrupt request flag register(IRQH, IRQL) is cleared to ¡È0¡È. 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).

Interrupt Enable Register Low

	7							0	
IENL	-	WDTR	BITE	-	-	-	-	-	R/W <00CCH>

Interrupt Enable Register High

	7							0	
IENH	KSCNE	INT1E	INT2E	-	T0E	T1E	T2E	-	R/W <00CEH>

Interrupt Request Register Low

	7							0	
IRQL	-	WDTR	BITE	-	-	-	-	-	R/W <00CDH>

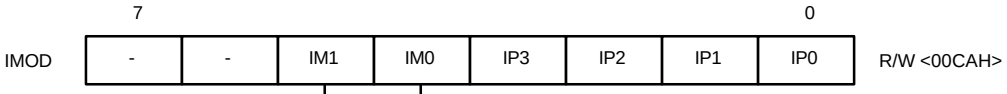
Interrupt Request Register High

	7							0	
IRQH	KSCNR	INT1R	INT2R	-	T0R	T1R	T2R	-	R/W <00CFH>

5.3 INTERRUPT ACCEPT MODE

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

Interrupt Mode Register



Assigning by interrupt accept mode bit

IM1	IM0	Priority
0	0	Fixed by H/W
0	1	Changeable by IP 3-0
1	*	Interrupt is inhibited

5.3.1 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 ¡È1¡È.

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 5.2 Interrupt Selection by IP3 - IP0

*In Reset state, these IP3 - IP0 registers become all ¡È0¡È.

5.3.2 Interrupt Timing

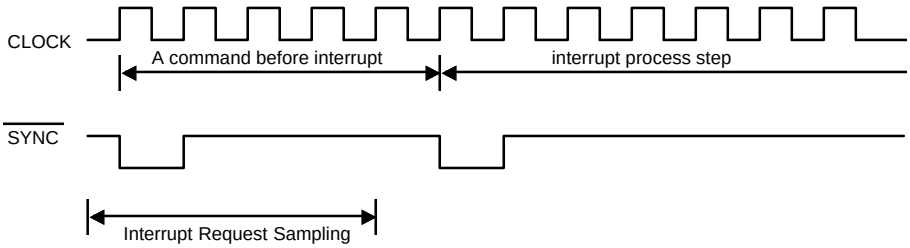


Fig. 5.2 Interrupt Enable Accept Timing

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

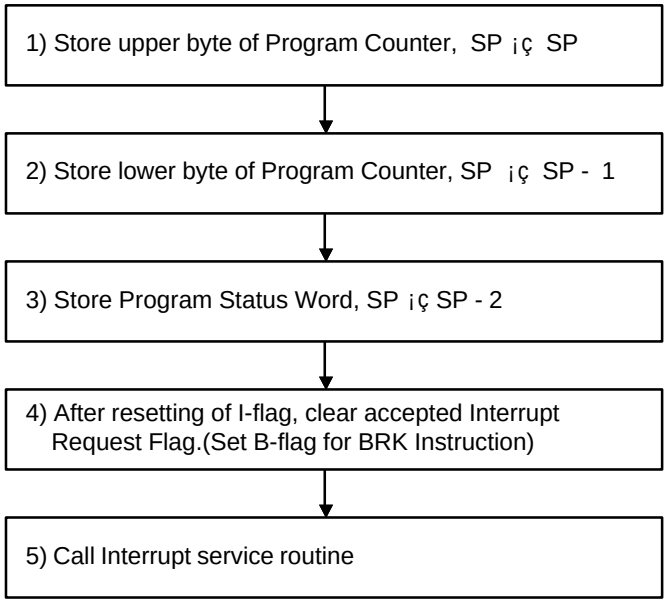
5.3.3 The valid timing after executing Interrupt control instructions

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.

5.4 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 saved in 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 (FFEOH-FFFFH) corresponding to each interrupt

Interrupt Processing Step



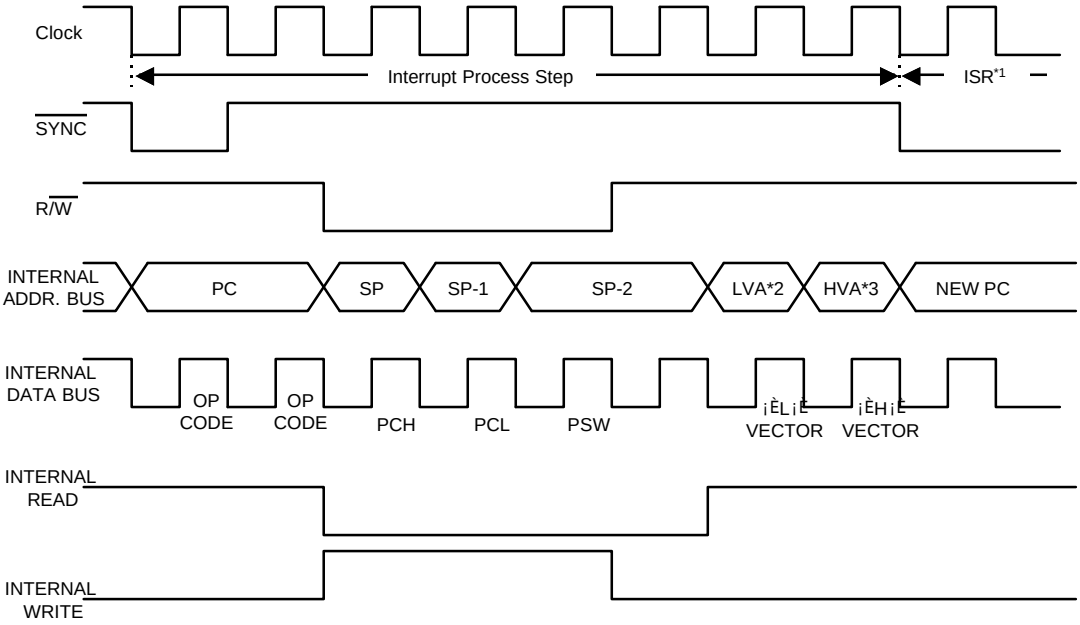


Fig. 5. 3 Interrupt Procesing Step Timing

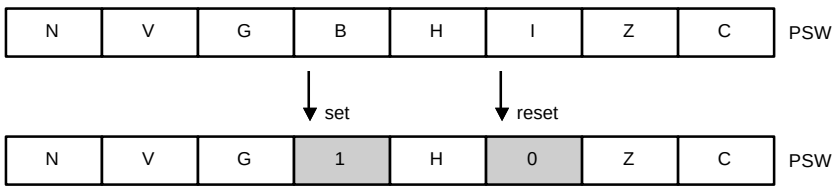
*1 ISR : Interrupt Service Routine
*2 LVA : Low Vector Address
*3 HVA : High Vector Address

5.1 SOFTWARE INTERRUPT

5.5.1 Interrupt by Break(BRK) Instruction

Software interrupt is available just by writing ¡ÈBreak(BRK)¡È 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.

Flag change by BRK execution



(Right after BRK execution)

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 Fig. 5.4 each processing routine is judged by contents of B flag.
There is no instruction to reset directly B flag.

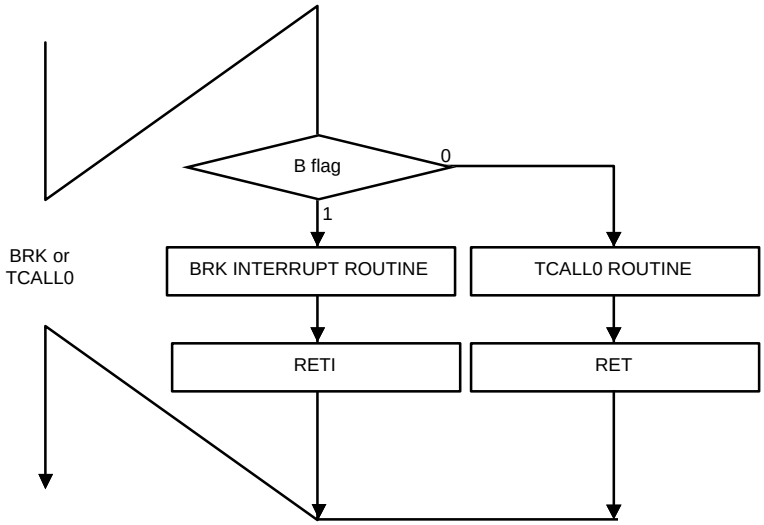


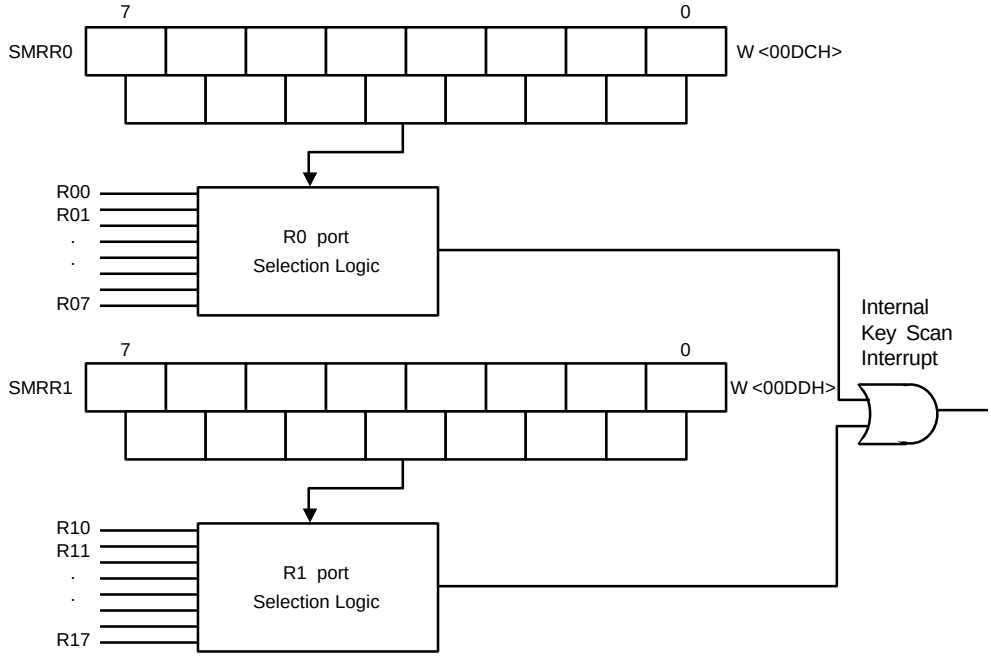
Fig. 5.4 Execution of BRK or TCALL0

5.6 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 ¡È1¡È, 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.

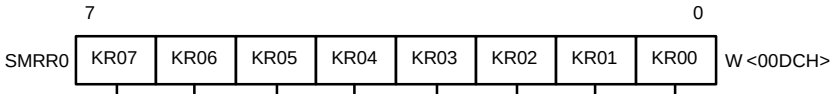
5.7 Key Scan Input Processing

Key Scan Interrupt is generated by detecting low Input from each Input pin (R0, R1) or standby(SLEEP, STOP) release signal. Key Scan ports are all 16bit which are controlled by Stand-by 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 ¡È00H¡È. So, there is no Key Input source.



<Key Scan Block>

SMRR0 Mode Register



KR00	Key Input Selection
0	no select
1	select

KR01	Key Input Selection
0	no select
1	select

KR02	Key Input Selection
0	no select
1	select

KR03	Key Input Selection
0	no select
1	select

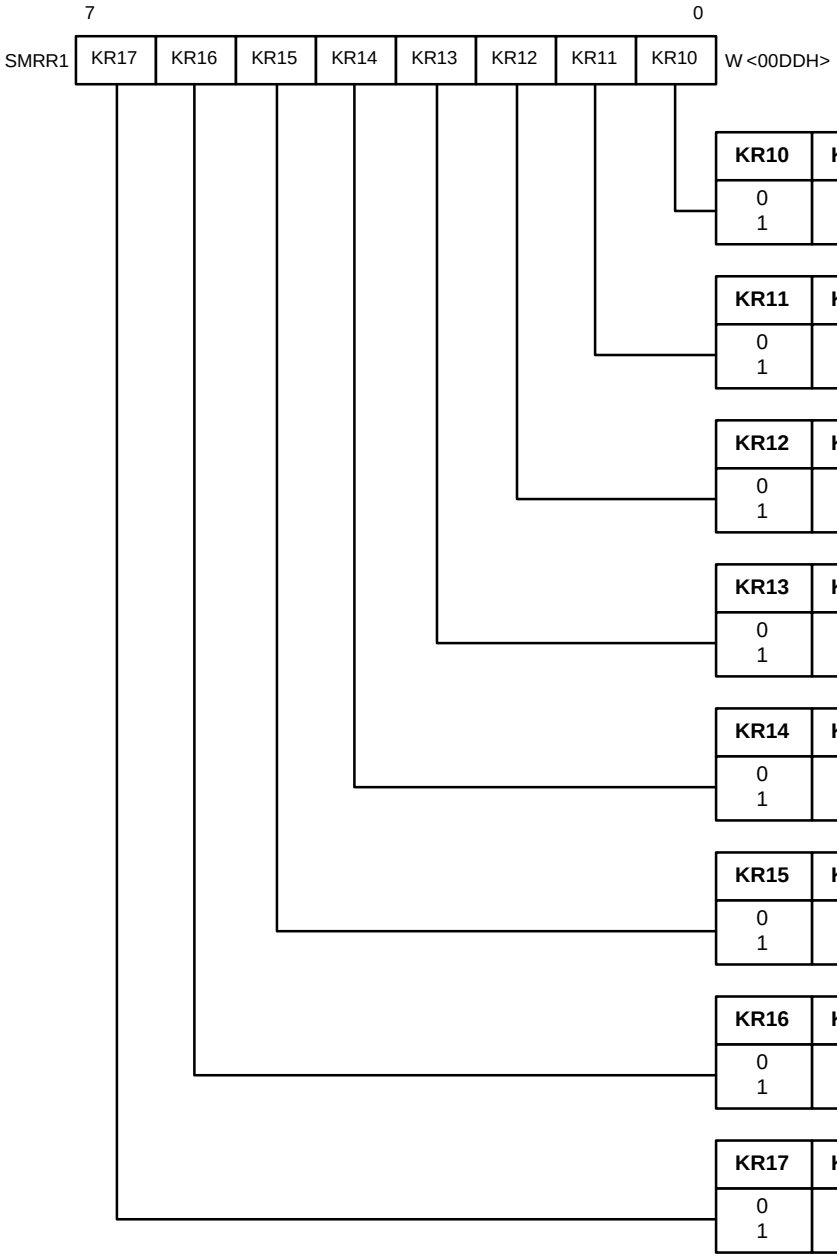
KR04	Key Input Selection
0	no select
1	select

KR05	Key Input Selection
0	no select
1	select

KR06	Key Input Selection
0	no select
1	select

KR07	Key Input Selection
0	no select
1	select

SMRR1 Mode Register



KR10	Key Input Selection
0	no select
1	select

KR11	Key Input Selection
0	no select
1	select

KR12	Key Input Selection
0	no select
1	select

KR13	Key Input Selection
0	no select
1	select

KR14	Key Input Selection
0	no select
1	select

KR15	Key Input Selection
0	no select
1	select

KR16	Key Input Selection
0	no select
1	select

KR17	Key Input Selection
0	no select
1	select

CHAPTER 6. STANDBY FUNCTION

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

6.1 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. ¡ÈNOP¡È instruction should be follows STOP instruction for rising precharge time of Data Bus line.

ex) STOP : STOP instructiion excution
NOP : NOP instruction

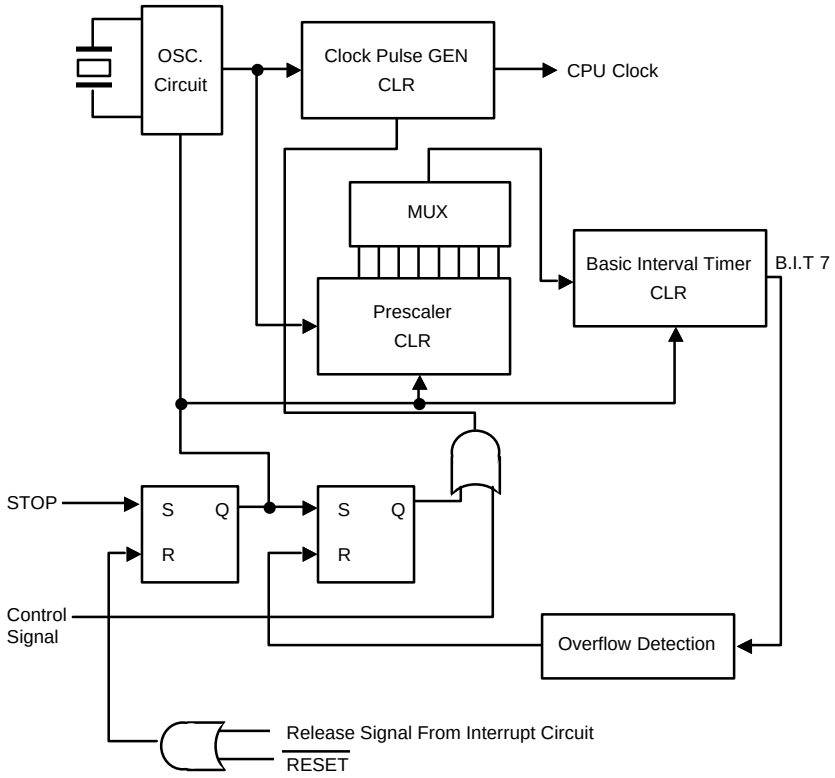


Fig. 6.1 Block Diagram of Standby Circuit

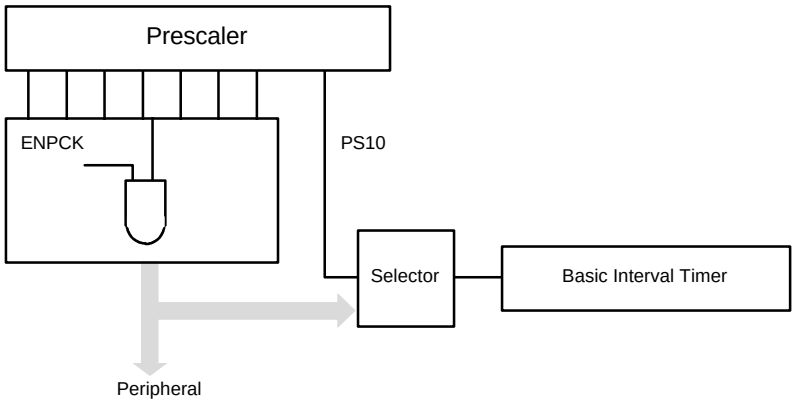


Fig. 6.2 ENPCK and Basic Interval Timer Clock

6.2 STANDBY MODE RELEASE

6.2.1 STOP Mode Release

Release of STANDBY mode is executed by $\overline{\text{RESET}}$ 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.

Table 6.1. Standby Mode Register

Release Signal	STOP
$\overline{\text{RESET}}$	0
KSCN (Key input)	0
INT1 - INT2	0

Table 6.2 Standby Mode Release

Release Factor	Release Method
$\overline{\text{RESET}}$ Pin	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.
INT 1 pin INT 2 pin	When external interrupt (INT1, INT2) enable flag is ¡È1¡È, 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.

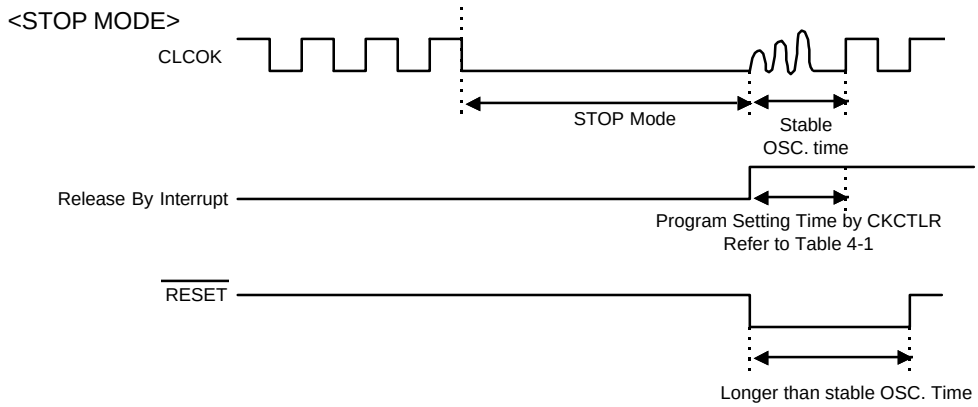


Fig. 6.3 Release Timing of Standby Mode

6.3 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(Fig. 6.3)

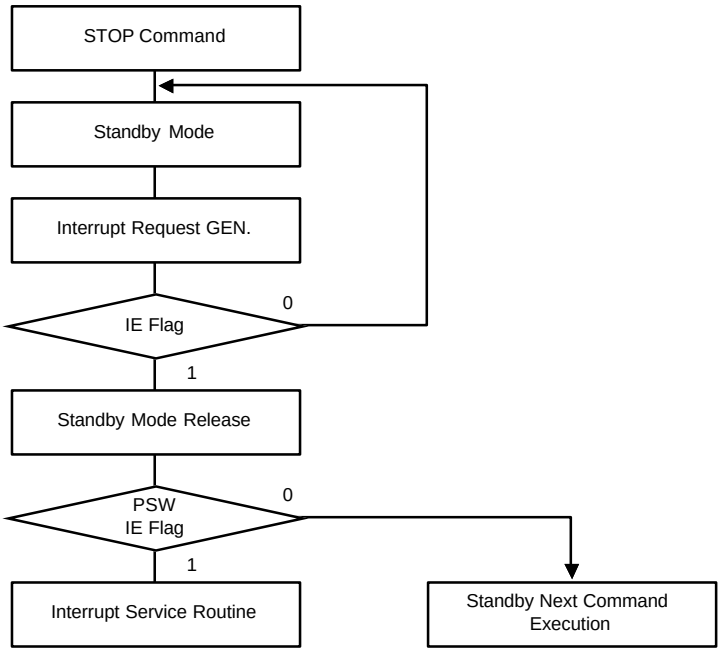
6.3.1 In Case of Interrupt Enable Flag(I) of PSW = 0

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

6.3.2 In Case of 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 ¡È1¡È, Standby mode is not entered.

Fig. 6.5 Standby Mode Release Flow



Internal circuit	STOP Mode
Oscillator	Stop
Internal CPU clock	Stop
Register	Retained
RAM	Retained
I/O port	Retained
Prescaler	Stop
Basic Interval Timer	Stop
Watch Dog Timer	Stop
Timer	Stop
Address Bus, Data Bus	Retained

Table 6.3 Operation State in Standby Mode

CHAPTER 7. RESET FUNCTION

7.1 EXTERNAL $\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ 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.

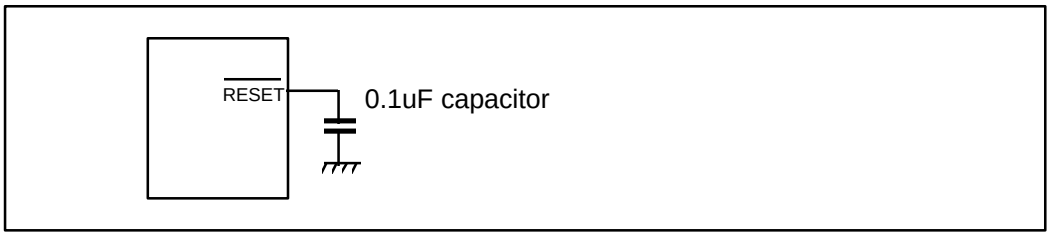


Fig 7.0 $\overline{\text{RESET}}$ Pin connection.

7.2 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, $\overline{\text{RESET}}$ terminal is maintained at ¡ÈL¡È 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 2^{19} (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.

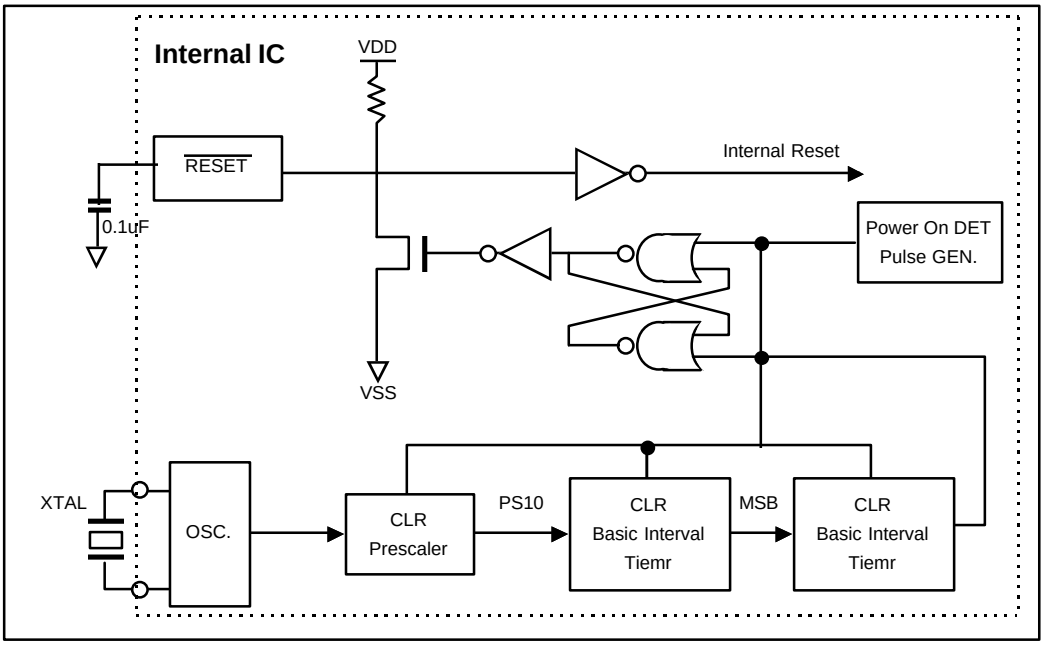


Fig. 7.1 Block Diagram of Power On Reset Circuit

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

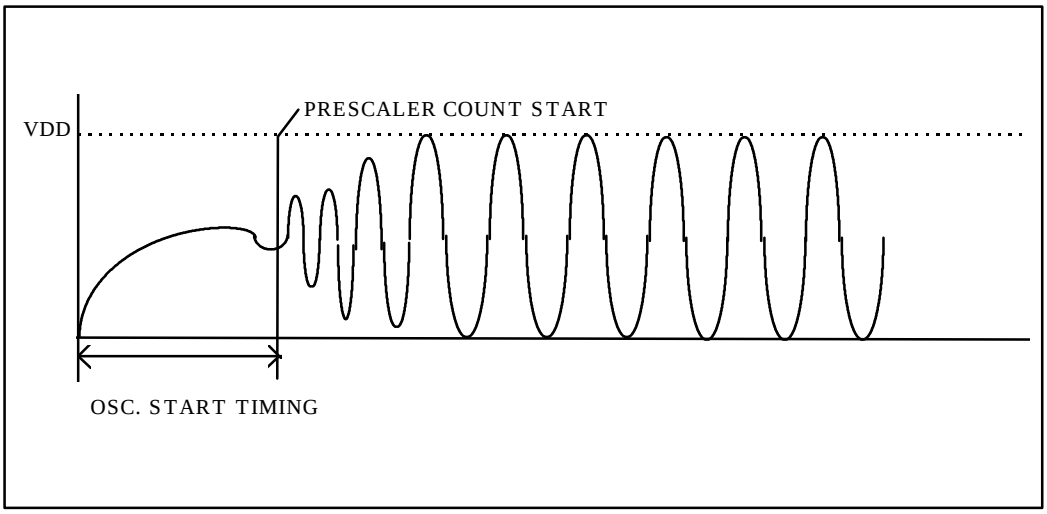


Fig. 7.2 Oscillator stabilization diagram

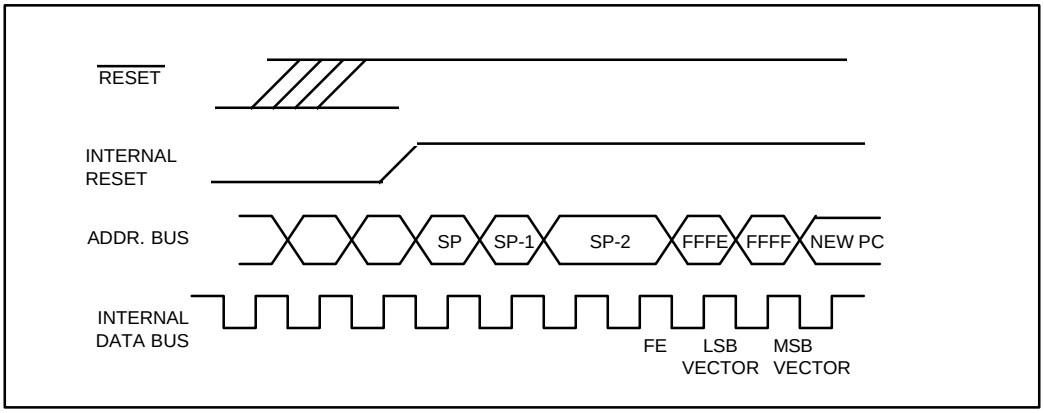


Fig. 7.3 Reset Timing by Diagram

7.3 Low Voltage Detection Mode

7.3.1 Low voltage detection condition

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

7.3.2 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.

7.3.3 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.

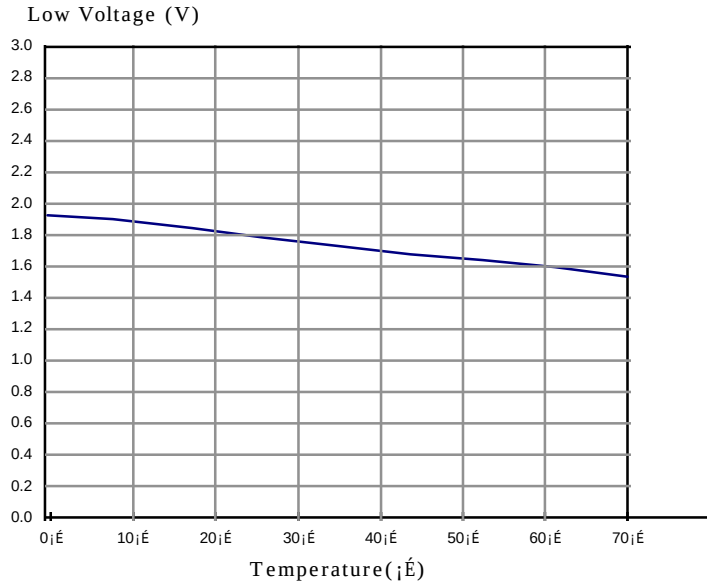
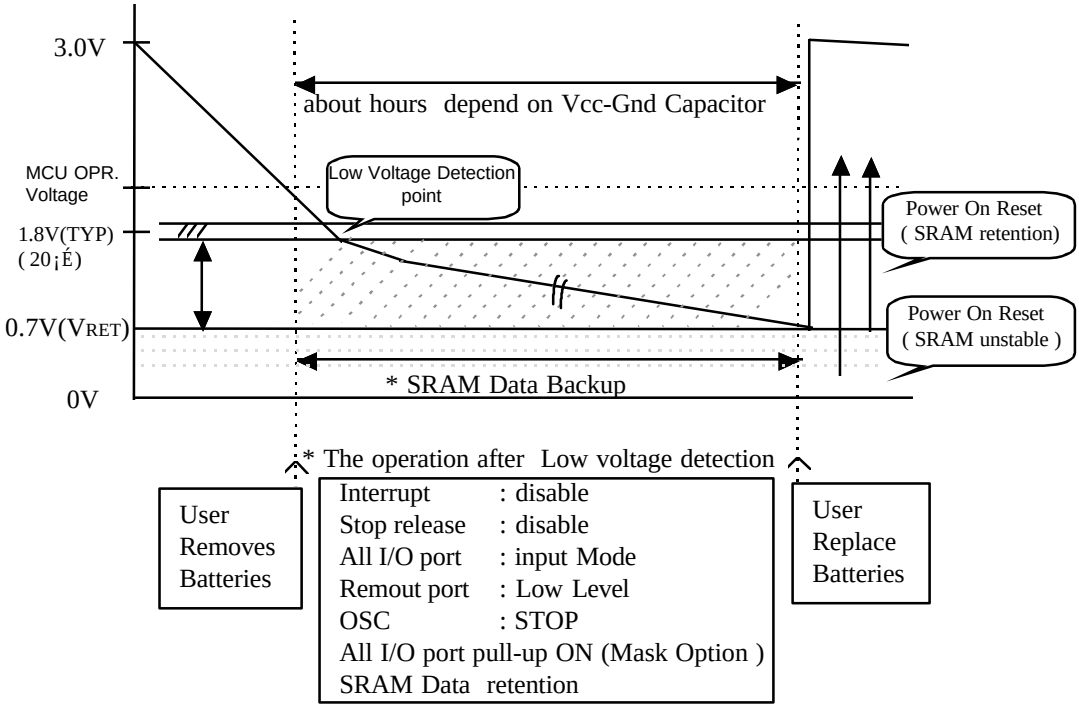
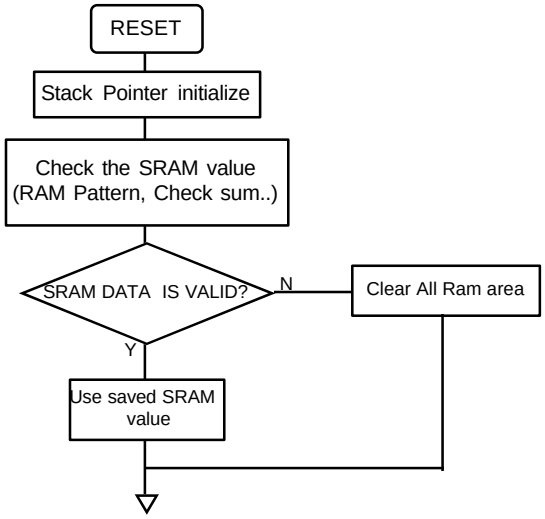


Fig 7.5 Low Voltage vs Temperature

*** SRAM BACK-UP after Low Voltage Detection.**



*** S/W flow chart example after Reset using SRAM Back-up**



APPENDIX A. INSTRUCTION SET TABLE

No.	MNEMONIC		OP CODE	Words	Exec. Cycle	OPERATION	Flag MVG HIZC
1	ADC	#imm	04	2	2	A = A + op + C	N V . . H . Z C
2	ADC	dp	05	2	3	¡È	N V . . H . Z C
3	ADC	dp+X	06	2	4	¡È	N V . . H . Z C
4	ADC	!abs	07	3	4	¡È	N V . . H . Z C
5	ADC	!abs+Y	15	3	5	¡È	N V . . H . Z C
6	ADC	[dp+X]	16	2	6	¡È	N V . . H . Z C
7	ADC	[dp]+Y	17	2	6	¡È	N V . . H . Z C
8	ADC	{X}	14	1	3	¡È	N V . . H . Z C
9	AND	#imm	84	2	2	A = A & op	N . . . . . Z .
10	AND	dp	85	2	3	¡È	N . . . . . Z .
11	AND	dp+X	86	2	4	¡È	N . . . . . Z .
12	AND	!abs	87	3	4	¡È	N . . . . . Z .
13	AND	!abs+Y	95	3	5	¡È	N . . . . . Z .
14	AND	[dp+X]	96	2	6	¡È	N . . . . . Z .
15	AND	[dp]+Y	97	2	6	¡È	N . . . . . Z .
16	AND	{X}	94	1	3	¡È	N . . . . . Z .
17	ASL	A	08	1	2	op = op << 1	N . . . . . Z C
18	ASL	dp	09	2	4	¡È	N . . . . . Z C
19	ASL	dp+X	19	2	5	¡È	N . . . . . Z C
20	ASL	!abs	18	3	5	¡È	N . . . . . Z C
21	BBC	A.bit, rel	y2	2	4/6	if (bit = 0)	. . . . .
22	BBC	dp.bit, rel	y3	3	5/7	then branch	. . . . .
23	BBS	A.bit, rel	x2	2	4/6	if (bit = 1)	. . . . .
24	BBS	dp.bit, rel	x3	3	5/7	then branch	. . . . .
25	BCC	rel	50	2	2/4	if (C=0) branch	. . . . .
26	BCS	rel	D0	2	2/4	if (C=1) branch	. . . . .
27	BEQ	rel	F0	2	2/4	if (Z=1) branch	. . . . .
28	BIT	dp	0C	2	4	Z = A & op	N N . . . . Z .
29	BIT	!abs	1C	3	5	¡È	N N . . . . Z .
30	BMI	rel	90	2	2/4	if (N=1) branch	. . . . .
31	BNE	rel	70	2	2/4	if (Z=0) branch	. . . . .
32	BPL	rel	10	2	2/4	if (N=0) branch	. . . . .
33	BRA	rel	2F	2	4	Branch	. . . . .
34	BRK		0F	1	8	S/W interrupt	. . . 1 . 0 . .
35	BVC	rel	30	2	2/4	if (V=0) branch	. . . . .
36	BVS	rel	B0	2	2/4	if (V=1) branch	. . . . .
37	CLR1	dp.bit	y1	2	4	op.bit = 0	. . . . .
38	CLRA1	A.bit	2B	2	2	¡È	. . . . .
39	CLRC		20	1	2	C = 0	 0
40	CLRG		40	1	2	G = 0	. . 0 . . . . .
41	CLRV		80	1	2	V = 0	. 0 . . 0 . . .

Appendix A. Instruction Set Table

No.	MNEMONIC		OP CODE	Words	Exec. Cycle	OPERATION	Flag MVG HIZC
42	CMP	#imm	44	2	2	Compare A, op	N . . . . . Z C
43	CMP	dp	45	2	3	¡È	N . . . . . Z C
44	CMP	dp+X	46	2	4	¡È	N . . . . . Z C
45	CMP	!abs	47	3	4	¡È	N . . . . . Z C
46	CMP	!abs+Y	55	3	5	¡È	N . . . . . Z C
47	CMP	[dp+X]	56	2	6	¡È	N . . . . . Z C
48	CMP	[dp]+Y	57	2	6	¡È	N . . . . . Z C
49	CMP	{X}	54	1	3	¡È	N . . . . . Z C
50	COM	dp	2C	2	4	dp = $\overline{dp}$	N . . . . . Z .
51	CMPX	#imm	5E	2	2	Compare X, op	N . . . . . Z C
52	CMPX	dp	6C	2	3	¡È	N . . . . . Z C
53	CMPX	!abs	7C	3	4	¡È	N . . . . . Z C
54	CMPY	#imm	7E	2	2	Compare Y, op	N . . . . . Z C
55	CMPY	dp	8C	2	3	¡È	N . . . . . Z C
56	CMPY	!abs	9C	3	4	¡È ¡È	N . . . . . Z C
57	DAA		DF	1	3		N . . . . . Z C
58	DAS		CF	1	3	Dec. adjustment (Add) Dec. adjustment (Sub)	N . . . . . Z C
59	DEC	A	A8	1	2		N . . . . . Z .
60	DEC	dp	A9	2	4	op = op -1	N . . . . . Z .
61	DEC	dp + X	B9	2	5	¡È	N . . . . . Z .
62	DEC	!abs	B8	3	5	¡È	N . . . . . Z .
63	DEC	X	AF	1	2	¡È	N . . . . . Z .
64	DEC	Y	BE	1	2	¡È ¡È	N . . . . . Z .
65	DIV		9B	1	12	Q:A, R:Y ¡ç YA/X	N V . . H . Z .
66	DI		60	1	3	I = 0	. . . . . 0 . .
67	EI		E0	1	3	I = 1	. . . . . 1 . .
68	EOR	#imm	A4	2	2	A = A ⊕ op	N . . . . . Z .
69	EOR	dp	A5	2	3	¡È	N . . . . . Z .
70	EOR	dp+X	A6	2	4	¡È	N . . . . . Z .
71	EOR	!abs	A7	3	4	¡È	N . . . . . Z .
72	EOR	!abs+Y	B5	3	5	¡È	N . . . . . Z .
73	EOR	[dp+X]	B6	2	6	¡È	N . . . . . Z .
74	EOR	[dp]+Y	B7	2	6	¡È	N . . . . . Z .
75	EOR	{X}	B4	1	3	¡È	N . . . . . Z .
76	INC	A	88	1	2	OP = OP + 1	N . . . . . Z C
77	INC	dp	89	2	4	¡È	N . . . . . Z .
78	INC	dp + X	99	2	5	¡È	N . . . . . Z .
79	INC	!abs	98	3	5	¡È	N . . . . . Z .
80	INC	X	8F	1	2	¡È	N . . . . . Z .
81	INC	Y	9E	1	2	¡È	N . . . . . Z .
82	JMP	!abs	1B	3	3	Branch	. . . . .
83	JMP	[!abs]	1F	3	5	¡È	. . . . .
84	JMP	[dp]	3F	2	4	¡È	. . . . .
85	CALL	!abs	3B	3	8	Subroutine call	. . . . .
86	CALL	[dp]	5F	2	8	¡È	. . . . .
87	PCALL	upage	4F	2	6	¡È	. . . . .
88	TCALL	n	nA	1	8	¡È	. . . . .

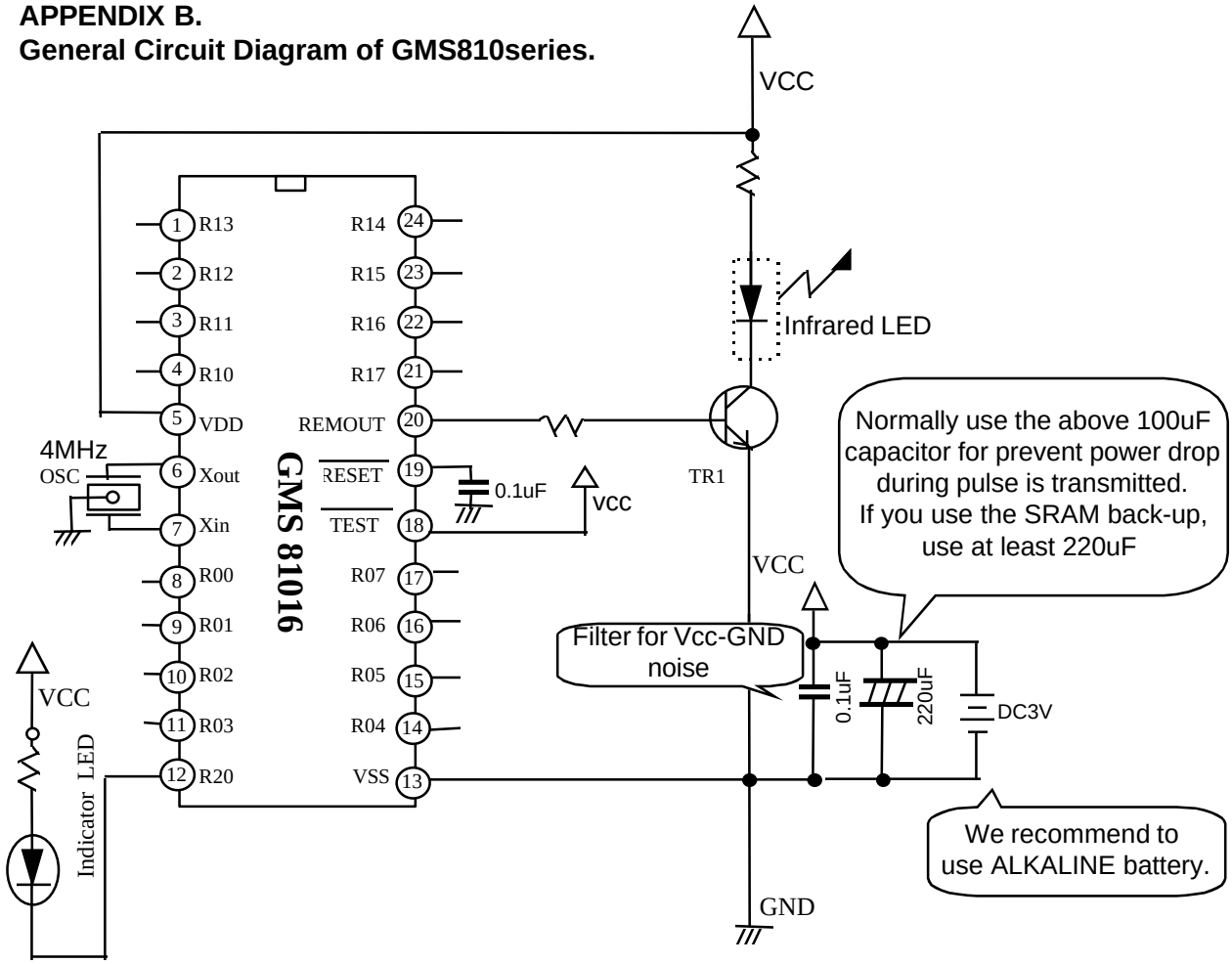
No.	MNEMONIC		OP CODE	Words	Exec. Cycle	OPERATION	Flag MVG HIZC
89	LDA	#imm	C4	2	2	A = op	N . . . . . Z .
90	LDA	dp	C5	2	3	¡È	N . . . . . Z .
91	LDA	dp+X	C6	2	4	¡È	N . . . . . Z .
92	LDA	!abs	C7	3	4	¡È	N . . . . . Z .
93	LDA	!abs+Y	D5	3	5	¡È	N . . . . . Z .
94	LDA	[dp+X]	D6	2	6	¡È	N . . . . . Z .
95	LDA	[dp]+Y	D7	2	6	¡È	N . . . . . Z .
96	LDA	{X}	D4	1	3	¡È	N . . . . . Z .
97	LDA	{X}+	DB	1	4	A = op, X = X+1	N . . . . . Z .
98	LDM	dp, #imm	E4	3	5	dp = #imm	. . . . .
99	LDX	#imm	1E	2	2	X = op	N . . . . . Z .
100	LDX	dp	CC	2	3	¡È	N . . . . . Z .
101	LDX	dp+Y	CD	2	4	¡È	N . . . . . Z .
102	LDX	!abs	DC	3	4	¡È	N . . . . . Z .
						Y = op	
103	LDY	#imm	3E	2	2	¡È	N . . . . . Z .
104	LDY	dp	C9	2	3	¡È	N . . . . . Z .
105	LDY	dp+X	D9	2	4		N . . . . . Z .
106	LDY	!abs	D8	3	4	¡È	N . . . . . Z .
107	LSR	A	48	1	2	op = op >>1	N . . . . . Z C
108	LSR	dp	49	2	4	¡È	N . . . . . Z C
109	LSR	dp + X	59	2	5	¡È	N . . . . . Z C
110	LSR	!abs	58	3	5	¡È	N . . . . . Z C
111	MUL		5B	1	9	YA = Y * A	N . . . . . Z .
112	NOP		FF	1	2	No operation	. . . . .
113	OR	#imm	64	2	2	A = A : op	N . . . . . Z .
114	OR	dp	65	2	3	¡È	N . . . . . Z .
115	OR	dp+X	66	2	4	¡È	N . . . . . Z .
116	OR	!abs	67	3	4	¡È	N . . . . . Z .
117	OR	!abs+Y	75	3	5	¡È	N . . . . . Z .
118	OR	[dp+X]	76	2	6	¡È	N . . . . . Z .
119	OR	[dp]+Y	77	2	6	¡È	N . . . . . Z .
120	OR	{X}	74	1	3	¡È	N . . . . . Z .
121	PUSH	A	0E	1	4	Push op, SP = SP - 1	. . . . .
122	PUSH	X	2E	1	4	¡È	. . . . .
123	PUSH	Y	4E	1	4	¡È	. . . . .
124	PUSH	PSW	6E	1	4	¡È	. . . . .
125	POP	A	0D	1	4	Pop op, SP = SP + 1	. . . . .
126	POP	X	2D	1	4	¡È	. . . . .
127	POP	Y	4D	1	4	¡È	. . . . .
128	POP	PSW	6D	1	4	¡È	(restored)
129	ROL	A	28	1	2	op = op << 1, with C	N . . . . . Z C
130	ROL	dp	29	2	4	¡È	N . . . . . Z C
131	ROL	dp+X	39	2	5	¡È	N . . . . . Z C
132	ROL	!abs	38	3	5	¡È	N . . . . . Z C
133	ROR	A	68	1	2	op = op >> 1, with C	N . . . . . Z C
134	ROR	dp	69	2	4	¡È	N . . . . . Z C
135	ROR	dp+X	79	2	5	¡È	N . . . . . Z C
136	ROR	!abs	78	3	5	¡È	N . . . . . Z C

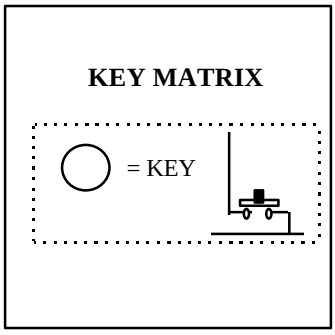
Appendix A. Instruction Set Table

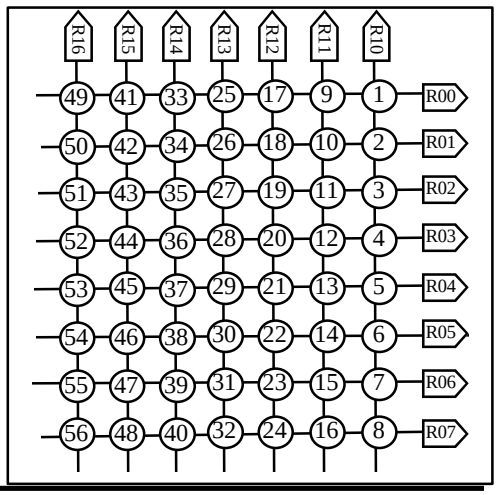
No.	MNEMONIC		OP CODE	Words	Exec. Cycle	OPERATION	Flag MVG HIZC
137	RETI		7F	1	6	Interrupt end	(restored)
138	RET		6F	1	5	Subroutine end	. . . . .
139	SBC	#imm	24	2	2	A = A - op - C	N V . . H . Z C
140	SBC	dp	25	2	3	¡È	N V . . H . Z C
141	SBC	dp+X	26	2	4	¡È	N V . . H . Z C
142	SBC	!abs	27	3	4	¡È	N V . . H . Z C
143	SBC	!abs+Y	35	3	5	¡È	N V . . H . Z C
144	SBC	[dp+X]	36	2	6	¡È	N V . . H . Z C
145	SBC	[dp]+Y	37	2	6	¡È	N V . . H . Z C
146	SBC	{X}	34	1	3	¡È	N V . . H . Z C
147	SETI	dp.bit	x1	2	4	op.bit = 1	. . . . .
148	SETA1	A.bit	0B	2	2	¡È	. . . . .
149	SETC		A0	1	2	C = 1	 1
150	SETG		C0	1	2	G = 1	. . 1 . . . .
151	STA	dp	E5	2	4	op = A	. . . . .
152	STA	dp+X	E6	2	5	¡È	. . . . .
153	STA	!abs	E7	3	5	¡È	. . . . .
154	STA	!abs+Y	F5	3	6	¡È	. . . . .
155	STA	[dp+X]	F6	2	7	¡È	. . . . .
156	STA	[dp]+Y	F7	2	7	¡È	. . . . .
157	STA	{X}	F4	1	4	¡È	. . . . .
158	STA	{X}+	FB	1	4	op = A, X=X+1	. . . . .
159	STOP		EF	1	3	CPU, OSC stop	. . . . .
160	STX	dp	EC	2	4	op = X	. . . . .
161	STX	dp+Y	ED	2	5	¡È	. . . . .
162	STX	!abs	FC	3	5	¡È	. . . . .
163	STY	dp	E9	2	4	op = Y	. . . . .
164	STY	dp+X	F9	2	5	¡È	. . . . .
165	STY	!abs	F8	3	5	¡È	. . . . .
166	TAX		E8	1	2	X= A	N . . . . Z .
167	TAY		9F	1	2	Y = A	N . . . . Z .
168	TST	dp	4C	2	3	Test dp = 0 or not	N . . . . Z .
169	TSPX		AE	1	2	X = SP	N . . . . Z .
170	TXA		C8	1	2	A = X	N . . . . Z .
171	TXSP		8E	1	2	SP = X	N . . . . . Z .
172	TYA		BF	1	2	A = Y	N . . . . Z .
173	XAX		EE	1	4	A ¡ê X	. . . . .
174	XAY		DE	1	4	A ¡ê Y	. . . . .
175	XCN		CE	1	5	A7-4 A3-0	N . . . . Z .
176	XMA	dp	BC	2	5	A ¡ê op	N . . . . Z .
177	XMA	dp+X	AD	2	6	¡È	N . . . . Z .
178	XMA	{X}	BB	1	5	¡È	N . . . . Z .
179	XYX		FE	1	4	X ¡ê Y	. . . . .

No.	MNEMONIC		OP CODE	Words	Exec. Cycle	OPERATION	Flag MVG HIZC
180	LDYA	dp	7D	2	5	YA = (dp+1)(dp)	N . . . . . Z .
181	STYA	dp	DD	2	5	(dp+1)(dp) = YA	. . . . .
182	INCW	dp	9D	2	6	(dp+1)(dp)++	N . . . . . Z .
183	DECW	dp	BD	2	6	(dp+1)(dp)--	N . . . . . Z .
184	ADDW	dp	1D	2	5	YA + (dp+1)(dp)	N V . . H . Z C
185	SUBW	dp	3D	2	5	YA - (dp+1)(dp)	N V . . H . Z C
186	CMPW	dp	5D	2	4	CP YA, (dp+1)(dp)	N . . . . . Z C
187	CBNE	dp, rel	FD	3	5/7	if (op !=A)	. . . . .
188	CBNE	dp+X, rel	8D	3	6/8	then branch	. . . . .
189	DBNE	dp, rel	AC	3	5/7	Dec op, if (Z=0)	. . . . .
190	DBNE	Y, rel	7B	2	4/6	then branch	. . . . .
191	NOT1	M.bit	4B	3	5	M.bit = $\overline{\text{M.bit}}$	. . . . .
192	OR1	M.bit	6B	3	5	C = $\overline{\text{M.bit}}$: C	 C
193	OR1B	M.bit	6B	3	5	C = $\overline{\text{(M.bit)}}$: C	 C
194	AND1	M.bit	8B	3	4	C = $\overline{\text{M.bit}}$ & C	 C
195	AND1B	M.bit	8B	3	4	C = (M.bit) & C	 C
196	EOR1	M.bit	AB	3	5	C= M.bit$\oplus$C	 C
197	EOR1B	M.bit	AB	3	5	C = $\overline{\text{(M.bit)}}\oplus$C	 C
198	LDC	M.bit	CB	3	4	C = M.bit	 C
199	LDCB	M.bit	CB	3	4	C = $\overline{\text{(M.bit)}}$	 C
200	STC	M.bit	EB	3	6	M.bit = C	. . . . .
201	TCLR1	!abs	5C	3	6	!abs = A & !abs	N . . . . . Z .
202	TSET1	!abs	3C	3	6	!abs = A : !abs	N . . . . . Z .

APPENDIX B.
General Circuit Diagram of GMS810series.



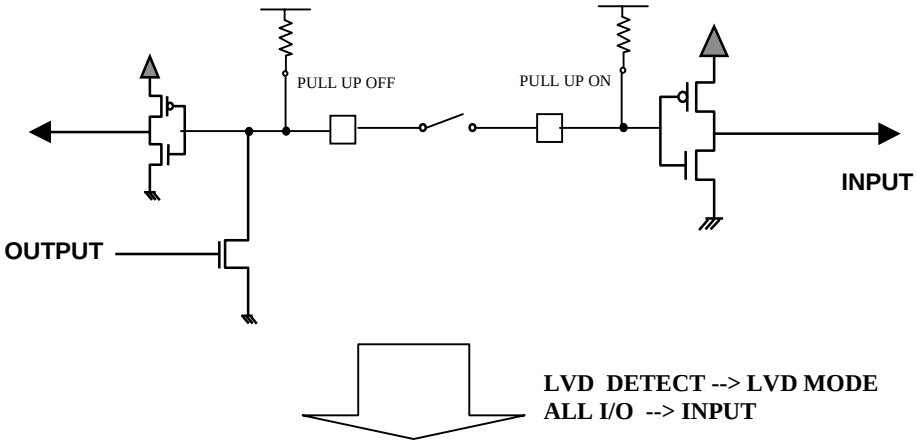




B-1 Circuit Diagram

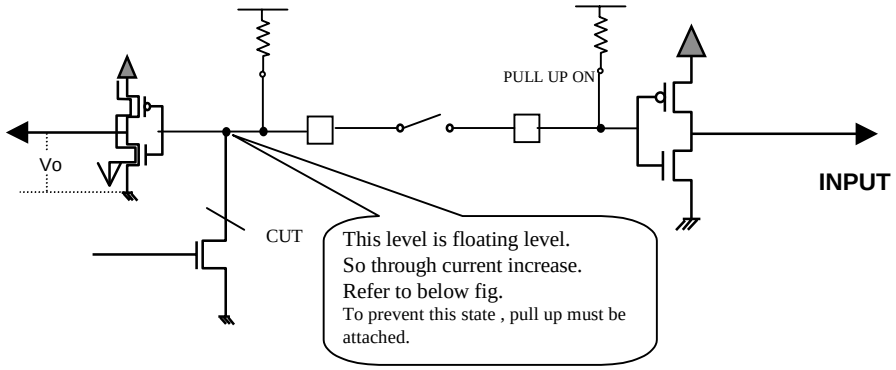
1.Normal state

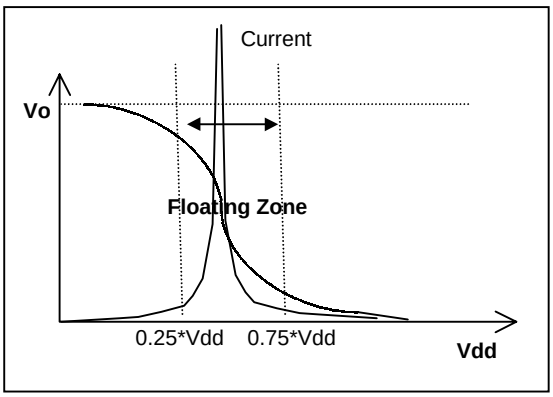
R10 PORT (OUTPUT) : pull up off & open drain R00 PORT (INPUT) : pull up on



2.LVD MODE state

R10 PORT(OUTPUT) --> INPUT





In this case , below option is right!!!

Port	R00
Y/N	Y
Y/N *0	Y

Port	R10
Y/N	N
Y/N *0	Y

Key Scan

- To secure the key board scanning , read the input port after minimum 60uS delay time from output port set to `Low `. This time delay is for the port rising time depend on the input pull-up resistor .

```
; program example ,See the circuit B-1

 .
ldm   R1,#1111_1110b      ;R10 port set to LOW
call   delay_60uS         ;60uS time delay routine
lda   R0                  ;R0 port Read
 .
 .
```

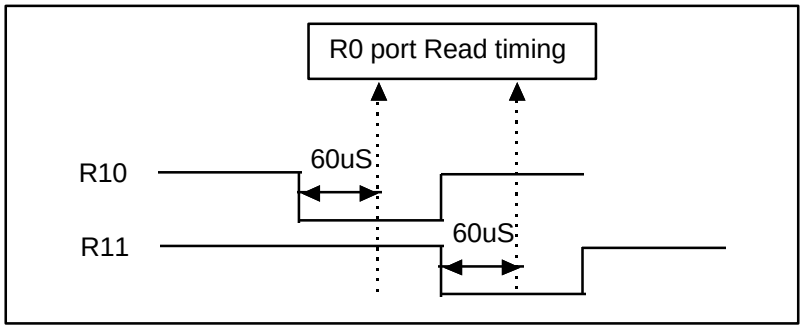


Fig B-2 , Input with pull-up port read time method

MASK ORDER & VERIFICATION SHEET

GMS810 □□

ROM CODE No.	UA

1. Customer Information

Company Name	
Application	Remote controller
Order Date	. .
TEL : FAX :	
Name & Signature :	

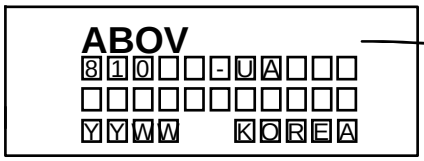
2. Device Information

Package	□ 20 SOP □ 24 SOP □ 28 SOP □ 20 PDIP □ 24 SKDIP □ 28 SKDIP □ 44 PLCC
Mask Data	□ Disk □ E-Mail ( )
File Name	. OTP
Check Sum	h □ 27256

* Set ***FFh*** in the unused area

3. Marking Specification

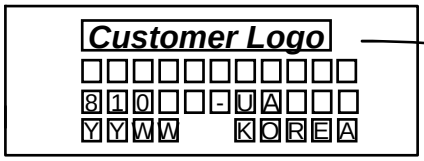
□



Customer part Numbe □□□

□□□□□□□□□□

□



□□□□□□□□□□

Customer part Number □□□

4. Delivery Schedule

	Date	Quantity	ABOV Confirmation
Customer Sample	. .	pcs	
Risk Order	. .	pcs	

5. ROM CODE Verification

ABOV Semiconductor Co., Ltd.write in below	*Customer write in below*
Verification Date : . .	Approval Date : . .
Please confirm our verification data.	***I agree with your verification data and confirm you to make mask set.***
Check Sum :	
TEL :82-2-2193-2200 FAX :82-2-508-6902	TEL : FAX :
Name & Signature	Name & Signature

GMS810 MASK OPTION LIST

Code Name : GMS810 - UA

1. Device & Package **Check Sum: h@27c256**

Please enter check marks as √

GMS81004	☐	GMS81024	☐
GMS81008	☐	GMS81032	☐
GMS81016	☐		
20PIN : SOP	☐	PDIP	☐
24PIN : SOP	☐	Skinny DIP	☐
28PIN : SOP	☐	Skinny DIP	☐
44PIN : PLCC	☐		

2. Inclusion of Pull up Resistor

- R0 PORT

Port	R00	R01	R02	R03	R04	R05	R06	R07
Y/N								
Y/N *0								

Y : Yes
N : No

- R1 PORT

Port	R10	R11	R12*2	R13*2	R14*2	R15*2	R16	R17
Y/N								
Y/N *0								

Y : Yes
N : No

- R2 PORT

Port	R20	R21*1	R22*1	R23*1	R24*1
Y/N					
Y/N *0					

Y : Yes
N : No

< NOTICE >

. *0 : must be selected for **Low Voltage detection Mode**..(HEI device has default LVD Circuit)

*1 : is not available for 20PIN & 24PIN. So, Default Option is Pull-Up.

. *2 : is not available for 20PIN. So Default Option is Pull-Up.

3. Low Voltage Detection for Ram Retention

Y/N	

Date :

Company Name :

Section Name :

Signature :