NOV. 1996 Rev. 2.1

8-BIT SINGLE CHIP MICROCOMPUTER

GMS84512 / 84524

USER'S MANUAL

HYUNDAI MicroElectronics

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

3. Peripheral Function

4. Control Function

5. Support Tool

6. Appendix

1. Overview

1. Features
2. Block Diagram
3. Summary Of Peripheral Function Register
4. Pin Assignment
5. Pin Description
6. Terminal Types

2. CPU

1. Registers
2. Memory Space

3. Peripheral Function

1. PORT
2. Clock Generation Circuit
3. Timer
4. A/D Comparator
5. Serial I/O
6. PWM
7. Interrupt Interval Measurement Circuit
8. On Screen Display

4. Control Function

1. Interrupts
2. Standby Function
3. Reset Function

5. Support Tool

1. Emulator
2. Debugger
3. Assembler
4. Linker
5. Font Editor
6. OTP Chip

6. Appendix

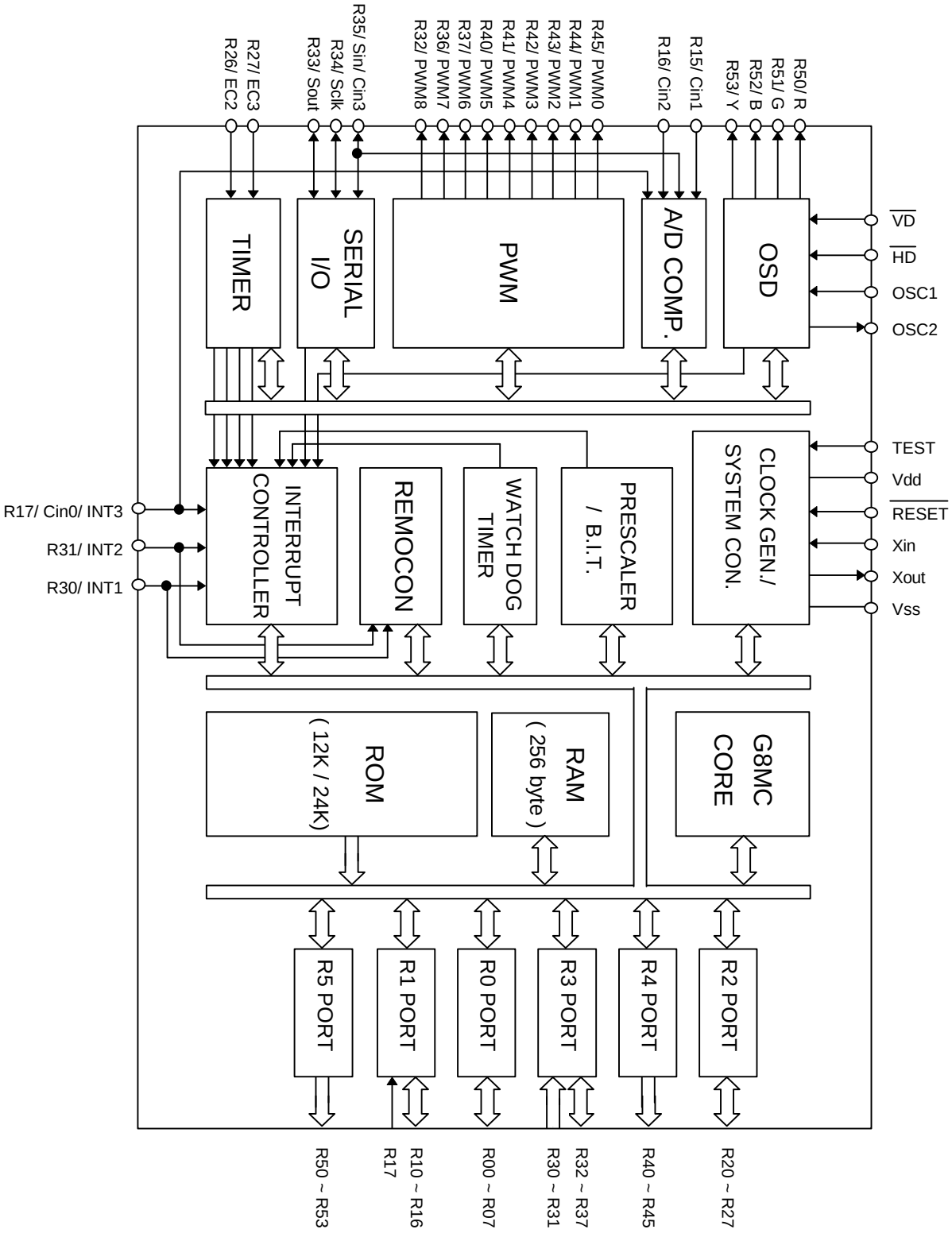
- Electrical Data
- Package Outline

An 8-bit microcomputer using the G8MC Core is a single-chip microcomputer including several peripheral functions such as Timer, I/O Comparator, Serial I/O, PWM, Watch-dog Timer and On-Screen Display.

1.1 FEATURES

- ROM — 12,288 Bytes (GMS84512) 24,576Bytes (GMS84524)
- RAM — 256 Bytes
- Minimum instruction execution time — 1 us (@ Xin = 4 MHz)
- I/O PORT — 42 (INPUT: 3, OUTPUT: 10, I/O: 29)
- Serial I/O — 8-bit X 1 ch. (1MHz, 500KHz, 250 KHz, Ext. clock)
- A/D Comparator — 5-bit X 4 ch. (max. 1 LSB)
- Pulse Width Modulation — 14-bit X 1 ch.
  7-bit X 8 ch.
- Timer
  - Timer/Counter — 8 bit X 4 ch. (16-bit X 2 ch is Acceptable)
  - Basic Interval Timer — 8 bit X 1 ch.
  - Watch Dog Timer
- Interrupt Interval estimation circuit — for Remocon signal receiving
- Interrupt Sources — 14 sources
- Pulse (T2048) Output Function — Period : 2,048 us, Duty: 50 %
- On Screen Display
  - Kinds of character — 128 kinds (include 2 test characters)
  - Construction of character — 14 dots X 18 dots
  - Size of character — 4 X 4 kinds
  - Number of display character — 22 Characters X 3 lines (Max. 12 lines)
  - Display colors — 8 kinds
  - Color Edge, Smoothing Function
- Power Save Mode — STOP mode
- Operating Voltage — 4.5 ~ 5.5 V
- Package — 52 SDIP
- OTP chip — GMS84512T/84524T

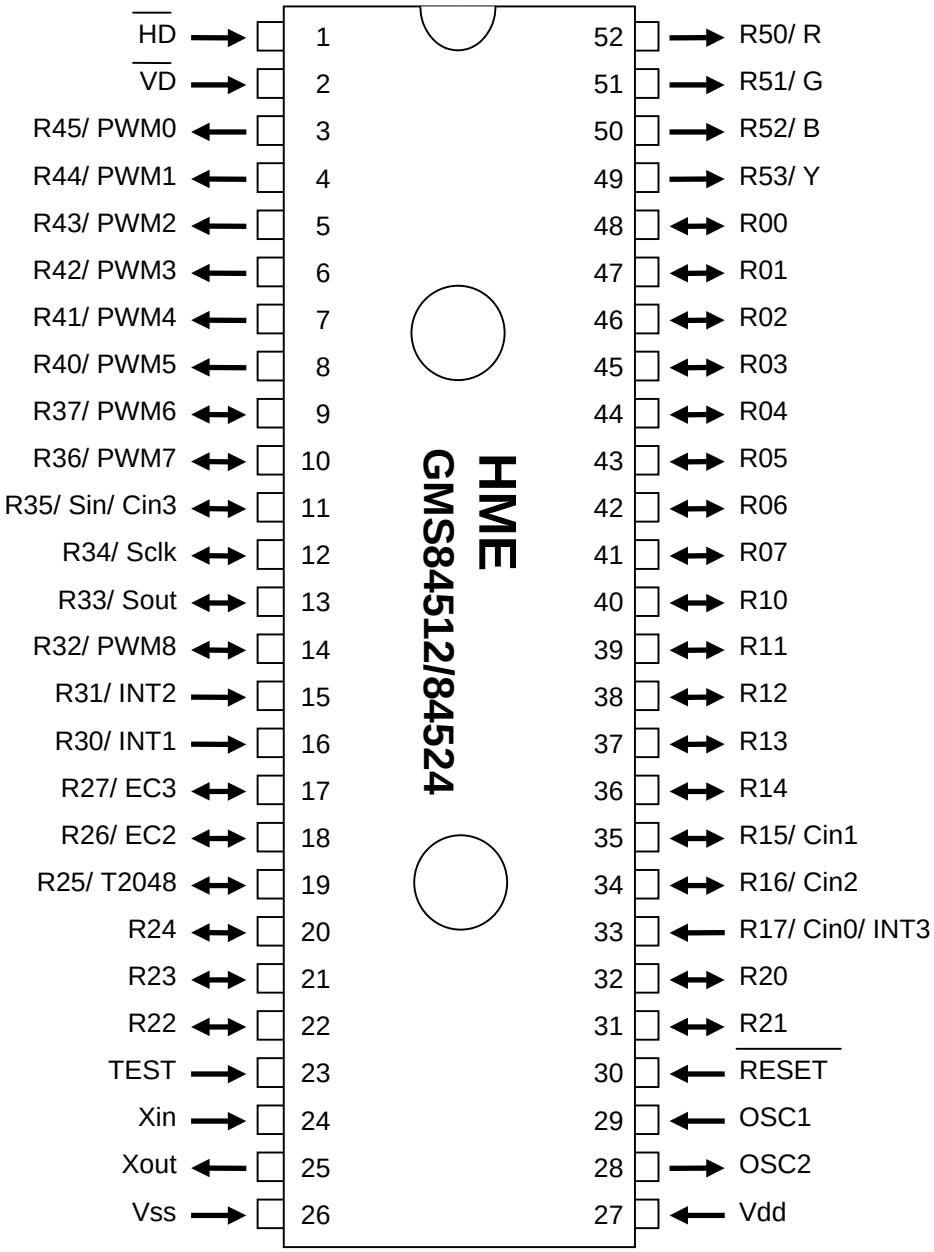
1.2 BLOCK DIAGRAM



1.2 Peripheral Function Overview

BLOCK	Function	INDEX
PRESCALER / B.I.T.	Prescaler is consists of 10 bits binary counter, and divide oscillation clock. The divided output from each bit of prescaler provided to peripheral hardware. B.I.T a 8 bit binary counter has a function such as security of oscillation stabilization time, generation of basic interval time interrupt as watch function, providing the clock for watch-dog timer	3 - 13
WATCH-DOG -TIMER	WDT is consist of 6-bit binary counter, WDTR(Watch-Dog Timer Register), and comparator, input clock of WDT is provided by Basic Interval Timer interrupt and maximum output cycle is 4 seconds. When WDTOM is '1', the output of WDT reset the Device.	3 -16
TIMER / COUNTER	Timer is an 8 bit binary counter and consisted of T0, T1, T2, T3. As an 8-bit binary counter, each T0, T1 can be used 16-bit interval Timer to connect each other. As an 8 bit binary counter/event counter each T2, T3 can be used 16-bit/event counter to connect each other. At 4 MHz oscillation, Maximum interval time of T0 is 8.192 ms, T1 is 2048 ms, T0-T1 is about 2 seconds, T2 is 2.048 ms, T3 is 512 uS,T2-T3 is about 0.5 seconds	3 - 19
A/D COMP-ARATOR	A/D Comparator has 5 bit resolution, and 4 input channel. It has sample and hold function of input. At 4 MHz it takes about 8 uS to compare. Error is less than 1/2 LSB.	3 - 26
SERIAL I/O	It is 8 bit clock synchronous serial interface unit, the clock transmission cycle is 1uS, 2uS,4uS Which can be selected external clock. When IOSW(Bit 6 Of Serial I/O Mode Register) is '1', R33 pin operates Sout at transmission mode, Sin at receiving mode.	3 - 28
PWM (Pulse Width Modulation)	PWM is consists of 14 bit PWM 1 ch and 7 bit PWM 8 ch. 14 bit PWM has 0.5 uS minimum resolution width, 8192 uS cycle time, 7 bit PWM has 8uS minimum resolution 8 uS,1024uS, cycle time. The polarity of PWM output can be assign by Software.	3 - 32
INTERRUPT INTERVAL MEASUREM-ENT CIRCUIT	Interrupt interval measurement circuit consists of 8 bit binary counter, interrupt interval saving circuit. It can select 32uS, 64uS as a measurement clock . Because it can select external signal edge, measurement of input signal cycle or pulse width is possible. So it can be used Remocon receiving.	3 - 38
OSD (On-Screen-Display)	Maximum number of character or symbol displayed in CRT is 128 basically displayed by 22 characters X 3 lines. Maximum 12 lines is possible with OSD interrupt. OSD clock can use 4 MHz ~ 8 MHz size of display character is 16 kinds, it can be used by line unit. The color of display character is 8 kinds it can be used by character unit. In display mode, there are character mode, background mode, color mode, and Blanking mode, it can be used by line unit especially smoothing function and OSD oscillator control function exists.	3 - 41

1.4 PIN ASSIGNMENT

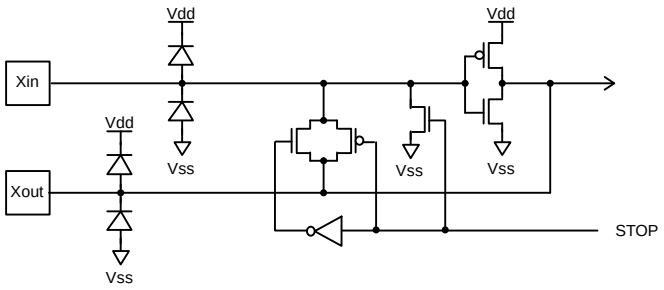
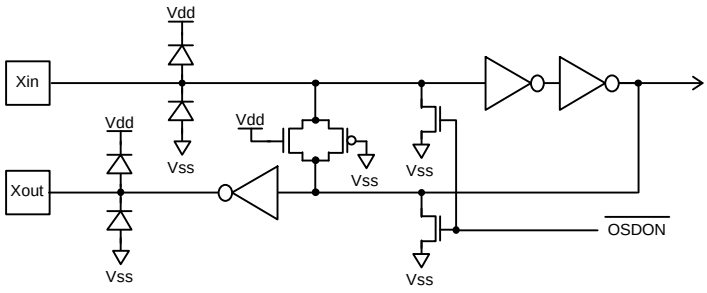
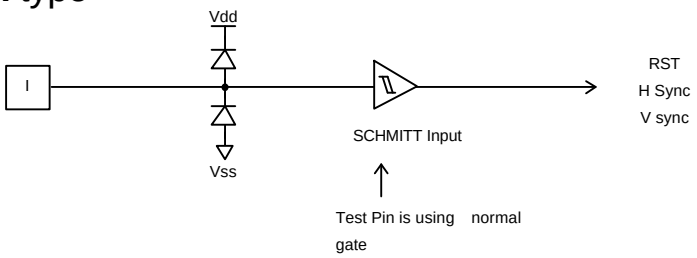
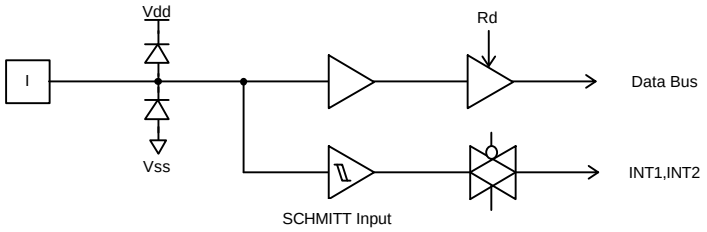
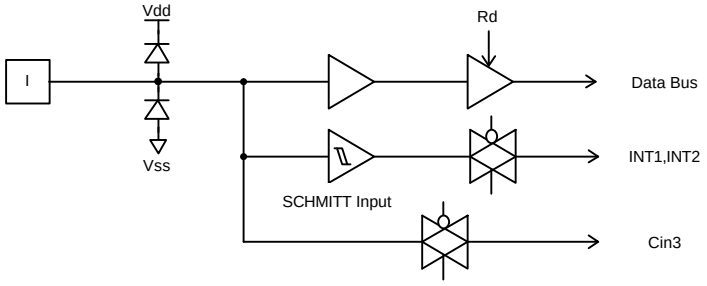


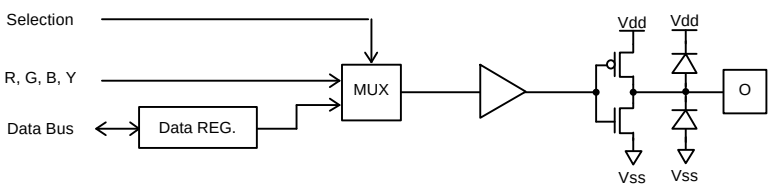
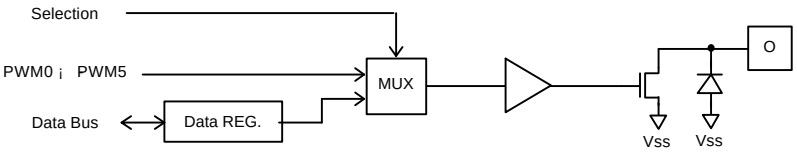
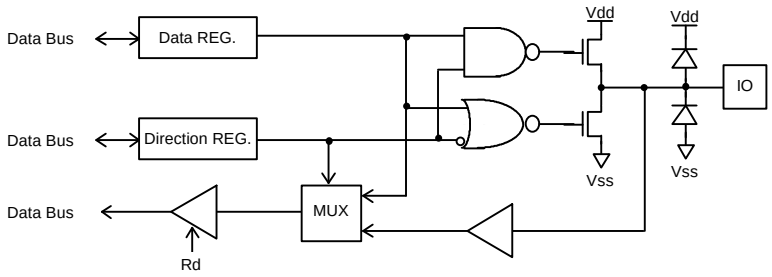
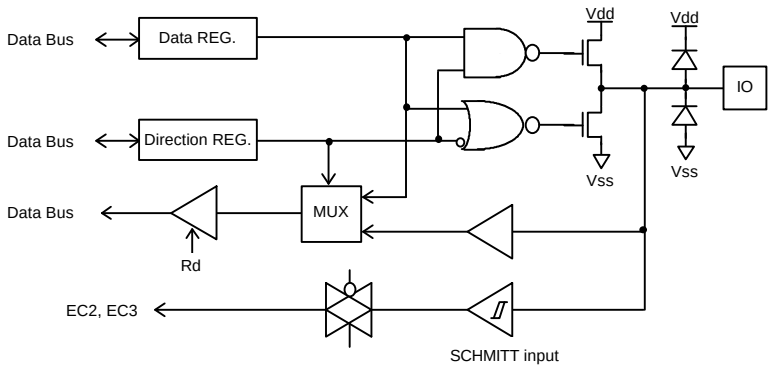
1.5 PIN DESCRIPTION

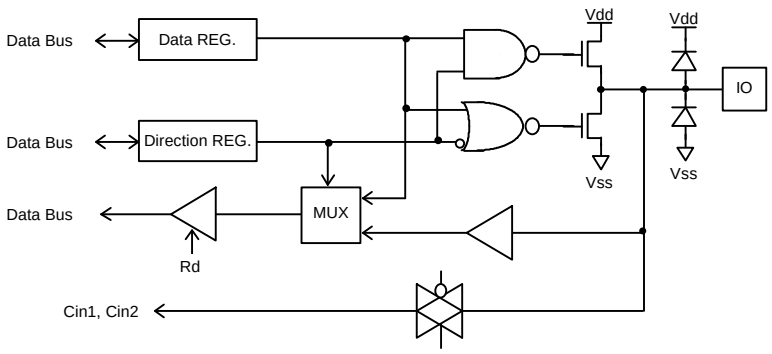
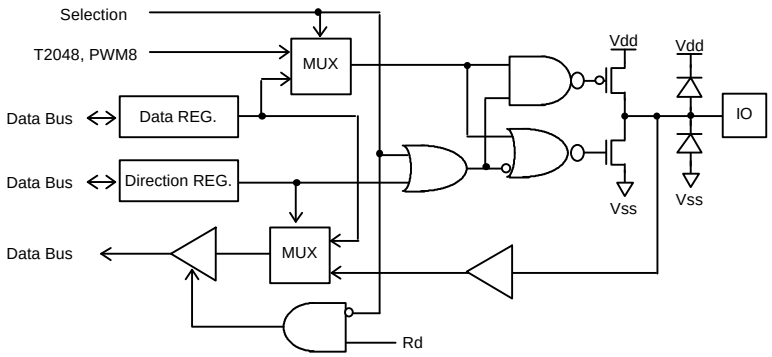
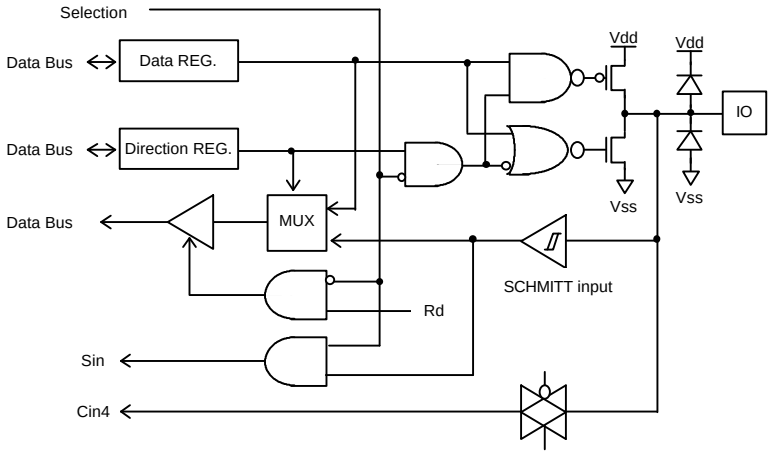
Classification	No.	Symbol	I/O	Function	Type	Remark
Power	27	Vdd	Input	Power supply (4.5~5.5V)		
	26	Vss	Input	Ground (0V)		
System Control or Clock	23	TEST	Input	TEST Input pin At 'L' input: SINGLE CHIP MODE At 'H' input : TEST MODE	**IA**	
	24	Xin	Input	CRYSTAL connection pin (with Xout) If an external clock is used, Xin pin should be connected external clock source		
	25	Xout	Output	CRYSTAL connection pin(with Xin) If an external clock used, Xout pin should be open		
	30	$\overline{\text{RESET}}$	Input	In the state of 'L' level, system enter the reset state	**IA**	
OSD	1	$\overline{\text{HD}}$	Input	Horizontal synchronizing signal input pin	**IA**	
	2	$\overline{\text{VD}}$	Input	Vertical synchronizing signal input pin		
	28	OSC2	Ouptut	Clock output for OSD		
	29	OSC1	Input	Clock input for OSD		
	49	Y	Output	Switching signal output pin	**OA**	R53 share
	50	B	Output	BLUE signal output pin		R52 share
	51	G	Output	GREEN signal output pin		R51 share
	52	R	Output	RED signal output pin		R50 share
PWM	3	PWM0	Output	Pulse width modulation output pin	**OB**	R45 share
	4	PWM1	Output	(7BIT PWM)		R44 share
	5	PWM2	Output			R43 share
	6	PWM3	Output			R42 share
	7	PWM4	Output			R41 share
	8	PWM5	Output			R40 share
	9	PWM6	Output		**IOF**	R37 share
	10	PWM7	Output			R36 share
	14	PWM8	Output	14BIT PWM output pin	**IOD**	R32 share
	19	T2048	Output	Pulse(2048uS) output pin		R25 share
SCI	11	Sin	Input	Serial Data Input pin	**IOE**	R35 share
	12	Sclk	I/O	Serial Clock I/O pin	**IOG**	R34 share
	13	Sout	Output	Serial Data output pin		R33 share
TIMER	17	EC3	Input	Event Counter input pin	**IOB**	R27 share

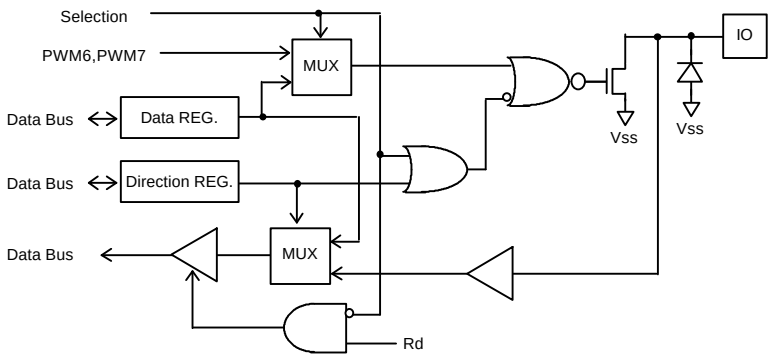
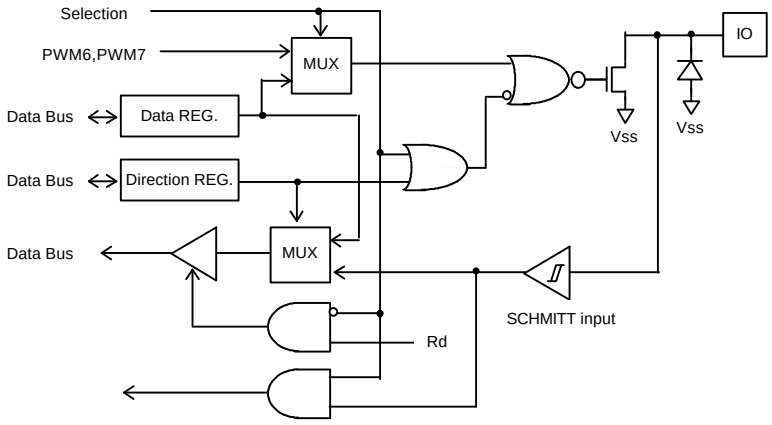
Classification	NO.	Symbol	I/O	Function	TYPE	Remarks
Interrupt	15	INT2	Input	External interrupt request input pin	**IB**	R31 share
	16	INT1	Input	(INT1,INT2 : Remocon input capture		R30 share
	33	INT3	Input	Input possible)	**IC**	R17 share
A/D Comparator	11	Cin3	Input	Analog input pin	**IOE**	R35 share
	33	Cin0	Input	(Default selection : Cin0)	**IC**	R17 share
	34	Cin2	Input		**IOC**	R16 share
	35	Cin1	Input			R15 share
I/O Port	41	R07	I/O	R0 Port		
	≀	≀		(Can assigned I/O state bit by bit by R0DD)	**IOA**	
	48	R00	I/O			
	33	R17	Input	R1 Port (R17 Input only)	**IC**	Cin0/INT3 share
	34	R16	I/O	(7 ports of R10~R16 can assigned I/O	**IOC**	Cin2 shrae
	≀	≀		state bit by bit by R1DD		Cin1 share
	40	R10	I/O		**IOA**	
	17	R27	I/O	R2 Port	**IOB**	EC3 share
	18	R26	I/O	(Can assigned I/O state bit by bit		EC2 share
	19	R25	I/O	by R2DD)	**IOD**	T2048 share
	20	R24	I/O			
	21	R23	I/O			
	22	R22	I/O		**IOA**	
	31	R21	I/O			
	32	R20	I/O			
	9	R37	I/O	R3 PORT	**IOB**	PWM6 share
	10	R36	I/O	(6 Bits of R31~R32 can assigned I/O		PWM7 share
	11	R35	I/O	state bit by bit by R3DD	**IOE**	Sin/Cin3 share
	12	R34	I/O		**IOG**	Sclk share
	13	R33	I/O			Sout share
	14	R32	I/O		**IOD**	PWM8 share
	15	R31	Input	(R30,R31 is input only)	**IB**	INT2 share
	16	R30	Input			INT1 share
	3	R45	Output	R4 Port		PWM0 share
	4	R44	Output	(6 bit output only)		PWM1 share
	5	R43	Output		**OB**	PWM2 share
	6	R42	Output			PWM3 share
	7	R41	Output			PWM4 share
	8	R40	Output			PWM5 share
	49	R53	Output	R5 Port		Y share
	50	R52	Output	(4 bit output only)	**OA**	B share
	51	R51	Output			G share
	52	R50	Output			R share

1.6 TERMINAL TYPES

PIN	TERMINAL TYPE	at RESET
Xin **Xout**		**Oscillation**
OSC1 **OSC2**		**Oscillation Stop**
$\overline{\text{RESET}}$ $\overline{\text{HD}}$ $\overline{\text{VD}}$ **TEST**	**IA** type 	**(“L”)** **Hi-Z**
R30/ INT1 **R31/ INT2**	**IB** type 	**Hi-Z**
R17/ Cin0 **/ INT3**	**IC** type 	**Hi-Z**

PIN	TERMINAL TYPE	at RESET
R50/ R **R51/ G** **R52/ B** **R53/ Y**	**OA** type 	**Hi-Z**
R45/ PWM0 **R44/ PWM1** **R43/ PWM2** **R42/ PWM3** **R41/ PWM4** **R40/ PWM5**	**OB** type 	**Hi-Z**
R00 ~ R07 **R10 ~ R14** **R20 ~ R24**	**IOA** type 	**Hi-Z**
R26/ EC2 **R27/ EC3**	**IOB** type 	**Hi-Z**

PIN	TERMINAL TYPE	at RESET
R15/ Cin1 **R16/ Cin2**	**IOC** type 	**Hi-Z**
R25/ T2048 **R32/ PWM8**	**IOD** type 	**Hi-Z**
R35/ Sin **/ Cin4**	**IOE** type 	**Hi-Z**

PIN	TERMINAL TYPE	at RESET
R36/ PWM7 **R37/ PWM6**	**IOF** type 	**Hi-Z**
R34/ Sclk **R33/ Sout**	**IOG** type 	**Hi-Z**

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

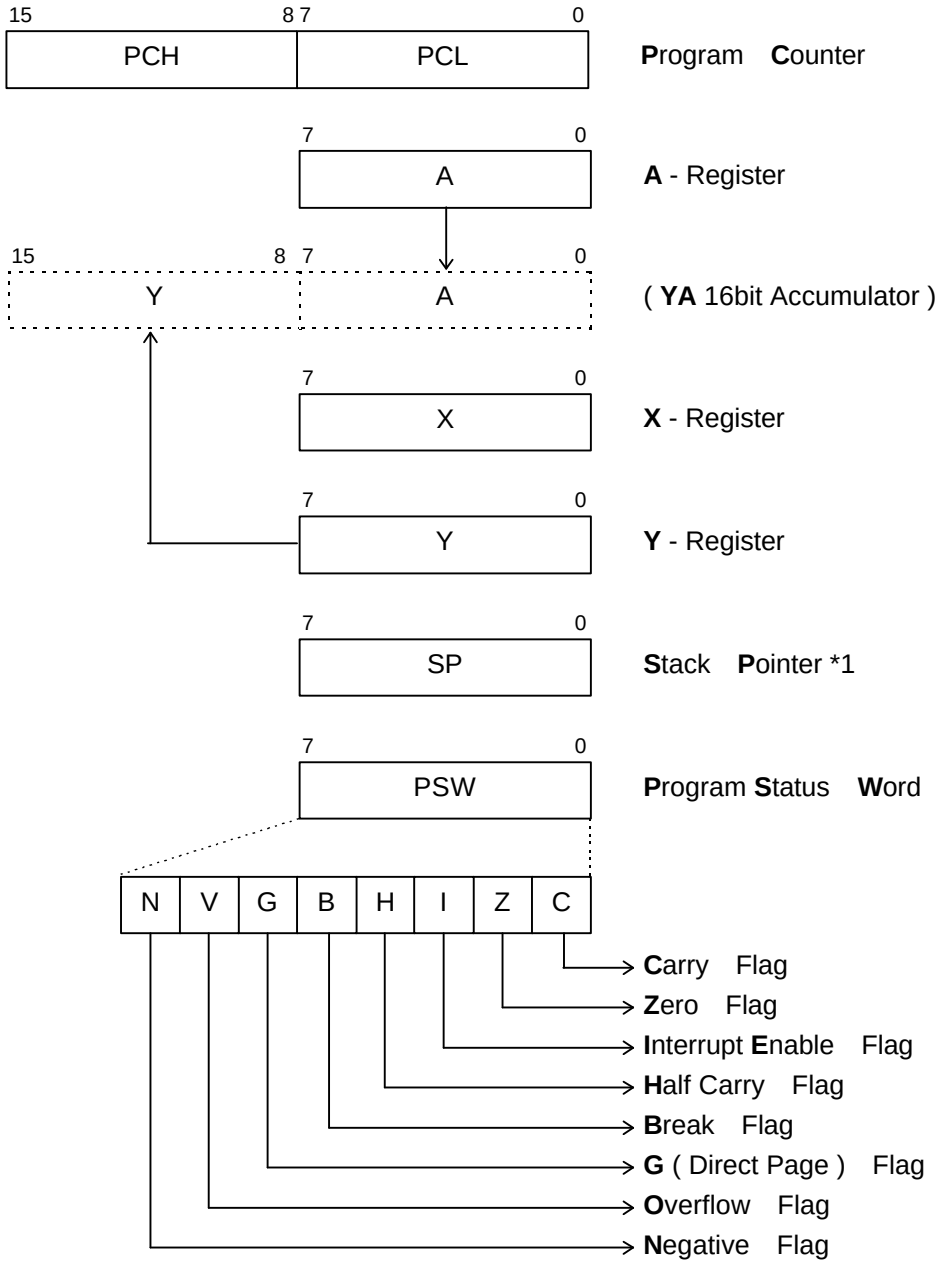
3. Peripheral Function

4. Control Function

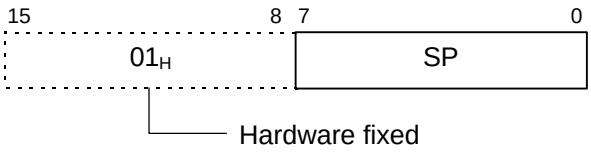
5. Support Tool

6. Appendix

2.1. REGISTERS



*1 STACK ADDRESS (0100_H ~ $013F_H$)



2.1.1. A - Register

- 8 bit Accumulator
- In the case of 16-bit operation, compose the lower 8-bit of YA (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 → YA)
- In the case of division instruction, execute as the lower 8-bit of dividend.
  After division operation, quotient enters. (YA ¡À X ¡æ Q: A , R: Y)

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 G mode operation, execute as destination address register.
  The operation result enters into memory indirectly addressed by X register.
- In the case of division instruction, execute as divisor register.

2.1.3. Y- Register

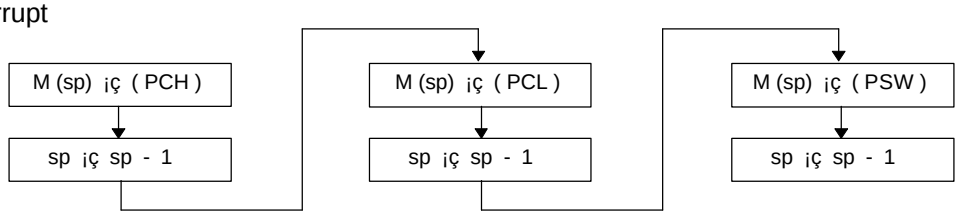
- General-purpose 8-bit register
- In the case of index addressing mode, execute as index register
- In the case of 16-bit operation instruction, execute as the upper 8-bit of YA (16-bit accumulator).
- In the case of multiplication instruction, execute as a multiplicand register.
  After multiplication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

2.1.4. Stack Pointer

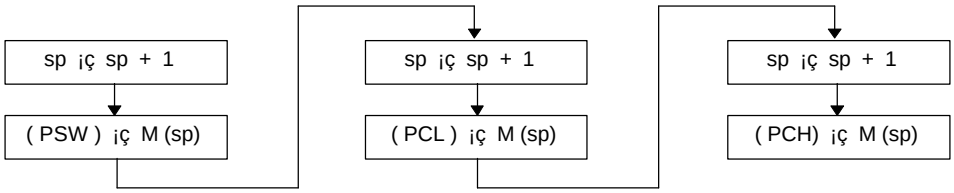
- In the cases of subroutine call, Interrupt and PUSH, POP, RETI, RET instruction, stack data on RAM or in the case of returning, assign the storage location having stacked data.
- Stack area is constrained within 1-page (00H-FFH). Page is fixed by H/W. User can only assign the lower address. At the initial stage, stack pointer should be initialized to

point to RAM area having H/W.

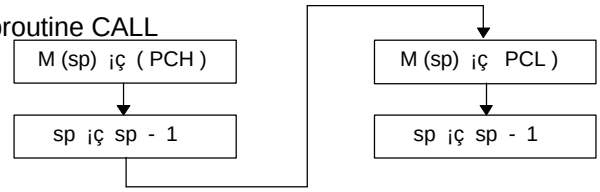
¨çInterrupt



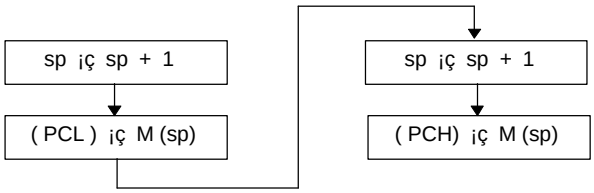
¨èRETI



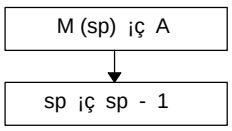
¨éSubroutine CALL



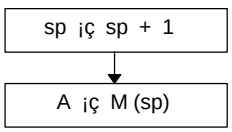
¨êRET



¨ëPUSH A (X, Y, PSW)



¨ìPOP A (X, Y, PSW)



2.1.5. Program Counter (**PC**)

- Program counter is a 16-bit counter consisted of 8-bit register PCH and PCL.
- Addressing space is 64K bytes.
- In reset state, Reset routine address in address FFFFH and FFFEH enter into PC.

2.1.6. Program Status Word(**PSW**)

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

PSW

7	6	5	4	3	2	1	0
N	V	G	B	H	I	Z	C

¨ ç 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.

¨ è Zero Flag (**Z**)

- After operation also including 16-bit operation, set if the result is “0”.
- Branch condition flag of BEQ, BNE.

¨ é Interrupt Enable Flag (**I**)

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

¨ ê 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.

¨ ë Break Flag (**B**)

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

¨ ì Direct Page Flag (**G**)

- Assign direct page (0-page, 1-page).
- Set and cleared by SETG, CLRG instruction.
- If used with PG2R($00FC_H$) it is enable to access 2-page (OSD RAM).

G-flag	PG2R	Direct Page
0	-	**0** - Page Access
1	0	**1** - Page Access
	1	**2** - Page Access

*NOTICE : Always after clearing, PG2R is enable to be accessed for it is the register of 0-page

¨ í Overflow Flag (**V**)

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

¨ î Negative Flag (**N**)

- N-flag is set whenever the result of a data transfer or operation is negative (bit7 isset to “1”).
- In the case of BIT instruction, bit7 of memory location is inputted to N-flag
- No CLEAR and SET instruction.
- Branch condition flag of BPL, BMI.

2.2 MEMORY SPACE

The memory space of GMS84512/84524 is 64K byte, it is equipped with RAM area, OSD RAM area, FONT ROM area and PROGRAM ROM area.

2.2.1. RAM area

0-PAGE (0000_H ~ $00FF_H$)

RAM 192 Bytes (0000_H ~ $00BF_H$) and peripheral function register($00C0_H$ ~ $00FF_H$)

1-PAGE (0100_H ~ $013F_H$)

RAM 64 Bytes (0100_H ~ $013F_H$) and STACK area

2-PAGE (0200_H ~ $02D5_H$)

OSD RAM 182 Bytes (0200_H ~ $02D5_H$)

2.2.2. FONT ROM area (2000_H ~ $3FFF_H$)

128 character OSD FONT

2.2.3. PROGRAM ROM area

Approximately ROM memory is 12 K bytes and it is domain of User Program.

The highest page($FF00_H$ ~ $FFFF_H$) is called U- Page and it is utilized domain as following.

PCALL area ($FF00_H$ ~ $FFBF_H$)

Domain of jumping at PCALL instruction

TCALL Vector area ($FFC0_H$ ~ $FFDF_H$)

Storage domain of vector address at TCALL instruction.

Interrupt Vector area ($FFE0_H$ ~ $FFFF_H$)

Storage domin of interrupt vector address,inclusive RESET

MEMORY MAP (GMS84512/84524)

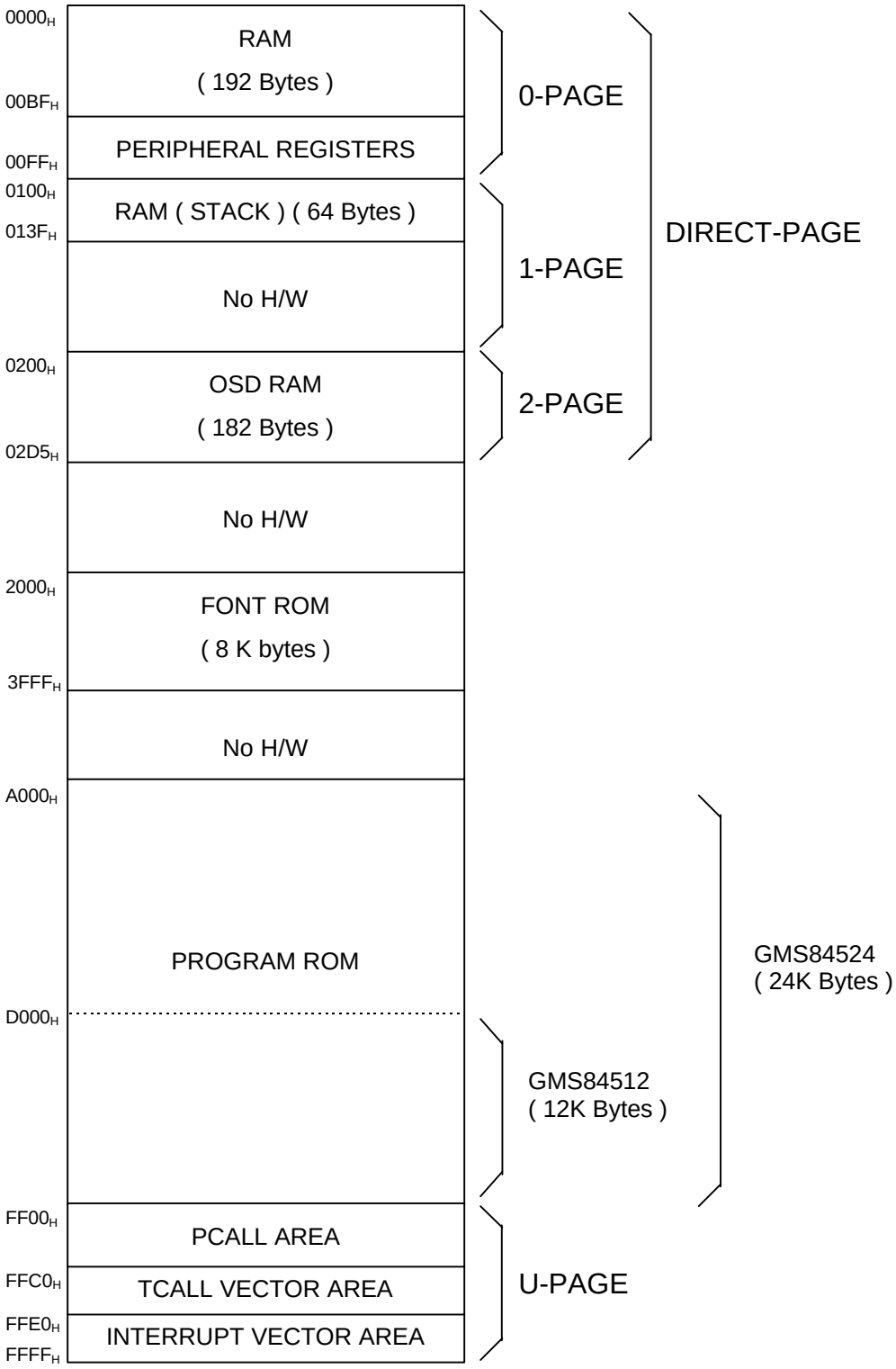


TABLE 2.1. PERIPHERALREGISTER LIST

Address	Register Name	SYMBOL	R/W	RESET VALUE								Page
				7	6	5	4	3	2	1	0	
$00C0_H$	R0 PORT DATA REGISTER	**R0**	R/W	Undefined								3 - 1
$00C1_H$	R0 PORT I/O DIRECTION REGISTER	**R0DD**	W	0	0	0	0	0	0	0	0	3 - 1
$00C2_H$	R1 PORT DATA REGISTER	**R1**	R/W	Undefined								3 - 2
$00C3_H$	R1 PORT I/O DIRECTION REGISTER	**R1DD**	W	-	0	0	0	0	0	0	0	3 - 2
$00C4_H$	R2 PORT DATA REGISTER	**R2**	R/W	Undefined								3 - 4
$00C5_H$	R2 PORT I/O DIRECTION REGISTER	**R2DD**	W	0	0	0	0	0	0	0	0	3 - 4
$00C6_H$	R3 PORT DATA REGISTER	**R3**	R/W	Undefined								3 - 6
$00C7_H$	R3 PORT I/O DIRECTION REGISTER	**R3DD**	W	0	0	0	0	0	0	-	-	3 - 6
$00C8_H$	R4 PORT DATA REGISTER	**R4**	R/W	-	-	Undefined						3 - 9
$00C9_H$	R5 PORT DATA REGISTER	**R5**	R/W	-	-	-	-	Undefined				3 - 10
$00CA_H$	PORT FUNCTION SELECTION REGISTER	**FUNC**	W	-	-	-	0	0	0	0	0	3 - 3
$00CB_H$	EXT. INTERRUPT EDGE SELECTION REGISTER	**IEDS**	W	-	-	0	0	0	0	0	0	3 - 39
$00CC_H$	OPERATION MODE REGISTER	**TMR**	W	-	-	-	-	-	0	0	0	¡ [a]
$00CE_H$	BASIC INTERVAL TIMER REGISTER	**BITR**	R	Undefined								3 - 16
	CLOCK CONTROL REGISTER	**CKCTLR**	W	-	-	0	1	0	1	1	1	3 - 13
$00CF_H$	WATCH-DOG TIMER REGISTER	**WDTR**	W	-	0	1	1	1	1	1	1	3 - 17
$00D0_H$	TIMER MODE REGISTER0	**TM0**	R/W	-	0	0	0	0	0	0	0	3 - 21
$00D1_H$	TIMER MODE REGISTER2	**TM2**	R/W	-	0	0	0	0	0	0	0	3 - 21
$00D2_H$	TIMER0 DATA REGISTER	**TDR0**	R/W	Undefined								3 - 21
$00D3_H$	TIMER1 DATA REGISTER	**TDR1**	R/W	Undefined								3 - 21
$00D4_H$	TIMER2 DATA REGISTER	**TDR2**	R/W	Undefined								3 - 21
$00D5_H$	TIMER3 DATA REGISTER	**TDR3**	R/W	Undefined								3 - 21
$00D6_H$	A/D COMPARATOR MODE REGISTER	**CMR**	W *6	0	0	-	0	0	0	0	0	3 - 27
$00D7_H$	A/D COMP. CHANNEL SELECTION REGISTER	**CIS**	W	-	-	-	-	-	-	0	0	3 - 27
$00D8_H$	SERIAL I/O MODE REGISTER	**SIOM**	R/W *0	-	0	0	0	0	0	0	1	3 - 29
$00D9_H$	SERIAL I/O DATA REGISTER	**SIOR**	R/W	Undefined								3 - 28
$00DA_H$	PWM0 DATA REGISTER	**PWMR0**	W	-	Undefined							3 - 35
$00DB_H$	PWM1 DATA REGISTER	**PWMR1**	W	-	Undefined							3 - 35
$00DC_H$	PWM2 DATA REGISTER	**PWMR2**	W	-	Undefined							3 - 35
$00DD_H$	PWM3 DATA REGISTER	**PWMR3**	W	-	Undefined							3 - 35

Address	Register Name	SYMBOL	R/W	RESET VALUE								Page
				7	6	5	4	3	2	1	0	
$00DE_H$	PWM4 DATA REGISTER	**PWMR4**	W	-	Undefined							3 - 35
$00DF_H$	PWM5 DATA REGISTER	**PWMR5**	W	-	Undefined							3 - 35
$00E0_H$	PWM6 DATA REGISTER	**PWMR6**	W	-	Undefined							3 - 35
$00E1_H$	PWM7 DATA REGISTER	**PWMR7**	W	-	Undefined							3 - 35
$00E2_H$	PWM8 DATA REGISTER HIGH	**PWM8H**	R/W	Undefined								3 - 36
$00E3_H$	PWM8 DATA REGISTER LOW	**PWM8L**	R/W	-	-	Undefined						3 - 36
$00E4_H$	PWM CONTROL REGISTER1	**PWMCR1**	R/W	0	0	0	0	0	0	0	0	3 - 37
$00E5_H$	PWM CONTROL REGISTER2	**PWMCR2**	R/W	-	-	-	0	0	0	0	0	3 - 37
$00E6_H$	INTERRUPT MODE REGISTER	**IMOD**	R/W	-	-	0	0	0	0	0	0	4 - 4
$00E8_H$	INTERRUPT ENABLE REGISTER LOW	**IENL**	R/W	0	0	0	0	0	-	-	-	4 - 3
$00E9_H$	INTERRUPT REQUEST FLAG REGISTER LOW	**IRQL**	R/W	0	0	0	0	0	-	-	-	4 - 4
$00EA_H$	INTERRUPT ENABLE REGISTER HIGH	**IENH**	R/W	0	0	0	0	0	0	0	0	4 - 3
$00EB_H$	INTERRUPT REQUEST FLAG REGISTER HIGH	**IRQH**	R/W	0	0	0	0	0	0	0	0	4 - 4
$00EC_H$	INTERRUPT INTERVAL DETERMINATION CONTROL REGISTER	**IDCR**	R/W	-	-	-	-	-	0	0	0	3 - 40
$00ED_H$	INTERRUPT INTERVAL DETERMINATION REGISTER	**IDR**	R	0	0	0	0	0	0	0	0	3 - 38
$00F0_H$	OSD 1st LINE HORIZONTAL POSITION REGISTER	**HDP1**	W	-	-	0	0	0	0	0	0	3 - 47
$00F1_H$	OSD 2nd LINE HORIZONTAL POSITION REGISTER	**HDP2**	W	-	-	0	0	0	0	0	0	3 - 47
$00F2_H$	OSD 3rd LINE HORIZONTAL POSITION REGISTER	**HDP3**	W	-	-	0	0	0	0	0	0	3 - 47
$00F3_H$	OSD 1st LINE VERTICAL POSITION REGISTER	**VDP1**	W	-	0	0	0	0	0	0	0	3 - 47
$00F4_H$	OSD 2nd LINE VERTICAL POSITION REGISTER	**VDP2**	W	-	0	0	0	0	0	0	0	3 - 47
$00F5_H$	OSD 3rd LINE VERTICAL POSITION REGISTER	**VDP3**	W	-	0	0	0	0	0	0	0	3 - 47
$00F6_H$	OSD 1st LINE DISPLAY MODE, CHARACTER SIZE, SMOOTHING FUNCTION SELECTION REGISTER	**DMSS1**	W	-	0	0	0	0	0	0	0	3 - 44
$00F7_H$	OSD 2nd LINE DISPLAY MODE, CHARACTER SIZE, SMOOTHING FUNCTION SELECTION REGISTER	**DMSS2**	W	-	0	0	0	0	0	0	0	3 - 44
$00F8_H$	OSD 3rd LINE DISPLAY MODE, CHARACTER SIZE, SMOOTHING FUNCTION SELECTION REGISTER	**DMSS3**	W	-	0	0	0	0	0	0	0	3 - 44
$00F9_H$	OSD OUTPUT and BACKGROUND CONTROL REGISTER	**OSDCON1**	W	0	0	0	0	0	0	0	0	3 - 48
$00FA_H$	I/O POLARITY CONTROL and OSD OSCILLATION CONTROL REGISTER	**OSDCON2**	W	0	0	0	0	0	0	0	0	3 - 48
$00FC_H$	OSD RAM (2 page) ACCESSABLE REGISTER	**PG2R****	R/W	-	-	-	-	-	-	-	0	3 - 43

¡ Ø -: Not used *0: READ only for bit 0 *6: READ only for bit 6

¡Ø Write Only Register can **not be accessed** by bit manipulation instruction.

**** :** OSD RAM area (2-page) can be accessed by LDM,SET1

VECTOR AREA

$FFC0_H$	TCALL 15	(L)
$FFC1_H$		(H)
$FFC2_H$	TCALL 14	(L)
$FFC3_H$		(H)
$FFC4_H$	TCALL 13	(L)
$FFC5_H$		(H)
$FFC6_H$	TCALL 12	(L)
$FFC7_H$		(H)
$FFC8_H$	TCALL 11	(L)
$FFC9_H$		(H)
$FFCA_H$	TCALL 10	(L)
$FFCB_H$		(H)
$FFCC_H$	TCALL 9	(L)
$FFCD_H$		(H)
$FFCE_H$	TCALL 8	(L)
$FFCF_H$		(H)
$FFD0_H$	TCALL 7	(L)
$FFD1_H$		(H)
$FFD2_H$	TCALL 6	(L)
$FFD3_H$		(H)
$FFD4_H$	TCALL 5	(L)
$FFD5_H$		(H)
$FFD6_H$	TCALL 4	(L)
$FFD7_H$		(H)
$FFD8_H$	TCALL 3	(L)
$FFD9_H$		(H)
$FFDA_H$	TCALL 2	(L)
$FFDB_H$		(H)
$FFDC_H$	TCALL 1	(L)
$FFDD_H$		(H)
$FFDE_H$	TCALL 0 *	(L)
$FFDF_H$		(H)

$FFE0_H$	not used	(L)
$FFE1_H$		(H)
$FFE2_H$	SERIAL I/O	(L)
$FFE3_H$		(H)
$FFE4_H$	Basic Interval Timer	(L)
$FFE5_H$		(H)
$FFE6_H$	Watch Dog Timer	(L)
$FFE7_H$		(H)
$FFE8_H$	EXT. INT 3	(L)
$FFE9_H$		(H)
$FFEA_H$	Timer 3	(L)
$FFEB_H$		(H)
$FFEC_H$	Timer1	(L)
$FFED_H$		(H)
$FFEE_H$	V-Sync Interrupt	(L)
$FFEF_H$		(H)
$FFF0_H$	1mS Interrupt	(L)
$FFF1_H$		(H)
$FFF2_H$	Timer 2	(L)
$FFF3_H$		(H)
$FFF4_H$	Timer 0	(L)
$FFF5_H$		(H)
$FFF6_H$	EXT. INT 2	(L)
$FFF7_H$		(H)
$FFF8_H$	EXT. INT1	(L)
$FFF9_H$		(H)
$FFFA_H$	On Screen Display	(L)
$FFFB_H$		(H)
$FFFC_H$	not used	(L)
$FFFD_H$		(H)
$FFFE_H$	RESET	(L)
$FFFF_H$		(H)

* This vector area is used in both BRK and TCALL 0 instruction

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

3. Peripheral Function

4. Control Function

5. Support Tool

6. Appendix

3.1 PORT

There are 6-ports in this device.

You can use these ports an digital I/O or 2nd function I/O

3.1.1 R0 PORT

8-bit I/O port including direction register and port data register (**IOA** Type)

- Register Structure and Description

Register Name	Symbol	R/W	Address	Initial Value
R0 I/O Direction Register	R0DD	W	$00C1_H$	0000 0000
R0 PORT Data Register	R0	R/W	$00C0_H$	Not initialized

R0 PORT I/O DIRECTION REGISTER

R0DD

<$00C1_H$>

W 7	W 6	W 5	W 4	W 3	W 2	W 1	W 0
R0DD7	R0DD6	R0DD5	R0DD4	R0DD3	R0DD2	R0DD1	R0DD0

Initial Value when RESET

[0000 0000]

Assign the direction of R0 port
(R0DD0) is assigned to R00 port)
0 : Input
1 : Output

R0 PORT DATA REGISTER

R0

<$00C0_H$>

R/W 7	R/W 6	R/W 5	R/W 4	R/W 3	R/W 2	R/W 1	R/W 0
R07	R06	R05	R04	R03	R02	R01	R00

Initial Value when RESET

[Not initialized]

Port R0 output data

If output mode port is read, the read data is R0 register data. And if input mode port is read, the read data is R0 pin data.

3.1.2 R1 PORT

You can use the R17 port as input mode only, but others as input or output mode.

		Selection Mode		
	Pin Name	Port Selection	2nd Function	Type
R1 PORT 0	R10	R10 (I/O)	R10 (I/O)	**IOA**
R1 PORT 1	R11	R11 (I/O)	R11 (I/O)	**IOA**
R1 PORT 2	R12	R12 (I/O)	R12 (I/O)	**IOA**
R1 PORT 3	R13	R13 (I/O)	R13 (I/O)	**IOA**
R1 PORT 4	R14	R14 (I/O)	R14 (I/O)	**IOA**
R1 PORT 5	R15/ Cin1	R15 (I/O)	Cin1 (I)	**IOC**
R1 PORT 6	R16/ Cin2	R16 (I/O)	Cin2 (I)	**IOC**
R1 PORT 7	R17/ Cin0/ INT3	R17 (I)	Cin0/ INT3 (I)	**IC**

- Register Structure and Description

Register Name	Symbol	R/W	Address	Initial Value
R1 I/O Directin Register	R1DD	W	$00C3_H$	0000 0000
R1 Port Data Register	R1	R/W	$00C2_H$	Not initialized
A/D COMP. Input CH. Selection Register	CIS	w	$00D7_H$	---- --00
Port Function Selection Register	FUNC	W	$00CA_H$	---0 0000

R1 PORT I/O DIRECTION REGISTER

R1DD

<$00C3_H$>

-	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
-	R1DD6	R1DD5	R1DD4	R1DD3	R1DD2	R1DD1	R1DD0

Initial Value when Reset

[-000 0000]

Not used

Assign the direction of R1 port

(R1DD0) is assigned to R10 port)

0 : Input

1 : Output

R1 PORT DATA REGISTER

R1

<00C2$_H$>

	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
	-	R16	R15	R14	R13	R12	R11	R10

Initial Value when Reset

[Not initialized]

Input data when read

Port R1 output data

A/D COMP. INPUT CHANNEL SELECTION

CIS

<00D7$_H$>

	-	-	-	-	-	-	W	W
	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	**CIS1**	**CIS0**

Initial Value when Reset

[---- --**00**]

Analog input channel selection
00 : CIN0
01 : CIN1
10 : CIN2
11 : CIN3

CIS1	CIS0	Channel	PORT Selection			
			R15/ Cin1	R16/ Cin2	R17/ Cin0/ INT3	R35/ Sin/ Cin3
0	0	Channel 0 (Cin0)	R15	R16	Cin0/ INT3	R35/ Sin
0	1	Channel 1 (Cin1)	Cin1	R16	R17/ Cin0	R35/ Sin
1	0	Channel 2 (Cin2)	R15	Cin2	R17/ Cin0	R35/ Sin
1	1	Channel 3 (Cin3)	R15	R16	R17/ Cin0	Cin3

PORT FUNCTION SELECTION REGISTER

FUNC

<00CA$_H$>

	-	-	-	W	W	W	W	W
	7	6	5	4	3	2	1	0
	-	-	-	EC3S	EC2S	**INT3S**	INT2S	INT1S

Initial value when MCU Reset

[---0 **0**000]

R30 / INT1 Selection
0 : R30 (Input)
1 : INT1 (Input)

R27 / EC3 Selection
0 : R27 (I/O)
1 : EC3 (Input)

R31/ INT2 Selection
0 : R31 (Input)
1 : INT2 (Input)

R26 / EC2 Selection
0 : R26 (I/O)
1 : EC2 (Input)

R17 / INT3 Selection
0 : R17 (Input)
1 : INT3 (Input)

3.1.3 R2 PORT

8- BIT I/O Port

		Pin Name	Selection Mode: Port Selection	Selection Mode: 2nd Functin	Type
R2 PORT	0	R20	R20 (I/O)	R20 (I/O)	**IOA**
	1	R21	R21 (I/O)	R21 (I/O)	**IOA**
	2	R22	R22 (I/O)	R22 (I/O)	**IOA**
	3	R23	R23 (I/O)	R23 (I/O)	**IOA**
	4	R24	R24 (I/O)	R24 (I/O)	**IOA**
	5	R25/ T2048	R25 (I/O)	T2048 (O)	**IOD**
	6	R26/ EC2	R26 (I/O)	EC2 (I)	**IOB**
	7	R27/ EC3	R27 (I/O)	EC3 (I)	**IOB**

- Register Structure and Description

Register Name	Symbol	R/W	Address	Initial Value
R2 I/O Direction Register	R2DD	W	$00C5_H$	0000 0000
R2 Port Data Register	R2	R/W	$00C4_H$	Not initialized
Port Function Selection Register	FUNC	W	$00CA_H$	---0 0000
PWM Control Register 2	PWMCR2	R/W	$00E5_H$	---0 0000

R2 PORT I/O DIRECTION REGISTER

R2DD

<$00C5_H$>

W 7	W 6	W 5	W 4	W 3	W 2	W 1	W 0
R2DD7	R2DD6	R2DD5	R2DD4	R2DD3	R2DD2	R2DD1	R2DD0

Initial value when MCU Reset

[0000 0000]

Assign the direction of R2 port
R2DD is assigned to R20 port

0 : Input
1 : Output

R2 PORT DATA REGISTER

R2

<00C4$_H$>

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R27	R26	R25	R24	R23	R02	R21	R20

Initial value when MCU Reset

[Not initialized]

Port R2 output data

PORT FUNCTION SELECTION REGISTER

FUNC

<00CA$_H$>

-	-	-	W	W	W	W	W
7	6	5	4	3	2	1	0
-	-	-	**EC3S**	**EC2S**	INT3S	INT2S	INT1S

Initial value when MCU Reset)

[---**0** **0**000]

R30 / INT1 Selection
0 : R30 (Input)
1 : INT1 (Input)

R31/ INT2 Selection
0 : R31 (Input)
1 : INT2 (Input)

R17 / INT3 Selection
0 : R17 (Input)
1 : INT3 (Input)

R27 / EC3 Selection
0 : R27 (I/O)
1 : EC3 (Input)

R26 / EC2 Selection
0 : R26 (I/O)
1 : EC2 (Input)

PWM OUTPUT CONTROL REGISTER 2

PWMCR2

<00E5$_H$>

-	-	-	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
-	-	-	**T2048**	POL2	POL1	EN7	EN6

Initial value when MCU Reset

[---**0** 0000]

R37/ PWM6 Selection
0 : R37
1 : PWM6

R36/ PWM7 Selection
0 : R36
1 : PWM7

14-bit PWM Output Polarity
0 : Positive poarity
1 : Negative Polarity

7-bit PWM OUTPUT Polarity
0 : Positive Polarity
1 : Negative Polarity

R25/ T2048 Selection
0 : R25
1 :T2048 (Output rectangular wave(T=2048uS)

3.1.4 R3 PORT

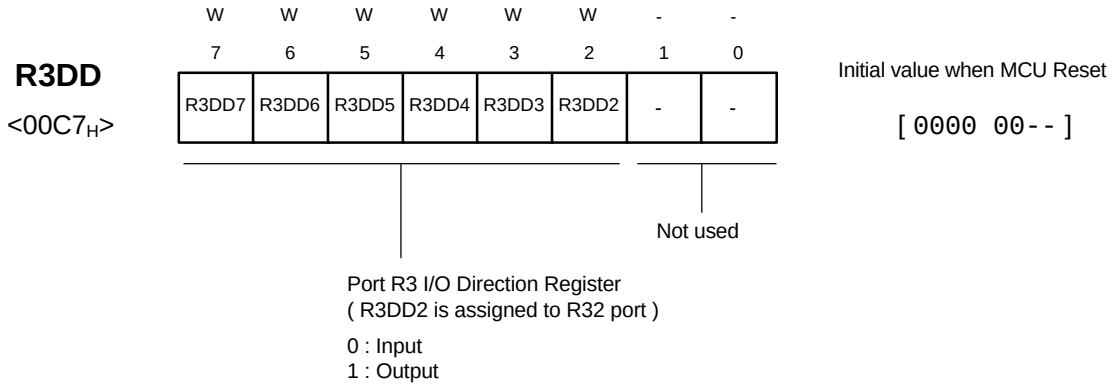
You can use lower 2-bits(R31, R30) of R3 port as input mode only. But others as input or output mode

			Selection Mode		
		Pin Name	Port Selection	2nd Function	Type
R3 PORT	0	R30/ INT1	R30 (I)	INT1 (I)	**IB**
	1	R31/ INT2	R31 (I)	INT2 (I)	**IB**
	2	R32/ PWM8	R32 (I/O)	PWM8 (O)	**IOD**
	3	R33/ Sout	R33 (I/O)	Sout (I/O)	**IOG**
	4	R34/ Sclk	R34 (I/O)	Sclk (I/O)	**IOG**
	5	R35/ Sin/ Cin3	R35 (I/O)	Sin/ Cin3 (I)	**IOE**
	6	R36/ PWM7	R36 (I/O)	PWM7 (O)	**IOF**
	7	R37/ PWM6	R37 (I/O)	PWM6 (O)	**IOF**

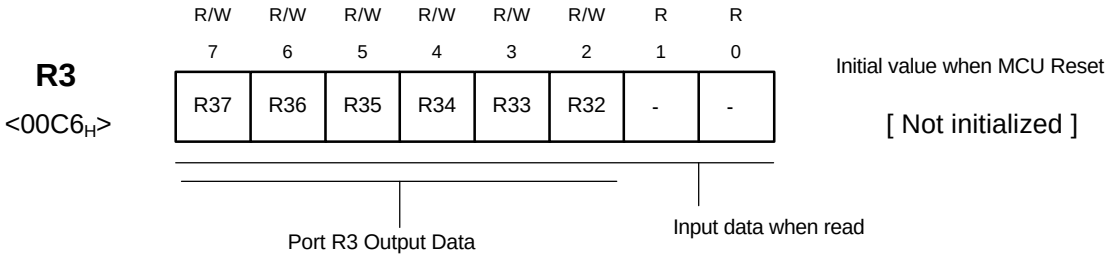
- Register Structure and Description

Register Name	Symbol	R/W	Address	Initial Value
R3 I/O Direction Register	R3DD	W	$00C7_H$	0000 0000
R3 Port Data Register	R3	R/W	$00C6_H$	Not Initialized
Port Function Selection Register	FUNC	W	$00CA_H$	---0 0000
Serial I/O Mode Register	SIOM	R/W	$00D8_H$	-000 0001
PWM Control Register 1	PWMCR1	R/W	$00E4_H$	0000 0000
PWM Control Regsiter 2	PWMCR2	R/W	$00E5_H$	---0 0000

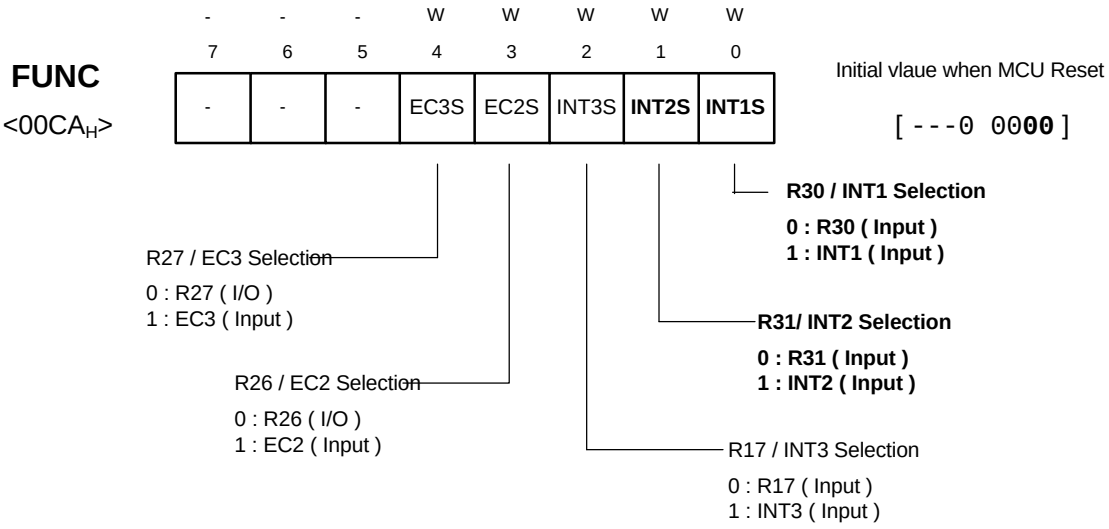
PORT R3 I/O DIRECTION REGISTER



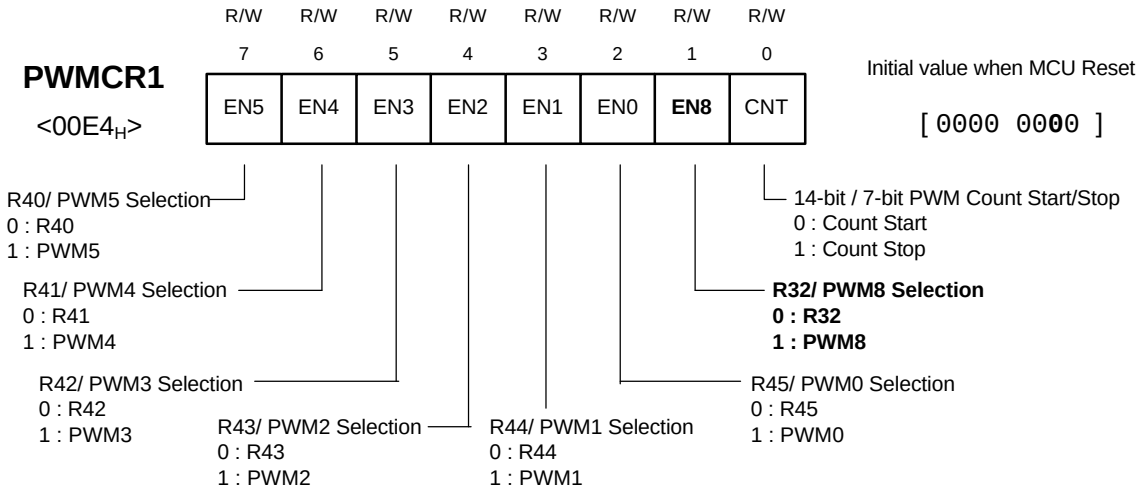
R3 PORT DATA REGISTER



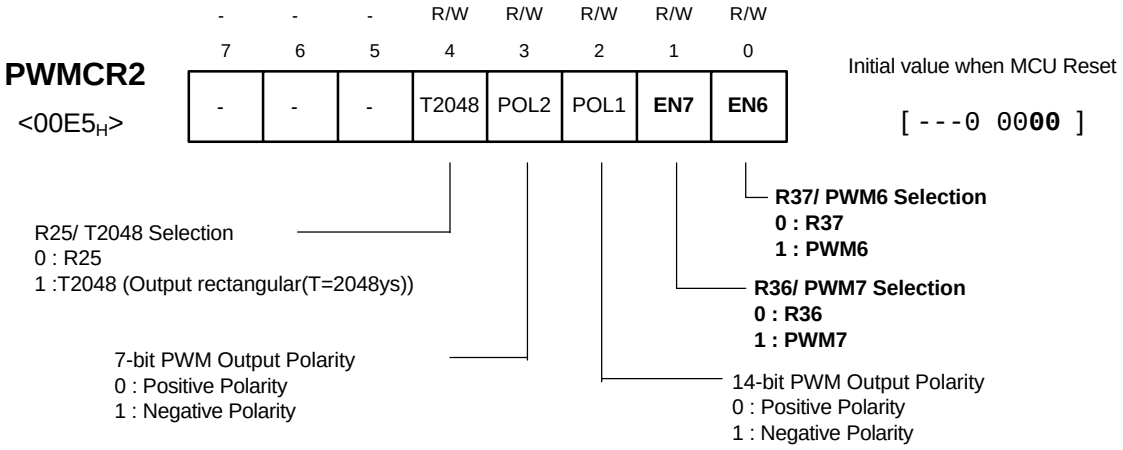
PORT FUNCTION SELECTION



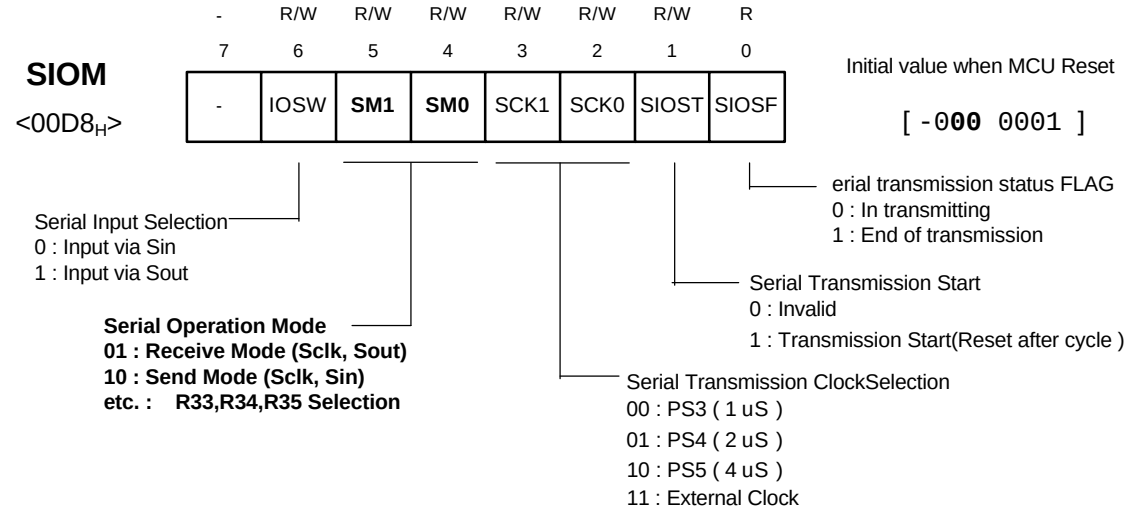
PWM CONTROL REGISTER 1



PWM CONTROL REGISTER 2



SERIAL I/O MODE REGISTER



SM1	SM0	Function Selection	Port Selection		
			R33/ Sout	R34/ Sclk	R35/ Sin/ Cin3 *
0	0	-	R33	R34	R35
0	1	Send Mode	Sout	Sclk	R35
1	0	Receive Mode	R33	Sclk	Sin
1	1	-	R33	R34	R35

R35 port will not operate, when Cin3 is operating as A/D input port.

3.1.5 R4 PORT

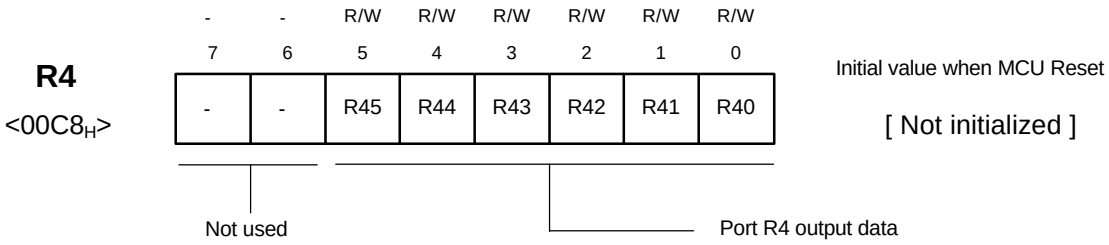
6-Bit output port.

		Selection Mode		
	Pin Name	Port Selection	Function Selection	Type
R4 PORT 0	R40/ PWM5	R40 (O)	PWM5 (O)	**OB**
R4 PORT 1	R41/ PWM4	R41 (O)	PWM4 (O)	**OB**
R4 PORT 2	R42/ PWM3	R42 (O)	PWM3 (O)	**OB**
R4 PORT 3	R43/ PWM2	R43 (O)	PWM2 (O)	**OB**
R4 PORT 4	R44/ PWM1	R44 (O)	PWM1 (O)	**OB**
R4 PORT 5	R45/ PWM0	R45 (O)	PWM0 (O)	**OB**

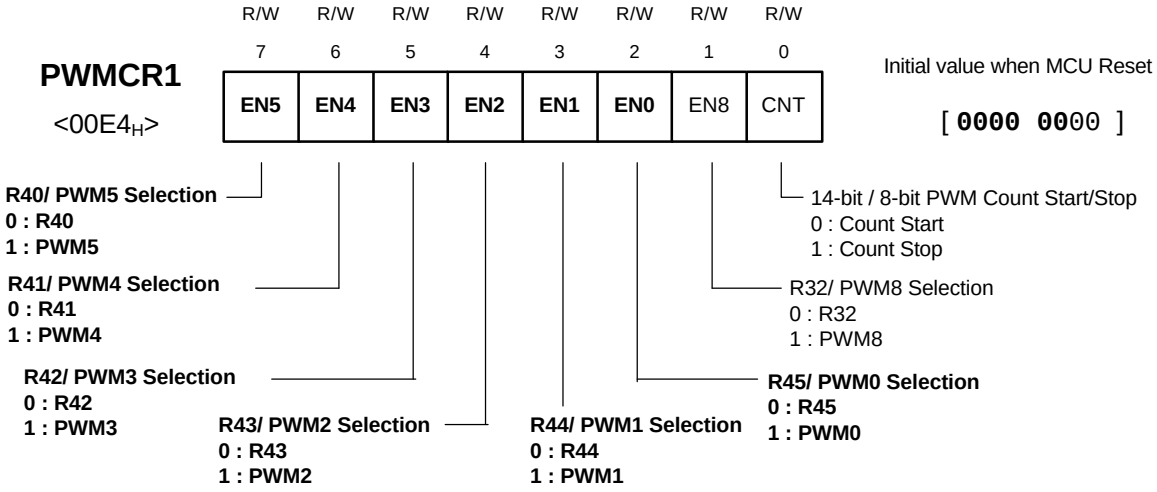
- Register Structure and Description

Register Name	Symbol	R/W	Address	Initial Value
R4 Port Data Register	R4	R/W	$00C8_H$	Not initialized
PWM Control Register	PWMCR1	R/W	$00E4_H$	0000 0000

R4 PORT DATA REGISTER



PWM CONTROL REGISTER 1



3.1.6 R5 PORT

4-Bit output only port.

		Selection Mode		
	Pin Name	Port Selection	2nd Function	Type
R5 PORT 0	R50/ R	R50 (O)	R (O)	**OA**
R5 PORT 1	R51/ G	R51 (O)	G (O)	**OA**
R5 PORT 2	R52/ B	R52 (O)	B (O)	**OA**
R5 PORT 3	R53/ Y	R53 (O)	Y (O)	**OA**

- Register Structure and Description

Register Name	Sumbol	R/W	Address	Initial Value
R5 Port Data Register	R5	R/W	$00C9_H$	Not initialized
OSD Output/ BACKGROUND Control Register	OSDCON1	W	$00F9_H$	0000 0000

R5 PORT DATA REGISTER

R5
<00C9$_H$>

-	-	-	-	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
-	-	-	-	R53	R52	R51	R50

Initial value when MCU Reset

[Not initialized]

Not used

Port R5 Output Data

OSD OUTPUT & BACKGROUND CONTROL

OSDCON1
<00F9$_H$>

W	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
OY	**OB**	**OG**	**OR**	OSDON	BB	BG	BR

Initial value when MCR Reset

[**0000** 0000]

R53/ Y Selection
0 : R53
1 : Y

R52/ B Selection
0 : R52
1 : B

R51/ G Selection
0 : R51
1 : G

R50/ R Selection
0 : R50
1 : R

OSD Output Control
0 : Disable
1 : Enable

Background and Edge Color Selecton

3.2 CLOCK GENERATION CIRCUIT

The clock generation circuit of GMS84512/84524 is consist of oscillation circuit for CPU clock, prescaler for peripheral clock and Basic Interval Timer Clock. Basic Interval Timer for reference time, and water Dog Timer for detecting S/W overrun.

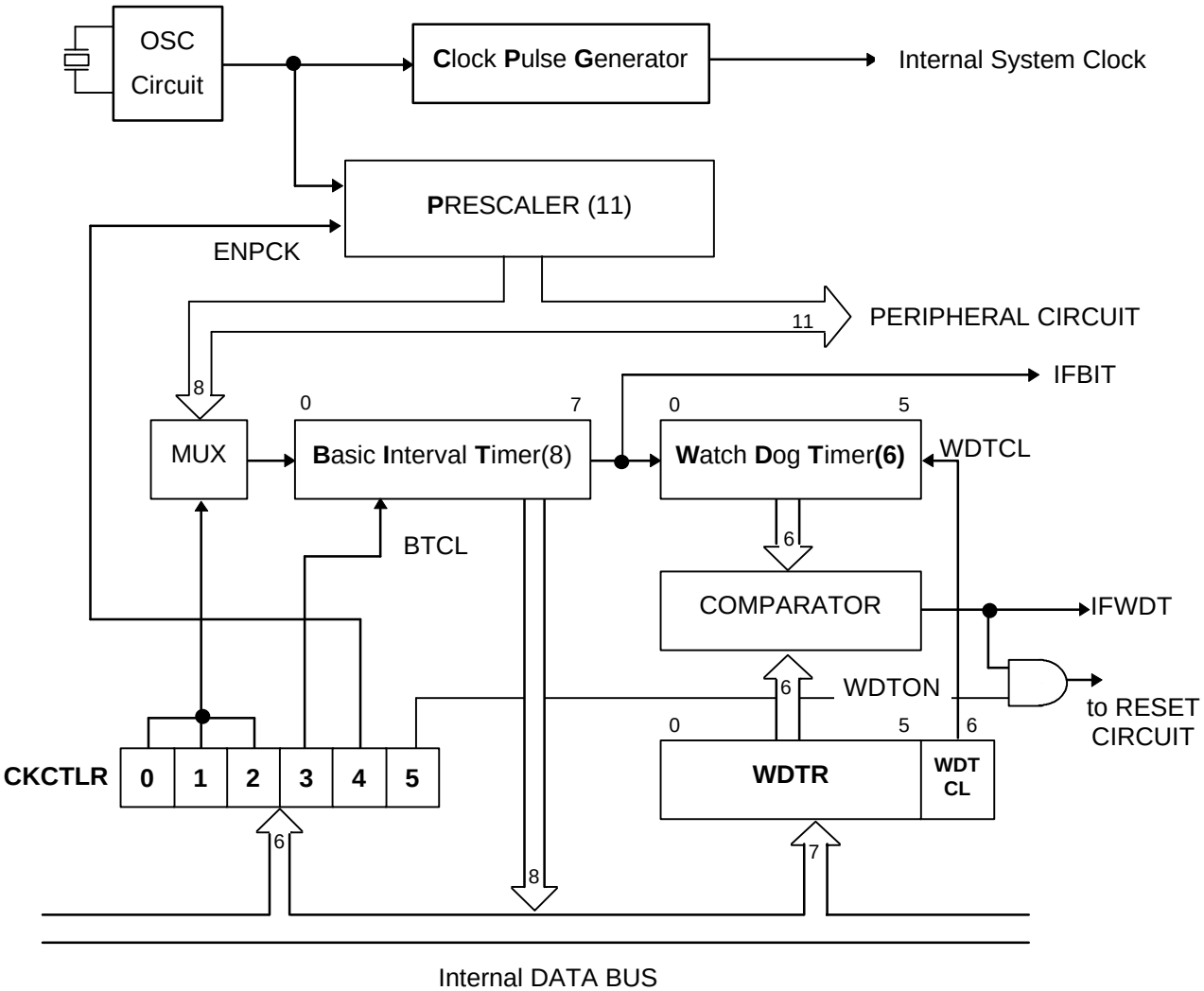


FIG.3.2.1 Clock Generation Circuit Block Diagram

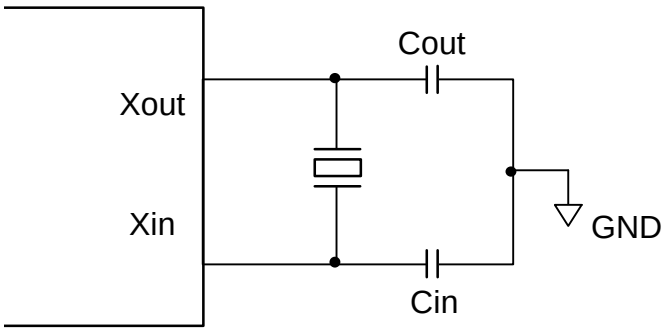
3.2.1 Oscillation Circuit

The clock signal incoming from crystal oscillator or ceramic resonator via Xin and Xout, or from external clock via Xin is supplied to Clock Pulse Generator and Prescaler

Internal System Clock for CPU is made by Clock Pulse Generator, and several peripheral clock is devided by prescaler

Clock Generation circuit of Crystal Oscillator or Ceramic Resonator is shown in Fig.3.2.2

¨ çClock Generation Circuit by Crystal Oscillator or Ceramic Resonator



¨ èClock generator circuit by external clock

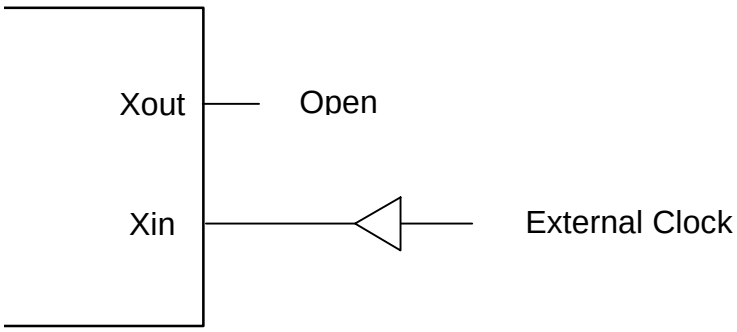


FIG. 3.2.2. Clock Generation Circuit

¡ ÚWhen **STOP** Mode, Oscillation Stops,
Xin pin is High-Impedance, and Xout pin is going to High level state.

3.2.2 PRESCALER

Prescaler is consisted of 11-bit binary counter, and input clock is supplied by oscillation circuit.

Frequency divided output from each bit of prescaler is used as peripheral clock.

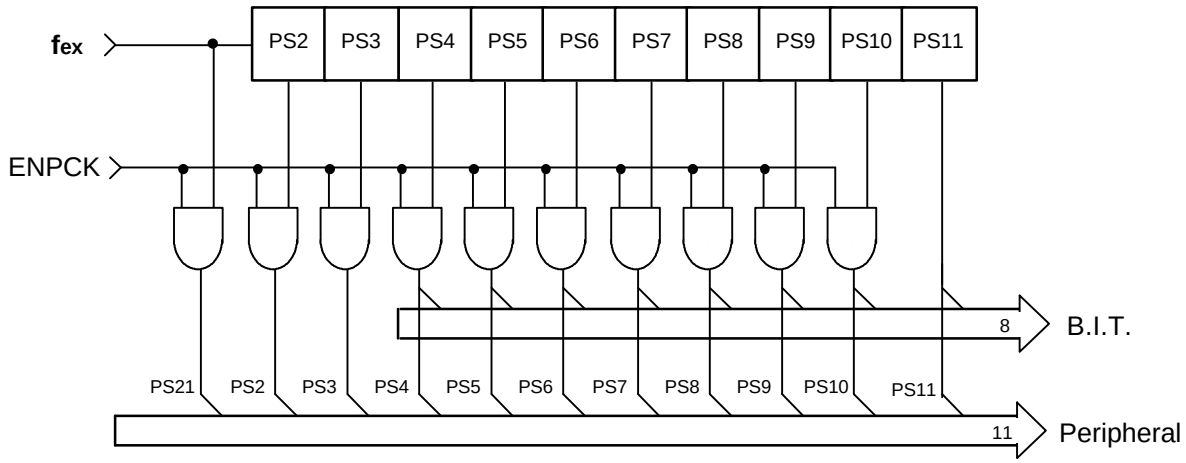
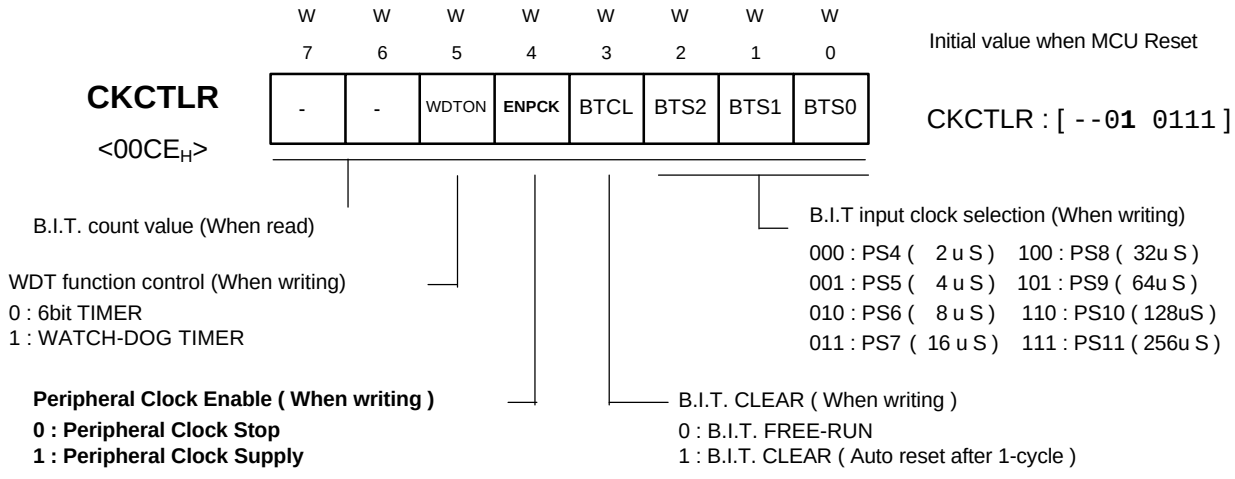


FIG. 3.2.3 Configuration of Prescaler

TABLE 3.2.1 Frequency-Divided Outputs of Prescaler

f_{EX} (§Ö)		PS1	PS2	PS3	PS4	PS5	PS6	PS7	PS8	PS9	PS10	PS11
4	Interval	4 §Ö	2 §Ö	1§Ö	500 § õ	250 § Õ	125 § õ	62.5 § õ	31.25 § õ	15.63 § õ	7.18 § õ	3.91 § Õ
	Period	250nS	500 n S	1 u S	2 u S	4 u S	8 u S	16 u S	32 u S	64 u S	128 u S	256 u S
6	Interval	6§Ö	3§Ö	1.5§Ö	750 §Õ	375 § õ	187.5 § õ	93.75 § õ	46.88 § õ	23.44 § õ	11.72 § õ	5.86 §
	Period	166.7 n S	333.3 n S	666.7 n S	1.3 u S	2.7 u S	5.3 u S	10.7 u S	21.3 u S	42.7 u S	85.3 u S	170.7 u S

CLOCK CONTROL REGISTER



Peripheral Hardware Clock control Function

Peripheral Clock supplied from prescaler can be stopped by ENPCK. Peripheral hardware clock control bit of CKCTLR Register.(However, PS11 cannot be stopped by ENPCK).

3.2.3 Basic Interval Timer

There is 8-bit binary counter is Basic Interval Timer. It operates as following function.

- Reference Time interval interrupt request as timer.
- **B.I.T.** can be read

  (Note; The writing at same address overwrites the CKCTLR.)
- Clock supply of **W**atch **D**og **T**imer.

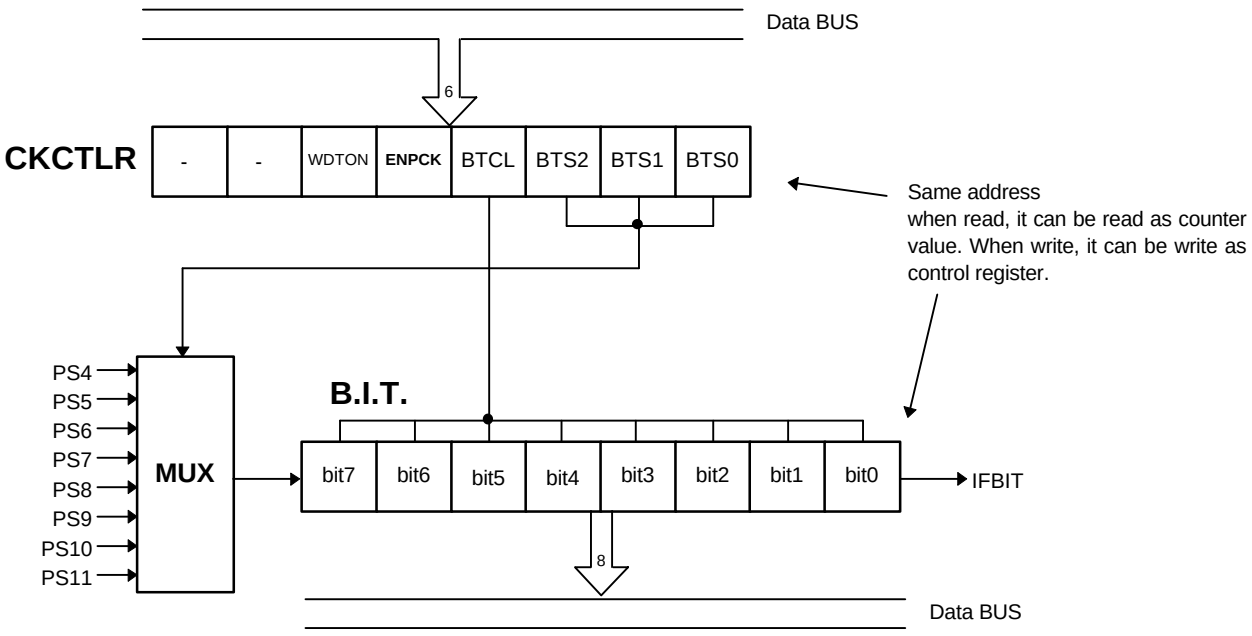
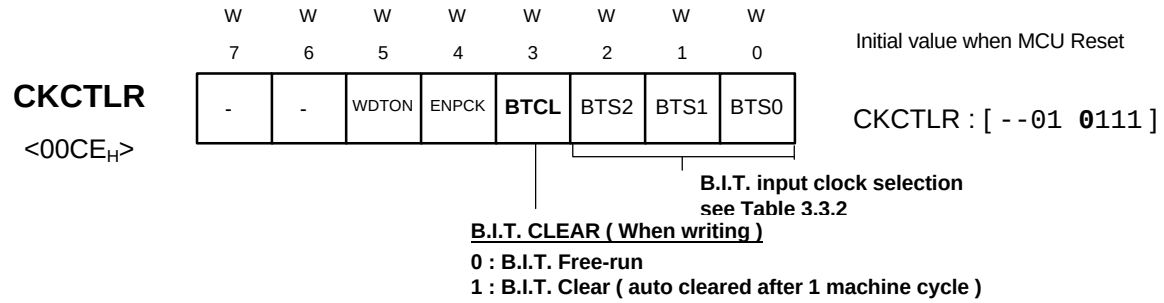


FIG. 3.2.4 Configuration of Basic Interval Timer

Control of Basic Interval Timer

Basic Interval Timer is Free Running Timer, but it can be cleared by setting **BTCL** (Bit 3 of clock control register). Initial state (after Reset) of BTCL is “0”, and if it is set to “1” it is auto-cleared after 1 machine cycle.

CLOCK CONTROL REGISTER



- Input clock selection of Basic Interval Timer and Reference Time interrupt interval

Input clock of Basic Interval Timer is selected by BTS2~BTS0(Bit2~0 of clock control register)among the prescaler outputs. Reference time interval interrupt is generated by BIT overflow.

CLOCK CONTROL REGISTER

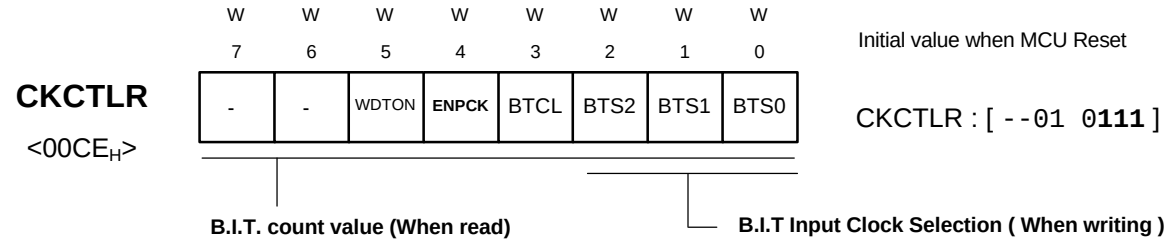


TABLE 3.2.2 Input clock selection of Basic Interval Timer and reference time interrupt interval (@**4**MHz)

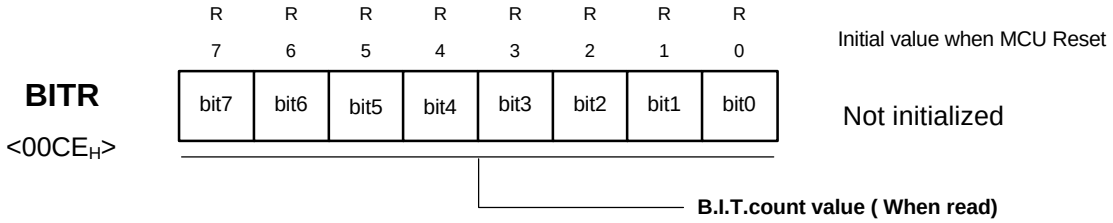
BTS2	BTS1	BTS0	B.I.T. Input Clock Period	Reference Time Interrupt Period
0	0	0	PS4 (2 u S)	512uS
0	0	1	PS5 (4 u S)	1,024uS
0	1	0	PS6 (8 u S)	2,048uS
0	1	1	PS7 (16 u S)	4,096uS
1	0	0	PS8 (32 u S)	8,192uS
1	0	1	PS9 (64 u S)	16,384uS
1	1	0	PS10 (128 u S)	32,768uS
1	1	1	PS11 (256 u S)	65,536uS

- Reading of Basic Interval Timer

  Basic Interval Timer Register can be read and interval up to 65ms can be measured

  (Note : The writing at same address overwrites the CKCTLR.)

BASIC INTERVAL TIMER REGISTER



3.2.4 Watch Dog Timer

Watch **D**og **T**imer is consist of 6-bit Binary Counter, 6-Bit Comparator and Watch Dog Timer Register(WDTR)

IFWDT is generated when counter value equals to WDTR, it can be used as S/W interrupt or MCU reset (Watch Dog Function) signal.

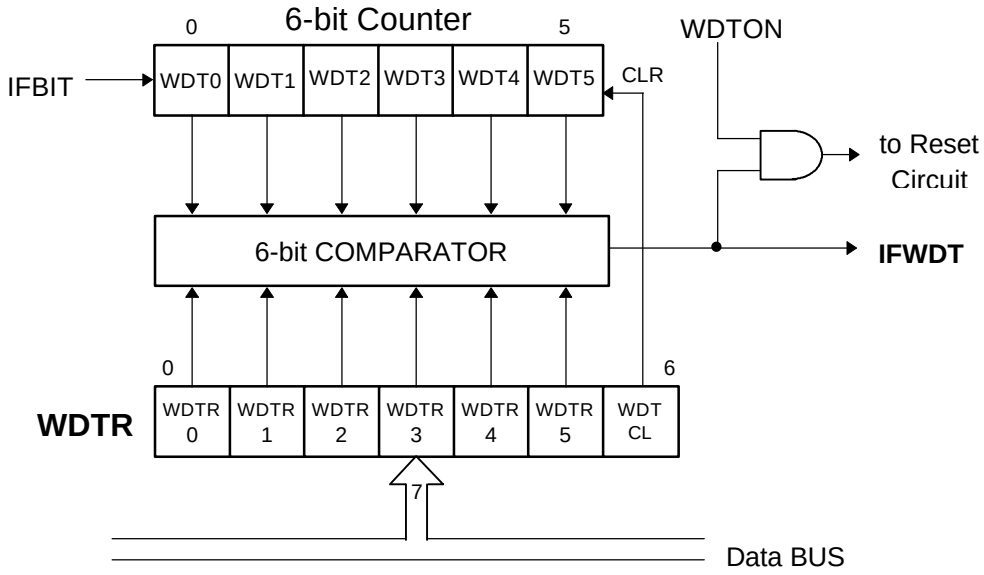


FIG. 3.2.4 Configuration Watch Dog Timer

- Control of WDT

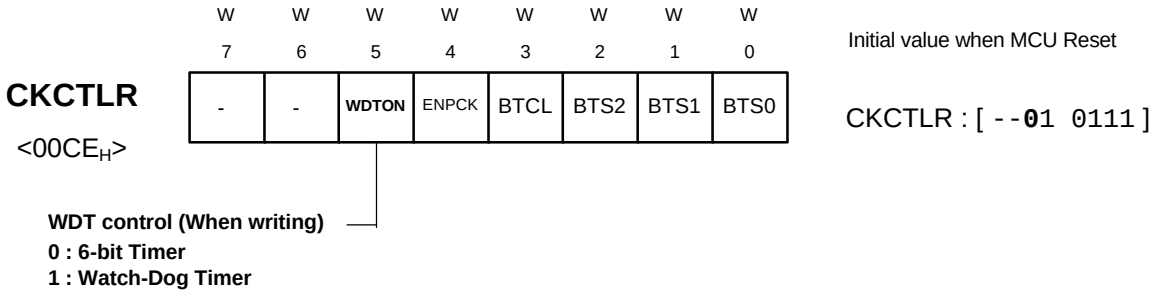
  WDTcan be used as 6-bit Timer or Watch Dog Timer according to WDTON (Bit 5 of CKCTLR).
  WDT is cleared by setting WDTCL (Bit 6 WDTR) to “1”.

  < Notice >

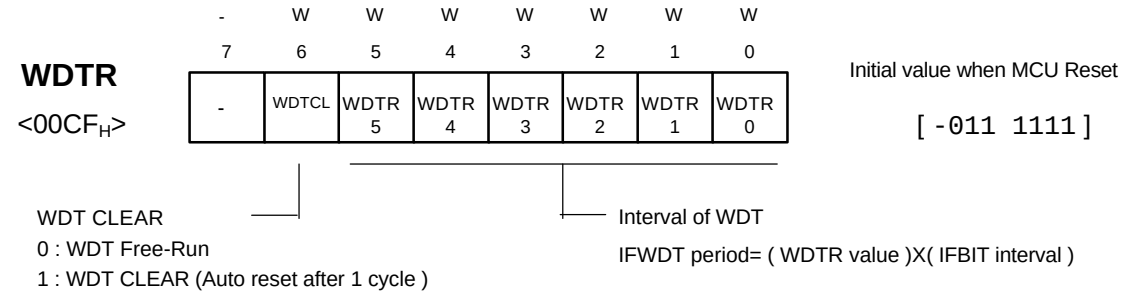
  1: After WDTON=1, maximum error of Timer is are one of period of IFBIT.

  2: Because 6-bit counter begin to count after MCU Reset
  the **Watch Dog Timer** should be enabled after clearing it.

CLOCK CONTROL REGISTER



WATCH-DOG TIMER REGISTER



- Interval of WDT Interrupt

  Interval of WDT Interrupt is decided by Basic Interval Timer Interrupt an WDTR
  That is, Interval of **= (WDTR value) X (IFBIT interval)**.

- Selection of WDT clock and maximum interval of WDT interrupt

  Input clock of WDT is IFBIT, so WDT interval is decided by BTS2~BTS1. Interval of WDT interrupt become maximum value.

  < Notice >

  Do not use WDTR=0 for MCU not to be Reset state always.

TABLE 3.2.2 Selection of WDT clock and maximum interval of WDT interrupt (@ 4MHz)

BTS2	BTS1	BTS0	B.I.T. Input Clock	WDT Input Clock	IFWDT max. interval
0	0	0	PS4 (2 u S)	512 uS	32,256 uS
0	0	1	PS5 (4 u S)	1,024 uS	64,512 uS
0	1	0	PS6 (8 u S)	2,048 uS	129,024 uS
0	1	1	PS7 (16 u S)	4,096 uS	258,048 uS
1	0	0	PS8 (32 u S)	8,192 uS	516,096 uS
1	0	1	PS9 (64 u S)	16,384 uS	1,032,192 uS
1	1	0	PS10 (128 u S)	32,768 uS	2,064,384 uS
1	1	1	PS11 (256 u S)	65,536 uS	4,128,768 uS

3.3 TIMER

Timer of GMS84512/84524 is 8-bit binary counter is consisted of Timer0(T0). Timer1(T1), Timer2(T2), Timer(T3), Timer Data Register(TDR0~TDR3). Timer Mode Register(TM0, TM2) and control circuit.

T0, T1 is each 8-bit interval Timer and can be used as a 16-bit intrval Timer.

T2, T3 is each 8-bit interval timer/event counter and can be used as a 16-bit interval timer/event counter

3.3.1 OPERATION MODE OF TIMER

- Operating mode of T0, T1

T0	T1
• 8-bit Interval Timer	• 8-bit Interval Timer
• 16-bit Interval Timer	

- Operating mode ofT2, T3

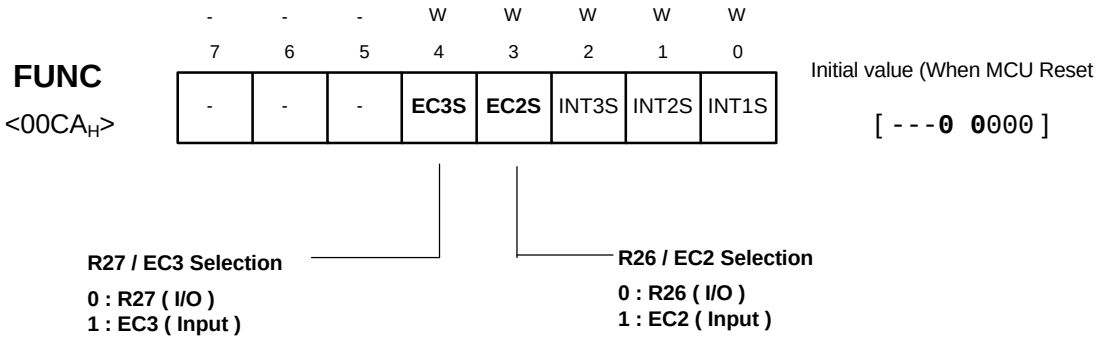
T2	T3
-. 8-bit Interval Timer -. 8-bit Event Counter	-. 8-bit Interval Timer -. 8-bit Event Counter
-. 16-bit Interval Timer -. 16-bit Event Counter	

When T2, T3 are used as event counter the relevant Port Mode Register Value should be assigned to select EC2 or EC3.

When T2, T3 are used as event counter, TDR value should be initialized to “FFH” because Timer count value is cleared if it equals to TDR value

Note) At the Reset Routine, TDR0 ~ TDR3 are should be initialized by software. (Except 00H)

PORT FUNCTION SELECTION REGISTER



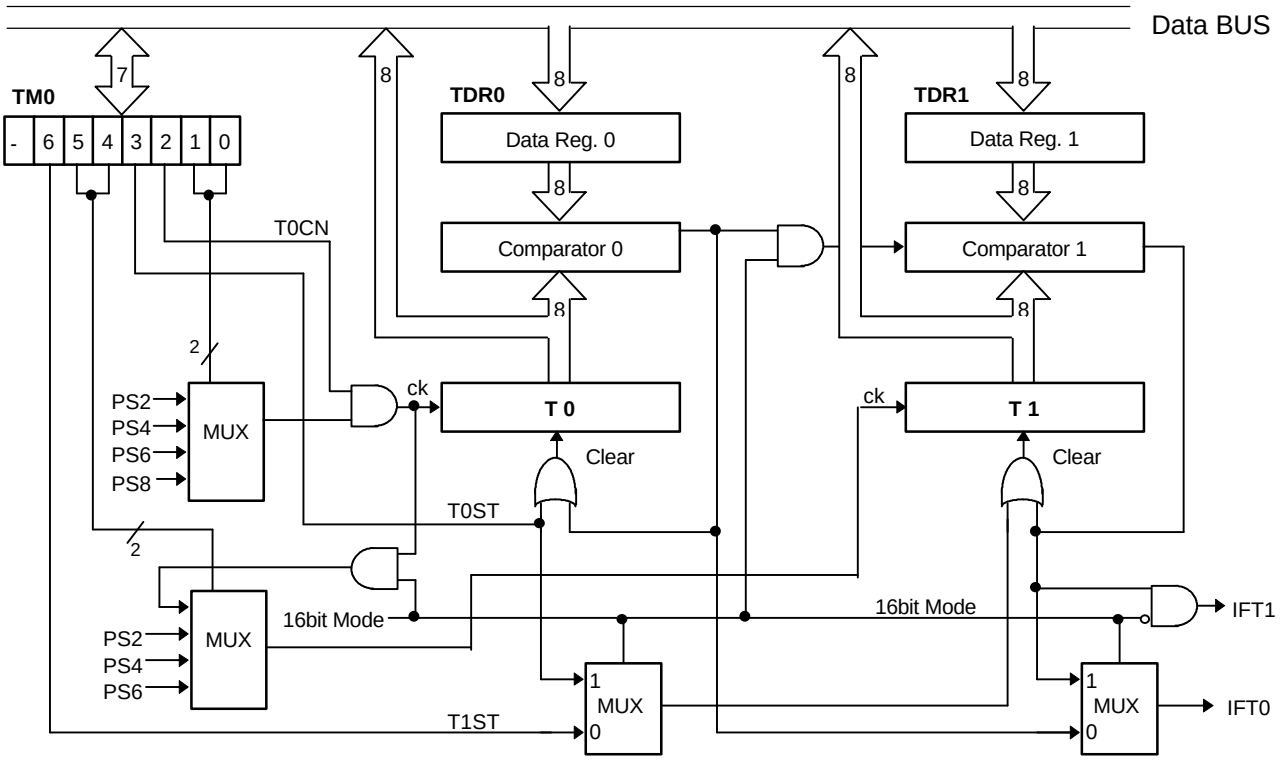


FIG. 3.3.1 Configuration TIMER0,TIMER1

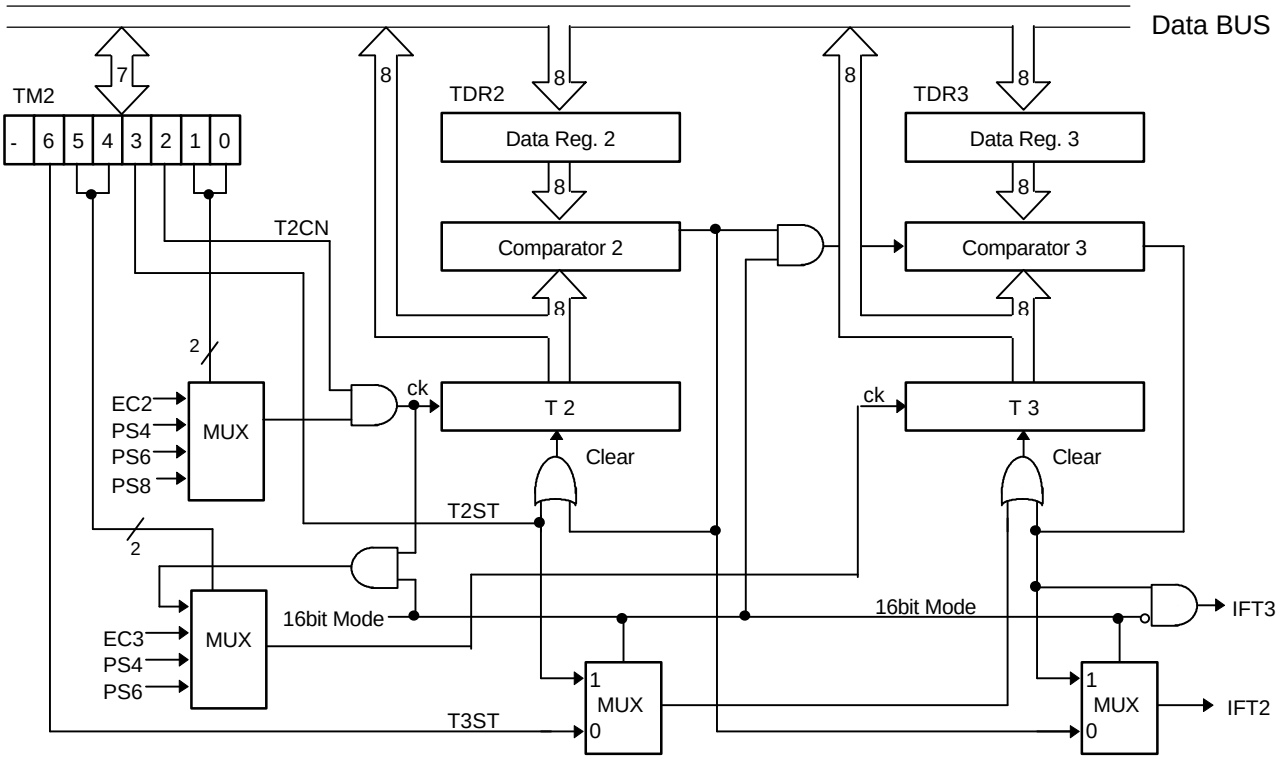


FIG. 3.3.2 Configuration of TIMER2,TIMER3

TIMER MODE REGISTER 0

TM0
<00D0$_H$>

-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
-	T1ST	T1SL1	T1SL0	T0ST	T0CN	T0SL1	T0SL0

Initial Value when MCU Reset

[-000 0000]

T1 Start/Stop control
0 : Cout Stop
1 : Counting start after clearing T1
* Don't care in 16bit Mode

T1 Input Clock Selection
00 : Connection to T0 (16bit Mode)
01 : PS2 (500 u S)
10 : PS4 (2 u S)
11 : PS6 (8 u S)

T0 Input Clock Selection
00 : PS2 (500 u S)
01 : PS4 (2 u S)
10 : PS6 (8 u S)
11 : PS8 (32 u S)

T0 Start/Stop control
0 : COUNT Stop
1 : COUNT Start

T0 Start/Stop control
0 : Count Stop
1 : Counting start after clearing T0

TIMER MODE REGISTER 2

TM2
<00D1$_H$>

-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
-	T3ST	T3SL1	T3SL0	T2ST	T2CN	T2SL1	T2SL0

Initial value when MCU Reset

[-000 0000]

T3 Start/Stop control
0 : Count Stop
1 : Counting start after clearing T3
* Don't care in 16bit Mode

T3 Input Clock Selection
00 : Connection to T2 (16bit Mode)
01 : Input external clock source(EC3)
10 : PS2 (500 u S)
11 : PS4 (2 u S)

T2 Input Clock Selection
00 : Input external clock source(EC2)
01 : PS2 (500 u S)
10 : PS4 (2 u S)
11 : PS6 (8 u S)

T2 Start/Stop Control
0 : Count Stop
1 : Count Start

T2 Start/Stop Control
0 : Count Stop
1 : Counting start after clearing T2

TIMER0 ~ TIMER3 DATA REGISTER

TDR0
<00D2$_H$>
TDR1
<00D3$_H$>
TDR2
<00D4$_H$>
TDR3
<00D5$_H$>

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TDR0 7	TDR0 6	TDR0 5	TDR0 4	TDR0 3	TDR0 2	TDR0 1	TDR0 0

Initial value when MCU Reset

[Not Initialized]

(READ)
T0 Count Value Read
T1 Count Value Read
T2 Count Value Read
T3 Count Value Read

(WRITE)
Modulo Data Write of T0
Modulo Data Write of T1
Modulo Data Write of T2
Modulo Data write T3

TABLE 3.3.1 Timer resolutin and maximum count at f_{IN}=4MHz

Timer	8-bit TIMER Mode		16-bit TIMER Mode	
	Resolution(ck)	max.count	Resolution(ck)	max. count
T0	PS2 (0.5 § Á)	128 § Á	PS2 (0.5 § Á)	32,768 §Á
	PS4 (2 § Á)	512 § Á	PS4 (2 § Á)	131,072 §Á
	PS6 (8 § Á)	2,048 § Á	PS6 (8 § Á)	524,288 §Á
	PS8 (32 § Á)	8,192 § Á	PS8 (32 § Á)	2,097,152 §Á
T2	PS2 (0.5 § Á)	128 § Á	PS2 (0.5 § Á)	32,768 §Á
	PS4 (2 § Á)	512 § Á	PS4 (2 § Á)	131,072 §Á
	PS6 (8 § Á)	2,048 § Á	PS6 (8 § Á)	524,288 §Á
T1	PS2 (0.5 § Á)	128 § Á	(Note)	
	PS4 (2 § Á)	512 §Á	Operation As Upper 8-Bit of T0	
	PS6 (8 § Á)	2,048 §Á		
T3	PS2 (0.5 § Á)	128 §Á	(Note)	
	PS4 (2 § Á)	512 §Á	Operation As Upper 8-Bit of T2	

3.3.2 Operation of TIMER0, TIMER1

T0 (T1) is consisted of 8-bit Binary Up-Counter. If T0 or T1 counter value become equal to Tdr0(or TDR1) value, it is cleared to 00H, and Interrupt request (IFT0 or IFT1) is generated.

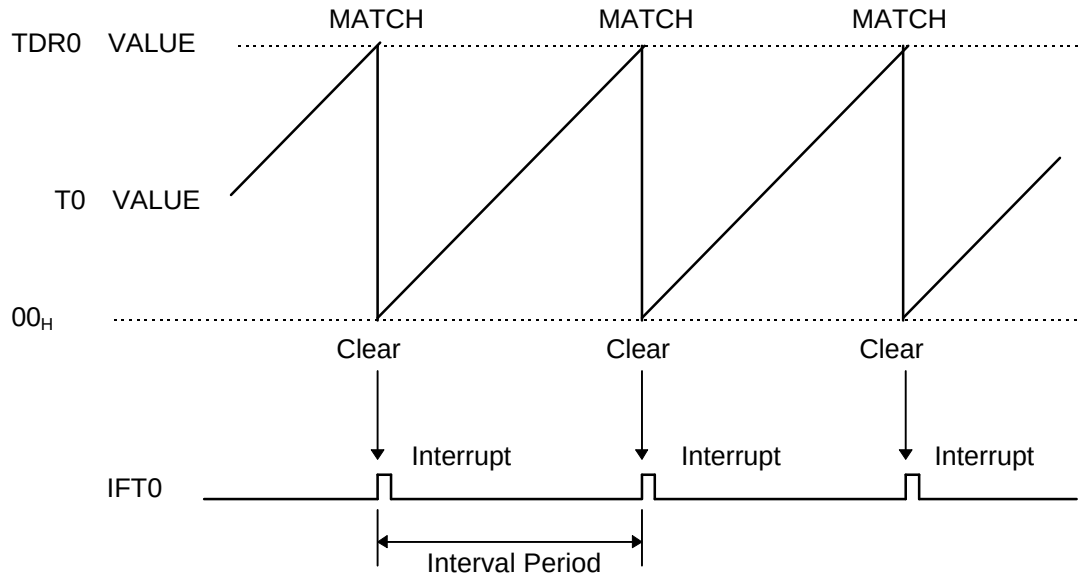


FIG 3.3.3 Operation of TIMER0 ,TIMER1

- Any of the PS2, PS4, PS6 or PS8 can be selected as the clock source of T0 by bit1(T0SLI) and bit0(T0SL0) of TM0. Andy of the PS2, PS4, PS6 or overflow of T0 can be selected as the clock source of T1 by bit5(T1SL1) and bit4(T1SL0) of TM0.

- The operation of T0, T1 is controlled by bit3(T0ST), bit2(T0CN) and bit6(T1ST) of TM0. T0CN controls count stop/start without clearing counter. T0ST and T1ST control count stop/start. In order to enable timer to count-up, T0CN, T0ST and T1St should become “1”. After clearing T0, T1 in order to count-up. T0st or T1ST should become “0” for a moment and return to “1”.

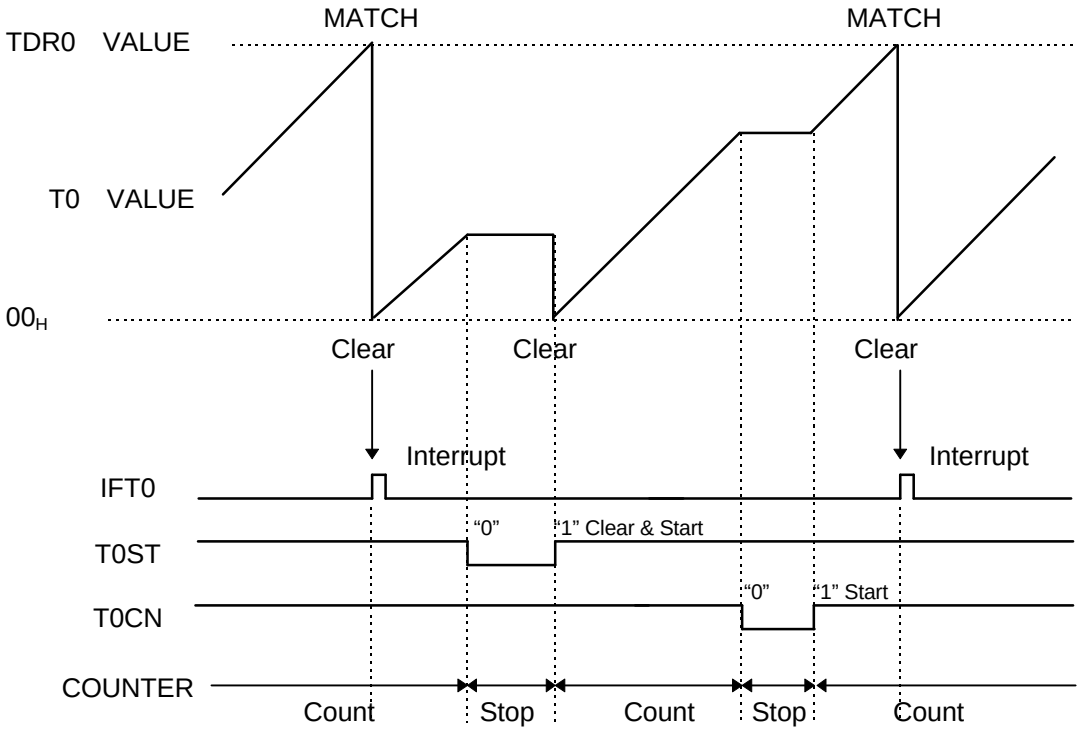


FIG 3.3.4 START/ STOP Control of Timer0

- The 16-bit interval timer is selected by assigning bit5(T1SL1) and bit4(T1SL0) to “0”.**)** At 16-bit timer mode, IFT0 interrupt only is valid. It is prefered to write to the TDR in non counting timer in order to protect undesirable interrupt.

3.3.2 Operation of TIMER2, TIMER3

- T2 (T3) is consisted of 8-bit Binary Up-Counter. If T2(T3) counter value become equal to TDR2(TDR3) value, it is cleared to 00H and interrupt request (IFT2 or IFT3) is generated.

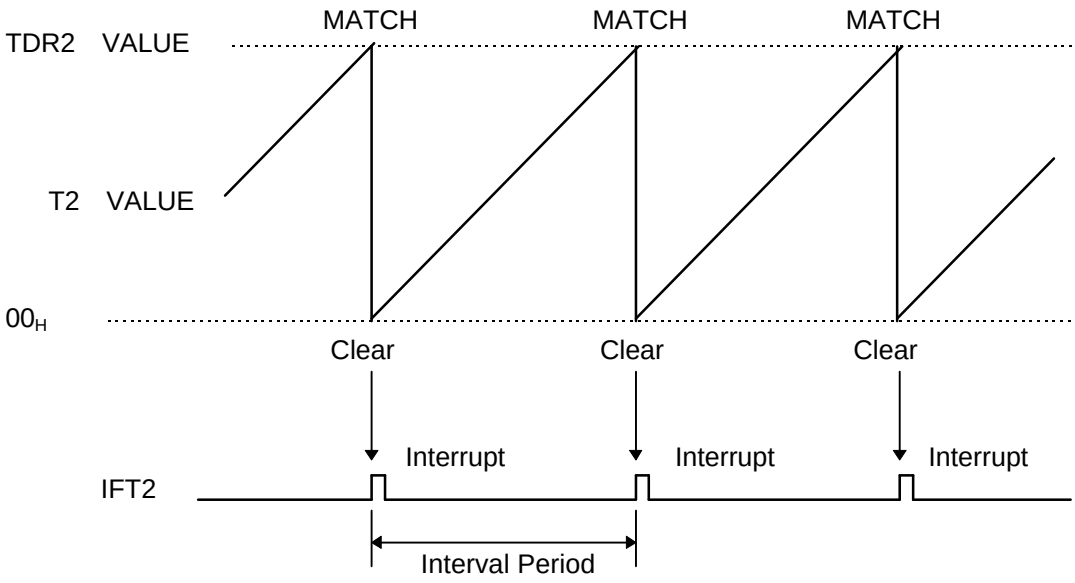


FIG 3.3.5 Operation of TIMER2 (or TIMER3)

- Any of the PS2, PS4, PS6 or external event input can be selected as the clock source of T2 by bit1(T2SL1) and bit0(T2SL0) of TM0. Any of the PS2, PS4 external event input or overflow of T2 can be selected as the clock source of T1 by bit5(T3SL1) and bit4(T3SL0) of TM0. If input clock is selected as external event input (EC2 or EC3), T2 and T3 operates as 8-bit event counter.

- The operation of T2, T2 is controlled by bit3(T2ST), bit2(T2CN) and bit6(T3ST) of TM2. controls count stop/start without clearing counter. T2ST and T3ST control count stop/start. order to enable timer to count-up T2CN, T2ST and T3ST should become “1”, After clearing T0,T1 in order to count-up. T2ST or T3ST should become “0” for a moment and return to “1”

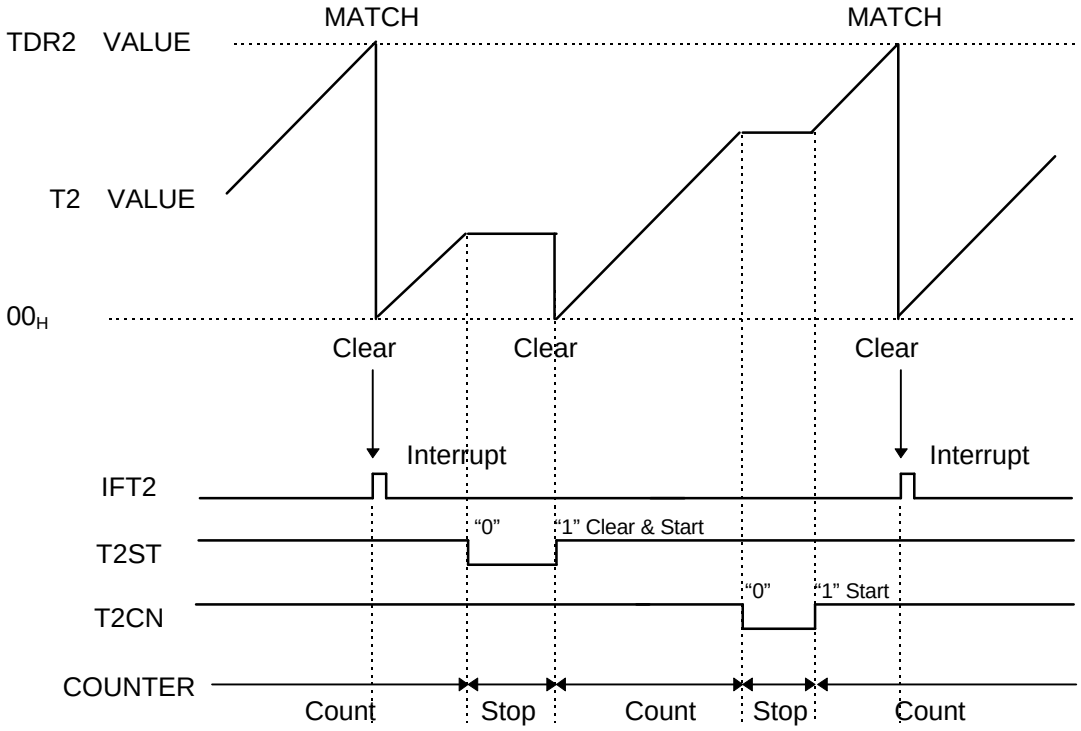


FIG 3.3.6 START/ STOP Control of Timer2

- The 16-bit interval timer is selected by assigning bit5(T3SL1) and bit4(T3SL0) to "0". At 16-bit timer mode, iIFT2 interrupt only is valid. It is prefered to write to the TDR in non-counting timer, in order to protect undesirable interrupt. If the input clock is selected among PS2, PS4 and PS6, T2 and T3 operate as 16-bit interval timer, while if EC2 operate as 16-bit event/counter.

< Notice >

1. On counting the reading value of TDR is counted value
2. 16-bit Mode, when data are read I the middle of Timer operation, the prior upper 8 bit data are read. Next the lower 8-bit data are read, and then the upper 8 bit data are read once again. If the earlier read upper 8-bit data are matched with the later read upper 8 bit data, 16-bit data are read correctly. 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**

3.4 A/D COMPARATOR

A/D comparator has an 5-bit resolution, and input is possible up to 4 channel. A/D comparator is consisted of analog input multiplexer, 5-bit D/A conversion circuit, sample & holder and control circuit

FIG.3.4.1 is a block diagram of A/D comparator

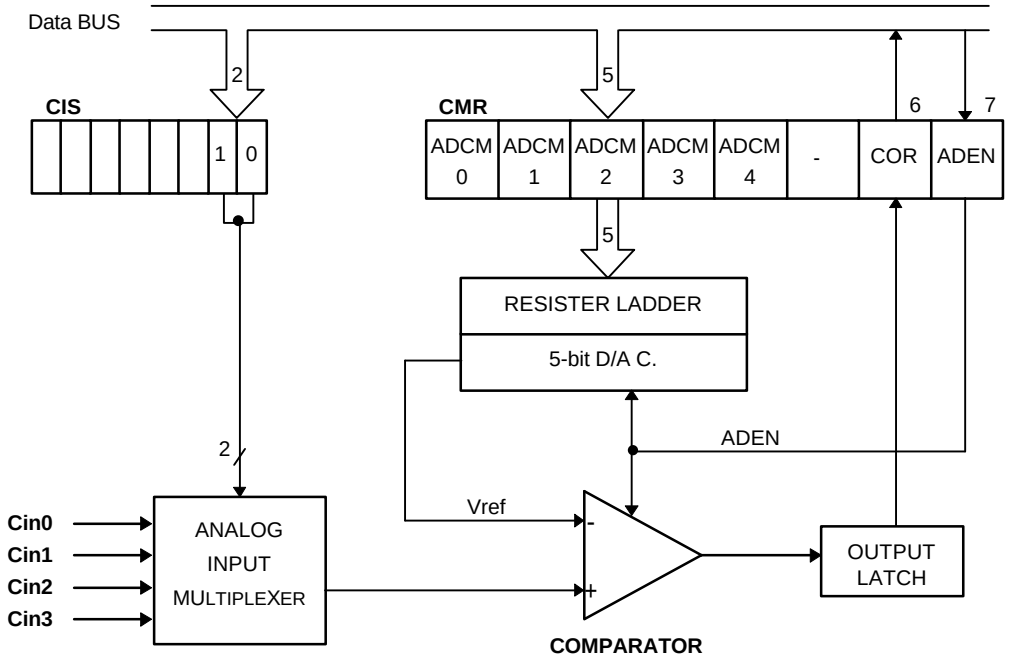


FIG. 3.4.1 Block Diagram of A/D Comparator

3.4.1 A/D COMPARATOR

Following produce is used.

- Write CIS register to select analog input channel.
- After writing CMR(ADCM0~4) to select reference voltage, set ADEN(bit7 of CMR) to “1” to start A/D comparision..

  < Notice > CMRcan’t be used with Bit Manipulation instruction and setting the reference voltage and starting A/D comparision can be used at same time..
- A/D Comparision processing needs 16machine cycle(8us)
- The result of comparision is stored in COR(bit6 of CMR).

  That is, if <input voltage > reference voltage>, COR=1

  if <input voltage < reference voltage>, COR=0

A/D COMP. INPUT CHANNEL SELECTION REGISTER

CIS

<00D7$_H$>

7	6	5	4	3	2	1	0
-	-	-	-	-	-	W	W
-	-	-	-	-	-	**CIS1**	**CIS0**

Initial value when MCU Reset

[---- --**00**]

Analog input channel selection
00 : CIN0
01 : CIN1
10 : CIN2
11 : CIN3

CIS1	CIS0	Function Selection	Port Selection			
			R15/ Cin1	R16/ Cin2	R17/ Cin0/ INT3	R35/ Sin/ Cin3
0	0	Channel 0 (Cin0)	R15	R16	Cin0/ INT3	R35/ Sin
0	1	Channel 1 (Cin1)	Cin1	R16	R17/ Cin0	R35/ Sin
1	0	Channel 2 (Cin2)	R15	Cin2	R17/ Cin0	R35/ Sin
1	1	Channel 3 (Cin3)	R15	R16	R17/ Cin0	Cin3

A/D COMPARATOR MODE REGISTER

CMR

<00D6$_H$>

7	6	5	4	3	2	1	0
W	R	-	W	W	W	W	W
ADEN	COR	-	ADCM 4	ADCM 3	ADCM 2	ADCM 1	ADCM 0

Initial value when MCU Reset

[00-0 0000]

A/D Comparision Control
0 : A/D Comparision Stop
1 ; A/D Comparision Start

A/D Comparison Result
0 : Input Voltage <Reference Voltage
1 ; Input Voltage >Reference Voltage

Reference Voltage Selection

00000 : Vdd/ 64	10000 : 33Vdd/ 64
00001 : 3Vdd/ 64	10001 : 35Vdd/ 64
00010 : 5Vdd/ 64	10010 : 37Vdd/ 64
|	|
01111 : 31Vdd/ 64	11111 : 63Vdd/ 64

- **The Calculation of Reference Voltage**

Reference Voltage (Vref) = { 2 X (Value of ADCM) + 1 } X Vdd / 64

3.5 Serial I/O

The serial I/O is 8-bit clock sychronous type and is consisted of serial I/O register, serial I/O mode register, clock selection circuit octal counter and control circuit. The Sout pin is degined to Input and Output. So serial I/O interface can be operated with minimum two pin.

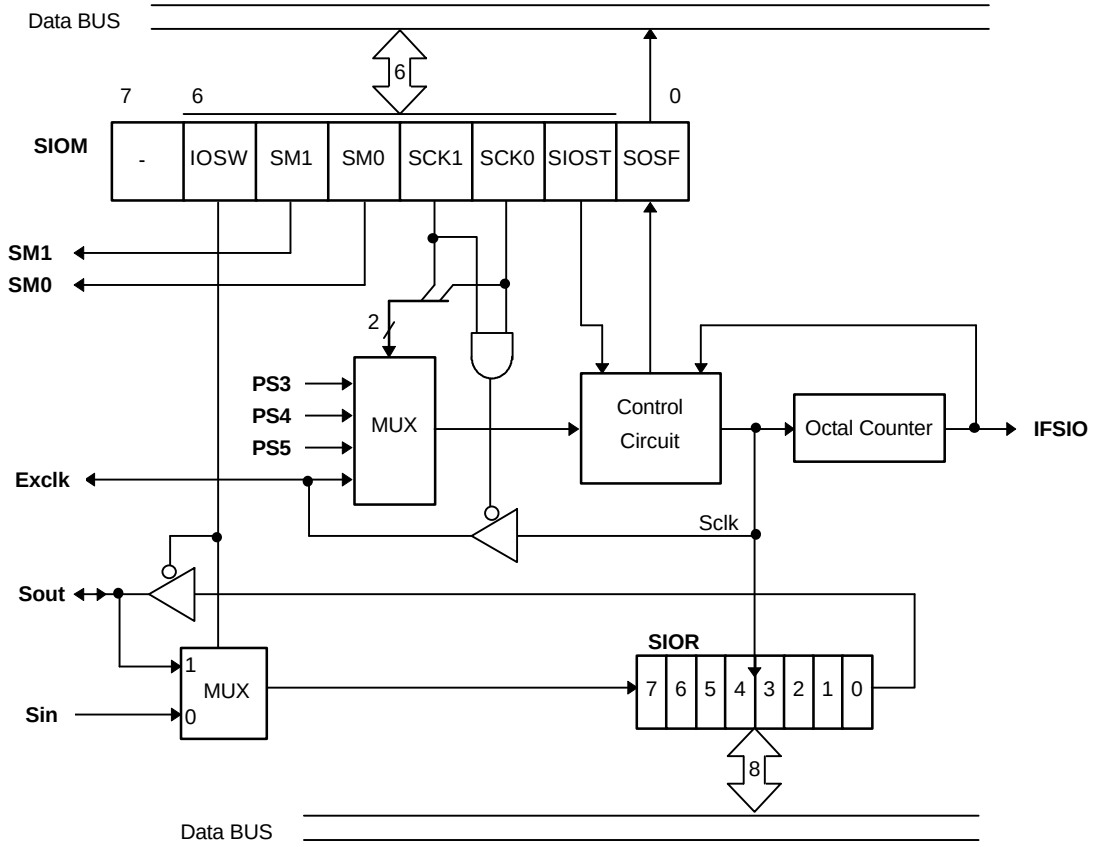
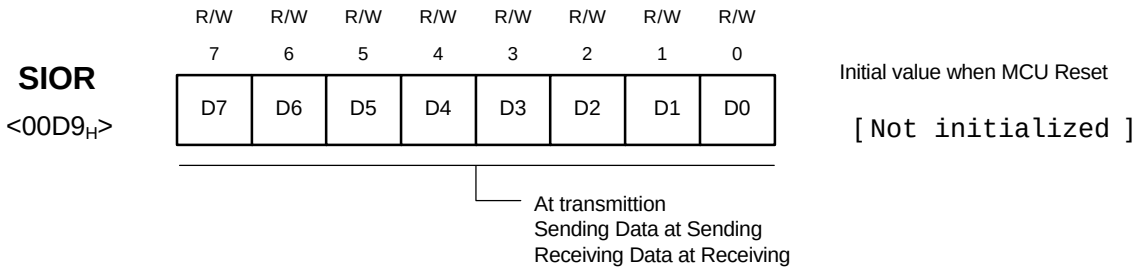


FIG. 3.5.1 Block Diagram of Serial I/O

3.5.1 Serial I/O Data Register

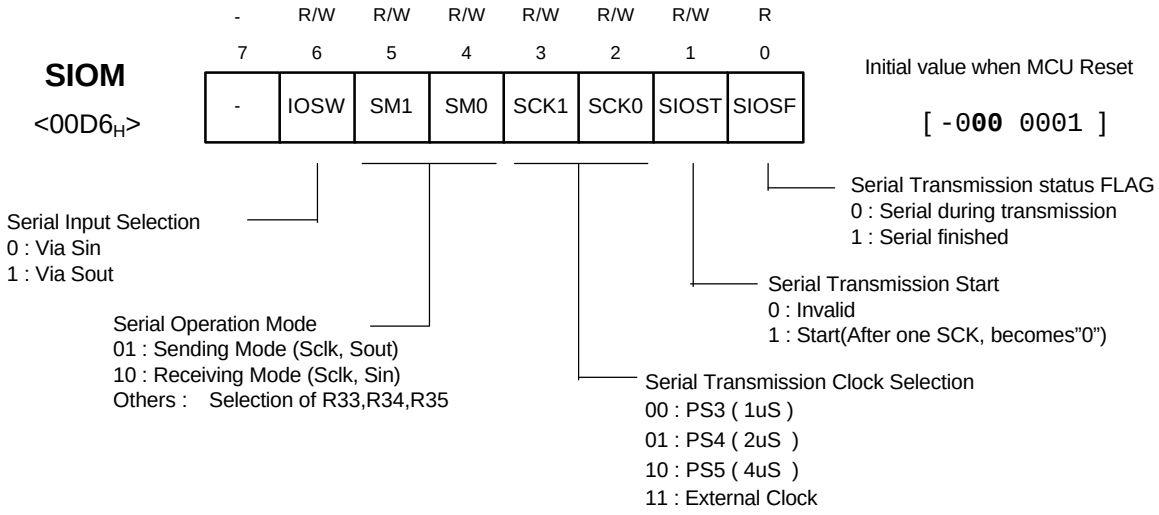
Serial I/O Data Register **SIOR**is a 8-bit Shift Register. First LSB is send or is received.



3.5.2 Serial I/O Mode Register

This register controls serial function. According to SCK1, SCK0 internal clock or external clock can be used.

SERIAL I/O MODE REGISTER



- **Port Selection According to Serial I/O Mode**

SM1	SM0	Function Selection	Port Selection		
			R33/ Sout	R34/ Sclk	R35/ Sin/ Cin3 *
0	0	-	R33	R34	R35
0	1	Sending Mode	Sout	Sclk	R35
1	0	Receiving Mode	R33	Sclk	Sin
1	1	-	R33	R34	R35

* If Cin3 is used as A/D comparator input channel, R35 port do not operate as output..

- **Selection of Serial input pin with the IOSW**

When receiving mode, serial input pin is selected by IOSW. That, if IOSW=0, R35/Sin is selected. If IOSW=1, R33/Sout

3.5.3 Data Transmission/Receving Timing

Serial transmission is started by setting SIOST(bit1 SIOM) to :"1". After one cycle of SCK, SIOST is cleared automatically to "0". serial output data from 8-bit shift register is output at folloing edge of Sclk. and input data is latched at rising edge of Sclek. When transmission Clock is counted 8times, serial I/O counter is cleared as "0". Transmission clock is halted in "H" state and serial I/O interrupt (IFSIO) occursg.

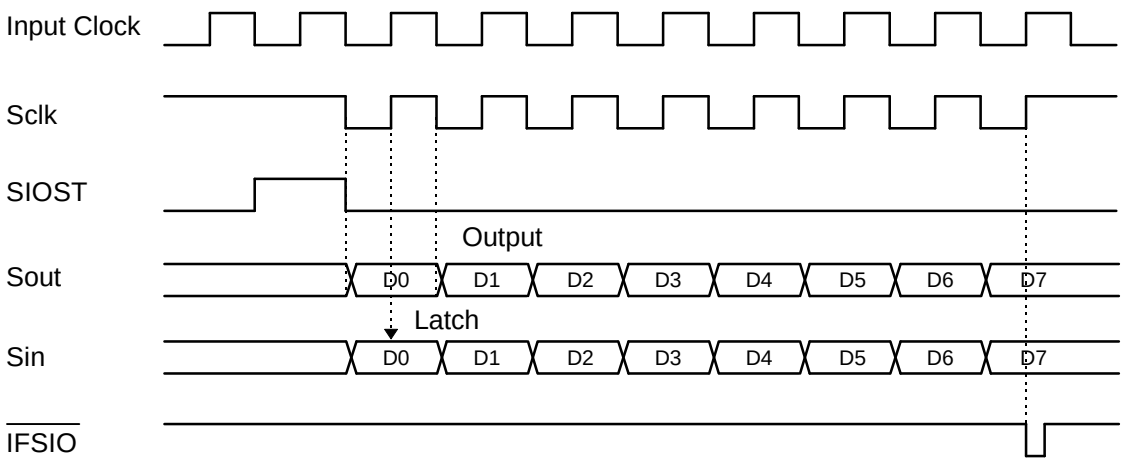


FIG. 3.5.1 Timing Diagram of Serial I/O

3.5.4 Data Transmission/Receiving Method

- Select transmission/receiving mode

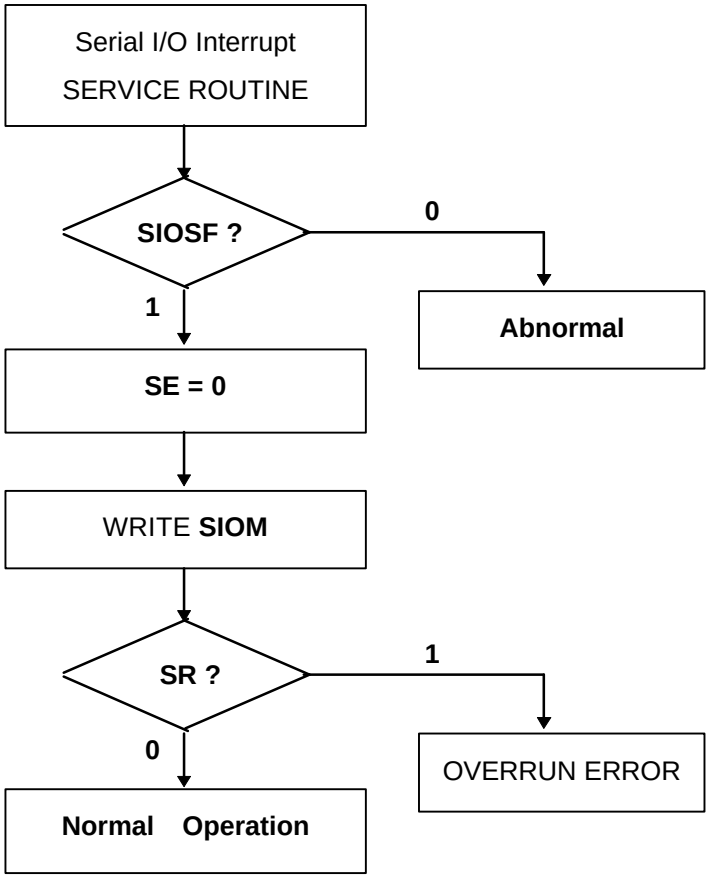
  <Notice>

  When external clock is used, the frequency should be less than 1MH and recommanded duty is 50%.
- When sending Data to be send is written at SIOR.
- Set SIOST to "1" to start serial transmission.

  <Notice >

  If both transmission mode selection and starting transmission is performed simultaneouslyit makes error.
- IFSIO is generated at completion and SIOSF is set to "1". In SIO interrupt service routine correct transmission should be tested.
- When receiving, receiving data is acquired by reading the SIOR.

3.5.5 The Method to Test Correct Transmission with S/W



Note) **SE:** Interrupt Enable Regist Low **IENL (Bit3)**

SR : Interrupt Request Flag Regist Low **IRQL (Bit3)**

FIG. 3.5.4 Serial Method to Teset Transmission.

3.6 Pulse Width Modulation (PWM)

The GMS84512/84524 is equipped with one 14-bit PWM(PWM8) and eight 7-bit PWM(PWM0~PWM7).

The 14-bit resolution gives PWM8 the minium resolution bit width of 500ns(PS2=500ns, if Xin=4MHz) and repeat period of 8,192uS. Each PWM0~PWM7 has a 7-bit resolution with min. resolution bit width of 8uS (PS6) and repeat period of 1,024uS.

- PWM Specification Table (@ Xin =4MHz)

Specification	14-bit PWM	7-bit PWM
Resolution	14 bits	7 bits
Input Clock	0.5uS	8uS
1 Frame Cycle	8,192uS	1,024uS

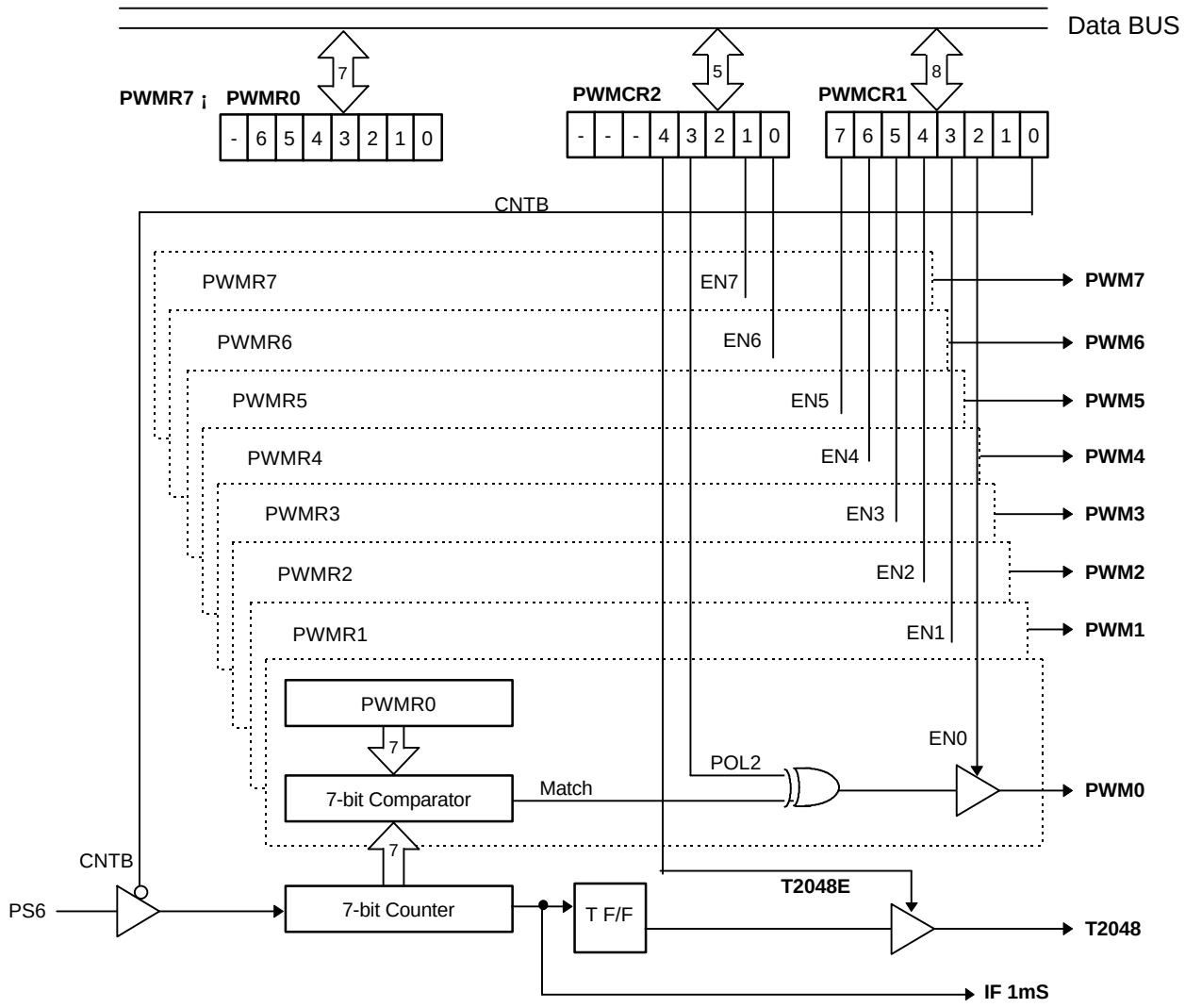


FIG. 3.6.1 Block Diagram of 7-bit PWM & T2048

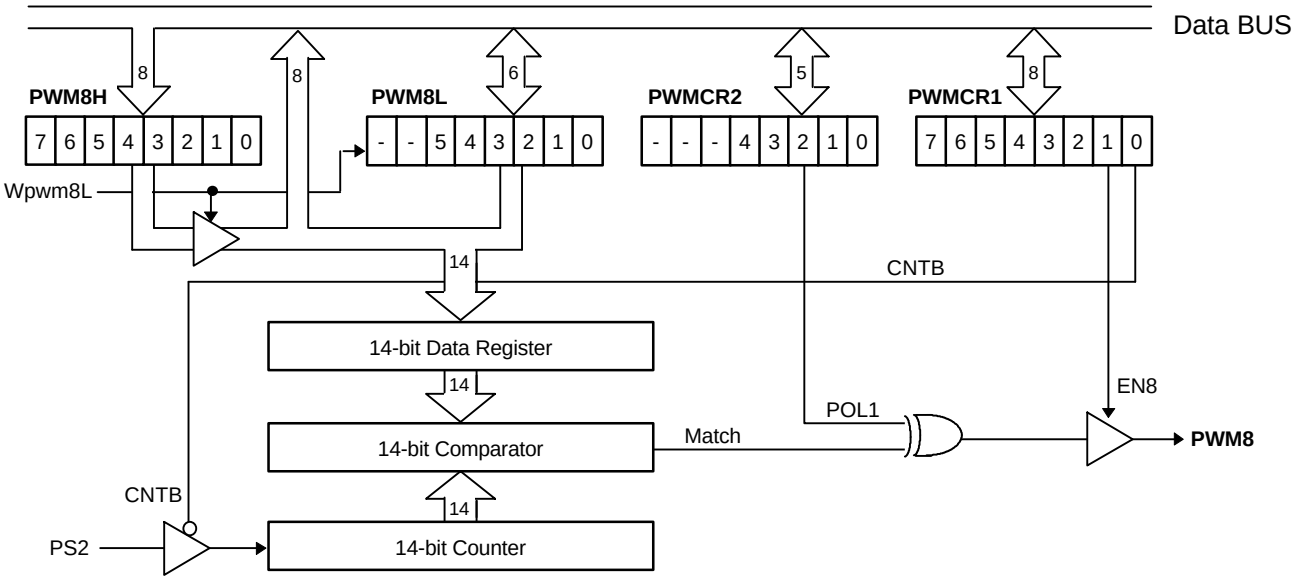


FIG. 3.6.2 Block Diagram of 14-bit PWM

3.6.1 PWM8 (14-bit PWM)

When the PWM8 is used for output, first set the higher 8-bit of the PWM8H register, then the lower 6-bit of the PWM6 register. The 14-bit data of PWM8 can be compare with the 14-bit comparator after lower 6-bit data of PWM8 is transferred**.**

Data setting for PWM8(14-bit PWM)

- The PWM output pulse period is consist with 64(=2^6) high level area which is consist with 256(=2^8)low-level area. First, the basic pulse is made by the data of PWM8H, then the position of pulse as long as low level width(0.5uS) among 64 short pulse is determined by contents of PWM8L.

TABLE 3.6.1

BIT	Sub-Frame(short pulse) position as long as low level area	Pulse
Bit0 = "1"	S32,	1
Bit1 = "1"	S16, S48	2
Bit2 = "1"	S8 , S24, S40, S56	4
Bit3 = "1"	S4 , S12, S20, S28, S36, S44, S52, S60	8
Bit4 = "1"	S2 , S6 , S10, S14, S18, S22, S26, S30, S34, S38, S42, S46, S50, S54, S58, S62	16
Bit5 = "1"	S1 , S3 , S5 , S7, S9, S11, S13, S15, S17, S19, S21, S23, S25, S27, S29, S31, S33, S35, S37, S39, S41, S43, S45, S47, S49, S51, S53, S55, S57, S59, S61, S63	32

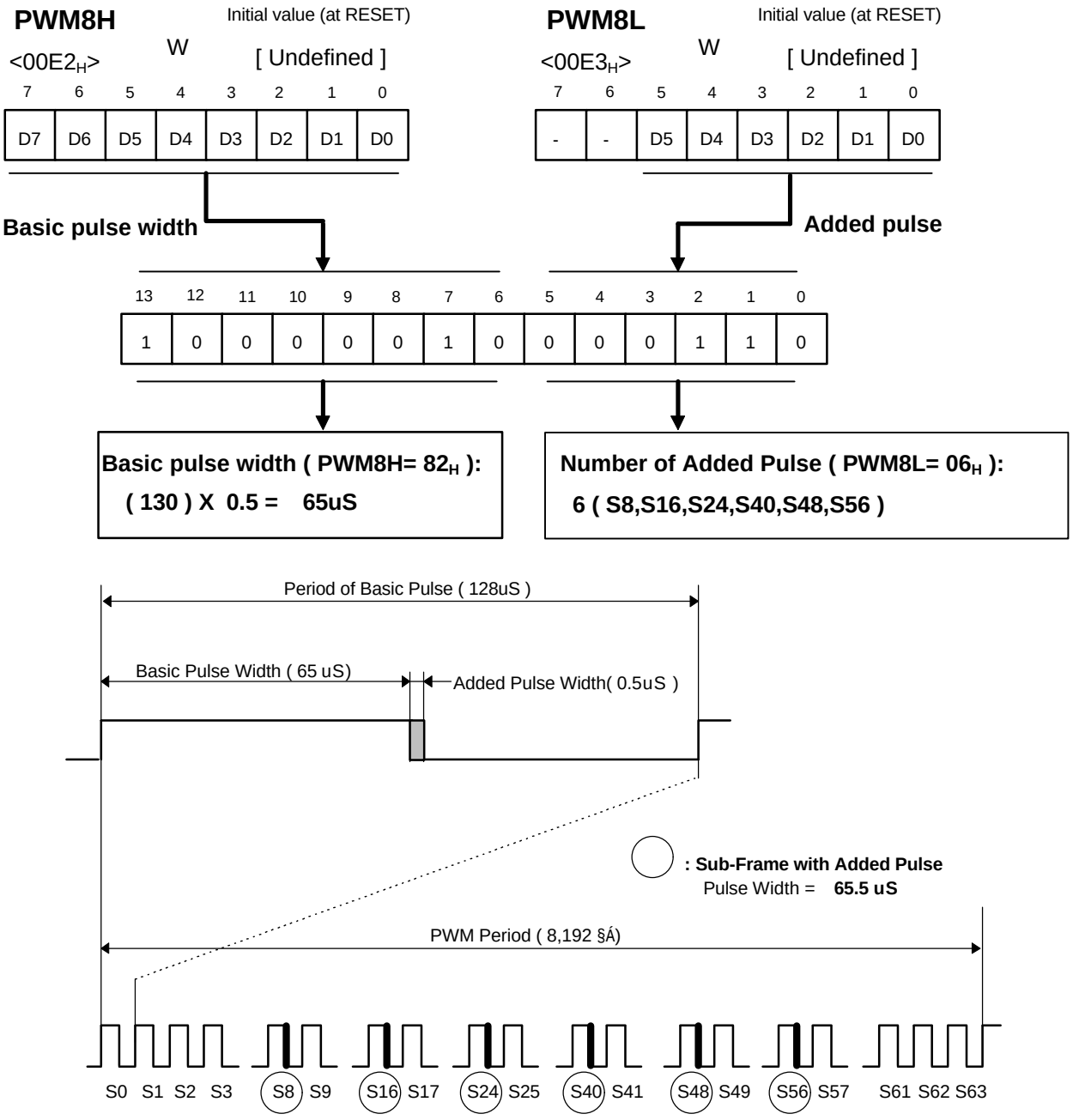


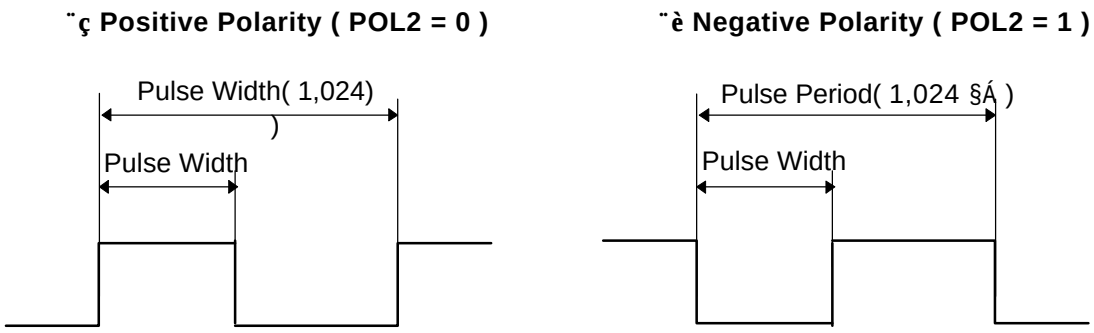
FIG.3.6.3 Example output of the 14-bit PWM (Polarity : Positive)

3.6.2 PWM0¡-PWM7 (7-bit PWM ¡¿ 8 CH.)

Each PWM0~PWM7 can be used for different PWM output by each 7-bit data register (PWMR0~PWM7). The PWM pulse period is 1,024§Á and the width is (PWMR+1)¡¿T/128 . (0<PWMR<127: Value of 7-bit PWM register data)

PWM0~PWM7 is positive, negative for output. The start point of output is spreaded wide, so the flow of current is proper.

PWM0~PWM7 is port is N-MOS open drain.



Pulse Width = (PWMR +1)£¯128 ¡¿ 1,024 [§Á]

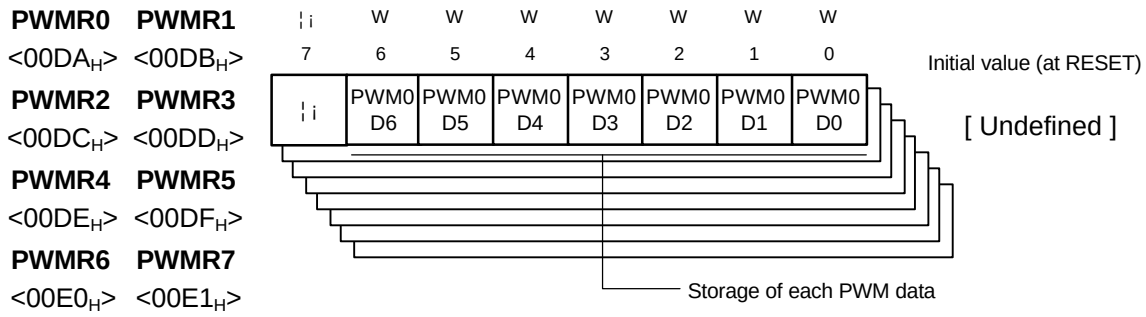
DUTY CYCLE = (PWMR +1)£¯128 ¡¿ 100 [%]

FIG. 3.6.4 Output Pulse of PWM0 ¡- PWM7

3.6.3 PWMR0¡-PWMR7 REGISTER

PWMR0 ¡-PWMR7 are the data register to define 7-bit PWM pulse width and it has only write . They are undefined at reset state.

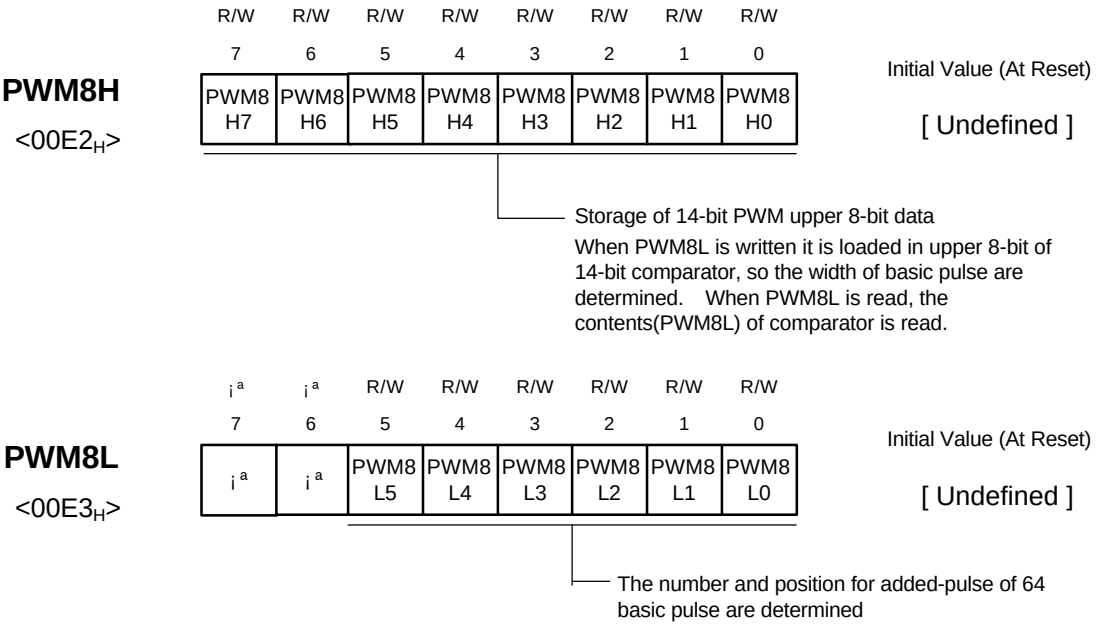
PWMR0 ¡-PWMR7 DATA REGISTER



3.6.4 PWM8H, PWM8L REGISTER

PWM register (PWM8H, PWM8L) are the data register to define 14bit PWM pulse width and it is enable R/W. They are not fixed at reset state.

PWM8H , PWM8L DATA REGISTER



3.6.5 The control of 14-bit PWM

¨ç Write upper data to **PWM8H** (definition of basic pulse width)

¨è Write lower 6-bit data to **PWM8L**

¨é When data are written PWM8L, the 14-bit data of PWM8 is written to compare register so, even if you will change the upper 8-bit of output data, you will write 6bit data to PWM8L again. But when you will change the lower 6-bit of output data, you need not write the upper 8-bit to PWM8H again..

¨ê Output polarity is determined by POL1.(bit2 of PWM control register 2)

Default is positive polarity.(POL1=0)

¨ë PWM8 port is selected by setting EN8(bit1 of PWM ontrol register1) to “1”, so the wave of PWM is to be output.

¨ì If CNTB(bit6 of PWMCR1) is "0", Counter is operating, on the contrary if it is "1" count stops.

This have an effect on both of them.(14-bit PWM/7-bit PWM counter)

3.6.6 The Control of 7-bit PWM

¨ç Write 7-bit data to each PWM data register (PWMR0~PWMR7).

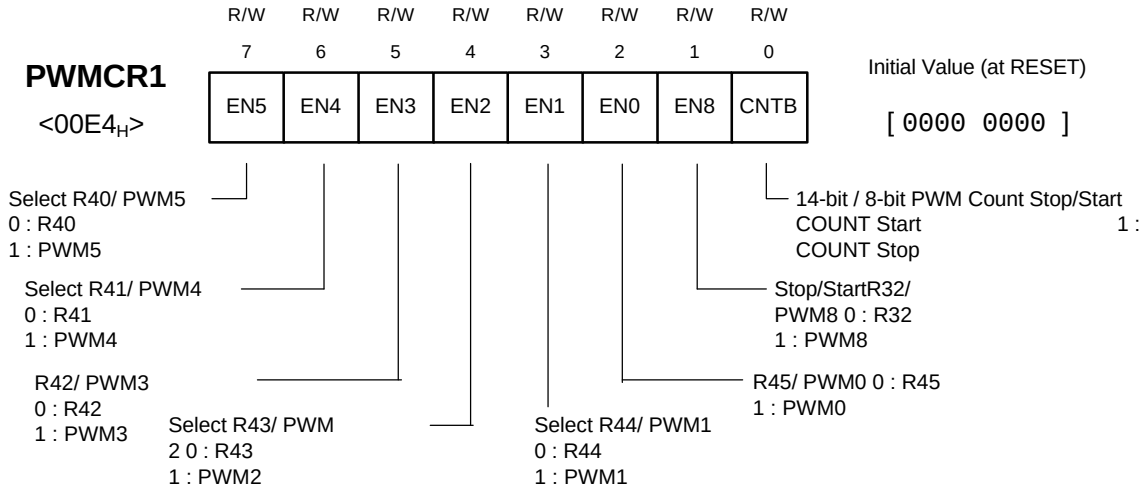
¨è**Define output polarity by POL2 (bit3 of PWMCR2(PWM control register2))**

Positive polarity is determine by default. also this has effect an all 7bit PWM output

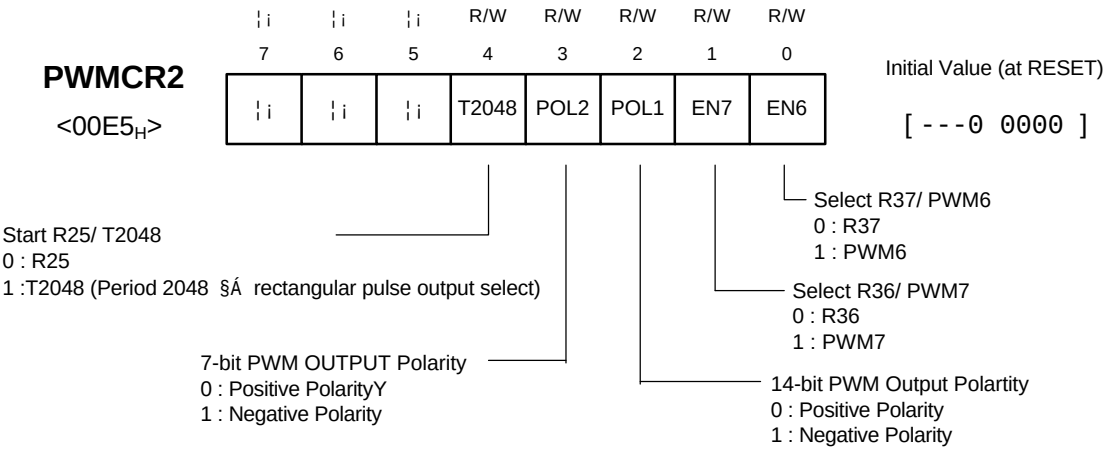
¨é If each PWM port is selected by setting EN0~EN7(bit2~bit7 of PWMCR1, bit1 and bit2 ofPWMCR2) to "1".

¨ê If CNTB(bit0 of PWMCR1) is "0", counter is operating, on the contrary if it is “1” counter stops. This has effect on all 14-bit PWM/7-bit PWM counter.

PWM CONTROL REGISTER 1



PWM CONTROL REGISTER 2



3.7. Interrupt Interval Measurement Circuit

GMS84512/84524 is equipped with distinct edge of input signal for 2 channel external interrupt (INT1, INT2) and interrupt interval measurement circuit of evaluating distinct edge interval.

Interrupt interval measurement circuit is equipped with interrupt input multiplexer, 8bit binary up-counter measurement clock selection circuit, interrupt interval storage circuit and interrupt interval measurement control register

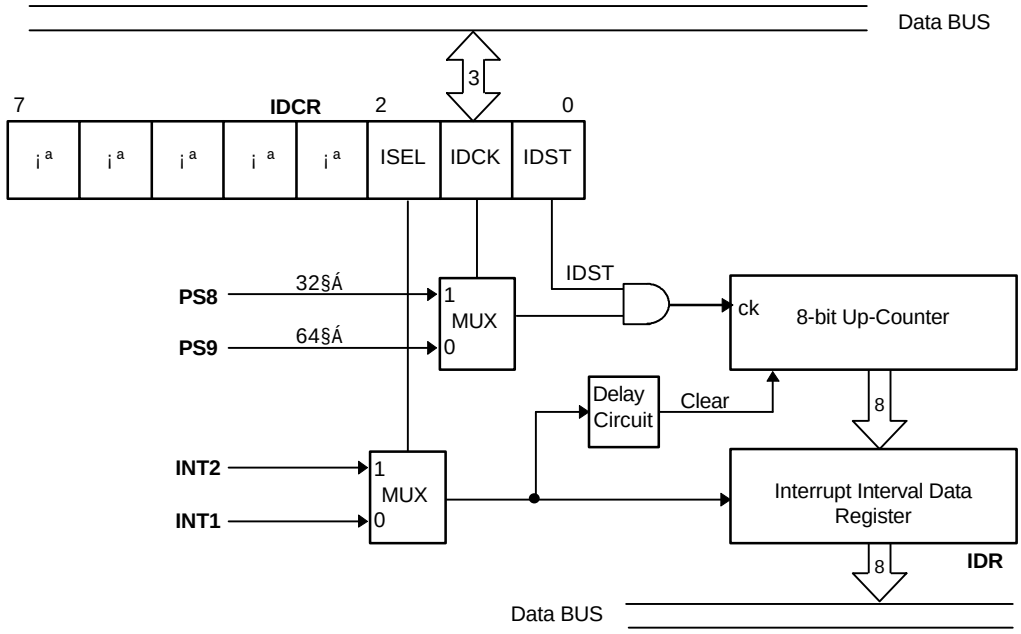


FIG. 3.7.1 Configruation of interrupt interval measurement circuit

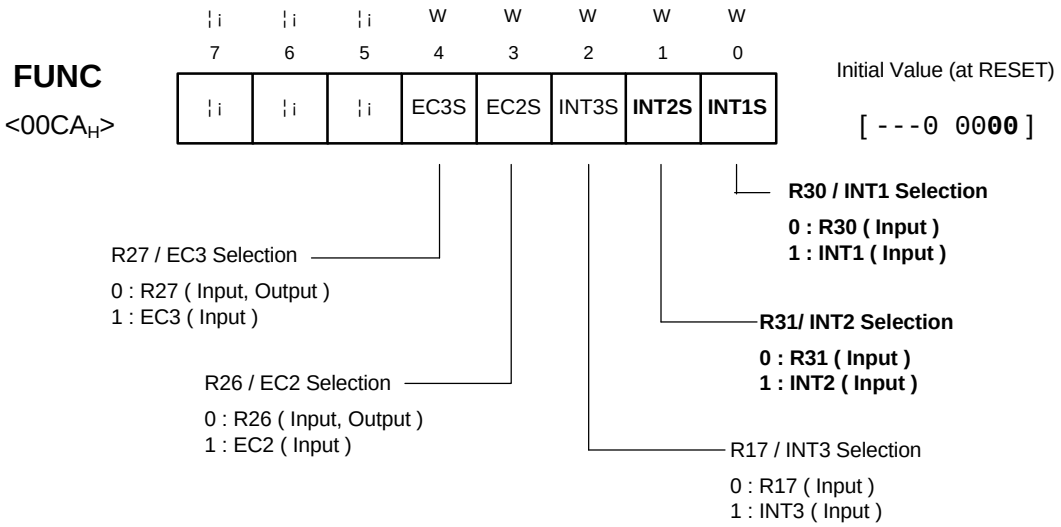
3.7.1 Operation of Interrupt Interval Measurement Circuit

Interrupt interval measurement circuit stores the count value of 8-bit up counter to IDR(interrupt interval data register) by selected edge of external interrupt input. And then it may clear 8-bit up-counter, go on counting again. And the counter value of 8-bit up-counter is stored to IDR by selected edge of second external interrupt input.

So, selected edge interval of external interrupt input is measured to PS8(32§Á) or PS9(64§Á). Rising/Falling edge of interrupt input signal is selected by IEDS(External Interrupt Signal Edge Selection)and width or period of input signal is measured by combination of selected edge. External interrupt input signal is selected by FUNC(port function selection register)

Fig 3.7.2 and TABLE 3.7.1 show interrupt input signal edge selection and measurement interval.

PORT FUNCTION SELECTION REGISTER



EXT. INTERRUPT EDGE SELECTION REGISTER

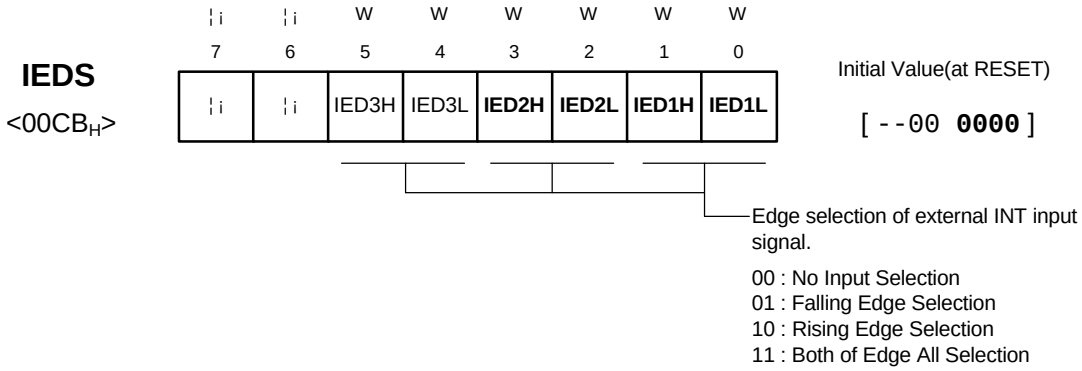


TABLE 3.7.1 Measurement Interrupt Interval and Edge Selection

	Symbol	IED*H	IED*L
Period	¨Í	1	0
	¨Î	0	1
Width of Pulse	¨Ï	1	1
	¨Ð	1	1

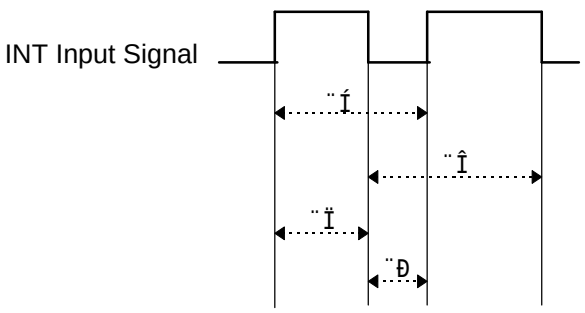


FIG 3.7.2 The Kind of Interrupt Interval

3.7.2 Interrupt Interval Measurement Method

The following is a interrupt interval measurement method.

¨ç Select interrupt input port to be used by writing data to FUNC($00CA_H$)

¨è To measure interrupt interval, select the edge of interrupt input signal by writing data to IEDS($00CB_H$)

¨é Control to write data to IDCR (Interrupt interval measurement control register) .

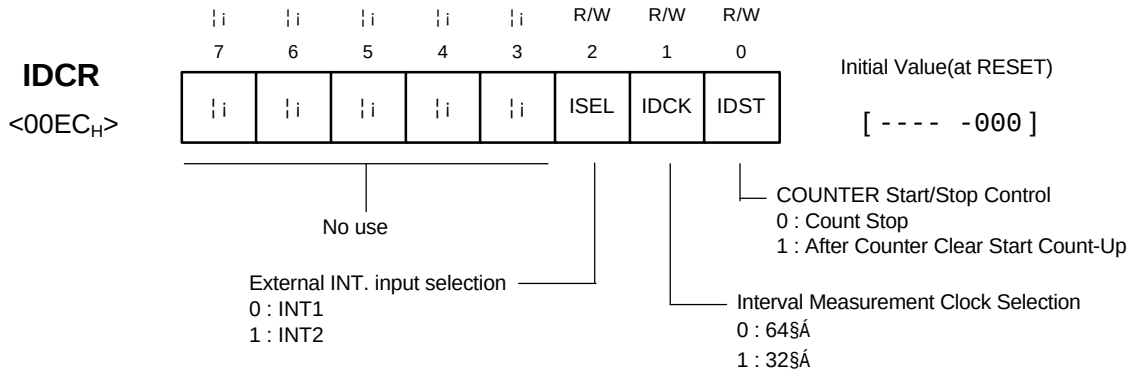
When IDST(bit0 fo IDCR) is "1", counter is operating. If IDCK (bit1 of IDCR) selecting measurement clock is "0", PS9(64§Á) is selected, otherwise PS8(32§Á) is selected.

If ISEL(bit of IDCR) selecting external interrupt input is "0", INT1 is selected, otherwise INT2 is selected.

¨ê If using edge of interrupt input signal is to be input automatically the value of counter is stored to IDR(00EDH), after 1 machine cycle, counter is to be clear and go on count-up.

So, interrupt interval is measured continuously.

INTERRUPT INTERVAL DETERMINATION CONTROL REGISTER



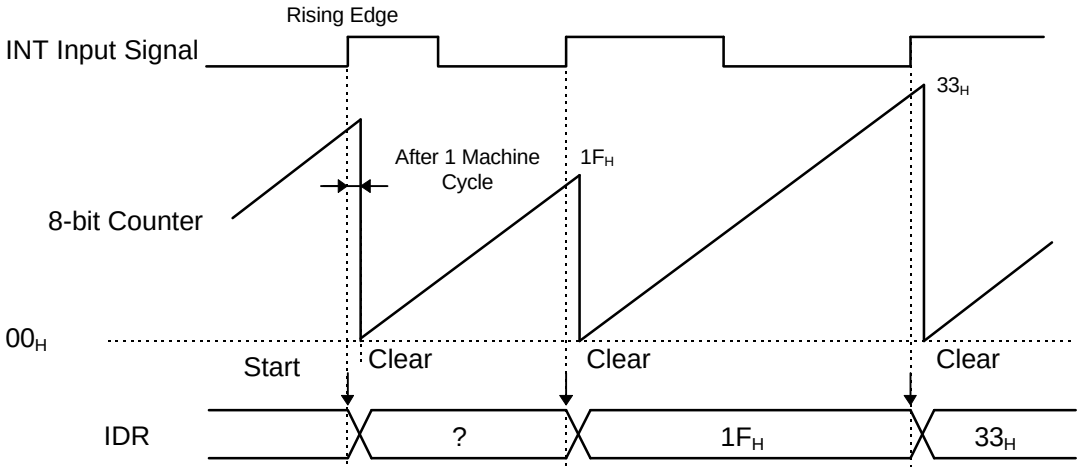


FIG 3.7.3 The example of Interrupt Interval Measurement

3.8 On Screen Display (OSD)

3.8.1 OSD Overview

The OSD of GMS84512/84524 can display maxium 128 kinds of character or symbol to CRT screen, basically GMS84512/84524 incorporates a 22 characters ¡¿3lines CRT display control circuit. If OSD interrupt is to be used, maxium 12 lines can be displayed.

Especially, GMS84512/84524 is equipped with smoothing function and color edge function.

3.8.2 Feature of OSD

- OSD CLOCK : **4 §Ů ¡- 8 §Ů**
- The Nnmber of Character : **128 characters** (include 2 test characters)
- Display Ability : **22 Character ¡¿ 3 lines**
  (use OSD Interrupt : enable 12 lines)
- Character Size : **16 kinds** (every line unit)
- Character Color : **8 kinds** (every character unit)
- Font Configulation : **14 ¡¿18 Dots**
- Display Position : **Horizontal 61 steps, Vertical 128 step (every line unit)**
- Display MOde : **Character Mode, Background Mode**
  Color Edge Mode, Blanking Mode
  (every line unit)
- Background Size : **Domain of total screen, domain of line unit**
- Background and Edge Color : **8 kinds**
- **Smoothing Function**
- **OSD Oscillator Control Function**

3.8.3 Configuration of OSD

The OSD of GMS84512/84524 is equipped with OSD oscillator, timing circuit, display position register (HDP1, HDP2, HDP3, VDP1, VDP2, VDP3) display mode register (DMSS1, DMSS2, DMSS3), display control register (OSDCON1, OSDCON2), character ROM storing 128 kinds of character font, display RAM (22 character ¡¿ 3 lines) storing font address and color data of display character and output control circuit.

Fig 3.8.1 is a block of OSD circuit of GMS84512/84524

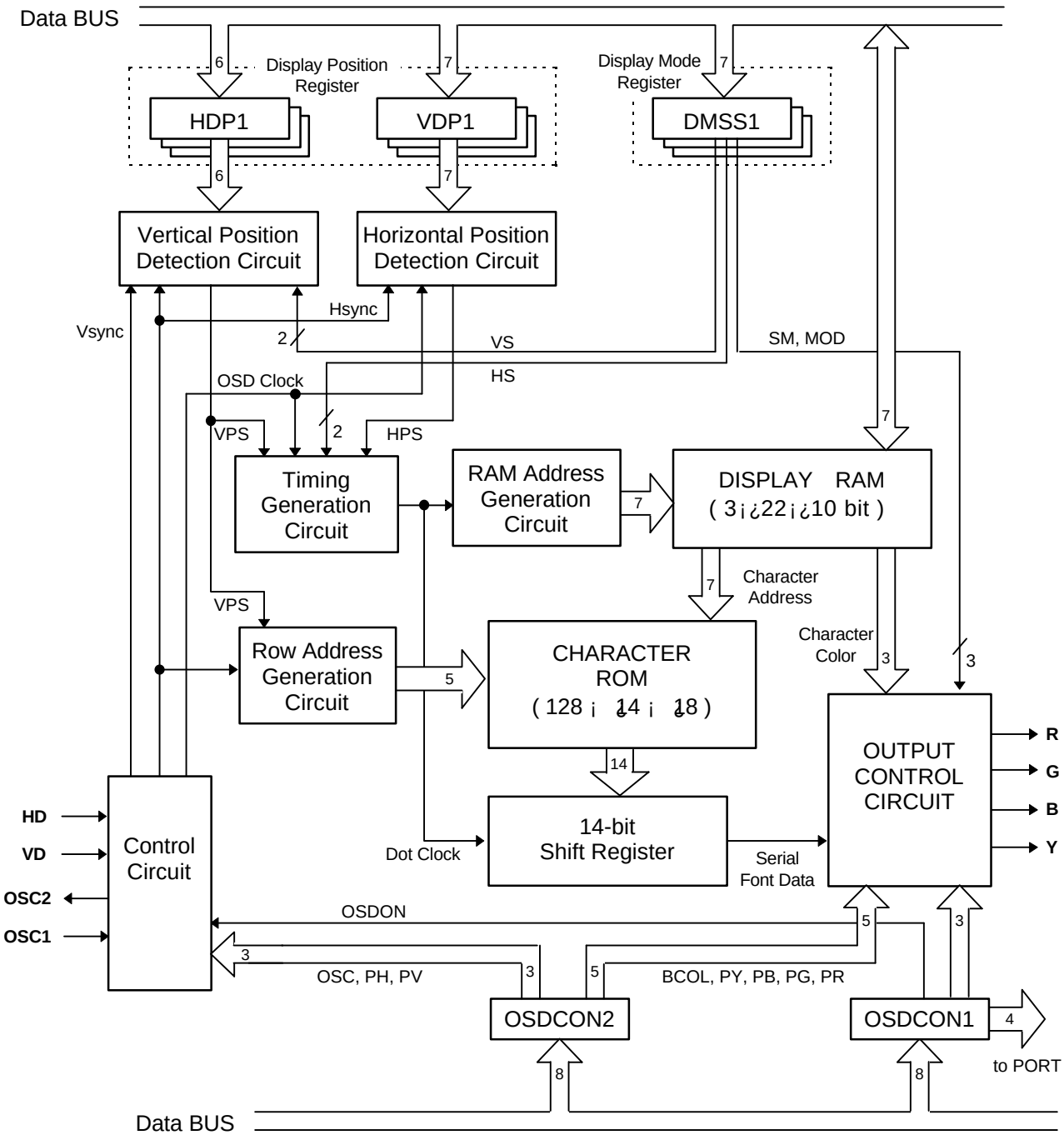


FIG. 3.8.1 Configuration of OSD Block

3.8.4 OSD DISPLAY RAM (2-page, 0200_H ¡-$02DF_H$)

OSD DISPLAY RAM is storing 3lines¡¿ 22 characters of character address and color, gives data to character font ROM and output control circuit in order to do OSD output.

When data are input, OSD display RAM separates character address and color and accesses twice. When data are output, data(10bits) are output once.

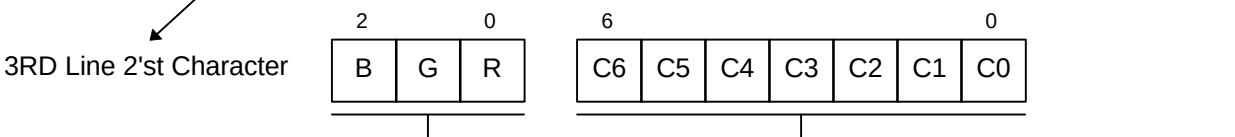
If OSD RAM (2page) accessable register PG2R(00FCH) is set "1" instruction of direct page addressing mode can be used to OSD DISPLAY RAM.

TABLE 3.8.1 Direct Page Access Method

G-Flag = 0	G-Flag = 1	
	PG2R = 0	PG2R = 1
0 Page	1 Page	2 Page

TABLE 3.8.2 OSD DISPLAY RAM ADDRESS (2-page, 0200_H ¡-$02DF_H$)

1ST LINE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHARACTER	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
COLOR	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95
2ND LINE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHARACTER	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35
COLOR	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5
3RD LINE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHARACTER	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55
COLOR	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5



Color	Color Data		
	B	G	R
Black	0	0	0
Red	0	0	1
Green	0	1	0
Yellow	0	1	1
Blue	1	0	0
Purpul	1	0	1
Cyan	1	1	0
White	1	1	1

FIG. 3.8.2 OSD DISPLAY RAM and DATA Configuration

3.8.5 OSD DISPLAY MODE REGISTER (DMSS1, DMSS2, DMSS3)

OSD Display mode register is register to control **Display Character Size, Display Mode, Smoothing function.**

OSD display mode register is every display line, so OSD display mode is determined by line.

OSD MODE REGISTER

DMSS1
<00F6$_H$>
DMSS2
<00F7$_H$>
DMSS3
<00F8$_H$>

¦¡	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
¦¡	1SM	1MOD 1	1MOD 0	1HS1	1HS0	1VS1	1VS0

Initial Value (At Reset)

[-000 0000]

Character size definition

vs	00	01	10	11
00	1×1	2×1	3×1	4×1
01	1×2	2×2	3×2	4×2
10	1×3	2×3	3×3	4×3
11	1×4	2×4	3×4	4×4

Display Mode Definition
00 : Blanking Mode
01 : Character Mode
10 : Color Edge Mode
11 : Background Mode

Smoothing Function definition
0 : Smoothing Off
1 : Smoothing On

- **Character Size Definition (every line unit)**

Each character basic configuration is a 14¡¿ 18 dots. The dot size of vertical direction is determined by VS1, VS0(bit1, 0 of DMSS) and the dot size of Horizontal direction is determined by HS1, HS0(bit3, 2 of DMSS). So, the size of display character is to be changable.

TABLE 3.8.3 Character Size Definition

HS1, HS0 / VS1, VS0	00	01	10	11
00	1¡¿1	2¡¿1	3¡¿1	4¡¿1
01	1¡¿2	2¡¿2	3¡¿2	4¡¿2
10	1¡¿3	2¡¿3	3¡¿3	4¡¿3
11	1¡¿4	2¡¿4	3¡¿4	4¡¿4

- **Display Mode Definition (every line unit)**

Display Mode is defined by MOD1, MOD0(bit5, 4 of DMSS) and is blanking mode, character mode, color edge mode, background mode and so on.

Background domain is determined by BCOL(bit6 of OSDCON2).

If BCOL is "0" line domain is determined on the contrary, if BCOL is "1" total screen is determined.

Notes : When BCOL is "1" only background mode is enable (Refer to Fig 3,8,3)

Mode	Blanking Mode	Character Mode	Color Edge Mode	Background Mode
MOD1,MOD0	00	01	10	11
BCOL=0		ABCD	**ABCD**	ABCD
BCOL=1		ABCD		

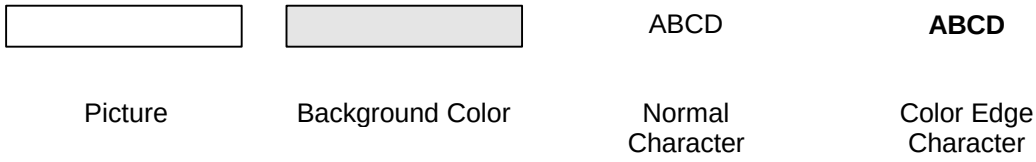


FIG. 3.8.3 DISPLAY MODE

- **Smoothing Function Definition (every line unit)**

  When the size of display character is over 2-times than normal size, smoothing function can smooth the rectangular part.

  Smoothing function is defined by SM (bit 6 of DMSS)

  (If SM is "1" function is ON, otherwise function is OFF)

  Fig 3.8.4 shows color edge function and smoothing function

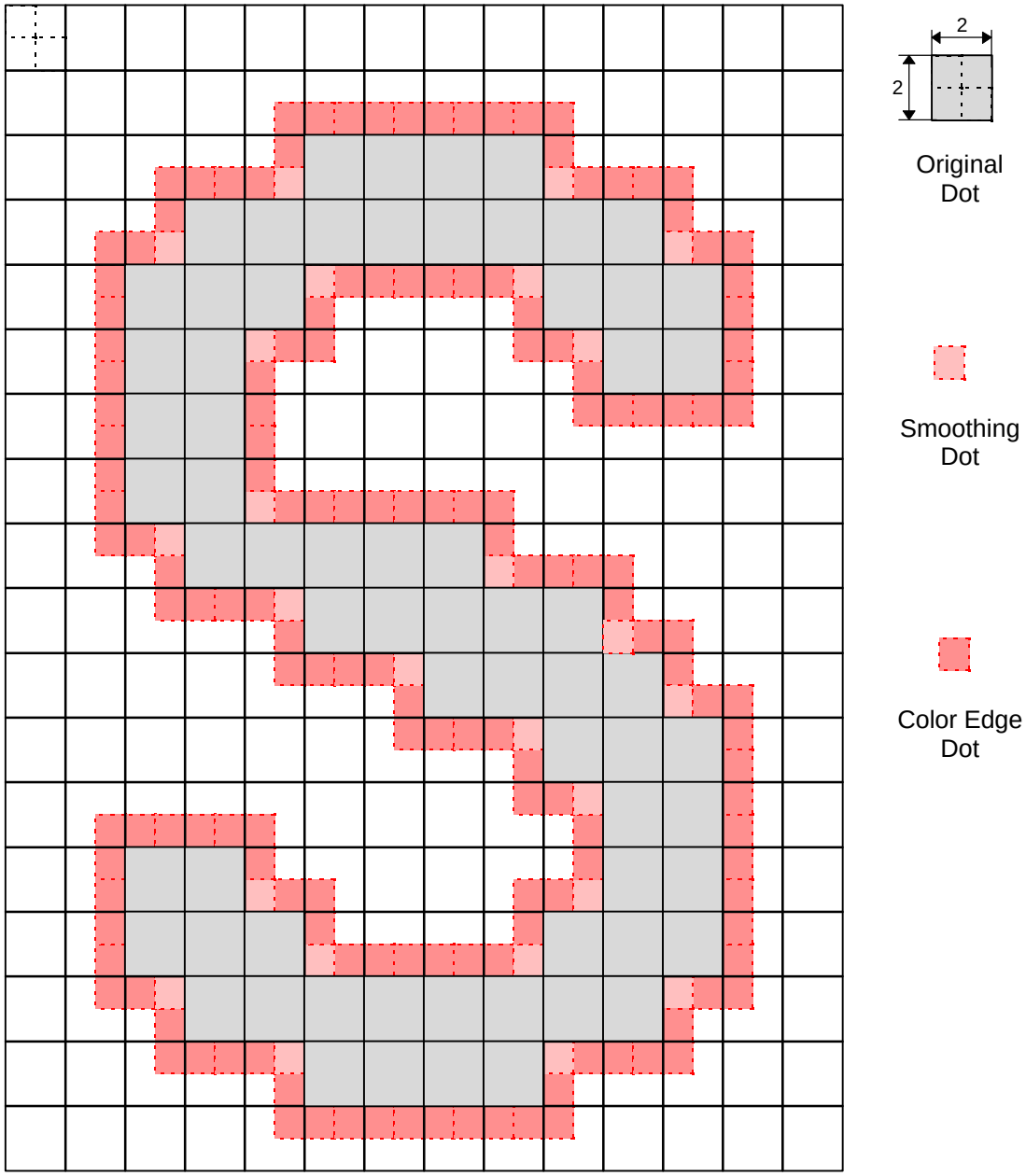


FIG. 3.8.4 Color Edge Function and Smoothing Function

3.8.6 OSD Display Position Register(HDP1¡-HDP3, VDP1¡-VDP3)

OSD Display Position Register defines horizontal, vertical position of screen every each display line.

HDPi, VDPi define display position of i-line first character. (only I= 1,2,3)

<Note> The value of HDPi is to be over four, the value of VDPi don't have to be included in domain of previous line.

OSD HORIZONTAL POSITION REGISTER

HDP1 <00F0$_H$>
HDP2 <00F1$_H$>
HDP3 <00F2$_H$>

¦i	¦i	W	W	W	W	W	W
7	6	5	4	3	2	1	0
¦i	¦i	1HP5	1HP4	1HP3	1HP2	1HP1	1HP0

Initial Value(at Reset)

[--00 0000]

Each Line Horizontal Position Definition
HSi = To ¡¿ { 4 ¡¿ (HDPi) £« 10 }
[To = OSD Clock Period, i = 1, 2, 3]

OSD VERTICAL POSITION REGISTER

VDP1 <00F3$_H$>
VDP2 <00F4$_H$>
VDP3 <00F5$_H$>

¦i	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
¦i	1VP6	1VP5	1VP4	1VP3	1VP2	1VP1	1VP0

Initial Value (at Reset)

[-000 0000]

Each Line Vertical Position Definition
VSi = 2 ¡¿ H ¡¿ (VDPi)
[H orizontal Synchronous Signal Period,,
i = 1, 2, 3]

- **Horizontal Display Position (HSi)**

HSi = To ¡¿ { 4 ¡¿ (HDPi) £« 10 }

[To = OSD Clock Period, i = 1, 2, 3]

- **Vertical Display Position (VSi)**

VSi = 2 ¡¿ H ¡¿ (VDPi)

[H = Horizontal Synchronous Signal Period, i = 1, 2, 3]

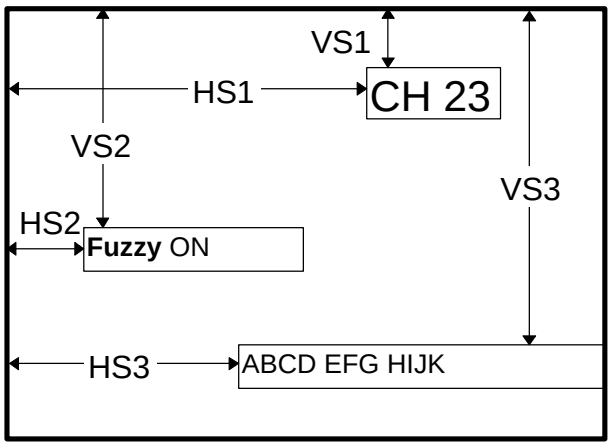


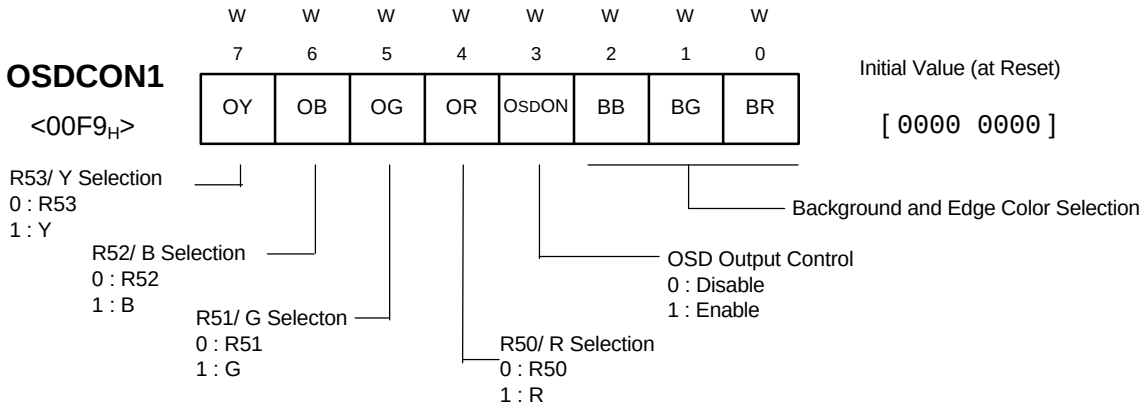
FIG. 3.8.5 CRT Screen and Display Position

3.8.7 OSD output Control Register(OSDCON1, OSDCON2)

- **OSDCON1 ($00F9_H$)**

OSD output control register 1(**OSDCON1**)defines background and edge color (BB, BG, BR) and is enable/disable OSD output(OSDON) and defines function of output port. (OY, OB, OG, OR)

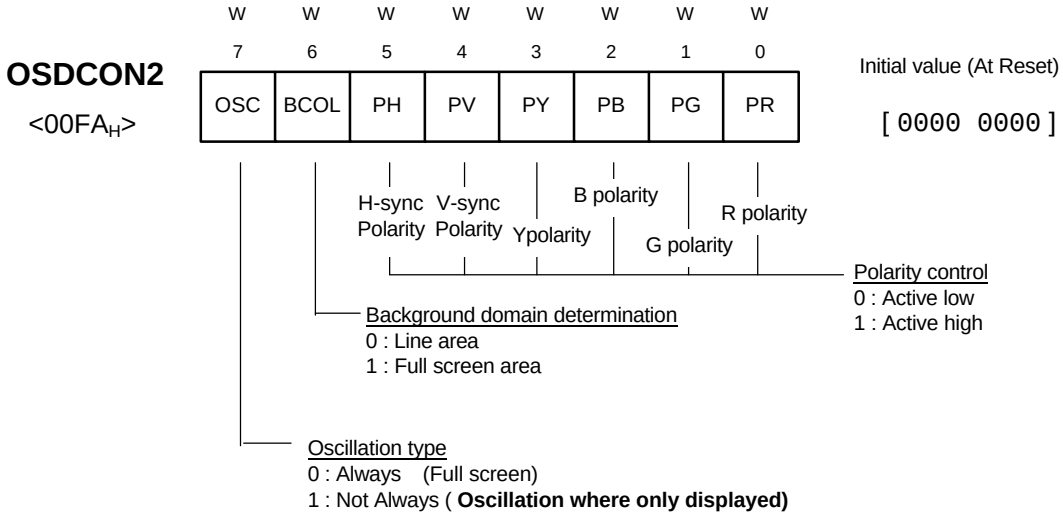
OSD OUTPUT & BACKGROUND CONTROL REGISTER



- **OSDCON2 ($00FA_H$)**

OSD output control register 2 (OSDCON2) defines the polarity of OSD output(PY,PB,PG,PR) and selects the polarity of input HD, VD(PH,PV) and defines background domain(BCOL) and defines the type of OSD oscillation (OSC)**.**

OSD I/O POLARITY & OSCILLATION CONTROL REGISTER



3.8.8 MULTI LINE DISPLAY

The OSD function of GMS84512/84524 is basically enable 3-line display, but if OSD interrupt is used maximum up to 12 lines can be displayed.

OSD interrupt request occurs when OSDON(bit3 of OSDCON1) is "1" and each line display finishes, and OSD interrupt happens when OSD interrupt request occurs, at this time I-Flag(bit2 of PSW) and OSDE(bit7 of IENH($00EA_H$)) has to set "1".

OSD Display allows multiple lines(more than 3 lines) to be displayed on the screen by OSD interrupt, each time one line is displayed and rewriting display RAM data, display position register (HDPi, VDPi) and display mode register in the OSD interrupt service routine for which display is terminated.

- 6 Line Display Occasion

¨ç 1'st Line Dispaly ¡æ Load 4'th line data(contents, position, mode) to 1'st line RAM and register.

¨è 2'nd Line Dispaly ¡æ Load 5'th line data(contents, position, mode) to 2'nd line RAM and register

¨é 3'rd Line Display ¡æ Load 6'th line data(contents, position, mode) to 3'rd line RAM and register

¨ê 4'th Line Dispaly

¨ë 5'th LIne Dispaly

¨ì 6'th Ling Dispaly

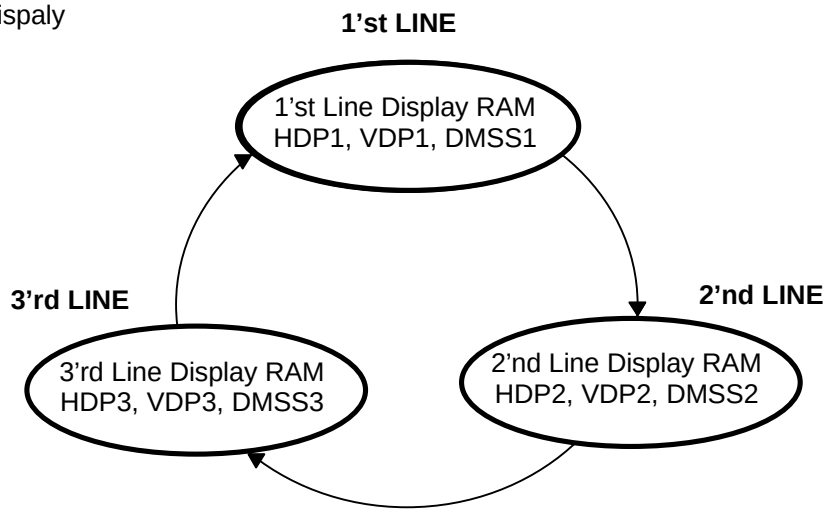


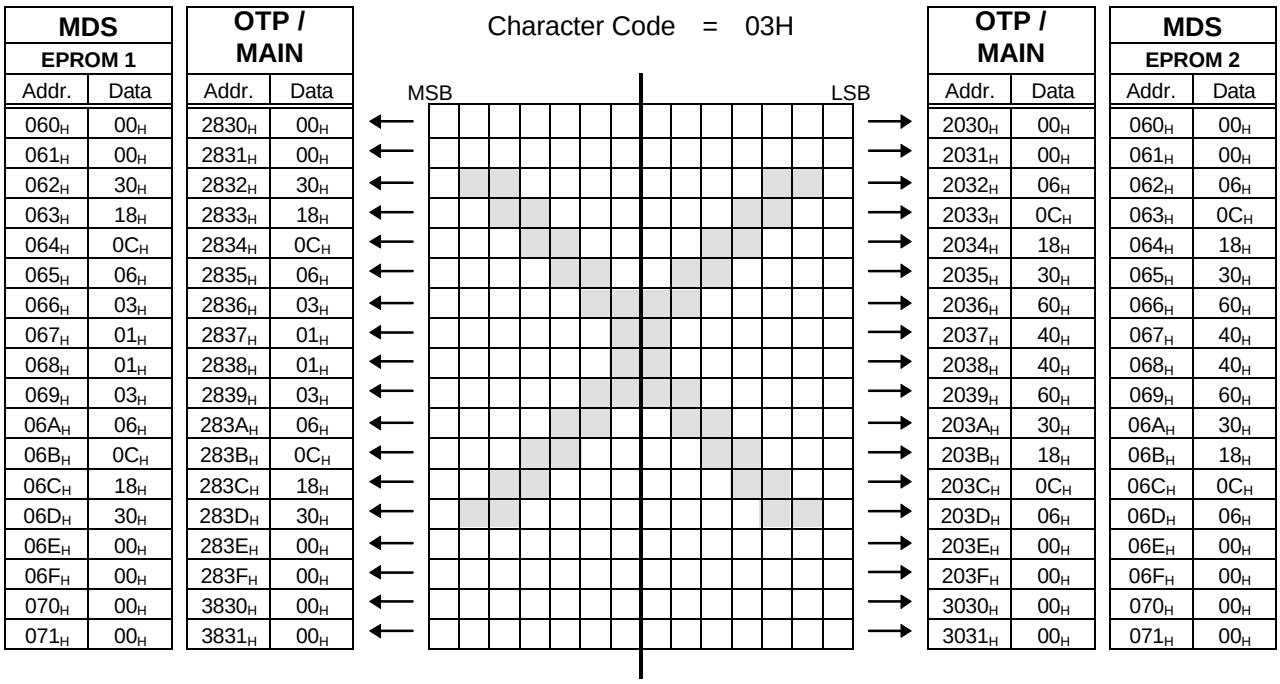
FIG. 3.8.6 OSD Display Method

3.8.9 Character ROM

The character ROM of GMS84512/84524 stores 128 kinds of font dot pattern data.

36bytes of dot pattern data needs to display one character.

Fig 3.8.7 is a example of character dot pattern, TABLE 3.8.4 is a relation about character code and character dot pattern address.



MDS EPROM 1		OTP / MAIN		OTP / MAIN		MDS EPROM 2	
Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
060_H	00_H	2830_H	00_H	2030_H	00_H	060_H	00_H
061_H	00_H	2831_H	00_H	2031_H	00_H	061_H	00_H
062_H	30_H	2832_H	30_H	2032_H	06_H	062_H	06_H
063_H	18_H	2833_H	18_H	2033_H	$0C_H$	063_H	$0C_H$
064_H	$0C_H$	2834_H	$0C_H$	2034_H	18_H	064_H	18_H
065_H	06_H	2835_H	06_H	2035_H	30_H	065_H	30_H
066_H	03_H	2836_H	03_H	2036_H	60_H	066_H	60_H
067_H	01_H	2837_H	01_H	2037_H	40_H	067_H	40_H
068_H	01_H	2838_H	01_H	2038_H	40_H	068_H	40_H
069_H	03_H	2839_H	03_H	2039_H	60_H	069_H	60_H
$06A_H$	06_H	$283A_H$	06_H	$203A_H$	30_H	$06A_H$	30_H
$06B_H$	$0C_H$	$283B_H$	$0C_H$	$203B_H$	18_H	$06B_H$	18_H
$06C_H$	18_H	$283C_H$	18_H	$203C_H$	$0C_H$	$06C_H$	$0C_H$
$06D_H$	30_H	$283D_H$	30_H	$203D_H$	06_H	$06D_H$	06_H
$06E_H$	00_H	$283E_H$	00_H	$203E_H$	00_H	$06E_H$	00_H
$06F_H$	00_H	$283F_H$	00_H	$203F_H$	00_H	$06F_H$	00_H
070_H	00_H	3830_H	00_H	3030_H	00_H	070_H	00_H
071_H	00_H	3831_H	00_H	3031_H	00_H	071_H	00_H

Fig. 3.8.7 The Example of Character Dot Pattern

TABLE 3.8.4 The relation of Character Code and Dot Pattern Address

Character CODE	OTP / MAIN CHIP		MDS
	Upper 7 - Bits	Lower 7 - Bits	Upper 7-bit,Lower 7-bit
00_H	2800_H ¡ - $280F_H$, 3800_H , 3801_H	2000_H ¡ - $200F_H$, 3000_H , 3001_H	000_H ¡ - 011_H
01_H	2810_H ¡ - $281F_H$, 3810_H , 3811_H	2010_H ¡ - $201F_H$, 3010_H , 3011_H	020_H ¡ - 031_H
02_H	2820_H ¡ - $282F_H$, 3820_H , 3821_H	2020_H ¡ - $202F_H$, 3020_H , 3021_H	040_H ~ 051_H
03_H	2830_H ¡ - $283F_H$, 3830_H , 3831_H	2030_H ¡ - $203F_H$, 3030_H , 3031_H	060_H ¡ - 071_H
XX_H	(2800_H + $XX0_H$) ¡ - (2800_H + XXF_H) (3800_H + $XX0_H$) , (3800_H + $XX1_H$)	(2000_H + $XX0_H$) ¡ - (2000_H + XXF_H) (3000_H + $XX0_H$) , (3000_H + $XX1_H$)	(20_H * XX_H + 00_H) ¡ - (20_H * XX_H + 11_H)
$7E_H$ *	$2FE0_H$ ¡ - $2FEF_H$, $3FE0_H$, $3FE1_H$	$27E0_H$ ¡ - $27EF_H$, $37E0_H$, $37E1_H$	$FC0_H$ ¡ - $FD1_H$
$7F_H$ *	$2FF0_H$ ¡ - $2FFF_H$, $3FF0_H$, $3FF1_H$	$27F0_H$ ¡ - $27FF_H$, $37F0_H$, $37F1_H$	$FE0_H$ ¡ - $FF1_H$

* these addresses are reserved for test (user not available)

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

3. Peripheral Function

4. Control Function

5. Support Tool

6. Appendix

4.1 INTERRUPTS

GMS84512/84524 has the following function to process interrupt request from the peripheral and external interrupt pin.

- Interrupt Source : 14
- Interrupt Vector : 14
- Multi Interrupt Possible.
- Programmable Interrupt Mode
  - ¨ç Hardware Priority Mode
  - ¨è Software Selection Mode
- R/W of Interrupt Request Flag is possible and in Interrupt Accept, automalically resetted.

4.1.1 Interrupt Circuit Configuration and Kinds

GMS84512/84524 Interrupt circuits is consist of Interrupt Enable Register (IENH,IENL), Interrupt Request Register (IRQH,IRQL), priority circuit and selecting circuit. Configuration of Interrupt circuit is shown in Fig. 4.1.1

The Interrupt sources are external interrupt source(INT1, INT2, INT3, V-sync), peripheral function source(OSD,T0,T1,T2,T3,1ms,WDT,BIT,Serial I/O) and software interrupt source(BRK).

In the case of reset input($\overline{\text{RESET}}$), the program execution at the start address located in vector table address like general interrupt.

The classification of interrupt source is shown in Table 4.1.1.

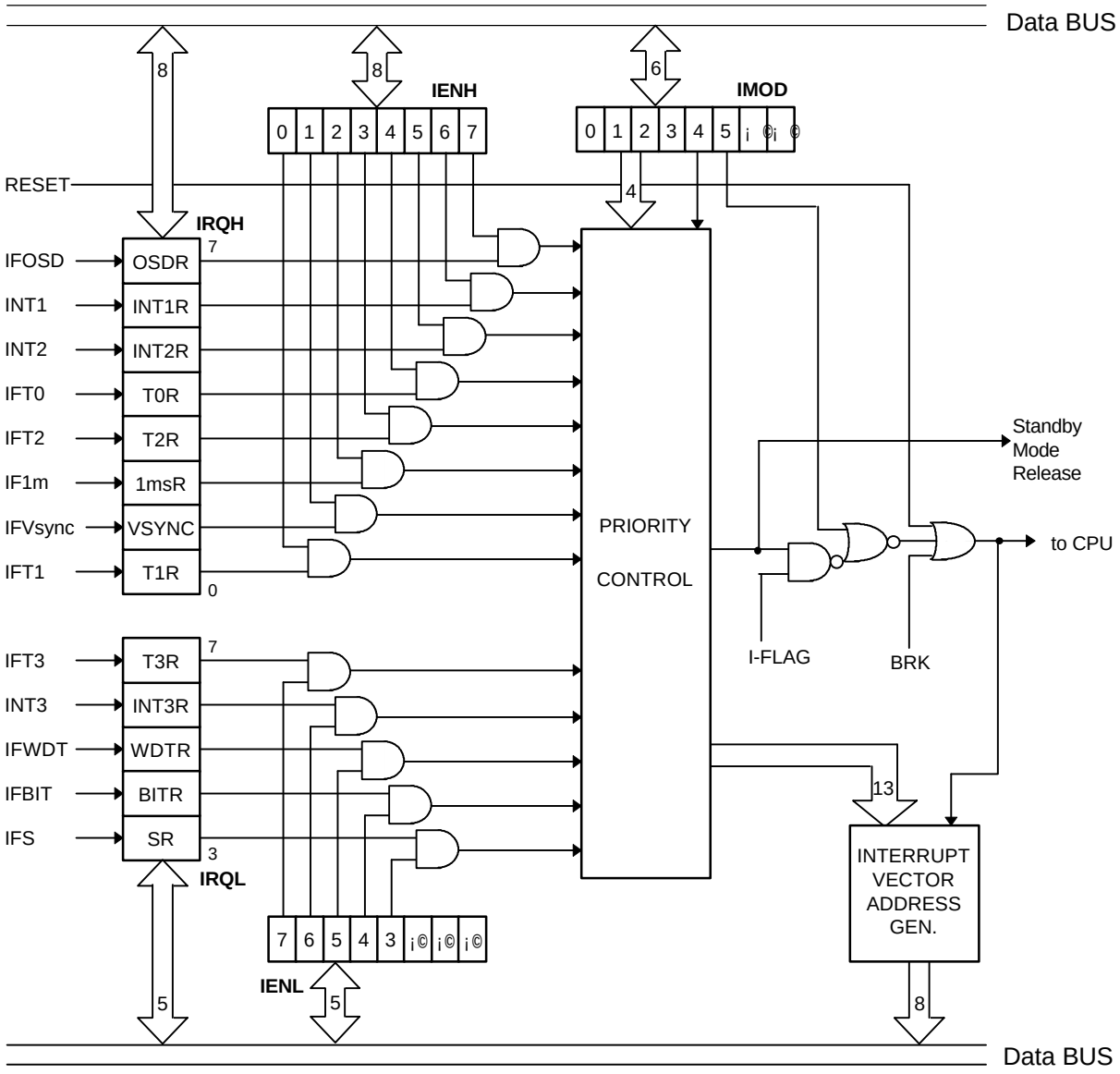


FIG. 4.1.1 Interrupt Function Block Diagram

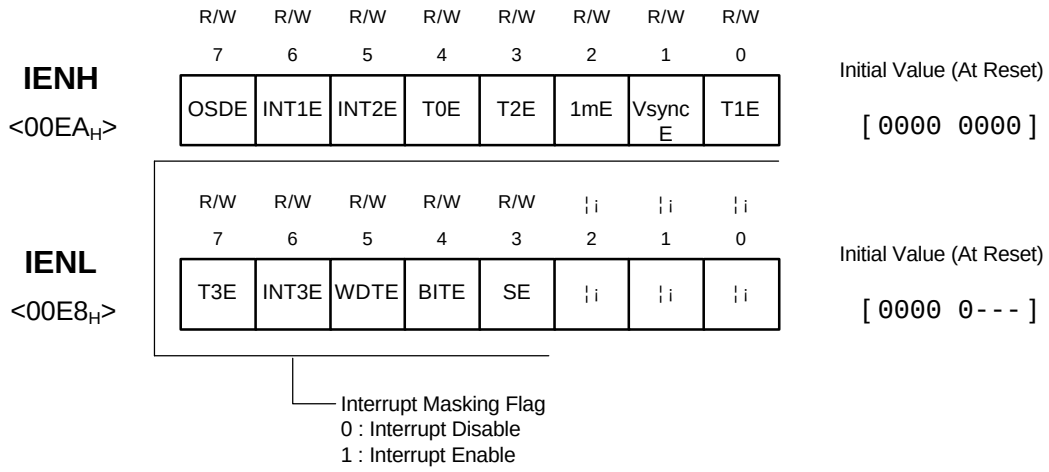
TABLE 4.1.1 Interrupt Request Source

Type	Mask	Priority	Interrupt Request Source	Vector H	Vector L
Hardware Interrupt	Non Maskable	1	RST Reset Pin	$FFFF_H$	$FFFE_H$
	Mask Enable	2	OSD On Screen Display	$FFFB_H$	$FFFA_H$
		3	INT1R External Interrupt 1	$FFF9_H$	$FFF8_H$
		4	INT2R External Interrupt 2	$FFF7_H$	$FFF6_H$
		5	T0R Timer 0	$FFF5_H$	$FFF4_H$
		6	T2R Timer 2	$FFF3_H$	$FFF2_H$
		7	1ms 1§Â Interrupt	$FFF1_H$	$FFF0_H$
		8	VSYNC V-sync Interrupt	$FFEF_H$	$FFEE_H$
		9	T1R Timer 1	$FFED_H$	$FFEC_H$
		10	T3R Timer 3	$FFEB_H$	$FFEA_H$
		11	INT3R External Interrupt 3	$FFE9_H$	$FFE8_H$
		12	WDTR Watch Dog Timer	$FFE7_H$	$FFE6_H$
		13	BITR Basic Interval Timer	$FFE5_H$	$FFE4_H$
		14	SR Serial I/O	$FFE3_H$	$FFE2_H$
S/W Interrupt	Non Maskable	¡ª	BRK Break Instruction	$FFDF_H$	$FFDE_H$

4.1.2 Interrupt Control

- To process interrupt, set the interrupt master enable flag I-Flag(3'rd bit of PSW). when I-Flag="0" all interrupts are disable except RESET and S/W interrupt.
- Interrupt Enable Register (**IENH, IENL**) includes interrupt enable bits of each interrupt source, and interrupt is accepted when the interrupt enable bit and the interrupt request bit are both "1".

INTERRUPT ENABLE REGISTER H, L



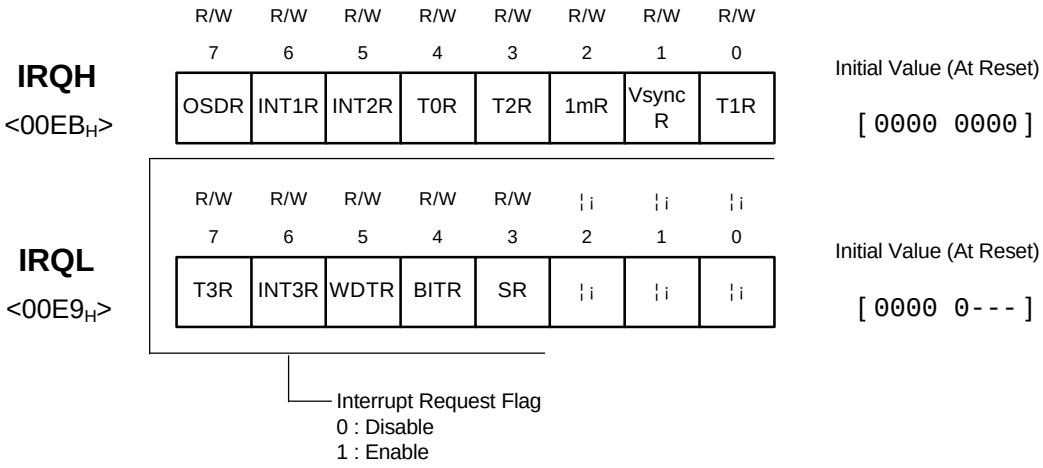
- Interrupt Request Flag Register (**IRQH, IRQL**)

  When interupt occurs, interrupt request flag is set. The accepted interrupt request flag is automatically cleared by interrupt process cycle.

  As long as the interrupt request flag which is set to "1" is not cleared by program, it maintains '1" until interrupt is accepted.

  Interrupt Request Flag Register (**IRQH, IRQL**) is Read/ Write Register. So, it is possible to be checked and changed by program.

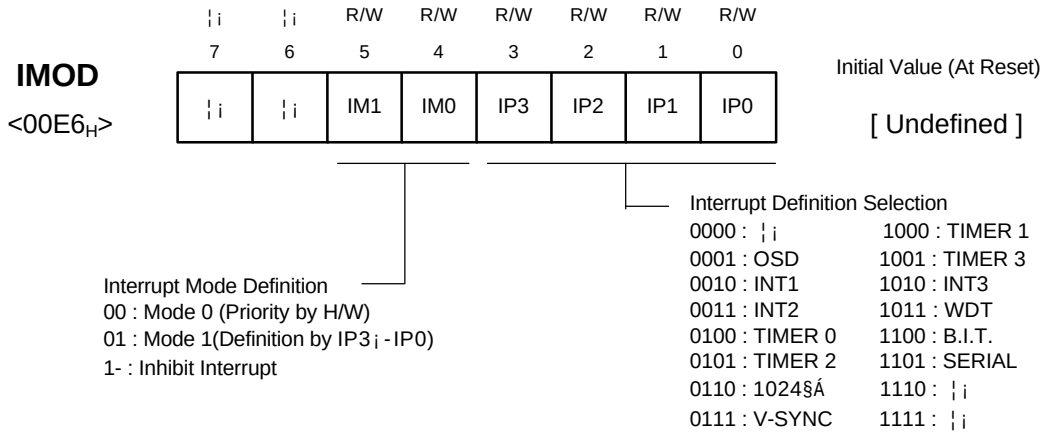
INTERRUPT REQUEST FLAG REGISTER H, L



- Interrupt Mode Register (**IMOD**)

  Interrupt Mode Register determines interrupt priority which can be selected by hardware or program.

INTERRUPT MODE REGISTER



- Interrupt Mode

¨Í Mode 0 (Priority by H/ W)

OSDR£¾ INT1R£¾INT2R£¾ T0R£¾T2R£¾1mR£¾VSYNCR£¾T1R£¾T3R£¾ INT3R£¾WDTR£¾BITR£¾SR

¨Î Mode 1 (Selection by IP3 ¡- iP0)

TABLE 4.1.2 Selection of Interrupt by IP3 ¡- IP0

IP3	IP2	IP1	IP0	Selection Interrupt
0	0	0	0	¡ª
0	0	0	1	OSDR On Screen Display
0	0	1	0	INT1R External Interrupt 1
0	0	1	1	INT2R External Interrupt 2
0	1	0	0	T0R Timer 0
0	1	0	1	T2R Timer 2
0	1	1	0	1msR 1§Â Interrupt
0	1	1	1	VSYNCR External V-sync Interrupt
1	0	0	0	T1R Timer 1
1	0	0	1	T3R Timer 3
1	0	1	0	INT3R External Interrupt 3
1	0	1	1	WDTR Watch Dog Timer
1	1	0	0	BITR Basic Interval Timer
1	1	0	1	SR Serial I/O
1	1	1	0	¡ª
1	1	1	1	¡ª

- Interrupt Accept Timing

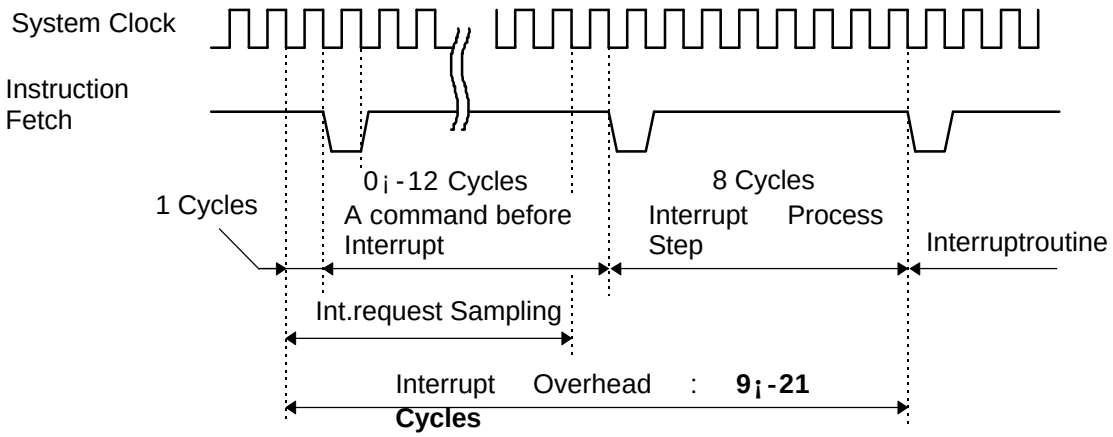


FIG. 4.1.2 Interrupt Accept Timing

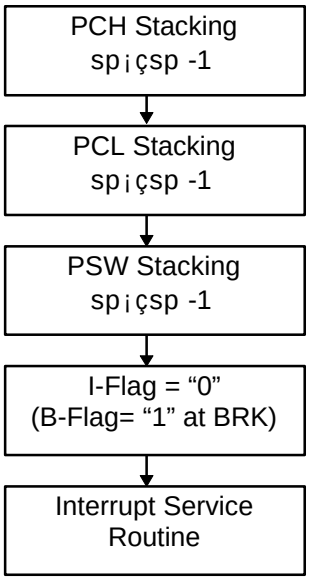
- The vlaid timing after executing Interrupt control Flag
  - ¨Í I-Flag is valid, after EI, DI executed
  - ¨Î **IENH, IENL** register is valid after next instruction

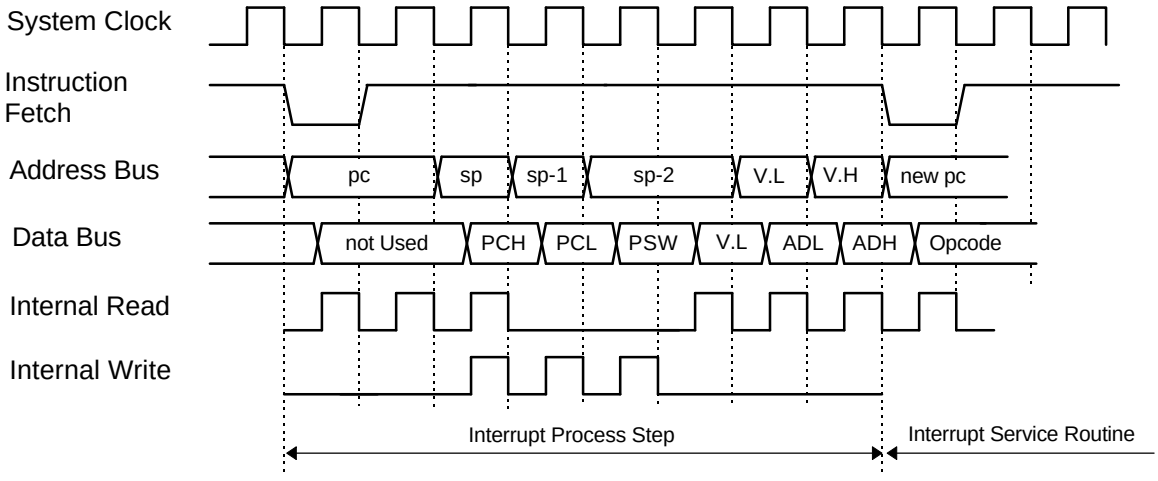
4.1.3 INTERRUPT SEQUENCE

When interrupt is accepted, the execution program is stopped, a certain of interrupt processing step is passed, and interrupt sevice routine is started.

By last instruction of interrupt service routine(RETI) return to original program.

- Interrupt Process Sequence





V.L, V.H is Vector Address, ADL, ADH is start Address of Interrupt Service Routine as Vector Contents

FIG. 4.1.3 Interrupt Process Step Timing

4.1.4 Software Interrupt

Software interrupt is interrupted by BRK instruction. In interrupt processing step I-Flag is cleared. B-flag is setted.

Interrupt vector of BRK instruction is shared with the vector of table call 0, when both instruction of BRK and TCALL 0 are used, each processing routine is executable through looking at the contents at B-Flag.

There is no instruction to Reset B-Flag directly.

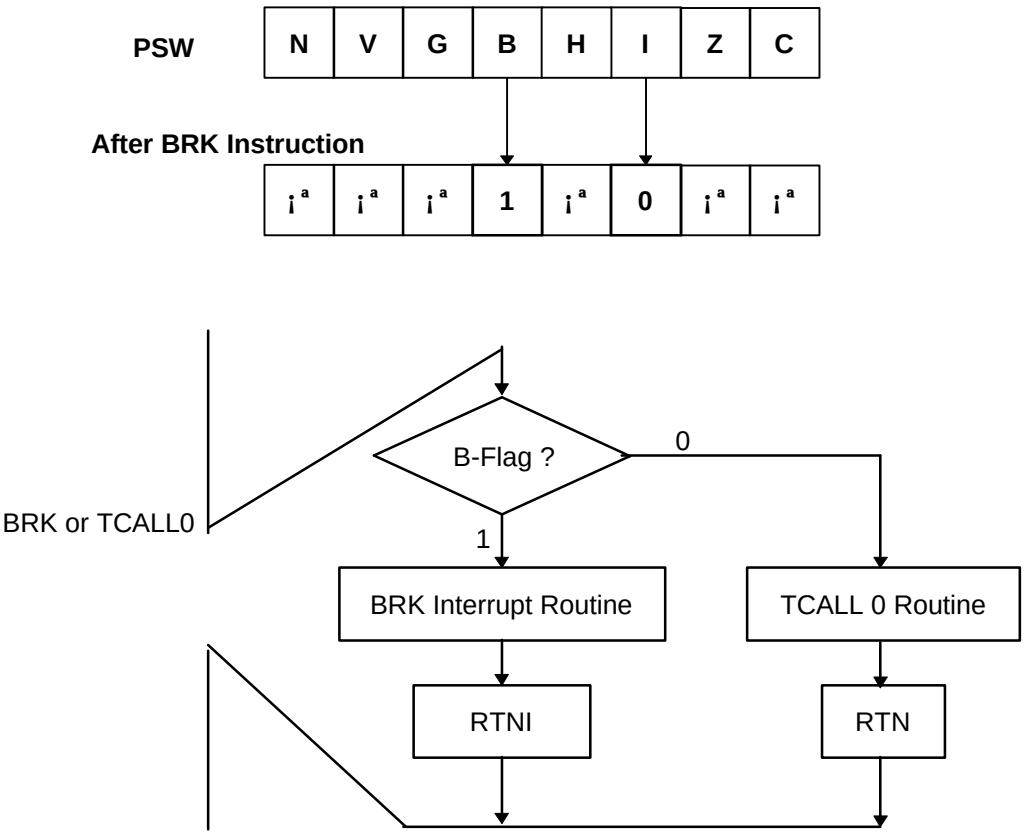


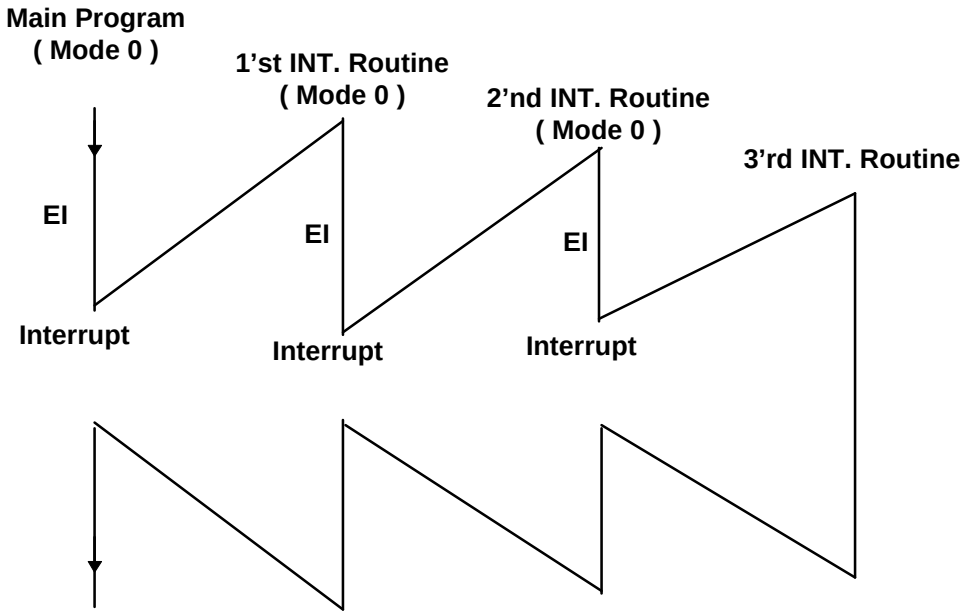
FIG. 4.1.4 Execution of BRK/ TCALL0

4.1.4 Multiple Interrupt

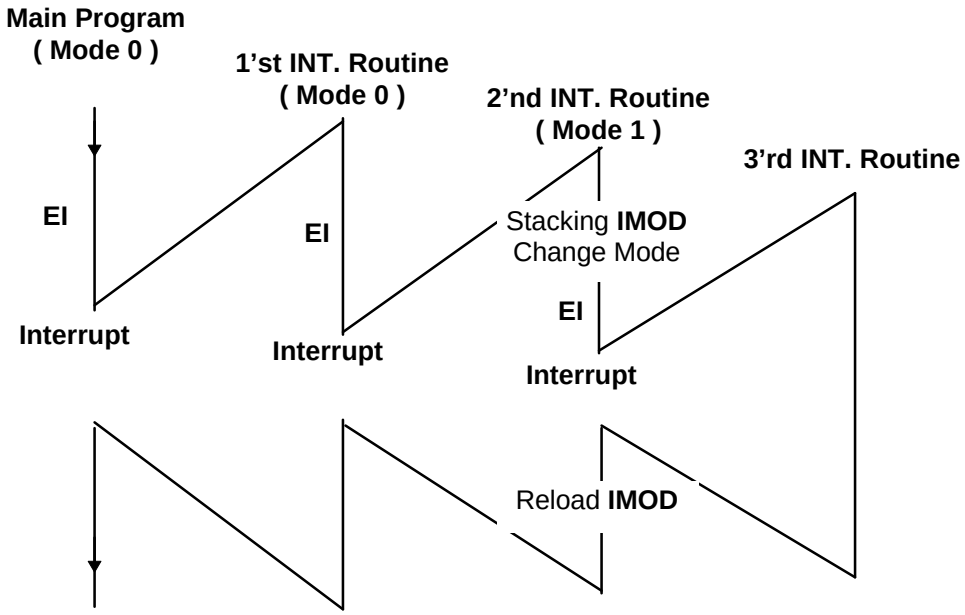
If there is an interrupt, interrupt enable flag is automalically resetted entering the interrupt service routine. After then, no interrupt is accepted. If EI instruction is executed, mask enable bit becomes "1", and each enable bit can accept the interrupt as a reply to 1's interrupt request. If multiple of interrupt request occurs at same time, the one with a higher priority is accepted and the other with lower priority are retained.

When multiple interrupt is accepted, it is possible to change Interrupt Accept Mode.

- Case of multiple interrupt at hardware priority accept mode(Mode0)



- Case of multiple interrupt nest H/W priority accept mode (Mode0) and S/W selection accept mode(Mode1)



4.2. Standby Function

To save the consuming power of device,GMS84512/84524 has Stop Mode.

In this mode,the execution of program stop. Stop Mode can be entered by Stop instruction.

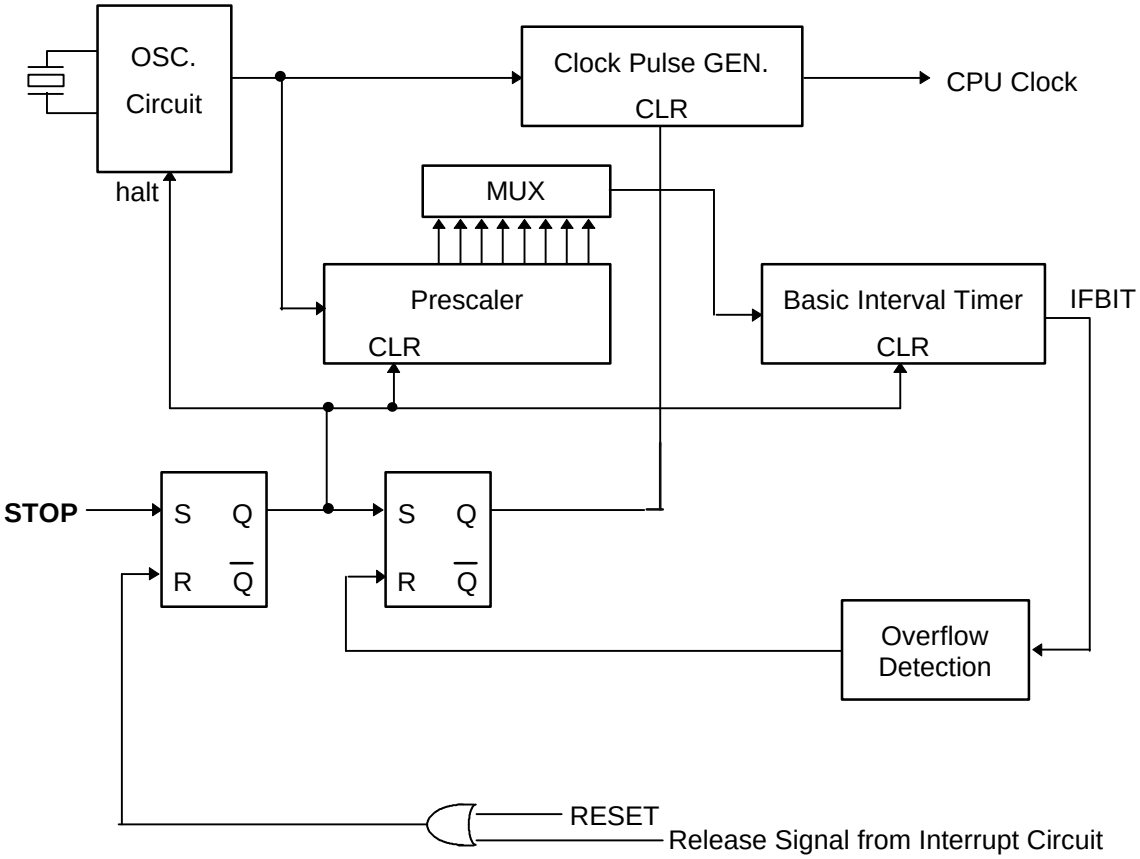


FIG. 4.2.1 STOP Mode Circuit Diagram

TABLE 4.2.1 At STOP Mode Device Operation State.

Peripheral Function	STOP Mode
Oscillator	¡¿
CPU Clock	¡¿
RAM, Register	Retain
I/O Port	Retain
Prescaler	¡¿
Basic Interval Timer	¡¿
Serial I/O	Operation(External Clock Selection)
WDT, Timer, A/D Comp., PWM, OSD, Interrupt Interval Mesurment Circuit	¡¿

4.2.1 STOP Mode

STOP Mode can be entered by STOP instruction during program execution. In STOP mode, oscillator is stopped to make all clocks stop, which leads to the mode requring much less power consumption. All register and RAM data are preserved.

4.2.2 STOP Mode Release

Release of STOP mode is done by reset input or interrupt. When there is a release signal of STOP mode, the instruction execution is started after stabilization oscillation time set by program. After releasing STOP mode, instruction execution is different by I-Flag(bit 2 of PSW)

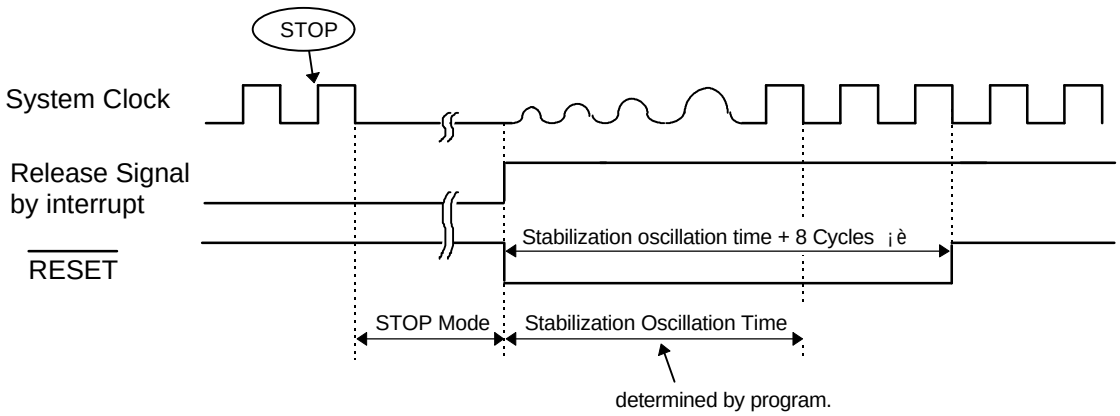
If I-Flag = “1” entered Interrupt Service Routine,

If I-Flag = “0” execute program from next instruction of STOP instruction.

TABLE 4.2.1 STOP Mode Release

Release Factor	Release Method	STOP
$\overline{\text{RESET}}$	By $\overline{\text{RESET}}$ pin=Low level, and Device is initialzed.	○
INT1,INT2 INT3, V-sync	In the state of enable flag=1 corrosponding to each interrupt at the edge.	○
Serial I/O (IFSIO)	When SE="1" and serial I/O is executing by external clock, interrupt occurs by serial I/O operation completed	○

- Release Timing of STOP Mode



When release the STOP Mode, to secure oscillation stabilization time,we use a B.I.T. So before execution STOP instruction, we must select suitable BIT clock for oscillation stabillization time.

Otherwise, It is possible to release by only RESET input.

- Because STOP mode is released by interrupt, even if both of interrupt enable bit(IE) and interrupt request flag is "1", STOP mode can not be executed.

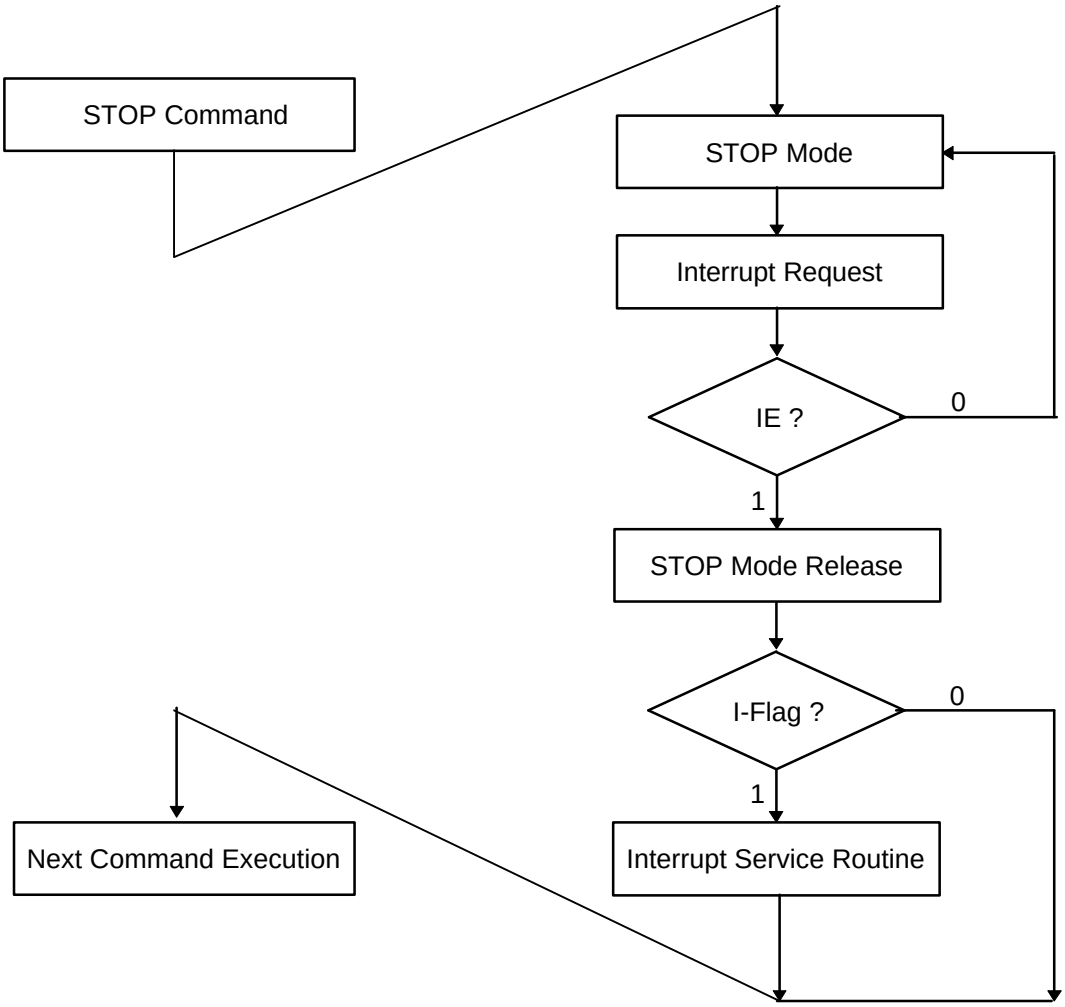


FIG. 4.2.2 STOP Mode Releasing Flow

4.3. Reset Function

To reset the device, maintain the $\overline{\text{RESET}}$="L" at least 8 machine cycle after power supplying and oscillation stabilization.

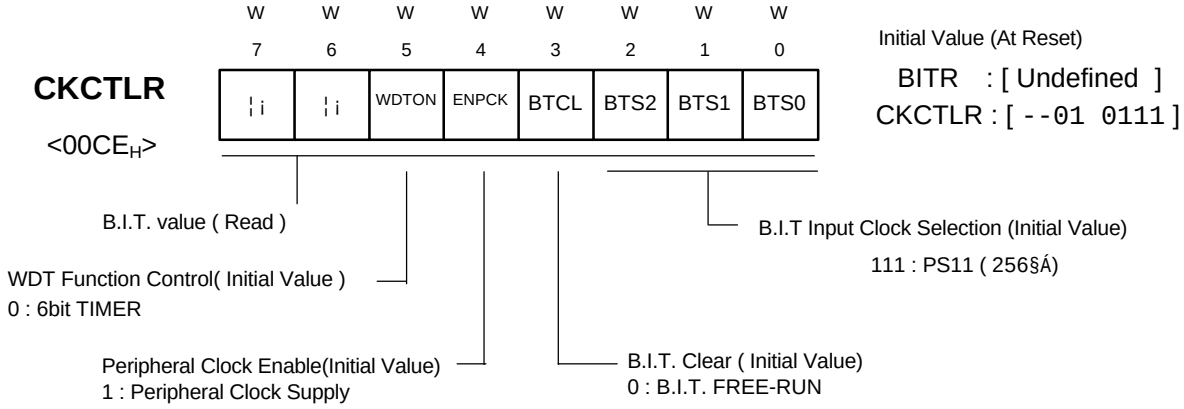
RESET terminal is organized as schmitt input.

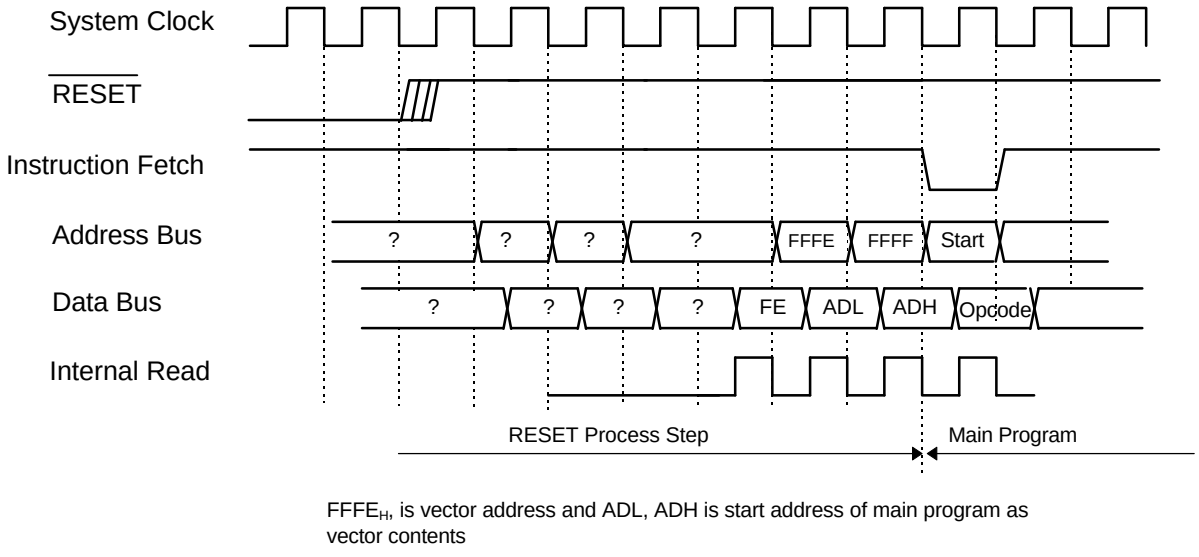
TABLE 4.3.1 is, at Reset, initial value of each register, if initial value is undefined it is needed initialize by a S/W.

Fig 4.3.1 is Timing of Reset Operation (Simular as interrupt instruction)

- **CLOCK CONTROL REGISTER**

CLOCK CONTROL REGISTER





$FFFE_H$, is vector address and ADL, ADH is start address of main program as vector contents

FIG. 4.1.1 RESET Operation Timing

TABLE 4.3.1 Initial state of each register at reset

BLOCK	Symbol	Address	Register Name	R/W	Initial Value								PAGE
					7	6	5	4	3	2	1	0	
CPU	**A**	¡ª	A - Register	¡ª	Undefined								2 - 2
	X	¡ª	X- Register	¡ª	Undefined								2 - 2
	Y	¡ª	Y - Register	¡ª	Undefined								2 - 2
	PSW	¡ª	Program Status Word	¡ª	0	0	0	0	0	0	0	0	2 - 3
	PC	¡ª	Program Counter	¡ª	Undefined								2 - 3
	SP	¡ª	Stack Pointer	¡ª	Undefined								2 - 2
PORT	**R0**	$00C0_H$	R0 Port Data Register	R/W	Undefined								3 - 1
	R0DD	$00C1_H$	R0 Port I/O Direction Register	W	0	0	0	0	0	0	0	0	3 - 1
	R1	$00C2_H$	R1 Port Data Register	R/W	Undefined								3 - 2
	R1DD	$00C3_H$	R1 Port I/O Direction Register	W	0	0	0	0	0	0	0	0	3 - 2
	R2	$00C4_H$	R2 Port Data Register	R/W	Undefined								3 - 4
	R2DD	$00C5_H$	R2 Port I/O Direction Register	W	0	0	0	0	0	0	0	0	3 - 4
	R3	$00C6_H$	R3 Port Data Register	R/W	Undefined								3 - 6
	R3DD	$00C7_H$	R3 Port I/O Direction Register	W	0	0	0	0	0	0	-	-	3 - 6
	R4	$00C8_H$	R4 Port Data Register	R/W	-	-	Undefined						3 - 9
	R5	$00C9_H$	R5 Port Data Register	R/W	-	-	-	-	Undefined				3 - 10
	FUNC	$00CA_H$	Port Function Selection Register	W	-	-	-	0	0	0	0	0	3 - -3
	IEDS	$00CB_H$	External Interupt Edge Selection Register	W	-	-	0	0	0	0	0	0	3 - 39
TIMER	**BITR**	$00CE_H$	Basic Interval Timer Register	R	Undefined								3 - 16
	CKCTLR		Clock Control Register	W	-	-	0	1	0	1	1	1	3 - 13
	WDTR	$00CF_H$	Watch Dog Timer Register	W	-	0	1	1	1	1	1	1	3 - 17
	TM0	$00D0_H$	Timer Mode Register0	R/W	-	0	0	0	0	0	0	0	3 - 21
	TM2	$00D1_H$	Timer Mode Register2	R/W	-	0	0	0	0	0	0	0	3 - 21
	TDR0	$00D2_H$	Timer0 Data Register	R/W	Undefined								3 -21
	TDR1	$00D3_H$	Timer1 Data Register	R/W	Undefined								3 - 21
	TDR2	$00D4_H$	Timer2 Data Register	R/W	Undefined								3 - 21
	TDR3	$00D5_H$	Timer3 Data Register	R/W	Undefined								3 - 21
A/D COMP.	**CMR**	$00D6_H$	A/D Comparator Mode Register	W*6	0	0	-	0	0	0	0	0	3 - 27
	CIS	$00D7_H$	A/D Comparator Channel Selection Register	W	-	-	-	-	-	-	0	0	3 - 27
SERIAL I/O	**SIOM**	$00D8_H$	Serial I/O Mode Register	R/W *0	-	0	0	0	0	0	0	1	3 - 29
	SIOR	$00D9_H$	Serial I/O Data Register	R/W	Undefined								3 - 28
INTERRUPT	**IMOD**	$00E6_H$	Interrup Mode Register	R/W	-	-	0	0	0	0	0	0	4 - 4
	IENL	$00E8_H$	Interrupt Enable Register Low	R/W	0	0	0	0	0	-	-	-	4 - 3
	IRQL	$00E9_H$	Interrupt Request Flag Register Low	R/W	0	0	0	0	0	-	-	-	4 - 4
	IENH	$00EA_H$	Interrupt Enable Register High	R/W	0	0	0	0	0	0	0	0	4 - 3
	IRQH	$00EB_H$	Interrupt Request Flag Register High	R/W	0	0	0	0	0	0	0	0	4 - 4
	IEDS	$00CB_H$	External Interupt Edge Selection Register	W	-	-	0	0	0	0	0	0	3 - 39
INTERRUPT INTERVAL D.	**IDCR**	$00EC_H$	Interrupt Interval Determination Control Register	R/W	-	-	-	-	-	0	0	0	3 - 40
	IDR	$00ED_H$	Interrupt Interval Determination Register	R	0	0	0	0	0	0	0	0	3 - 38

TABLE 4.3.1 Initial State of Each Register at Reset

BLOCK	SYMBOL	Address	Register Name	R/W	Initial Value								PAGE
					7	6	5	4	3	2	1	0	
PWM	**PWM0**	$00DA_H$	PWM0 Data Register	W	-	Undefined							3 - 35
	PWM1	$00DB_H$	PWM1 Data Register	W	-	Undefined							3 - 35
	PWM2	$00DC_H$	PWM2 Data Register	W	-	Undefined							3 - 35
	PWM3	$00DD_H$	PWM3 Data Register	W	-	Undefined							3 - 35
	PWM4	$00DE_H$	PWM4 Data Register	W	-	Undefined							3 - 35
	PWM5	$00DF_H$	PWM5 Data Register	W	-	Undefined							3 - 35
	PWM6	$00E0_H$	PWM6 Data Register	W	-	Undefined							3 - 35
	PWM7	$00E1_H$	PWM7 Data Register	W	-	Undefined							3 - 35
	PWM8H	$00E2_H$	PWM8 Data Register High	R/W	Undefined								3 - 36
	PWM8L	$00E3_H$	PWM8 Data Register Low	R/W	-	-	Undefined						3 - 36
	PWMCR1	$00E4_H$	PWM Control Register1	R/W	0	0	0	0	0	0	0	0	3 - 37
	PWMCR2	$00E5_H$	PWM Control Register2	R/W	-	-	-	0	0	0	0	0	3 - 37
OSD	**HDP1**	$00F0_H$	OSD 1st Line Horizontal Position Register	W	-	-	0	0	0	0	0	0	3 - 47
	HDP2	$00F1_H$	OSD 2nd Line Horizontal Position Register	W	-	-	0	0	0	0	0	0	3 - 47
	HDP3	$00F2_H$	OSD 3rd Line Horizontal Position Register	W	-	-	0	0	0	0	0	0	3 - 47
	VDP1	$00F3_H$	OSD 1st Line Vertical Position Register	W	-	0	0	0	0	0	0	0	3 - 47
	VDP2	$00F4_H$	OSD 2nd Line Vertical Position Register	W	-	0	0	0	0	0	0	0	3 - 47
	VDP3	$00F5_H$	OSD 3rd Line Vertical Position Register	W	-	0	0	0	0	0	0	0	3 - 47
	DMSS1	$00F6_H$	OSD 1st Line Display Mode, Character Size, Smoothing Function Selection Register	W	-	0	0	0	0	0	0	0	3 - 44
	DMSS2	$00F7_H$	OSD 2nd Line Display Mode, Character Size, Smoothing Function Selection Register	W	-	0	0	0	0	0	0	0	3 - 44
	DMSS3	$00F8_H$	OSD 3rd Line Display Mode, Character Size, Smoothing Function Selection Register	W	-	0	0	0	0	0	0	0	3 - 44
	OSDCON1	$00F9_H$	OSD Output and Background Control Register	W	0	0	0	0	0	0	0	0	3 - 48
	OSDCON2	$00FA_H$	OSD I/O Polarity Control and OSD Oscilllation Control Register	W	0	0	0	0	0	0	0	0	3 - 48
	PG2R	$00FC_H$	OSD RAM (2 page) Accessable Register	R/W	-	-	-	-	-	-	-	0	3 - 43

¡Ø -: Not use, *0: bit 0 is READ only, *6: bit 6 is READ only.

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

3. Peripheral Function

4. Control Function

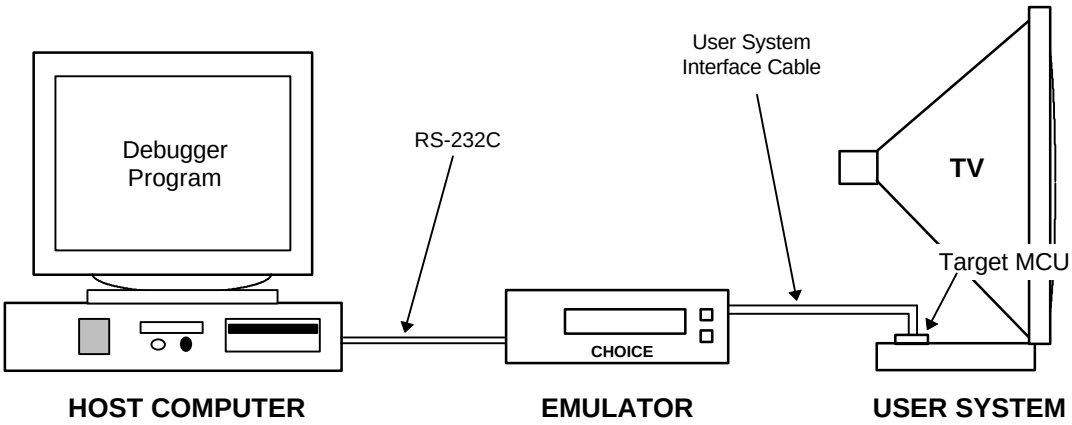
5. Support Tool

6. Appendix

5.1 EMULATOR { Ref. GMS800 SERIES MDS MANUAL }

The CHOICE emulator is a hardware debugging tool for developing user program via the 8 bit core G8MC family from HYUNDAI MicroElectrincs Co., Ltd.

[Fig 5.1.1] Environment for developing user program



5.1.1 Configuration of Emulator

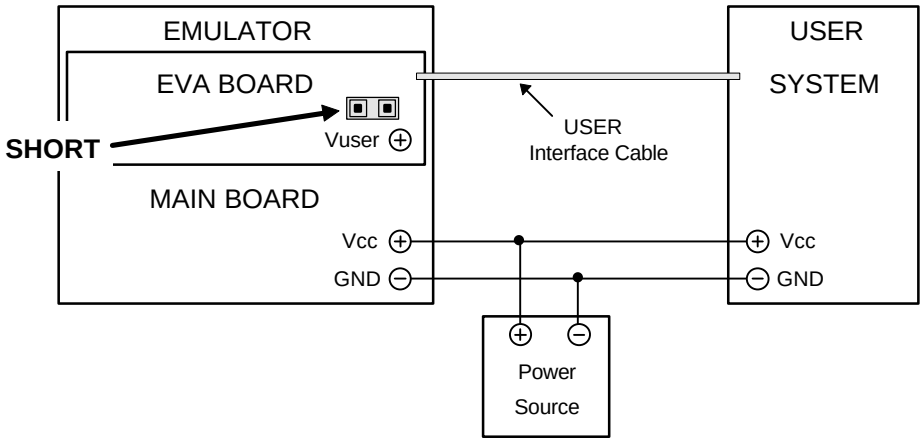
Emulator CHOICE is constructed by “ADD-ON BOARD” architecture.
MAIN Board, CONTROL Board are the base boards of CHOICE and EVA. Board is GMS84512 EVA Board for TV application.

- EVA. Chip (**GMS84512EVA :** 156 pin PGA)
  EVA Chip is special chip for the target MICOM. EVA chip supports target MICOM’s all function & includes interface logic with the emulator hardware.

5.1.2 Cautionary notes

1. When changing board at expansion slot. You must ensure that power is off.
2. Check and ensure that power for emulator & user system is supplied separately and if this is so, EVA power option jumper must be open before using.
3. Check polarity & connection location when connecting power cable, serial cable or interface cable to board.

¨ **ÎIf Emulator & USER System use same Power Source**



¨ **ÎIf Emulator & USER System use different Power Source**

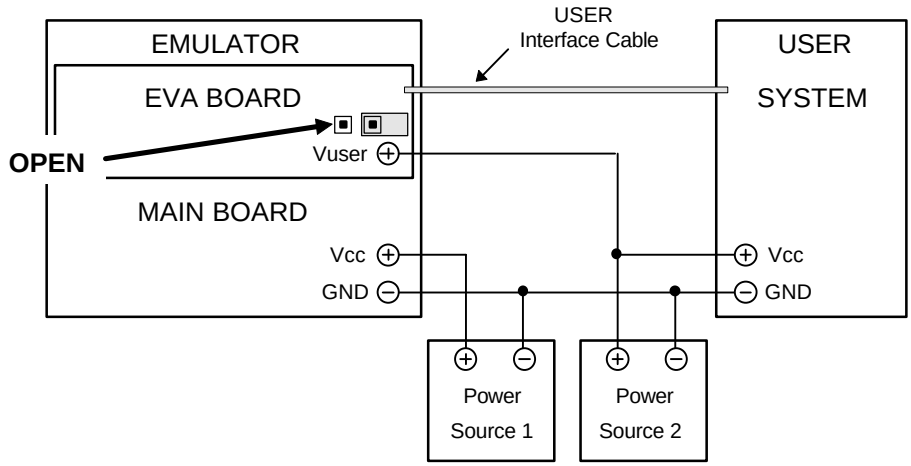


FIG 5.1.2 Connecting emulator & user system

5.1.3 USER INTERFACE SOCKET PIN ASSIGNMENT

a. ¢Ð Sign indicates unconnected pin.

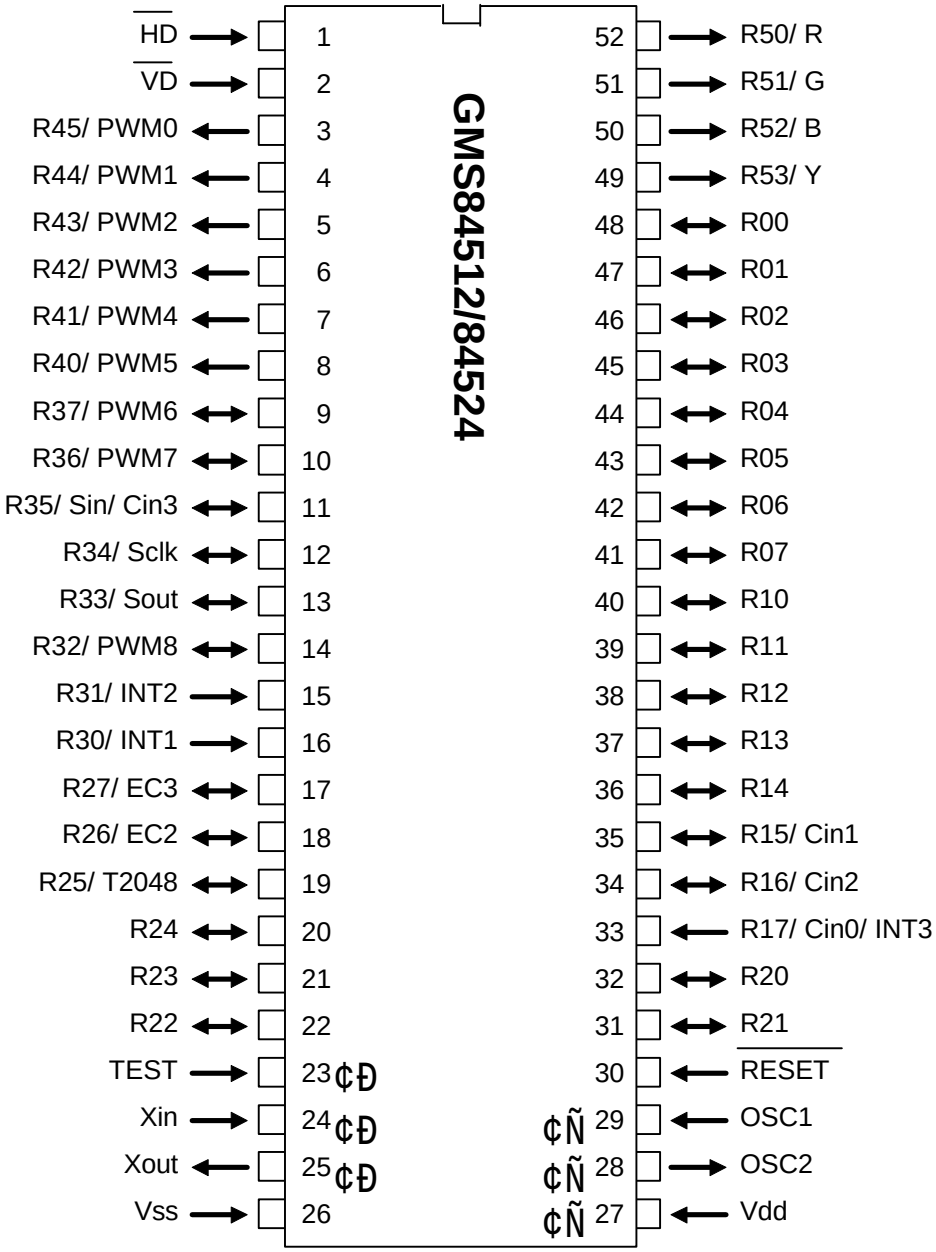


Fig 5.1.3. Pin assign of GMS84512 / 84524 interface socket

5.2 DEBUGGER { Ref. GMS800 SERIES MDS MANUAL }

The **G8MC Debugger** is a S/W tool for developing user programs for the HYUNDAI 8 bit core (G8MC family)**.**

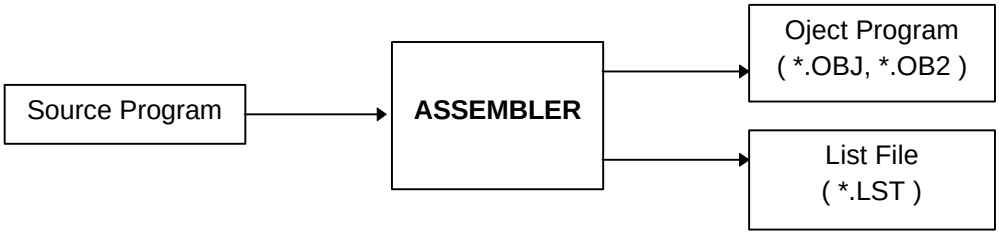
We prepared two types of debugger S/W.

One is for the MS-DOS and the other is for the MS-Windows (include MS-Win95).

5.3 ASSEMBLER { Ref. GMS800 SERIES MDS MANUAL }

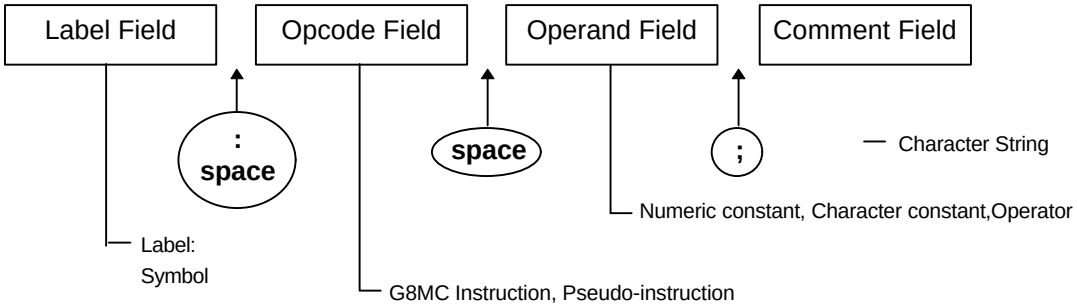
ASSEMBLER is a S/W which translates source code to object code.

Especially GMS800 Series is like with high level language.



5.3.1 Structure of Source Program List

Structure of source program list is shown below.



5.3.2 PSEUDO-INSTRUCTION

FUNCTION	**PSEUDO-INSTRUCTION**
Constant definition	**EQU (EQUAL)**
ROM data definition	**DB (DEFINE BYTE)** **DW (DEFINE WORD)**
Defining RAM symbol	**DS (DEFINE STORAGE)**
Address alteration	**ORG (ORIGIN)**
Program End	**END**
Inserting external files	**INCLUDE**
Refer to external symbols	**PUBLIC** **EXTRN (EXTERNAL)**
Outputting List files	**LIST** **NOLIST (NO LIST)** **TITLE** **PAGE**
Macro definition	**MACRO** **ENDM (END OF MACRO)**

5.3.3 STRUCTURED COMMANDS

- Assignment Statement (**=**)
- **IF** Statement

```
IF   <rel_expr>
   <statement>
ENDIF
```

```
IF   <rel_expr>
   <statement 1>
ELSE
   <statement 2>
ENDIF
```

- **FOR** Statement

```
FOR   <rel_expr>
   <statement>
NEXT
```

- **WHILE** Statement

```
DO
   <statement>
 WHILE   <rel_expr>
```

<rel_expr>
relational operator
£¼ (less)
£¾ (greater)
£¼£½(less or equal)
£¾£½(greater or equal)
£½£½(equal)
£¡£½(not equal)

- **SWITCH** Statement

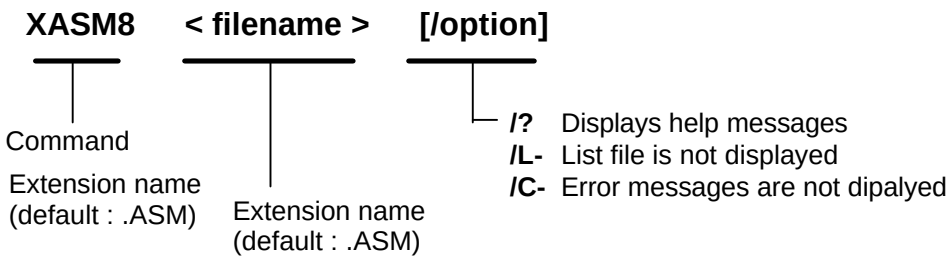
```
SWITCH <data>
    CASE <value 1>
          <statement 1>
          BREAK
    CASE <value 2>
          <statement 2>
          BREAK
            :
            :
    DEFAULT
          <statement 1>
          BREAK
ENDS
```

- **BREAK** Statement

5.3.4 Usage of Assembler

.

[syntex]



[example]

XASM8 TEST.ASM ↵

[Output files]

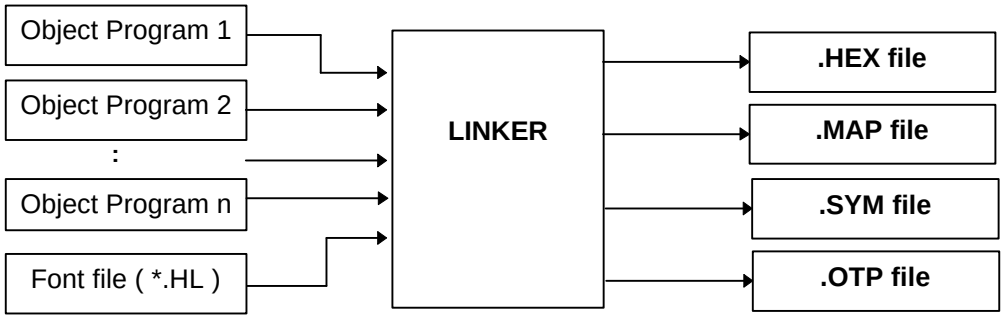
TEST.OB2 **:** Object file for OTP file

TEST.OBJ **:** Object file for HEX file

TEST.LST **:** List file

5.4 LINKER { Ref. GMS800 SERIES MDS MANUAL }

LINKER is a S/W that creates executable machine code from one or more object programs. G8MC Linker generates Motorola S-Format.



5.4.1 Usage of Linker

[syntex]

XLINK8 < filename 1> ... <filename n> [/option]

Command
Extension name
(default : . OBJ)

Extension name
(default : . OBJ)

/? Displays help messages
/M- .MAP file is not created
/S- .SYM file is not created
/CPU=<CPU_TYPE>Declaration target-MCU ROM size
/F Specify font file name

[example]

XLINK8 filename1 filename2 /F font.HL /CPU=84512

5.5 FONT EDITOR { Ref. GMS800 SERIES MDS MANUAL }

G8MC Font editor is OSD font editor for GMS84512.

MS-Windows version is supported.

Executable file name : FED8.EXE

- **OUTPUT FILES**

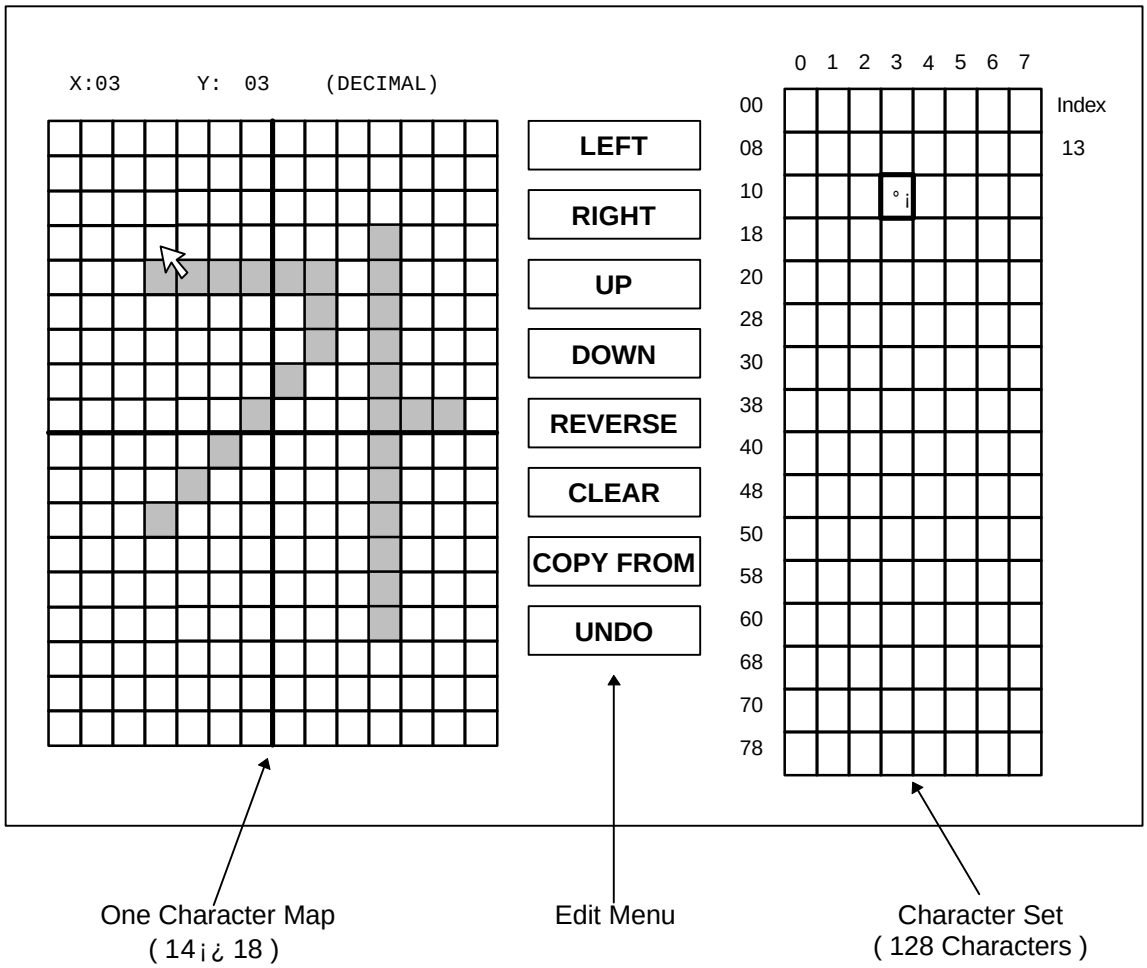
¨ç **£ª.DAT** **:** Data file for Font Editor

¨è **£ª.HL** **:** Hexa file for OTP (It is converted to otp file by link)

¨é **£ª.HI** **:** Font hexa file for emulator upper bits

¨ê **£ª.LO** **:** Font hexa file for emulator lower bits

¡ØNotice : OTP file must be converted from *.HL file by XLINK8 using by /F option.



5.6 OTP Socket Adapter (GMS84512/24 OTP-AD)

GMS84512T/84524T is equivalent with GMS84512/84524, and it include programmable ROM.

PROM writer socket adapter

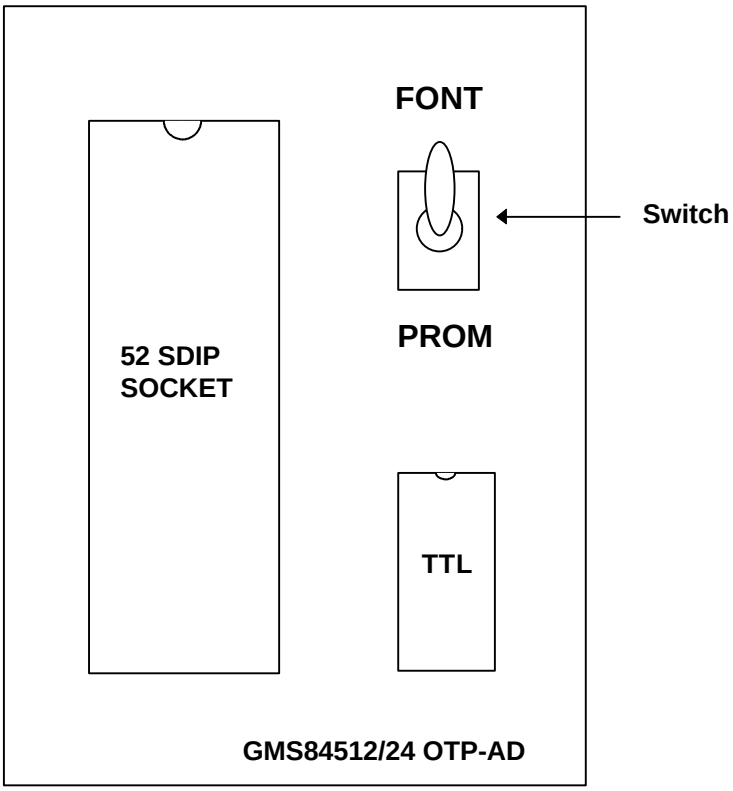
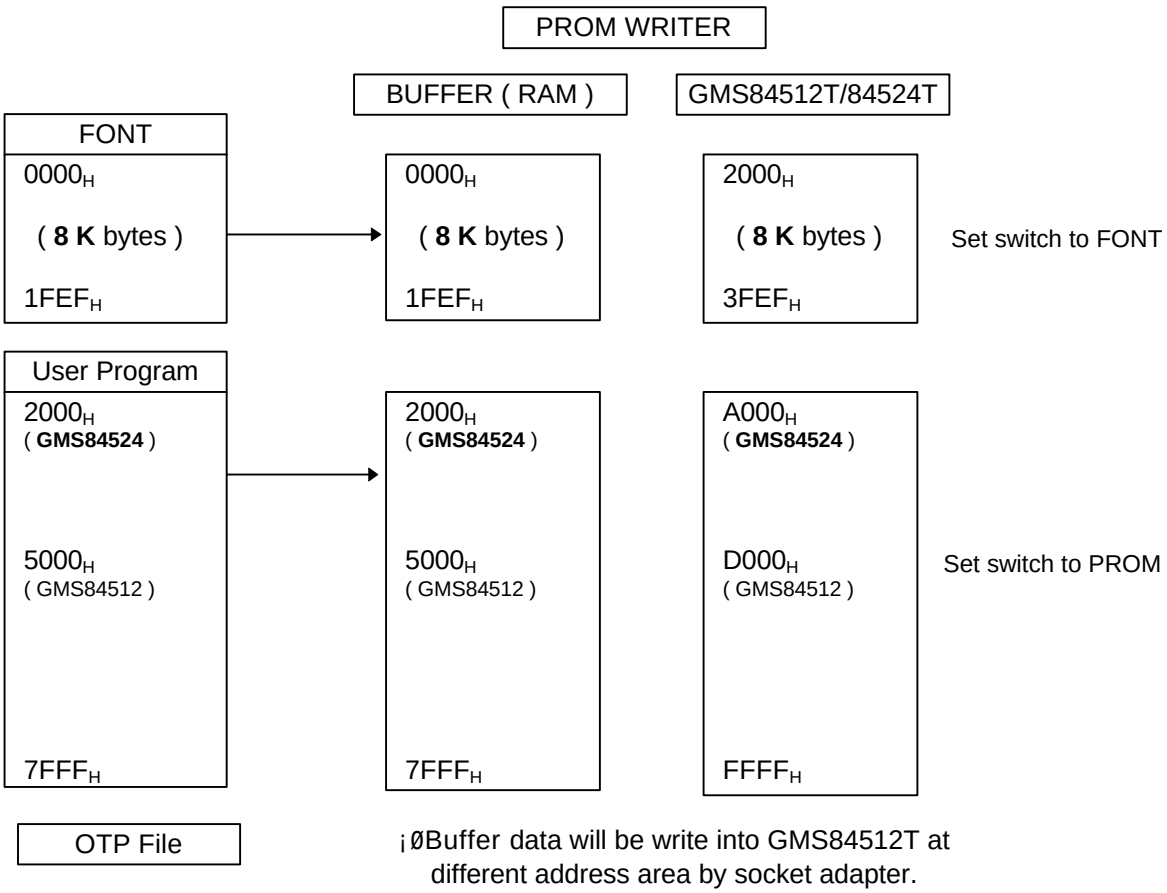


FIG.5.6.1 Top view of socker adapter

5.6.1 OTP CHIP WRITING METHOD

1. Programming Voltage (**Vpp**) is 12.5 V
2. If you write FONT ROM, then set the switch to FONT.
3. And if you write program ROM, then set the switch to PROM.
4. You must use **OTP file** not HEX file.
5. You must set EPROM type to 27C256 in PROM writer.
6. You must program OTP chip two times. Because FONT ROM and Program ROM are internally separated.



- Cautionary notes.

  For chip reliability, after write & aging, reading test is recommended.

GMS84512/84524 USER'S MANUAL

Table of Contents

1. Overview

2. CPU

3. Peripheral Function

4. Control Function

5. Support Tool

6. Appendix

1.ABSOULTE MAXIMUM RATINGS

(Ta = 25¡É)

NO.	Parameter		Symbol	Unit	Ratings	etc.
1	Supply Voltage		**Vdd**	**£Ö**	-0.3¡ -6.0	
2	Input Voltage	NMOS OPEN DRAIN	**Vin1**	**£Ö**	-0.3¡ -9.0	
		etc.	**Vin2**		-0.3¡ -Vdd+0.3	
3	Output Current	1 PORT PEAK	**Ipeak**	**§Ì**	MAX. ¦¢10¦¢	
		TOTAL	**¥ÒI avg**	**§Ì**	MAX. ¦¢50¦¢	
4	Operating temeperature		**Topr**	**¡É**	-10¡ -70	
5	Starage Temperature		**Tstg**	**¡É**	-40¡ -125	

2. DC CHARACTERISTICS (¥°)

(Vss = 0£Ö, Ta = -10¡ -70¡É, f (Xin) = 4 MHz)

NO.	Parameter	Symbol	Unit	Test Conditions	SPECIFICATION			etc.
					MIN.	TYP.	MAX.	
1	Supply Voltage	Vdd	£Ö		4.5	5	5.5	
2	Supply Voltage	Idd	mA	f(X_{IN}) =4 MHz, RESET state		10	20	
3	Supply Current in Stop Mode	Istop	§Ë	Input = Vss		10	300	*1

*1 : When port output current do not sink or source.

3. DC CHARACTERISTICS (¥±)

(Vdd = 5 V¡¾10%,Vss = 0£Ö, Ta = -10¡-70¡É,**f (Xin) = 4 §Ö)**

NO.	Parameter	Pin	Symbol	Unit	Test Condition	Specification			Etc.
						Min.	Typ.	Max.	
1	'H' Input Voltage	$\overline{\text{RESET}}$, TEST, Xin, Schmitt *1 Input	Vih	£Ö		0.8 Vdd		Vdd	
		Other Port *2				0.7 Vdd		Vdd	
2	'L' Input Voltage	$\overline{\text{RESET}}$, TEST, Xin, Schmitt *1 Input		£Ö		0		0.12 Vdd	
		Other Port *2				0		0.3 Vdd	
3	'H' Input Current	All input pins	Iih	§Ë	Vi = Vdd	-5.0		5.0	
4	'L' Input Current	All input pins	Iil	§Ë	Vi = Vss	-5.0		5.0	
5	Hysteresis	Schmitt *1 Input	Vt+ — Vt-	£Ö		0.3		1.0	
		$\overline{\text{RESET}}$				0.2		0.7	
6	'H' Output Voltage	R0, R10¡-R16, R2, R33¡-R35, R5	Voh	£Ö	Ioh = -5 §Ì	Vdd -1			Fig. A.1 Ref.
7	'L' Output Voltage	R0, R10¡-R16, R2, R32¡-R37, R4, R5	Vol	£Ö	Iol = 5 §Ì			1.0	Fig. A.1 Ref.
8	RAM DATA RETENSION	Vdd	Vram	£Ö	At Clock Stop	2.0			

*1 . Schmitt Input : EC3, EC2, Sin, Sclk, INT1¡-INT3, $\overline{\text{HD}}$, $\overline{\text{VD}}$

*2. Other Ports : R0, R1, R2, R3

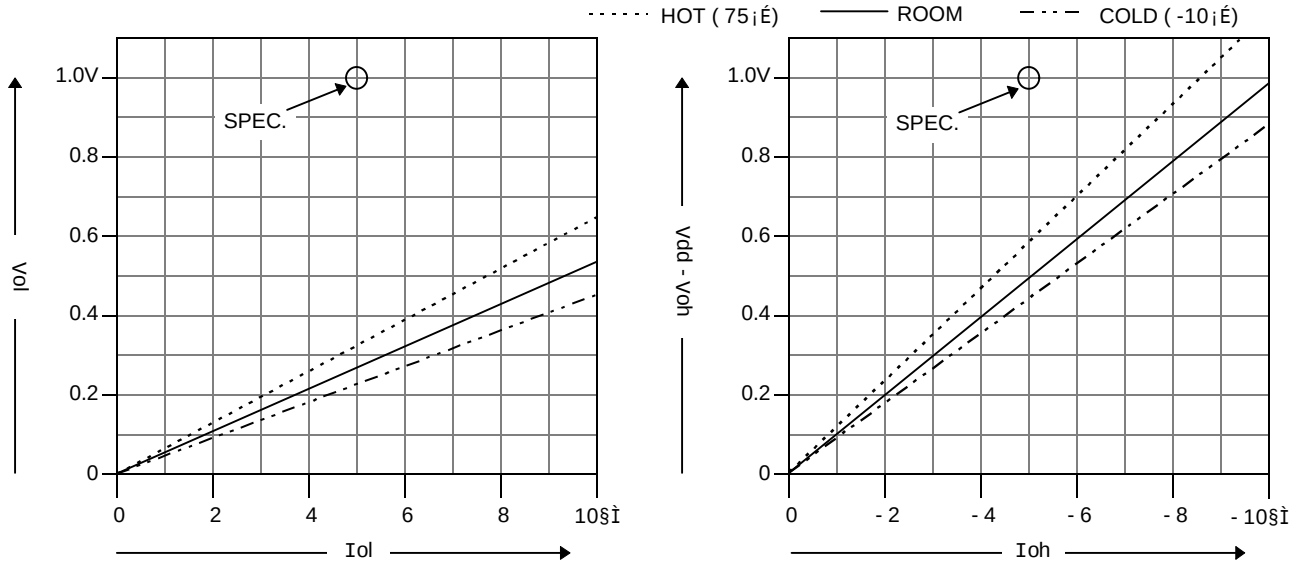


Fig. A.1 PORT Output Characteristics (@ Vdd = 5 V)

4. A/D COMPARATOR CHARACTERISTICS

(Vdd = 5 V¡¾10%,Vss = 0£Ö, Ta = -10¡-70¡É,**f (Xin) = 4 §Ô)**

NO.	Parameter	Pin	Symbol	Unit	SPECIFICATION			etc.
					MIN.	TYP.	MAX.	
1	Analog input voltage rage	Cin0¡-Cin3	VAIN	V	Vss		Vdd	
2	Accuracy			LSB			¡¾1	

5. AC CHARACTERISTICS

5.1 Input Conditions (MAIN CLOCK, RESET, INT, EC, OSD CLOCK, VD, HD)

(Vdd = 5 V¡¾10%,Vss = 0£Ö, Ta = -10¡-70¡É,**f (Xin) = 4 §Ô)**

NO.	Parameter	Pin	Symbol	Unit	SPECIFICATION			etc.
					MIN.	TYP.	MAX.	
1	Main Clock Frequency	Xin	fcp	MHz	3	4	5	
2	System Clock Cycle		tsys	ns	400	500	667	
3	Oscillation Stable Time	Xin, Xout	tST	ms			20	
4	External Clock Pulse Width	Xin	tcpw	ns	80			
5	External Clock Transition Timer	Xin	trcp, tfcp	ns	20			
6	Interrupt Pulse Width	INT1~INT3	tIW	tsys	2			
8	'L' Reset Input Pulse Width	$\overline{\text{RESET}}$	tRST	tsys	8			
9	Event Counter Pulse Width	EC2, EC3	tECW	tsys	2			
10	Event Counter Transition Timer	EC2, EC3	trEC, tfEC	ns	20			
11	Osd Clock Frequency	OSC1, OSC2	f_{OSD}	MHz	4	6	8	L-C Oscillator
12	V-sync Pulse Width	$\overline{\text{VD}}$	t_{VDW}	§Á	2			
13	H-sync Pulse Width	$\overline{\text{HD}}$	t_{HDW}	§	2			

¡ØFig. A-2 Ref.

5.2 SERIAL TRANSFER

(Vdd = 5 V¡¾10%,Vss = 0£Ö, Ta = -10¡ -70**¡É,f (Xin) = 4 §Ô)**

NO.	Parameter	Pin	Symbol	Unit	Test Condition	SPECIFICATION			etc.
						MIN.	TYP.	MAX.	
1	Serial Input Clock Cycle	Sclk	tscyc	§À		2tsys+ 200			
2	Serial Input Clock Pulse Width	Sclk	tsck	§À		tsys +70			
3	Serial Input Clock Transition Time	Sclk	trsck, tfsck	§À				30	
4	Serial Input Data Transition Time	Sin	trsin, tfsin	§À				30	
5	Serial Input Data Set-Up Time	Sin	tsus	§À	External Sclk	100			
					Internal Sclk	200			
6	Serial Input Data Hold Time	Sin	ths	§À	External Sclk	tsys +100			
7	Serial Output Clock Cycle	Sclk	tscyc	§À	LOAD = 50 §Ü	4tsys		16 tsys	
8	Serial Output Clock Pulse Width	Sclk	tsck	§À		2tsys - 30			
9	Serial Output Transition Time	Sclk	trsck, tfsck	§À				30	
10	Serial Output Data Delay Time	Sout	tds	§À				100	

¡ØFig. A-3 Ref.

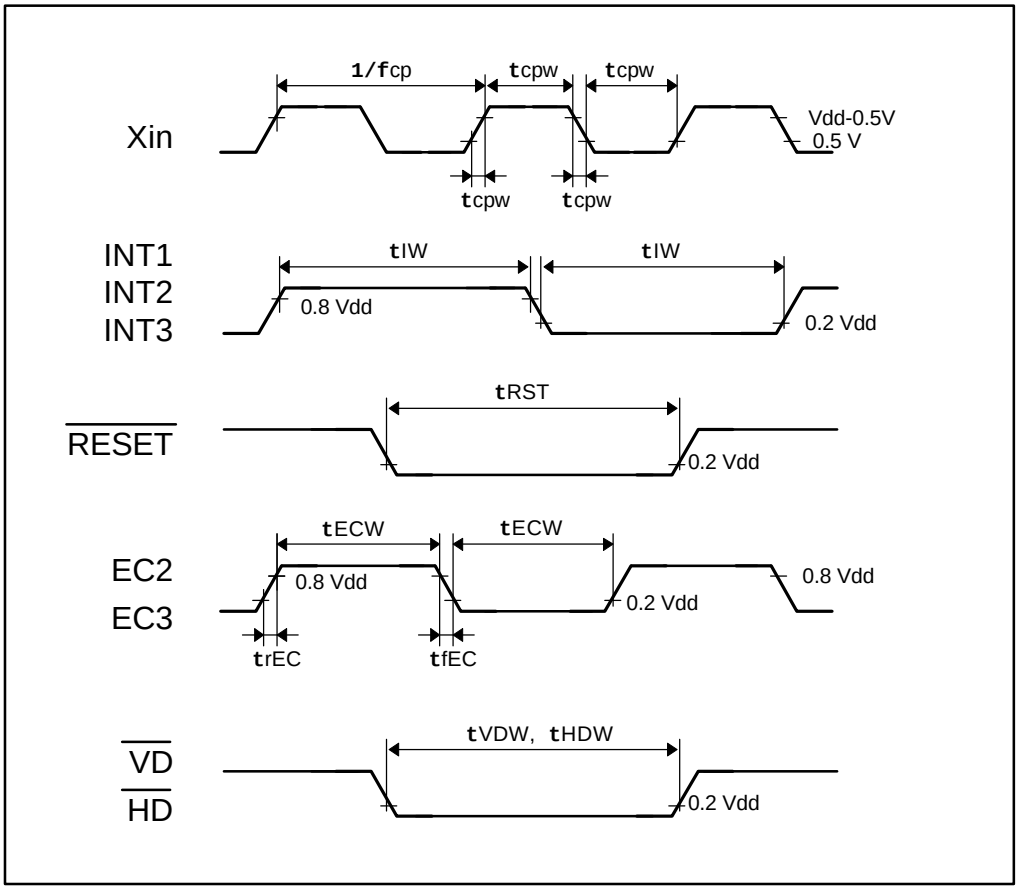


Fig. A.2 CLOCK, INT, RESET, EC, VD, HD Input Timing

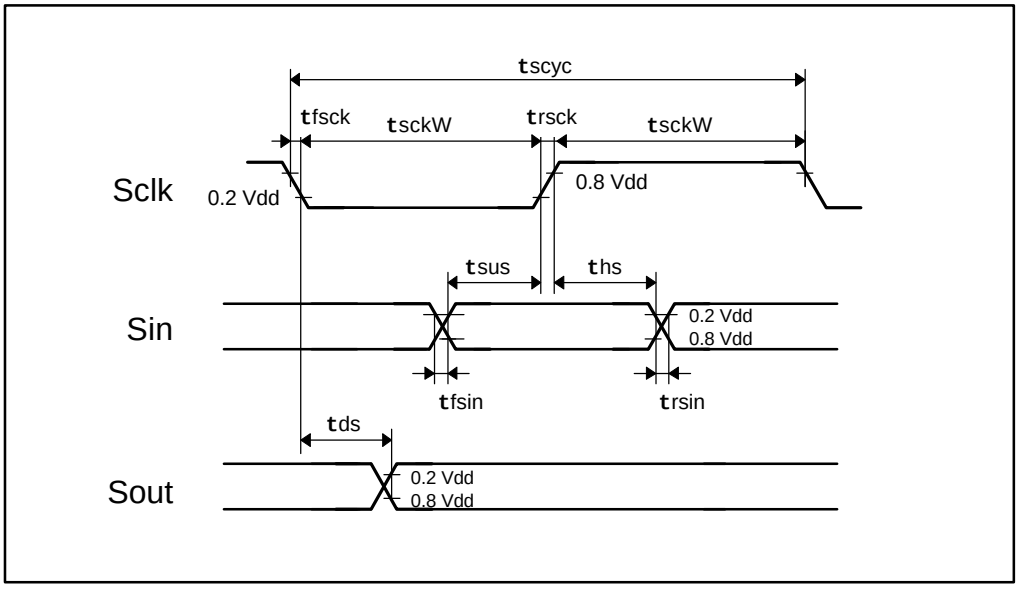


Fig. A.3 SERIAL I/O Timing

GMS84512 PACKAGE OUTLINE (52-PIN SDIP)

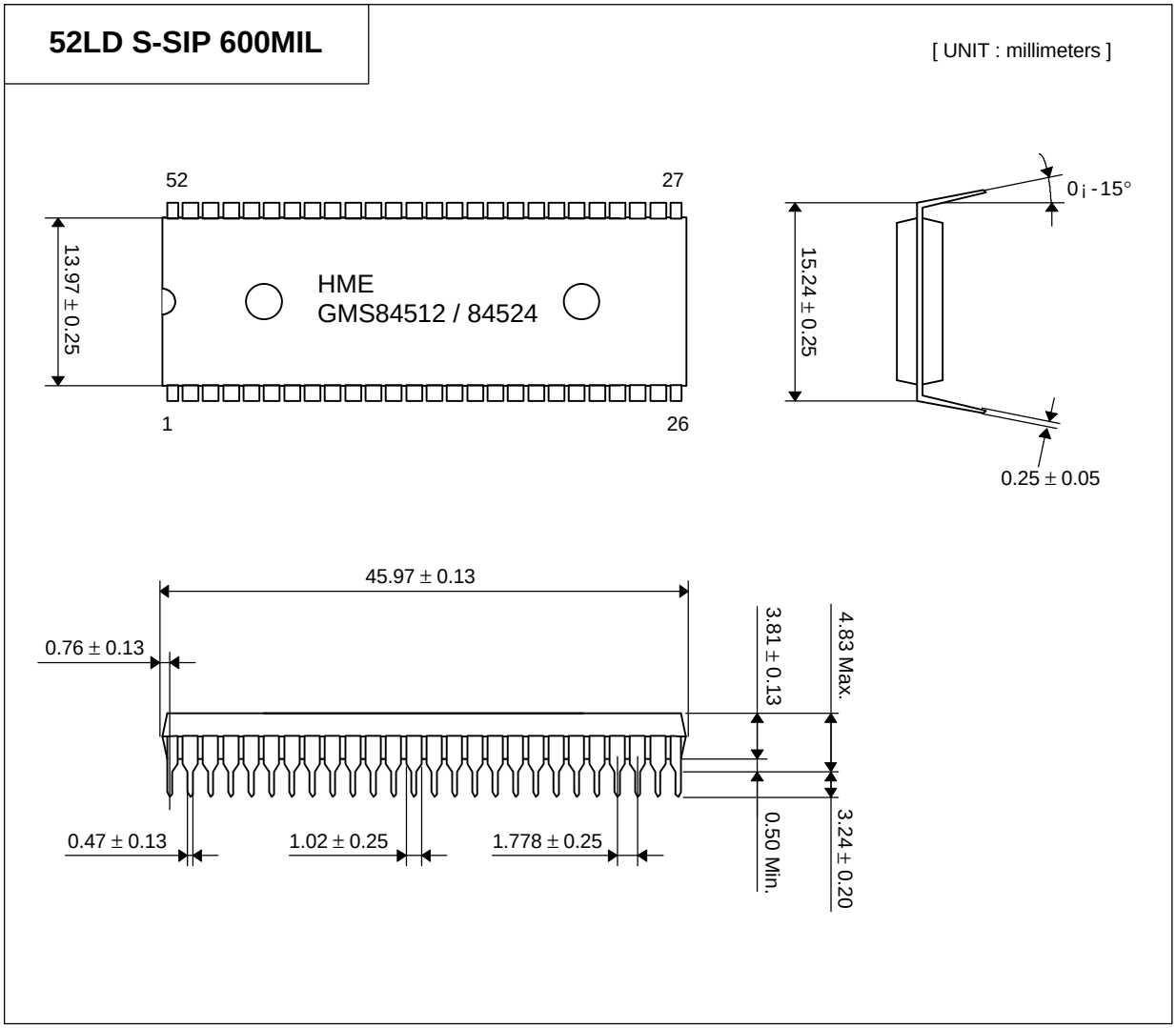


Fig. A.4 52-PIN S-DIP PACKAGE DIMENSION

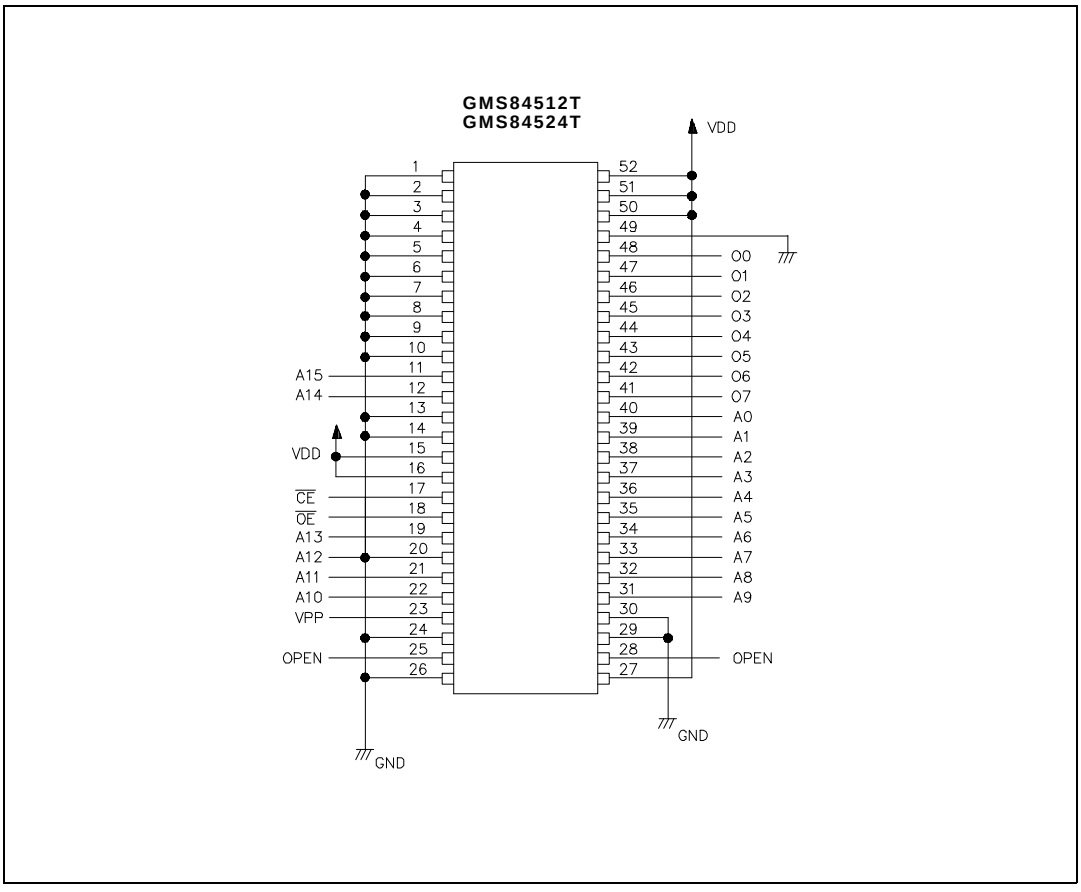
GMS84512T/24T PROGRAMMING MANUAL

1. DEVICE OVERVIEW

The GMS84512T/GMS84524T are high-performance CMOS 8-bit microcontroller with 12K/24K bytes of EPROM. The device is one of GMS800 family. The HYUNDAI GMS84512T/GMS8424T are powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. The GMS84512T/GMS84524T provides the following standard features: 12K/24K bytes of EPROM, 256 bytes of RAM, 42 I/O lines, 16-bit or 8-bit timer/counter, On Screen Display, 5-bit A/D comparotor, PWM, on-chip oscillator and clock circuitry.

DEVICE NAME	ROM SIZE
GMS84512T	12K PROM
GMS84524T	24K PROM

2. PIN CONFIGURATION



52SDIP Package for GMS84512T/GMS84524T

Pin No.	MCU Mode		OTP Mode	
1	$\overline{HD}$	I	(1)	I
2	$\overline{VD}$	I	(1)	I
3	R45/PWM0	O	(1)	O
4	R44/PWM1	O	(1)	O
5	R43/PWM2	O	(1)	O
6	R42/PWM3	O	(1)	O
7	R41/PWM4	O	(1)	O
8	R40/PWM5	O	(1)	O
9	R37/PWM6	I/O	(1)	I
10	R36/PWM7	I/O	(1)	I
11	R35/Sin/Cin3	I/O	A15	I
12	R34/Sclk	I/O	A14	I
13	R33/Sout	I/O	(1)	I
14	R32/PWM8	I/O	(1)	I
15	R31/INT2	I	(2)	I
16	R30/INT1	I	(2)	I
17	R27/EC3	I/O	$\overline{CE}$	I
18	R26/EC2	I/O	$\overline{OE}$	I
19	R25/T2048	I/O	A13	I
20	R24	I/O	A12	I
21	R23	I/O	A11	I
22	R22	I/O	A10	I
23	TEST	I	V_{PP}	-
24	X_{IN}	I	(1)	I
25	X_{OUT}	O	(3)	O
26	V_{SS}	I	V_{SS}	-

Pin No.	MCU Mode		OTP Mode	
27	V_{DD}	I	V_{DD}	-
28	OSC2	O	(3)	O
29	OSC1	I	(1)	I
30	$\overline{RESET}$	I	(1)	I
31	R21	I/O	A9	I
32	R20	I/O	A8	I
33	R17/Cin0/INT3	I	A7	I
34	R16/Cin2	I/O	A6	I
35	R15/Cin1	I/O	A5	I
36	R14	I/O	A4	I
37	R13	I/O	A3	I
38	R12	I/O	A2	I
39	R11	I/O	A1	I
40	R10	I/O	A0	I
41	R07	I/O	O7	I/O
42	R06	I/O	O6	I/O
43	R05	I/O	O5	I/O
44	R04	I/O	O4	I/O
45	R03	I/O	O3	I/O
46	R02	I/O	O2	I/O
47	R01	I/O	O1	I/O
48	R00	I/O	O0	I/O
49	R53/Y	O	(1)	O
50	R52/B	O	(2)	O
51	R51/G	O	(2)	O
52	R50/R	O	(2)	O

NOTES:

(1) These pins must be connected to V_{SS}, because these pins are input ports during programming, program verify and reading
(2) These pins must be connected to V_{DD}.
(3) X_{OUT} pin must be opened during programming.

I/O: Input/Output Pin
I: Input Pin
O: Output Pin

3. PIN FUNCTION (OTP Mode)

V_{PP} (Program Voltage)

V_{PP} is the input for the program voltage for programming the EPROM.

$\overline{CE}$ (Chip Enable)

CE is the input for programming and verifying internal EPROM.

$\overline{OE}$ (Output Enable)

OE is the input of data output control signal for verify.

A_0~A_{15} (Address Bus)

A_0~A_{15} are address input pins for internal EPROM.

O_0~O_7 (EPROM Data Bus)

These are data bus for internal EPROM.

4. PROGRAMMING

The GMS84512T/GMS84524T has two address ranges, Font area and PGM area. Therefore, the programmer should be program twice, each Font and PGM.

4.1 GMS84512T Programming

The Font address ranges are from 0000_H to $1FF1_H$ and PGM addres ranges are from 5000_H to $7FFF_H$ in OTP file. Firstly, the programmer program the Font data into the GMS84512T OTP device (from 2000_H to $3FF1_H$), and then program the PGM data into the GMS84512T OTP device (from $D000_H$ to $FFFF_H$).
When the programmer write the Font data 0000_H to $1FF1_H$, consequently, the data actually will be written into address 2000_H to $3FF1_H$of the GMS84512T OTP device. And PGM data 5000_H to $7FFF_H$, consequently, the data actually will be written into address $D000_H$ to $FFFF_H$of the GMS84512T OTP device.

1. The data format to be programmed is made up of Motorola S1 format.

Ex) "Motorola S1" format;

```
S00800006D633634615C
S1130800FFFFFCF0E3E7E7E7E7E7E7E7E7E3F0FC10
S1051800FFFFE4
S1130000FFFF9F87E3F3F3F3F3F3F3F3F3E3879F44
:
S1051FF0FFBF2D
S11307F0FEFFFFFFFFFFFFFFFFFFFFFFFFFFFF06
S10517F0FFFEF6
S12250009FE1C0711B003F1B003E1B043D1BF83C1B213B1BD13A1B00391B63381BC037C5
S123501F1BF0361B011B1BFB1A1B40053BFF1A061A2F1A2E1B232D1B922C1B8F2B1B002A1C
:
S1217FE020012001200120012001200120012001E6012001DA021803BB0320010A1420017B
S1057FFEFF2F4F
S9030000FC
```

FONT
0000_H ~ $1FF1_H$

PGM
5000_H ~ $7FFF_H$

2. Down load above data into programmer from PC.

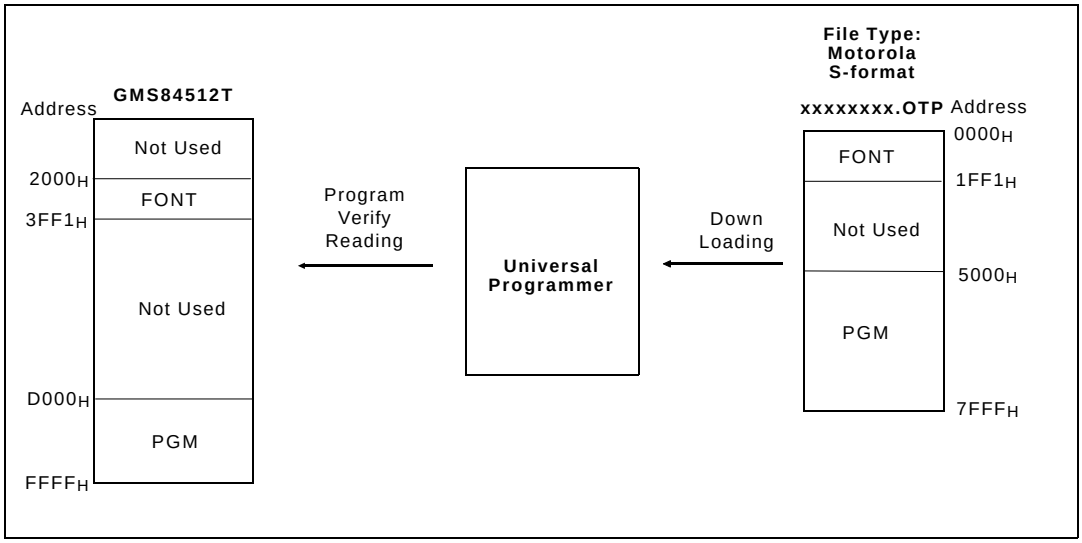
3. Programming the data from address 0000_H to $1FF1_H$ and 5000_H to $7FFF_H$ into OTP MCU, the data must be turned over respectively, and then record the data.

Ex) 00→FF, 76→89, FF→00 etc.

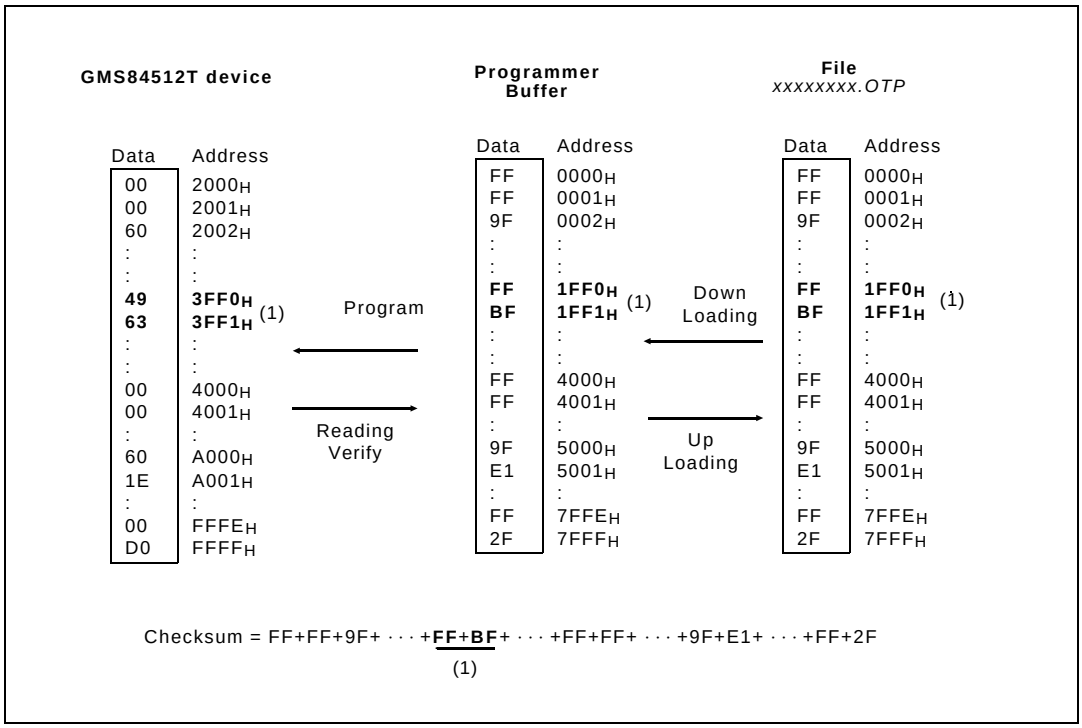
4. Of course, the check sum is result of the sum of whole data from address 0000_H to $7FFF_H$ in the file (not reverse data of OTP MCU).

Caution : In the OTP file, data **FF_H, BF_H** are written in **$1FF0_H$ and $1FF1_H$** respectively. However, the OTP chip was written already **49_H, 63_H** in that addresses. Therefore, **that addresses should be skipped in programming or blank check.** And **should be treated FF_H, BF_H in checksum calculation (READ or VERIFY).**
On the checksum calculation, **not used area data should be regarded as FF_H in OTP file**.
When GMS84512T shipped, the blank data of GMS84512T is initially 00_H (not FF_H).

Programming Flow



Programming Example



4.2 GMS84524T Programming

The Font address ranges are from 0000_H to $1FF1_H$ and PGM addres ranges are from 2000_H to $7FFF_H$ in OTP file. Firstly, the programmer program the Font data into the GMS84524T OTP device (from 2000_H to $3FF1_H$), and then program the PGM data into the GMS84524T OTP device (from $A000_H$ to $FFFF_H$).
When the programmer write the Font data 0000_H to $1FF1_H$, consequently, the data actually will be written into address 2000_H to $3FF1_H$of the GMS84524T OTP device. And PGM data 2000_H to $7FFF_H$, consequently, the data actually will be written into address $A000_H$ to $FFFF_H$of the GMS84524T OTP device.

1. The data format to be programmed is made up of Motorola S1 format.

Ex) "Motorola S1" format;

```
S00700006D61696E53
S1130800FFFFFCF0E3E7E7E7E7E7E7E7E7E3F0FC10
S1051800FFFFE4
S1130000FFFF9F87E3F3F3F3F3F3F3F3F3E3879F44
:
:
S1130FF0BFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3D
S1051FF0FFBF2D
S12220009FE1C0711B003F1B003E1B183D1B803C1B283B1BD33A1B00391B63381BFF3711
S123201F1BF0361BFC1B1BFB1A1B40051B0F061BFF433A415B00BA4A8FE9BBE72FED3A4083
:
:
S1217FE0670B670B670BFF5F670B670B670BCE0C670BFA0BB30BA00D670B991F670B1D
S1057FFEFF5F1F
S9030000FC
```

FONT
0000_H ~ $1FF1_H$

PGM
2000_H ~ $7FFF_H$

2. Down load above data into programmer from PC.

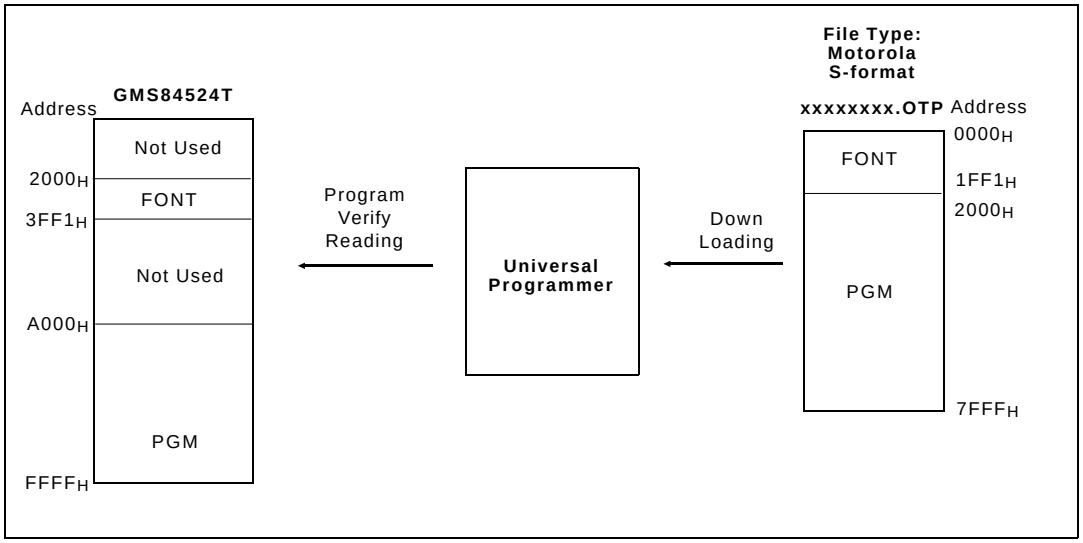
3. Programming the data from address 0000_H to $1FF1_H$ and 2000_H to $7FFF_H$ into OTP MCU, the data must be turned over respectively, and then record the data.

Ex) 00→FF, 76→89, FF→00 etc.

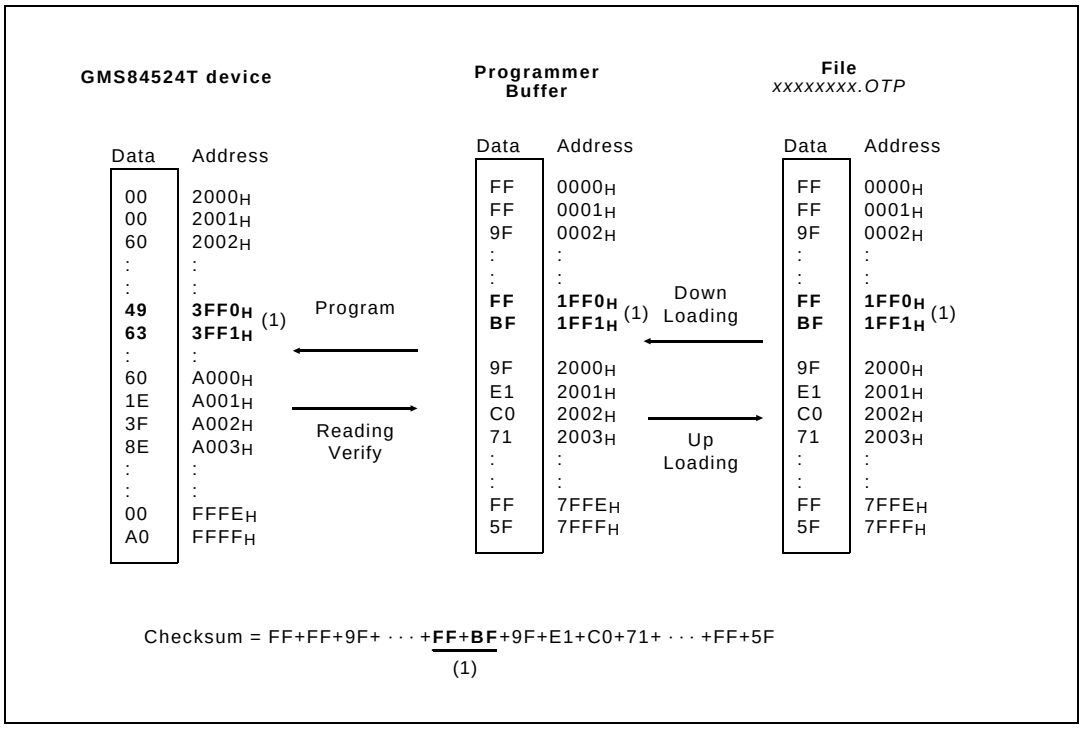
4. Of course, the check sum is result of the sum of whole data from address 0000_H to $7FFF_H$ in the file (not reverse data of OTP MCU).

Caution : In the OTP file, data **FF_H, BF_H** are written in **$1FF0_H$ and $1FF1_H$** respectively. However, the OTP chip was written already **49_H, 63_H** in that addresses. Therefore, **that addresses should be skipped in programming or blank check.** And **should be treated FF_H, BF_H in checksum calculation (READ or VERIFY).**
On the checksum calculation, **not used area data should be regarded as FF_H in OTP file**.
When GMS84524T shipped, the blank data of GMS84524T is initially 00_H (not FF_H).

Programming Flow



Programming Example



5. DEVICE OPERATION MODE

(T_A = 25°C ±5°C)

Mode	$\overline{CE}$	$\overline{OE}$	A0~A15	VPP	VDD	O0~O7
Read	X		X	V_{DD}	5.0V	D_{OUT}
Output Disable	V_{IH}	V_{IH}	X	V_{DD}	5.0V	Hi-Z
Programming	V_{IL}	V_{IH}	X	V_{PP}	V_{DD}	D_{IN}
Program Verify	X		X	V_{PP}	V_{DD}	D_{OUT}

NOTES:
1. X = Either V_{IL} or V_{IH}
2. See DC Characteristics Table for V_{DD} and V_{PP} voltages during programming.

6. DC CHARACTERISTICS

(V_{SS}=0 V, T_A = 25°C ±5°C)

Symbol	Item	Min	Typ	Max	Unit	Test condition
V_{PP}	Intelligent Programming	12.0	-	13.0	V	
V_{DD}(1)	Intelligent Programming	5.75	-	6.25	V	
I_{PP} (2)	V_{PP} supply current			50	mA	$\overline{CE}$=V_{IL}
I_{DD} (2)	V_{DD} supply current			30	mA	
V_{IH}	Input high voltage	0.8 V_{DD}			V	
V_{IL}	Input low voltage			0.2 V_{DD}	V	
V_{OH}	Output high voltage	V_{DD}-1.0			V	I_{OH} = -2.5 mA
V_{OL}	Output low voltage			0.4	V	I_{OL} = 2.1 mA
I_{IL}	Input leakage current			5	uA	

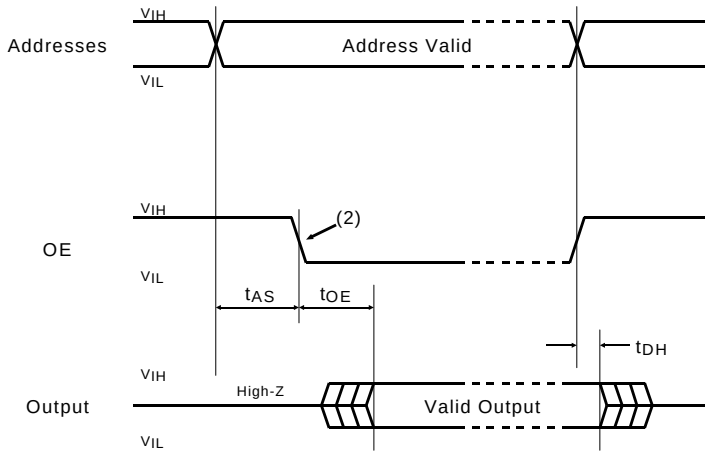
NOTES:

1. V_{DD} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The maximum current value is with outputs O_0 to O_7 unloaded.

SWITCHING WAVEFORMS

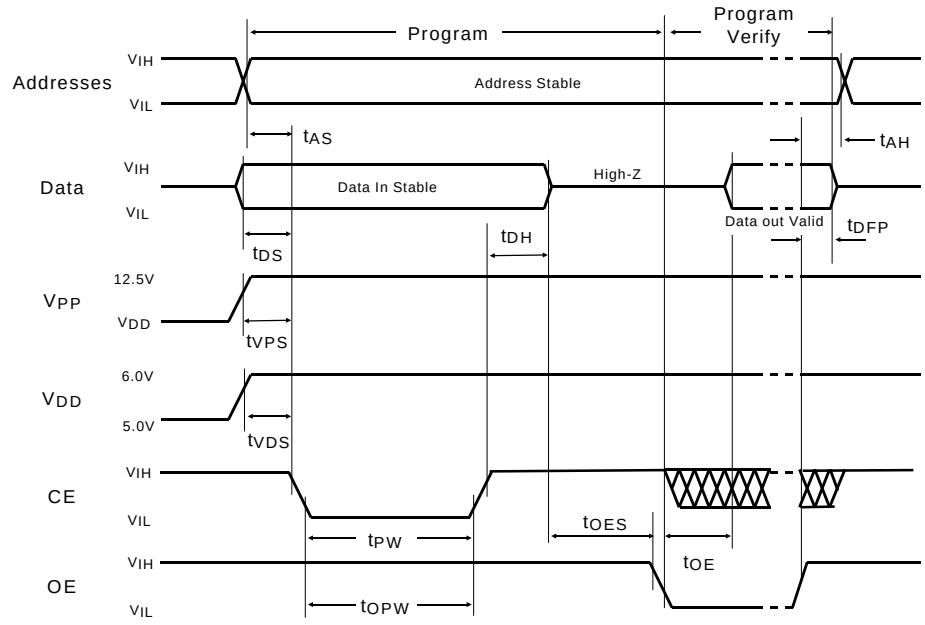
WAVEFORM	INPUTS	OUTPUTS
	Must be steady	Will be steady
	May change from H to L	Will be changing from H to L
	May change from L to H	Will be changing from L to H
	Do not care any change permitted	Changing state unknown
	Does not apply	Center line is high impedance "Off" state

READING WAVEFORMS



NOTES:
1. The input timing reference level is 1.0 V for a V_{IL} and 4.0V for a V_{IH} at V_{DD}=5.0V
2. To read the output data, transition requires on the $\overline{OE}$ from the high to the low after address setup time t_{AS}.

PROGRAMMING ALGORITHM WAVEFORMS



NOTES:

1. The input timing reference level is 1.0 V for a V_{IL} and 4.0V for a V_{IH} at V_{DD}=5.0V

7. AC READING CHARACTERISTICS

(V_{SS}=0 V, T_A = 25°C ±5°C)

Symbol	Item	Min	Typ	Max	Unit	Test condition
t_{AS}	Address setup time	2			us	
t_{OE}	Data output delay time			200	ns	
t_{DH}	Data hold time	0			ns	

NOTES:

1. V_{DD} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.

8. AC PROGRAMMING CHARACTERISTICS

(V_{SS}=0 V, T_A = 25°C ±5°C; See DC Characteristics Table for V_{DD} and V_{PP} voltages.)

Symbol	Item	Min	Typ	Max	Unit	Condition* (Note 1)
t_{AS}	Address set-up time	2			us	
t_{OES}	$\overline{OE}$ set-up time	2			us	
t_{DS}	Data setup time	2			us	
t_{AH}	Address hold time	0			us	
t_{DH}	Data hold time	1			us	
t_{DFP}	Output disable delay time	0			us	
t_{VPS}	V_{PP} setup time	2			us	
t_{VDS}	V_{DD} setup time	2			us	
t_{PW}	Program pulse width	0.95	1.0	1.05	ms	Intelligent
t_{OPW}	$\overline{CE}$ pulse width when over programming	2.85		78.75	ms	(Note 2)
t_{OE}	Data output delay time			200	ns	

*AC CONDITIONS OF TEST
Input Rise and Fall Times (10% to 90%) . . . 20 ns
Input Pulse Levels . . . . . 0.45V to 4.55V
Input Timing Reference Level . . . . . 1.0V to 4.0V
Output Timing Reference Level . . . . . 1.0V to 4.0V

NOTES:

1. V_{DD} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse may vary from 2.85 msec to 78.75 msec as a function of the iteration counter value X (Intelligent Programming Algorithm only). Refer to page 13.

Intelligent Programming Algorithm