
GMS81508A

GMS81516A

USER'S MANUAL

Revision History

Rev 2.2 (Dec. 1998)

Add the package dimension for 64LQFP on page 3-1, 4-1.

Rev 2.1 (Nov. 1998)

Operating Temperature, -10~75°C is extended to -20~85°C.

Add the unused port guidance on page 55.

Correct errata for opcode of "EOR [dp+X], EOR [dp]+Y, EOR {X}" in "Instruction Set".

Add the OTP device programming guidance, recommend using "Intelligent Mode".

Add the chapter for OTP programming manual as an appendix.

Rev 2.0 (Sep. 1997)

- CONTENTS -

| | |
|--|----------|
| 1. OVERVIEW | 1 |
| 1.1. FEATURES | 1 |
| 1.2. BLOCK DIAGRAM..... | 2 |
| 1.3. PIN ASSIGNMENT | 3 |
| 1.4. PACKAGE DIMENSION | 4 |
| 1.5. PIN DESCRIPTION..... | 5 |
| 2. FUNCTIONS | 7 |
| 2.1. REGISTERS | 7 |
| 2.1.1. A - Register..... | 8 |
| 2.1.2. X- Register..... | 8 |
| 2.1.3. Y- Register | 8 |
| 2.1.4. Stack Pointer | 8 |
| 2.1.5. Program Counter..... | 10 |
| 2.1.6. Program Status Word..... | 10 |
| 2.2. MEMORY SPACE..... | 12 |
| 2.2.1. RAM area | 12 |
| 2.2.2. Peripheral Register area | 12 |
| 2.2.3. Program ROM area | 12 |
| 2.2.4. Peripheral Register List | 14 |
| 2.3. CLOCK GENERATION CIRCUIT | 16 |
| 2.3.1. Oscillation Circuit | 16 |
| 2.3.2. Prescaler | 17 |
| 2.4. BASIC INTERVAL TIMER..... | 18 |
| 2.4.1. Control of Basic Interval Timer..... | 18 |
| 2.5. WATCH DOG TIMER..... | 19 |
| 2.5.1. Control of Watch Dog Timer | 19 |
| 2.5.2. The output of WDT signal..... | 20 |
| 2.6. TIMER..... | 21 |
| 2.6.1. Control of Timer | 23 |
| 2.6.2. Interval Timer..... | 24 |
| 2.6.3. Event Counter..... | 24 |
| 2.6.4. Pulse Output..... | 24 |
| 2.6.5. Input Capture..... | 24 |
| 2.7. EXTERNAL INTERRUPT..... | 26 |
| 2.8. A/D CONVERTER | 27 |
| 2.8.1. Control of A/D Converter..... | 27 |
| 2.9. SERIAL I/O | 29 |
| 2.9.1. Data Transmission/Receiving Timing | 31 |
| 2.9.2. The Serial I/O operation by Srdy pin | 31 |
| 2.9.3. The method of Serial I/O | 32 |
| 2.9.4. The Method to Test Correct Transmission with S/W..... | 32 |
| 2.10. PWM | 33 |
| 2.10.1. Controls of PWM | 33 |
| 2.11. BUZZER DRIVER..... | 35 |
| 2.11.1. Buzzer Driver Operation | 36 |
| 2.12. INTERRUPTS..... | 37 |
| 2.12.1. Interrupt Circuit Configuration and Kinds..... | 37 |
| 2.12.2. Interrupt Control..... | 38 |
| 2.12.3. Interrupt Priority | 39 |

| | |
|--|-----------|
| 2.12.4. <i>Interrupt Sequence</i> | 40 |
| 2.12.5. <i>Software Interrupt</i> | 41 |
| 2.12.6. <i>Multiple Interrupt</i> | 42 |
| 2.13. STANDBY FUNCTION | 44 |
| 2.13.1. <i>STOP Mode</i> | 45 |
| 2.13.2. <i>STOP Mode Release</i> | 45 |
| 2.14. RESET FUNCTION | 47 |
| 3. I/O PORTS..... | 48 |
| 3.1. R0 PORT | 48 |
| 3.2. R1 PORT | 49 |
| 3.3. R2 PORT | 50 |
| 3.4. R3 PORT | 51 |
| 3.5. R4 PORT | 52 |
| 3.6. R5 PORT | 53 |
| 3.7. R6 PORT | 54 |
| 3.8. TERMINAL TYPES | 56 |
| 4. ELECTRICAL CHARACTERISTICS..... | 60 |
| 4.1. ABSOLUTE MAXIMUM RATINGS..... | 60 |
| 4.2. RECOMMENDED OPERATING CONDITIONS..... | 60 |
| 4.3. A/D CONVERTER CHARACTERISTICS | 60 |
| 4.4. DC CHARACTERISTICS | 61 |
| 4.5. AC CHARACTERISTICS | 62 |
| 4.5.1. <i>Input Conditions</i> | 62 |
| 4.5.2. <i>Serial Transfer</i> | 63 |
| 4.5.3. <i>Microprocessor Mode I/O Timing</i> | 64 |
| 4.5.4. <i>Bus Holding Timing</i> | 65 |
| 5. INSTRUCTION SET | 66 |

1. OVERVIEW

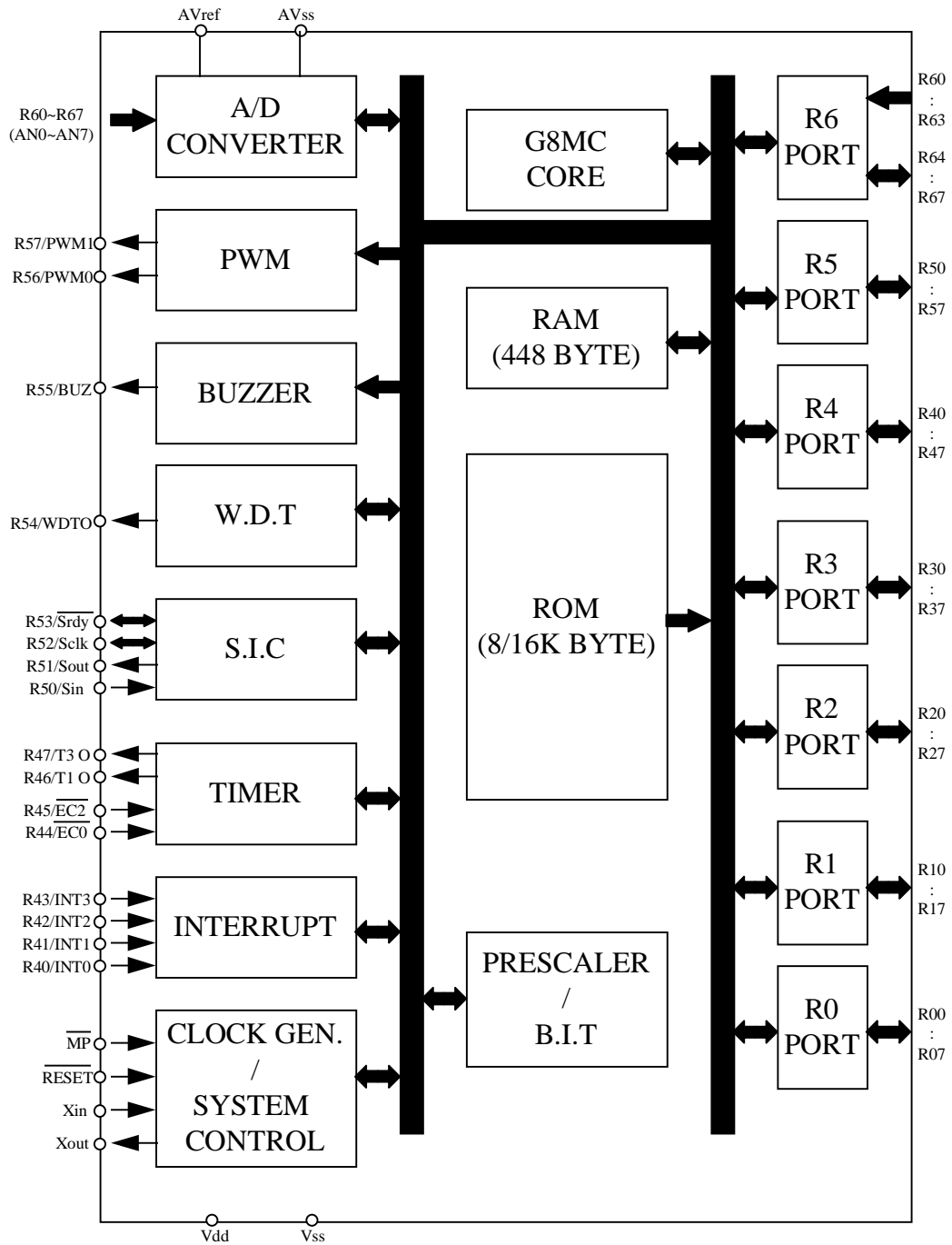
GMS81508/16 is a single chip microcomputer designed CMOS technology. The use of CMOS process enables extremely low power consumption.

This device using the G8MC Core includes several peripheral functions such as Timer, A/D Converter, Programmable Buzzer Driver, Serial I/O, Pulse Width Modulation Function, etc. ROM, RAM, I/O are placed on the same memory map in addition to simple instruction set.

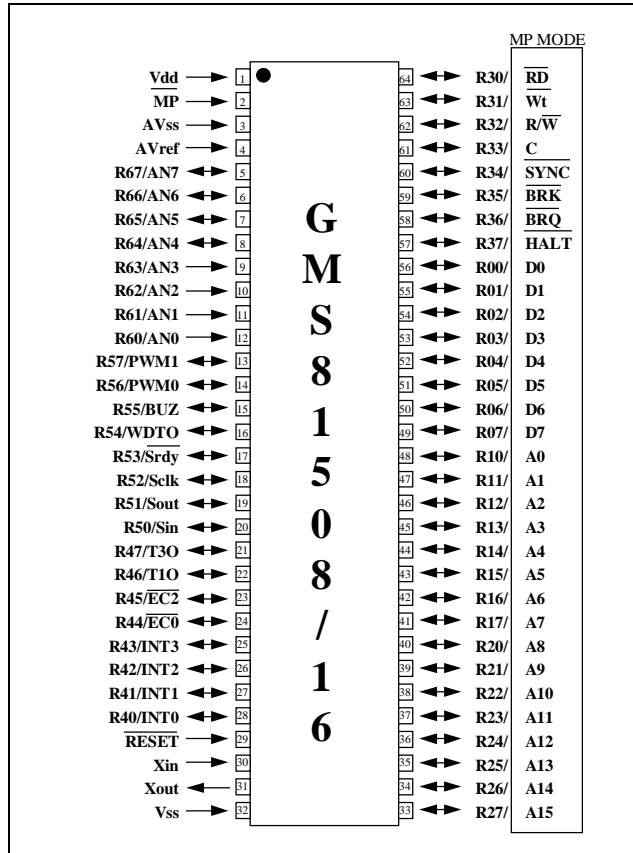
1.1. FEATURES

| | GMS81508 | GMS81516 |
|-----------------------------|-------------------------------------|-----------------|
| ROM(Bytes) | 8K | 16K |
| RAM(Bytes) | 448 bytes(includes stack area) | |
| Execution Time | 0.5us (@Xin=8MHz) | |
| Basic Interval Timer | 8bit X 1ch. | |
| Watch Dog Timer | 6bit X 1ch. | |
| Timer | 8bitX 4ch.(or 16bit X 2ch.) | |
| ADC | 8bit X 8ch. | |
| PWM | 8bit X 2ch. | |
| Serial I/O | 8bit X 1ch. | |
| External Interrupt | 4ch. | |
| Buzzer Driver | Programmable Buzzer Driving Port | |
| I/O Port | 4 - Input only 52 - Input/Output | |
| Power Save Mode | STOP Mode | |
| Operating Voltage | 4.5 ~ 5.5V (@ Xin=8MHz) | |
| Operating Frequency | 1 ~ 8MHz | |
| Package | 64SDIP, 64QFP | |
| OTP | GMS81516T | |
| Application | Home Appliances, LED Applications | |

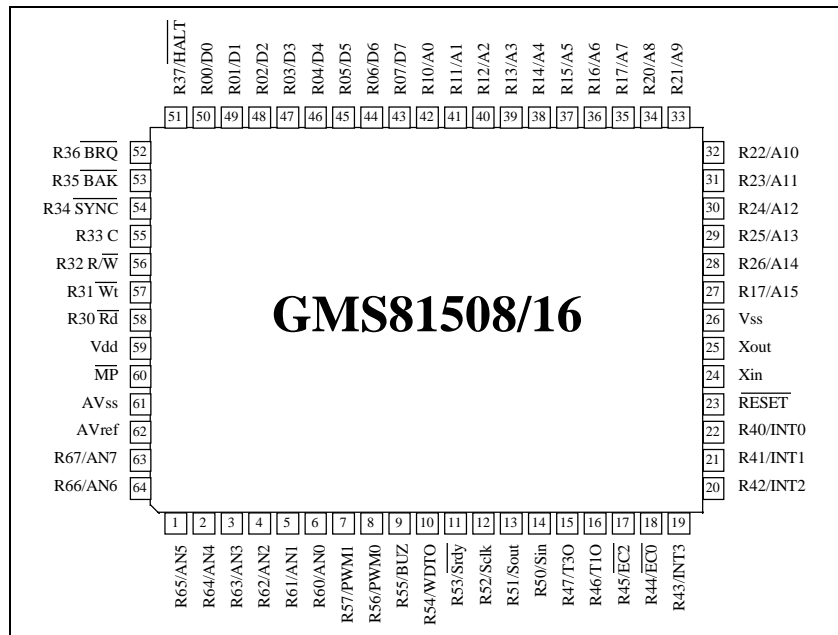
1.2. BLOCK DIAGRAM



1.3. PIN ASSIGNMENT

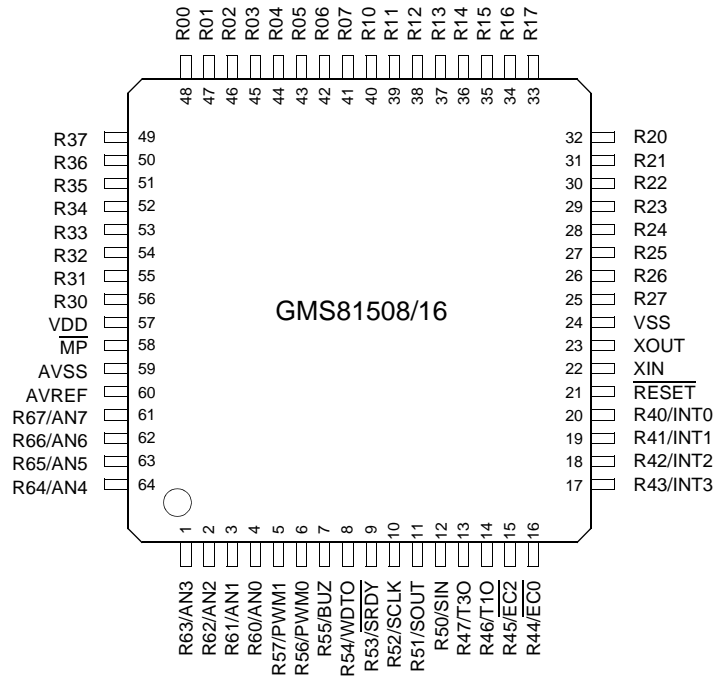


64 SDIP

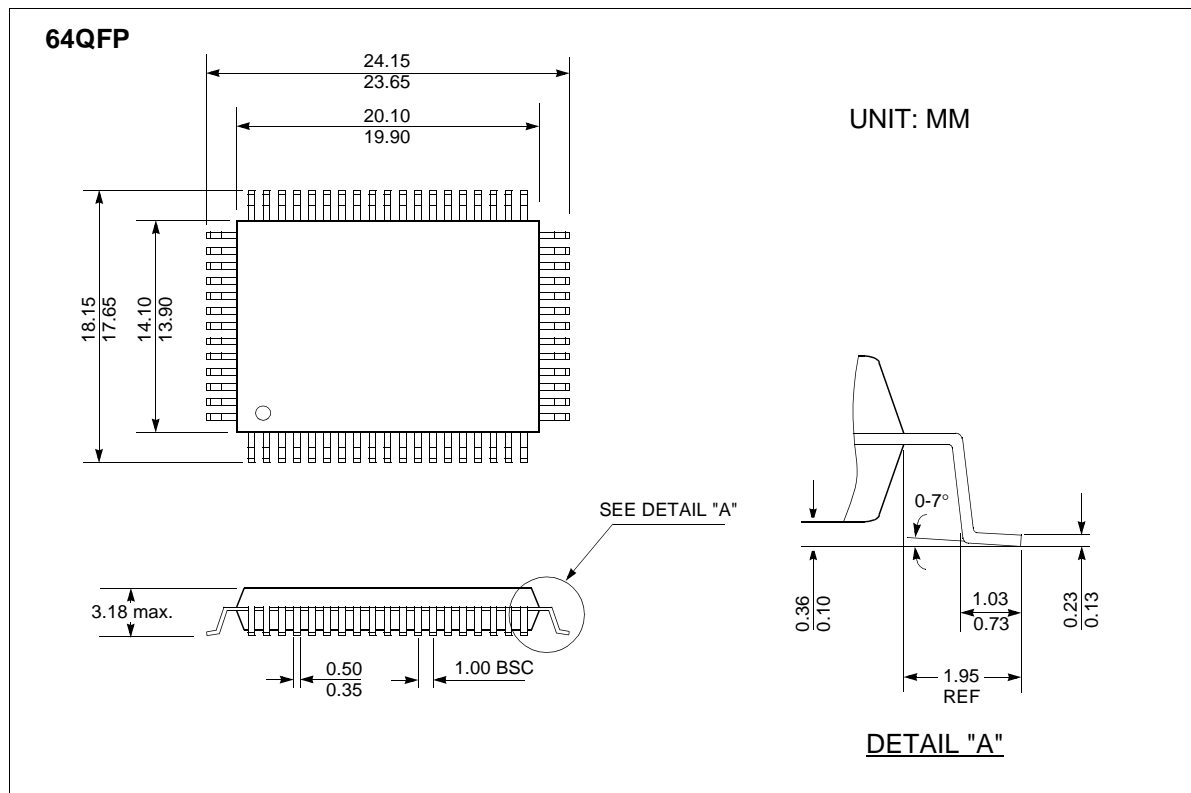
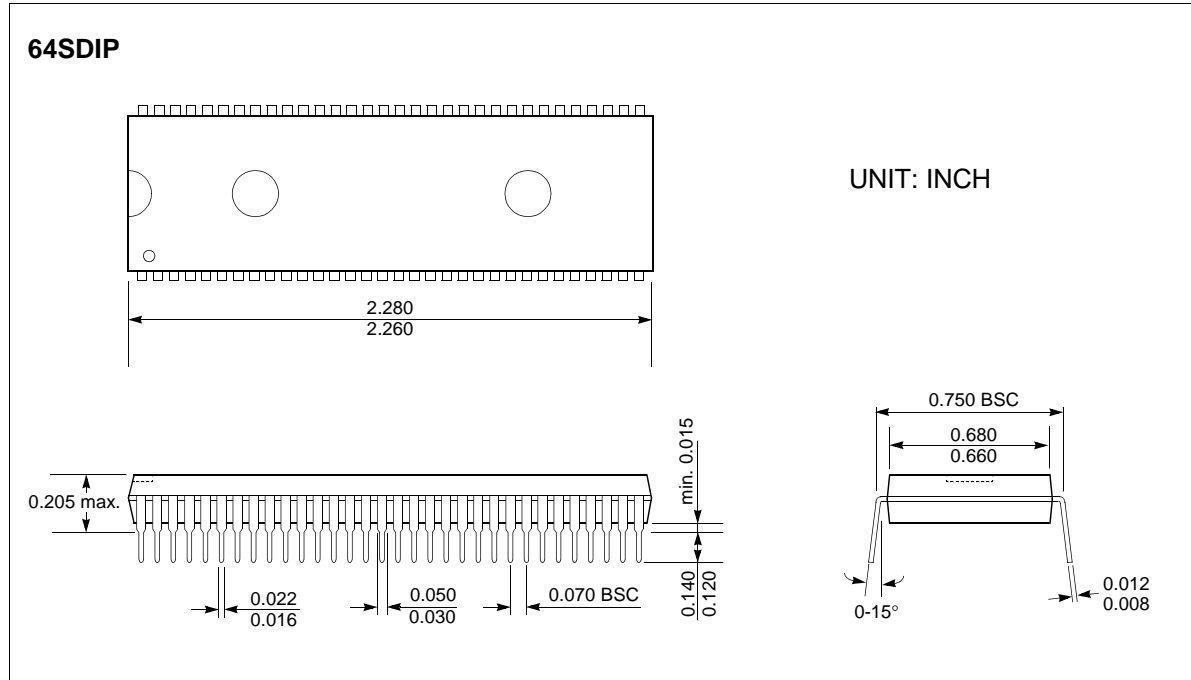


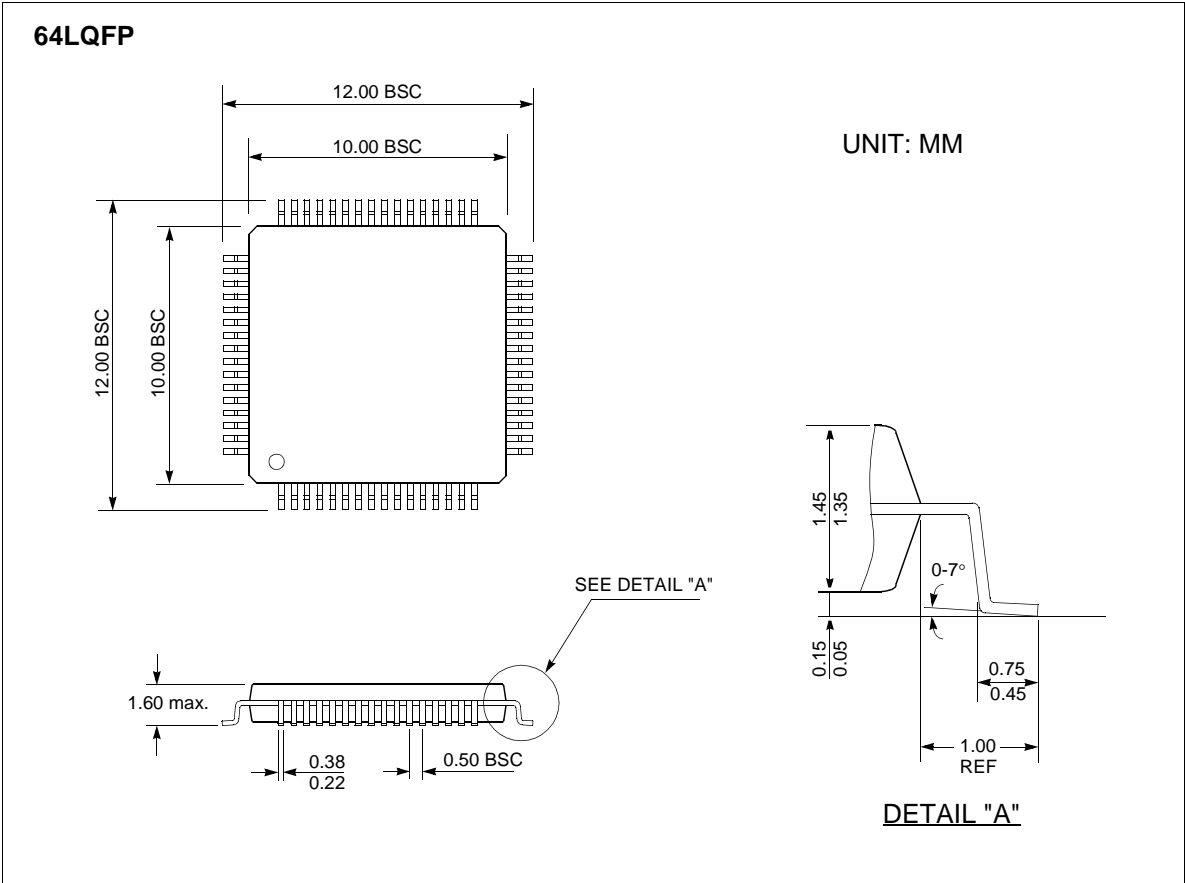
64 QFP

64LQFP



1.4 PACKAGE DIMENSION





1.5. PIN DESCRIPTION

| Classification | No. | Symbol | I/O | Descriptions |
|-------------------------------|-----|--------------------|-----|---|
| Power | 1 | Vdd | I | Power Supply Input Pin(4.5~5.5V) |
| | 32 | Vss | I | Ground(0V) |
| System Control or Clock | 2 | \overline{MP} | I | Controls Microprocess Mode of the Chip At "H" input : Single Chip Mode At "L" input : Microprocess Mode |
| | 29 | \overline{RESET} | I | In the state of "L" level, system enter to the reset state. |
| | 30 | Xin | I | This chip has an internal clock generating circuit. To control generating frequency, an external ceramic or a quartz crystal oscillator is connected between Xin and Xout pins. |
| | 31 | Xout | I | If external clock is used, the clock source should be connected to the Xin pin and the Xout pin should be left open. |
| Timer | 24 | $\overline{EC0}$ | I | Event Counter Source Clock Input Pin |
| | 23 | $\overline{EC2}$ | I | |
| | 22 | T1O | O | Timer Counter Overflow Output Pin |
| | 21 | T3O | O | |
| Ext. Interrupt | 28 | INT0 | I | External Interrupt Request Signal Input Pin |
| | 27 | INT1 | I | |
| | 26 | INT2 | I | |
| | 25 | INT3 | I | |
| A/D Converter | 4 | AVref | I | Reference Voltage Input Pin for A/D Converter |
| | 3 | AVss | I | Ground Level Input Pin for A/D Converter |
| | 12 | AN0 | I | Analog Voltage Input Pin for A/D Converter |
| | 11 | AN1 | I | |
| | 10 | AN2 | I | |
| | 9 | AN3 | I | |
| | 8 | AN4 | I | |
| | 7 | AN5 | I | |
| | 6 | AN6 | I | |
| 5 | AN7 | I | | |
| Serial I/O | 17 | \overline{Srdy} | I/O | Receive Enable Output Pin |
| | 18 | Sclk | I/O | Serial Clock Output Pin |
| | 19 | Sout | O | Serial Data Output Pin |
| | 20 | Sin | I | Serial Data Input Pin |
| P.W.M | 14 | PWM0 | O | PWM Pulse Output Pin |
| | 13 | PWM1 | O | |
| Buzzer | 15 | BUZ | O | Buzzer Driving Frequency Output Pin |
| W.D.T | 16 | WDTO | O | Watch dog Timer Overflow Output Pin |

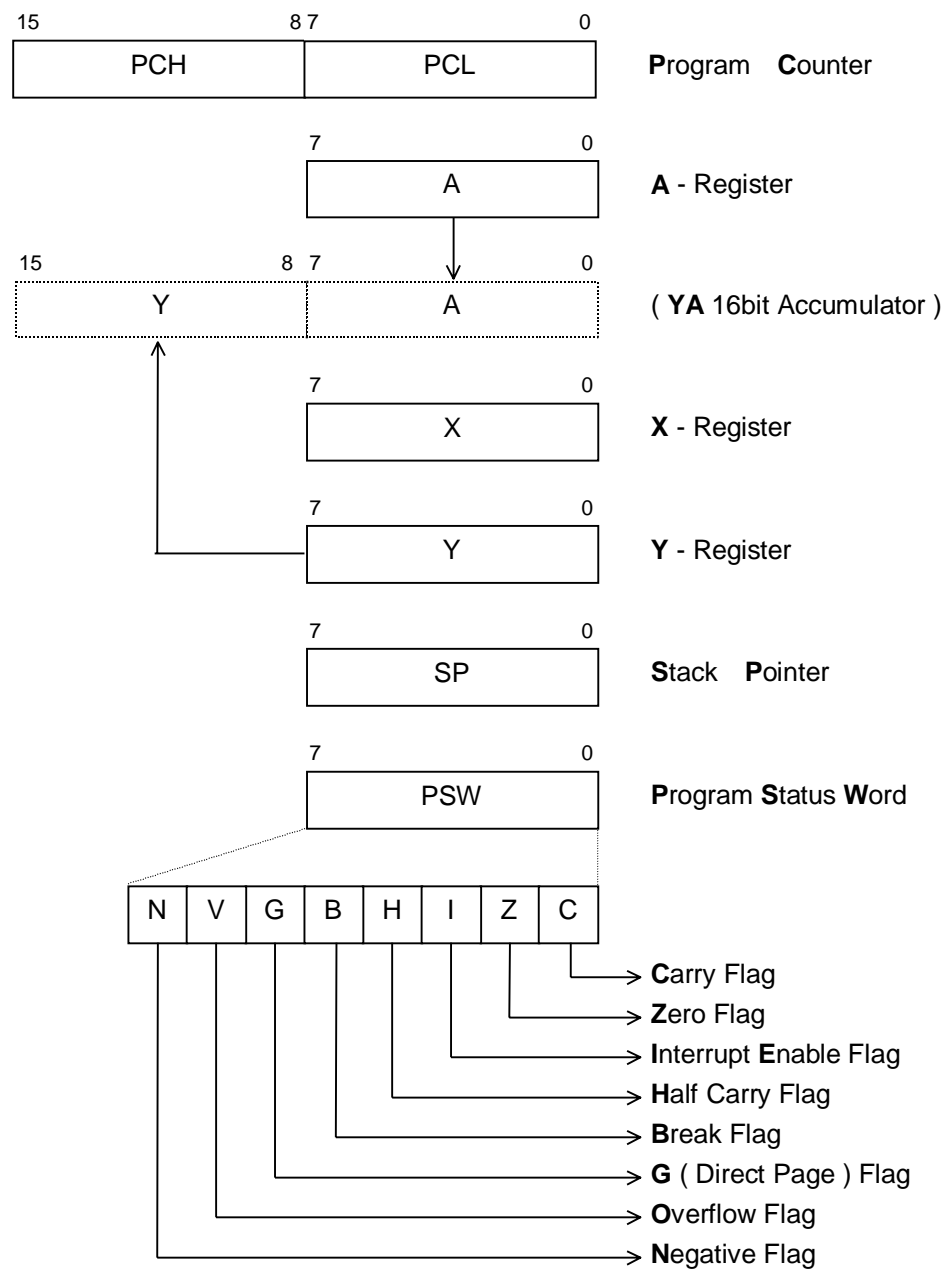
HYUNDAI MicroElectronics

| Classification | No. | Symbol | I/O | Description |
|----------------|---------------|-----------------|-----|---|
| I/O Port | 49 : 56 | R00 : R07 | I/O | R0 Port (Can be determined I/O by R0DD) In MP mode, This port functions as 8-bit data bus for the CPU. (D0~D7) |
| | 41 : 48 | R10 : R17 | I/O | R1 Port (Can be determined I/O by R1DD) In MP mode, This functions as 8-bit lower address output pins. (A0~A7) |
| | 33 : 40 | R20 : R27 | I/O | R2 Port (Can be determined I/O by R2DD) In MP mode, This functions as 8-bit higher address output pins.(A8~A15) |
| | 57 : 64 | R30 : R37 | I/O | R3 Port (Can be determined I/O by R3DD) In MP mode, This port functions as 8-bit control bus for the CPU. |
| | 28 : 21 | R40 : R47 | I/O | R4 Port (Can be determined I/O by R4DD) |
| | 20 : 13 | R50 : R57 | I/O | R5 Port (Can be determined I/O by R5DD) |
| | 12 : 9 | R60 : R63 | I | R6 Port Input Only |
| | 8 : 5 | R64 : R65 | I/O | R6 Port (Can be determined I/O by R6DD) |

2. FUNCTIONS

2.1. REGISTERS

6 registers are built-in the CPU of G8MC. Accumulator(A), Index register X, Y, Stack Pointer (SP) and Program Status Word(PSW) consists of 8-bit registers. Program Counter(PC) consists of 16-bit registers. The contents of these registers are undefined after RESET.



2.1.1. A - Register

The accumulator is the 8-bit general purpose register. This is used register for data operation, data transfer, temporary saves and conditional judgment.

Accumulator can be used as a 16-bit register with Y register and has a lower 8-bit data.

In case of multiplication instruction(MUL), it works as a multiplier. After execution of MUL instruction, Accumulator has lower 8-bit data of the results(16-bit).

In case of division instruction(DIV), it has the lower 8-bit of dividend (16-bit)

2.1.2. X- Register

In index addressing mode, this register is executed as a 8-bit index register within direct page(RAM area). also, In indirect addressing mode, it is destination address register.

This register can be used as a increment, decrement, comparison, and data transfer function.

In case of division instruction(DIV), it works as a divisor.

2.1.3. Y- Register

In index addressing mode, this register is executed as a index register.

In case of 16-bit operation instruction, this register has upper 8-bit of YA (16-bit accumulator).

In case of multiplication instruction(MUL), this register is executed as a multiplicand register. After multiplication operation, it has the upper 8-bit of the result.

In case of division instruction, it is executed as a dividend(upper 8-bit). After division operation, it has quotient.

This register can be used as a loop counter of conditional branch command. (e.g. DBNE Y, rel)

2.1.4. Stack Pointer

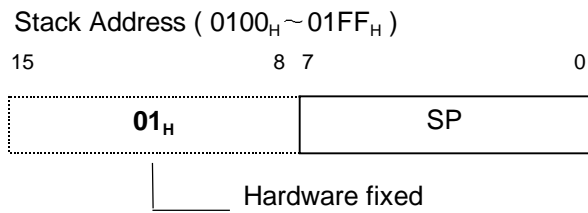
The stack pointer(SP) is an 8-bit register used during subroutine calling and interrupts.

When branching out from an on-going routine to subroutine or interrupt routine, it is necessary to remember the return address. normally, internal RAM is used for storing the return address and this area is called stack area. SP is pointer to show where the stack data are stored within the stack area.

The stack area is located in 1-Page of internal RAM. SP must be initialized by SW because the contents of SP is undefined after RESET.

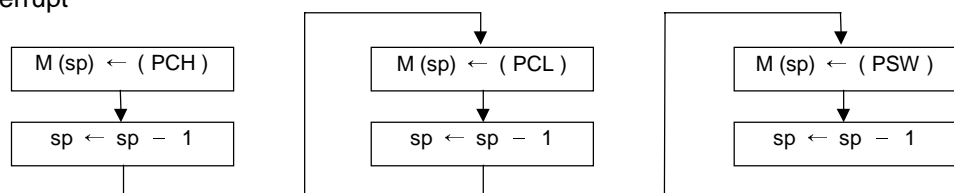
```
ex)   LDX          #0FEH          ;0FEH -> X register
      TXSP          ;X -> SP
```

caution) You can't use !01FFH as stack. If you use this area, mal-function would be occurred.

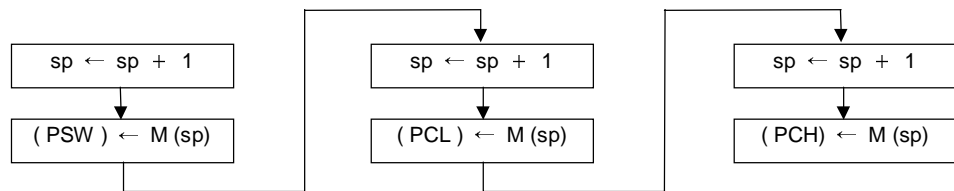


The belows shows data store and restore sequence to/from stack area.

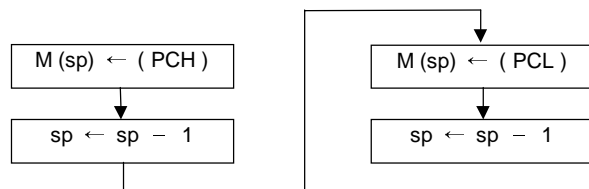
① Interrupt



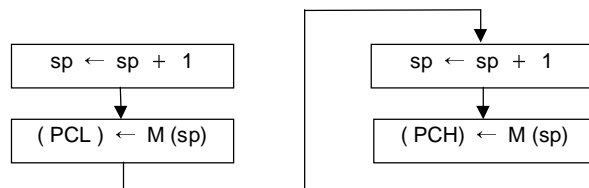
② RETI



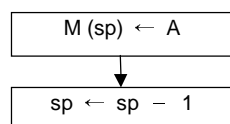
③ Subroutine CALL



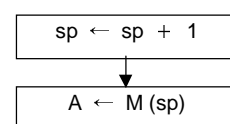
④ RET



⑤ PUSH A (X, Y, PSW)



⑥ POP A (X, Y, PSW)



2.1.5. Program Counter

The program counter(PC) is a 16-bit counter which consists of 8-bit register PCH and PCL. The addressing space is 64K bytes.

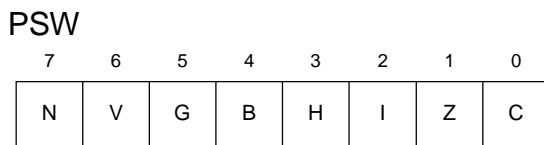
This counter indicates the address of the next instruction to be executed.

In reset state, the program counter (PC) has reset routine address in address FFFFH and FFFE H .

2.1.6. Program Status Word

PSW is an 8-bit register which is composed of flags to maintain the condition of the processor immediately after an operation.

After RESET, The contents of PSW is set to "00H".



① Carry Flag (C)

After an operation, it is set to "1" when there is a carry from bit7 of ALU or not a borrow.

SETC,CLRC instructions allow direct access for setting and resetting.

it can be used as a 1-bit accumulator.

It is a branch condition flag of BCS, BCC instructions.

② Zero Flag (Z)

After an operation including 16-bit operation, it is set to "1" when the result is "0".

It is a branch condition flag of BEQ, BNE.

③ Interrupt Enable Flag (I)

This flag is used to enable/disable all interrupts except interrupt caused by BRK instruction.

When this flag is "1", it means interrupt enable condition. When an interrupt is accept, this flag is automatically set to "0" thereby preventing other interrupts. also it is set to "1" by RETI instruction.

This flag is set and cleared by EI, DI instructions.

④ Half Carry Flag (H)

After an operation, it is set when there is a carry from bit3 of ALU or is not a borrow from bit4 of ALU.

It can not be set by any instruction. it is cleared by CLR V instruction like V flag.

⑤ Break Flag (**B**)

This flag is set by BRK (S/W interrupt) instruction to distinguish BRK and TCALL instruction having the same vector address.

⑥ Direct Page Flag (**G**)

This flag assign direct page (0-page, 1-page) for direct addressing mode. When G-flag is "0", the direct addressing space is in 0-page(0000H~00FFH). When G-flag is "1", the direct addressing space is in 1-page(0100H~01FFH).

It is set and cleared by SETG, CLRG instruction

⑦ Overflow Flag (**V**)

This flag functions when one word is added or subtracted in binary with the sign. When results exceeds +127 or -128, this flag is set.

When BIT instruction is executed, The bit6 of memory is input into V-flag.

This flag is cleared by CLRV instruction, but set instruction is not exist.

It is a branch condition flag of BVS, BVC.

⑧ Negative Flag (**N**)

N-flag is set when the result of a data transfer or operation is negative (bit7 is "1").

it means the bit-7 of memory is sign bit. thereby data is valid in the range of -128 ~ +127.

When BIT instruction is executed, The bit7 of memory is input into N-flag.

Set or clear instruction is not exist.

It is a branch condition flag of BPL, BMI instruction.

2.2. MEMORY SPACE

All RAM ,ROM,I/O, Peripheral Register are placed in the same memory area. Therefore, same instructions enable both data transfer and operation without the need to distinguish memory and I/O. The program counter of GMS81508/16 consists of 16-bit and memory addressing space is 64K byte.

2.2.1. RAM area

RAM(includes stack area) is 448 Bytes (0000_H~01FF_H).

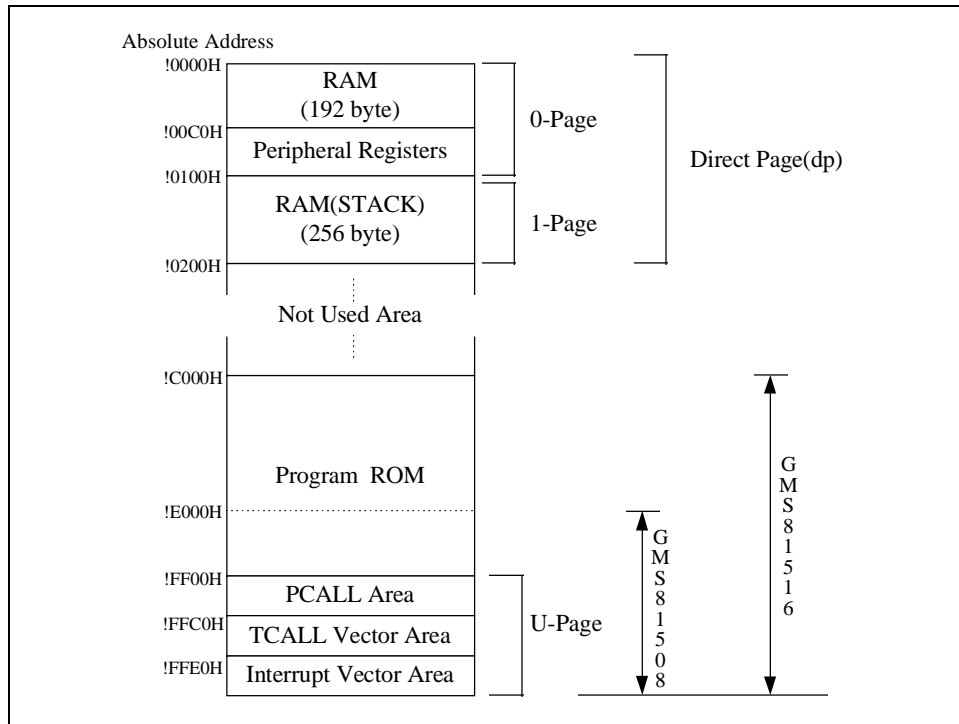
The internal RAM is used for data storage, subroutine calling or stack area when interrupts occur. When RAM is used as the stack area, the depth of the subroutine "nesting" and the interrupt levels should be kept in mind in order to avoid destruction of the RAM contents.

2.2.2. Peripheral Register area

Address 00C0_H~00FF_H are assigned for peripheral register.

2.2.3. Program ROM area

- ① PCALL subroutines must be located in **PCALL** area (FF00_H~FFBF_H).
- ② **TCALL** vector area (FFC0_H~FFDF_H) has the vector address corresponding to TCALL instruction.
- ③ **Interrupt Vector** area (FFE0_H~FFFF_H) has the vector address of interrupts, inclusive RESET.



■ VECTOR TABLE

| TCALL | | INTERRUPT | |
|---------------|----------|---------------|----------------------|
| Address | Vector | Address | Vector |
| FFC0H - FFC1H | TCALL 15 | FFE0H - FFE1H | not used |
| FFC2H - FFC3H | TCALL 14 | FFE2H - FFE3H | not used |
| FFC4H - FFC5H | TCALL 13 | FFE4H - FFE5H | Serial I/O |
| FFC6H - FFC7H | TCALL 12 | FFE6H - FFE7H | Basic Interval Timer |
| FFC8H - FFC9H | TCALL 11 | FFE8H - FFE9H | Watch Dog Timer |
| FFCAH - FFCBH | TCALL 10 | FFEAH - FFEBH | A/D Converter |
| FFCCH - FFCDH | TCALL 9 | FFECH - FFEDH | Timer 3 |
| FFCEH - FFCFH | TCALL 8 | FFEEH - FFEFH | Timer 2 |
| FFD0H - FFD1H | TCALL 7 | FFF0H - FFF1H | Timer 1 |
| FFD2H - FFD3H | TCALL 6 | FFF2H - FFF3H | Timer 0 |
| FFD4H - FFD5H | TCALL 5 | FFF4H - FFF5H | Ext. Int. 3 |
| FFD6H - FFD7H | TCALL 4 | FFF6H - FFF7H | Ext. Int. 2 |
| FFD8H - FFD9H | TCALL 3 | FFF8H - FFF9H | Ext. Int. 1 |
| FFDAH - FFDBH | TCALL 2 | FFFAH - FFFBH | Ext. Int. 0 |
| FFDCH - FFDDH | TCALL 1 | FFFCH - FFFDH | not used |
| FFDEH - FFDFH | TCALL 0 | FFFEH - FFFFH | RESET |

2.2.4. Peripheral Register List

| Address | Register Name | SYMBOL | R/W | RESET VALUE | | | | | | | | |
|-------------------|--------------------------------|---------------|-----|-------------|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 00C0 _H | R0 PORT DATA REGISTER | R0 | R/W | Undefined | | | | | | | | |
| 00C1 _H | R0 PORT I/O DIRECTION REGISTER | R0DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00C2 _H | R1 PORT DATA REGISTER | R0 | R/W | Undefined | | | | | | | | |
| 00C3 _H | R1 PORT I/O DIRECTION REGISTER | R0DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00C4 _H | R2 PORT DATA REGISTER | R0 | R/W | Undefined | | | | | | | | |
| 00C5 _H | R2 PORT I/O DIRECTION REGISTER | R0DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00C6 _H | R3 PORT DATA REGISTER | R0 | R/W | Undefined | | | | | | | | |
| 00C7 _H | R3 PORT I/O DIRECTION REGISTER | R0DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00C8 _H | R4 PORT DATA REGISTER | R4 | R/W | Undefined | | | | | | | | |
| 00C9 _H | R4 PORT I/O DIRECTION REGISTER | R4DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00CA _H | R5 PORT DATA REGISTER | R5 | R/W | Undefined | | | | | | | | |
| 00CB _H | R5 PORT I/O DIRECTION REGISTER | R5DD | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00CC _H | R6 PORT DATA REGISTER | R6 | R/W | Undefined | | | | | | | | |
| 00CD _H | R6 PORT I/O DIRECTION REGISTER | R6DD | W | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 00D0 _H | PORT R4 MODE REGISTER | PMR4 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00D1 _H | PORT R5 MODE REGISTER | PMR5 | W | - | - | 0 | 0 | - | - | - | - | - |
| 00D2 _H | TEST MODE REGISTER | TMR | W | - | - | - | - | - | 0 | 0 | 0 | 0 |
| 00D3 _H | BASIC INTERVAL REGISTER | BITR | R | Undefined | | | | | | | | |
| | CLOCK CONTROL REGISTER | CKCTLR | W | - | - | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 00E0 _H | WATCH DOG TIMER | WDTR | W | - | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 00E2 _H | TIMER MODE REGISTER 0 | TM0 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00E3 _H | TIMER MODE REGISTER 2 | TM2 | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00E4 _H | TIMER0 DATA REGISTER | TDR0 | R/W | Undefined | | | | | | | | |
| 00E5 _H | TIMER1 DATA REGISTER | TDR1 | R/W | Undefined | | | | | | | | |
| 00E6 _H | TIMER2 DATA REGISTER | TDR2 | R/W | Undefined | | | | | | | | |
| 00E7 _H | TIMER3 DATA REGISTER | TDR3 | R/W | Undefined | | | | | | | | |
| 00E8 _H | A/D CONVERTER MODE REGISTER | ADCM | R/W | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 00E9 _H | A/D CONVERTER DATA REGISTER | ADR | R | Undefined | | | | | | | | |
| 00EA _H | SERIAL I/O MODE REGISTER | SIOM | R/W | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Address | Register Name | SYMBOL | R/W | RESET VALUE | | | | | | | | |
|-------------------|--|--------------|-----|-------------|---|---|---|---|---|---|---|---|
| | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 00EB _H | SERIAL I/O REGISTER | SIOR | R/W | Undefined | | | | | | | | |
| 00EC _H | BUZZER DRIVER REGISTER | BUR | W | Undefined | | | | | | | | |
| 00F0 _H | PWM0 DATA REGISTER | PWMR0 | W | Undefined | | | | | | | | |
| 00F1 _H | PWM1 DATA REGISTER | PWMR1 | W | Undefined | | | | | | | | |
| 00F2 _H | PWM CONTROL REGISTER | PWMCR | W | 00 | | | | | | | | |
| 00F3 _H | INTERRUPT MODE REGISTER | IMOD | R/W | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00F4 _H | INTERRUPT ENABLE REGISTER LOW | IENL | R/W | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 00F5 _H | INTERRUPT REQUEST FLAG REGISTER LOW | IRQL | R/W | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 00F6 _H | INTERRUPT ENABLE REGISTER HIGH | IENH | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00F7 _H | INTERRUPT REQUEST FLAG REGISTER HIGH | IRQH | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 00F8 _H | EXT. INTERRUPT EDGE SELECTION REGISTER | IEDS | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

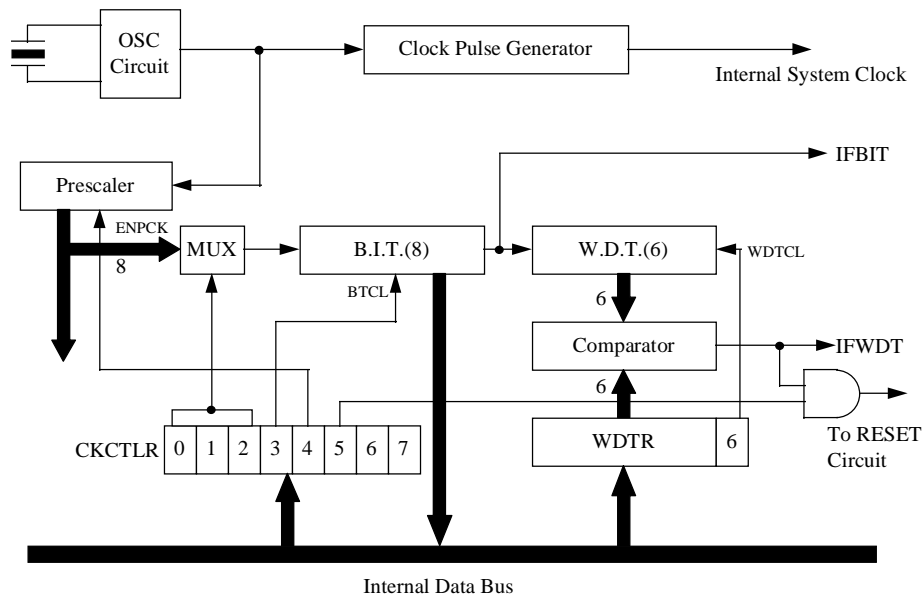
※ -: Not Used

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

2.3. CLOCK GENERATION CIRCUIT

The clock generation circuit of GMS81508/16 consists of oscillation circuit, prescaler, Basic Interval Timer.

The source clock of peripherals is provided by 11-bit prescaler.

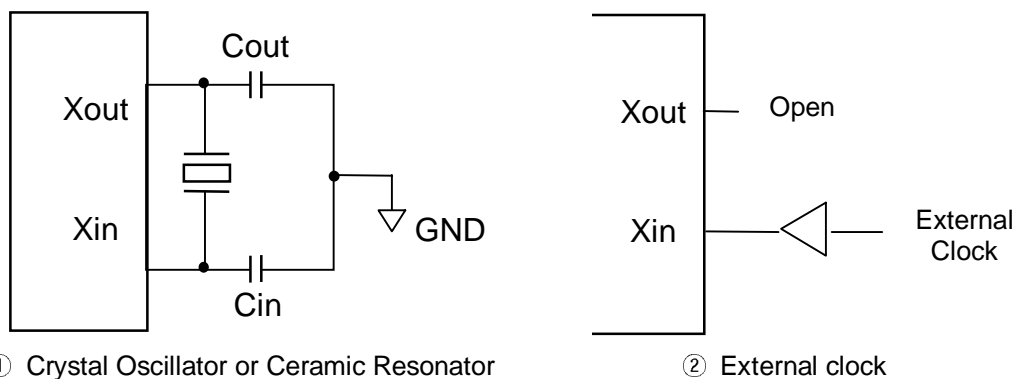


2.3.1. Oscillation Circuit

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.

The internal system clock for CPU is made by Clock Pulse Generator, and several peripheral clock is divided by prescaler.

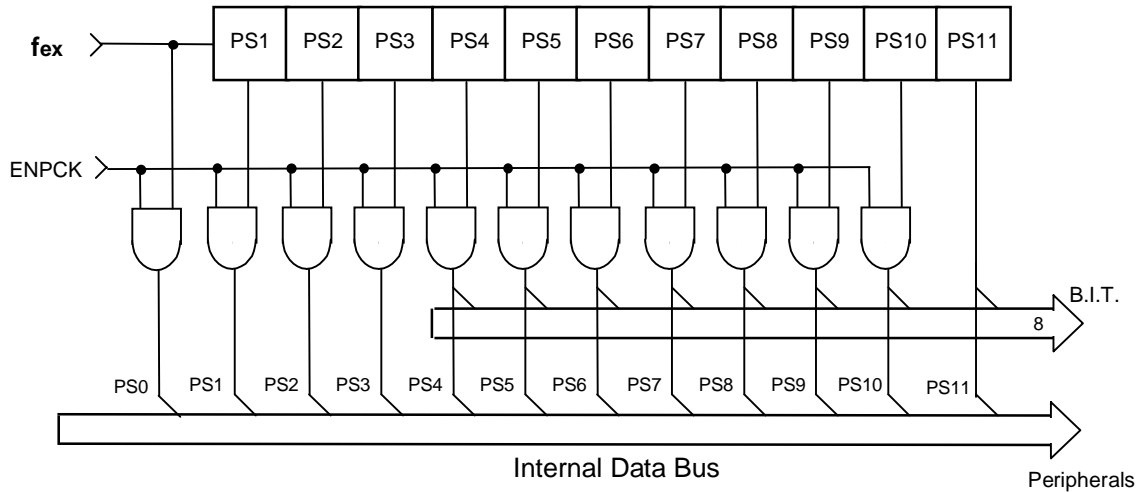
The clock generation circuit of crystal oscillator or ceramic resonator is shown in below.



★ In **STOP** Mode, The oscillation is stopped, Xin pin goes to "L" level status, and Xout pin goes to "H" level state.

2.3.2. Prescaler

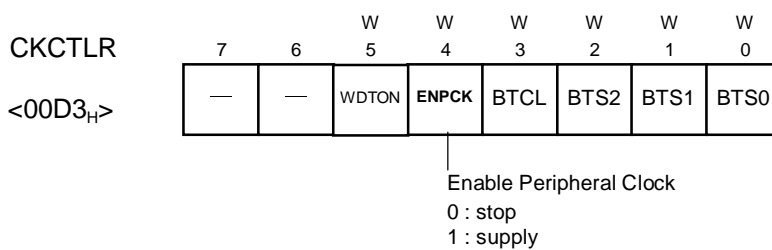
The prescaler consists of 11-bit binary counter, and input clock is supplied by oscillation circuit. The frequency divided by prescaler is used as a source clock for peripherals.



◆ Frequency-Divided Outputs of Prescaler

| f_{ex} (MHz) | PS1 | PS2 | PS3 | PS4 | PS5 | PS6 | PS7 | PS8 | PS9 | PS10 | PS11 | |
|----------------|----------|--------|--------|-----------|-----------|-----------|-----------|------------|------------|------------|-------------|-------------|
| 8 | Interval | 4 MHz | 2 MHz | 1 MHz | 500 kHz | 250 kHz | 125 kHz | 62.5 kHz | 31.25 kHz | 15.36 kHz | 7.18 kHz | 3.59 kHz |
| | Period | 250 ns | 500 ns | 1 μ s | 2 μ s | 4 μ s | 8 μ s | 16 μ s | 32 μ s | 64 μ s | 128 μ s | 256 μ s |

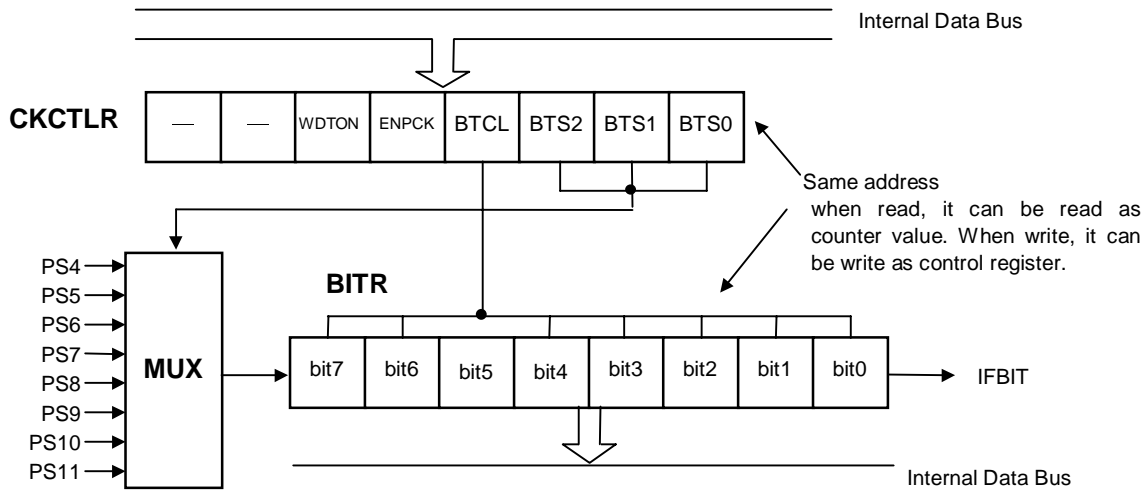
The peripheral clock supplied from prescaler can be stopped by ENPCK. (However, PS11 cannot be stopped by ENPCK)



2.4. BASIC INTERVAL TIMER

The Basic Interval Timer(B.I.T.) has 8-bit binary counter. The operations is shown below.

- . Generates reference time interval interrupt request as a timer.
- . The counting value of B.I.T. can be read.
(Note; The writing at same address overwrites the CKCTLR.)
- . The overflow of B.I.T be used the source clock of Watch Dog Timer.



2.4.1. Control of Basic Interval Timer

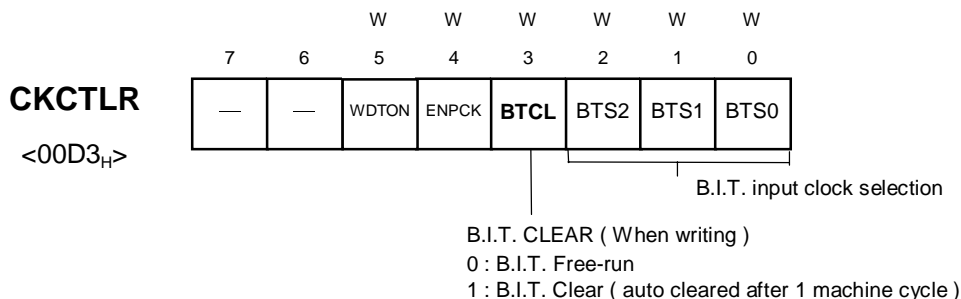
The Basic Interval Timer is free running timer. When the counting value is changed "0FFH" to "00H", The interrupt request flag is generated. The counter can be cleared by setting **BTCL** (Bit 3 of CKCTLR) and the BTCL is auto-cleared after 1 machine cycle. The initial state (after Reset) of BTCL is "0".

The input clock of Basic Interval Timer is selected by BTS2~BTS0 (Bit2~0 of CKCTLR) among the prescaler outputs (PS4~PS11).

The Basic Interval Timer Register (BITR) can be read.

The CKCTLR and the BITR have a same address (00D3H). So, If you write to this address, the CKCTLR would be controlled. If you read this address, the counting value of BITR would be read.

■ CLOCK CONTROL REGISTER



■ BASIC INTERVAL TIMER DATA REGISTER



2.5. WATCH DOG TIMER

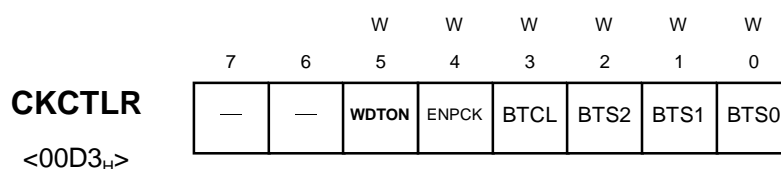
The Watch Dog Timer is a means of recovery from a system problem.

In this Device, the Watch Dog Timer consists of 6-bit binary counter, 6-bit comparator and watch dog timer register(WDTR). The source clock of WDT is overflow of Basic Interval Timer. The interrupt request of WDT is generated when the counting value of WDT equal to the contents of WDTR(bit0~5). This can be used as s/w interrupt or MICOM RESET signal(Watch Dog Function).

2.5.1. Control of Watch Dog Timer

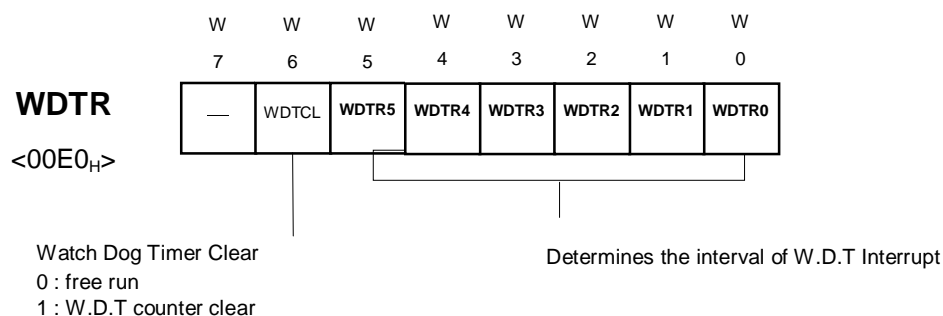
It can be used as 6-bit timer or WDT according to bit5(WDTON) of Clock Control Register (CKCTLR). The counter can be cleared by setting **WDTCL** (Bit 6 of WDTR) and the WDTCL is auto-cleared after 1 machine cycle. The initial state (after Reset) of WDTCL is "0".

■ CLOCK CONTROL REGISTER



WDT ON
0 : 6-bit Timer
1 : Watch Dog Timer

■ WATCH DOG TIMER REGISTER



HYUNDAI MicroElectronics

The interval of WDT interrupt is decided by the interrupt interval of Basic Interval Timer and the contents of WDTR.

$$\text{The interval of WDT} = \text{The contents of WDTR} \times \text{The interval of B.I.T.}$$

Caution) Do not use the contents of WDTR = "0"

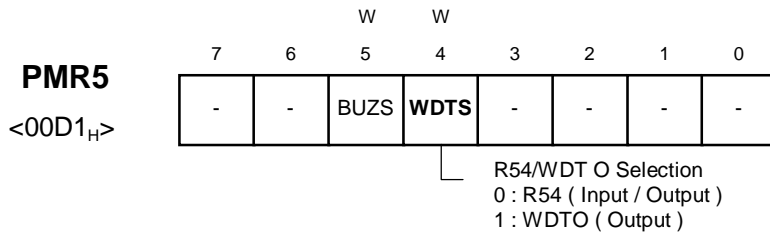
- ◆ The relationship between the input clock of B.I.T and the output of W.D.T. (@8MHz)

| BTS2 | BTS1 | BTS0 | B.I.T. Input Clock | The cycle of B.I.T. | The cycle of W.D.T.(max) |
|------|------|------|----------------------|---------------------|--------------------------|
| 0 | 0 | 0 | PS4 (2 μ S) | 512 μ S | 32,256 μ S |
| 0 | 0 | 1 | PS5 (4 μ S) | 1,024 μ S | 64,512 μ S |
| 0 | 1 | 0 | PS6 (8 μ S) | 2,048 μ S | 129,024 μ S |
| 0 | 1 | 1 | PS7 (16 μ S) | 4,096 μ S | 258,048 μ S |
| 1 | 0 | 0 | PS8 (32 μ S) | 8,192 μ S | 516,096 μ S |
| 1 | 0 | 1 | PS9 (64 μ S) | 16,384 μ S | 1,032,192 μ S |
| 1 | 1 | 0 | PS10 (128 μ S) | 32,768 μ S | 2,064,384 μ S |
| 1 | 1 | 1 | PS11 (256 μ S) | 65,536 μ S | 4,128,768 μ S |

2.5.2. The output of WDT signal

The overflow of WDT can be output through R54/WDT O port by setting bit4 of PMR5(WDTS) to "1".

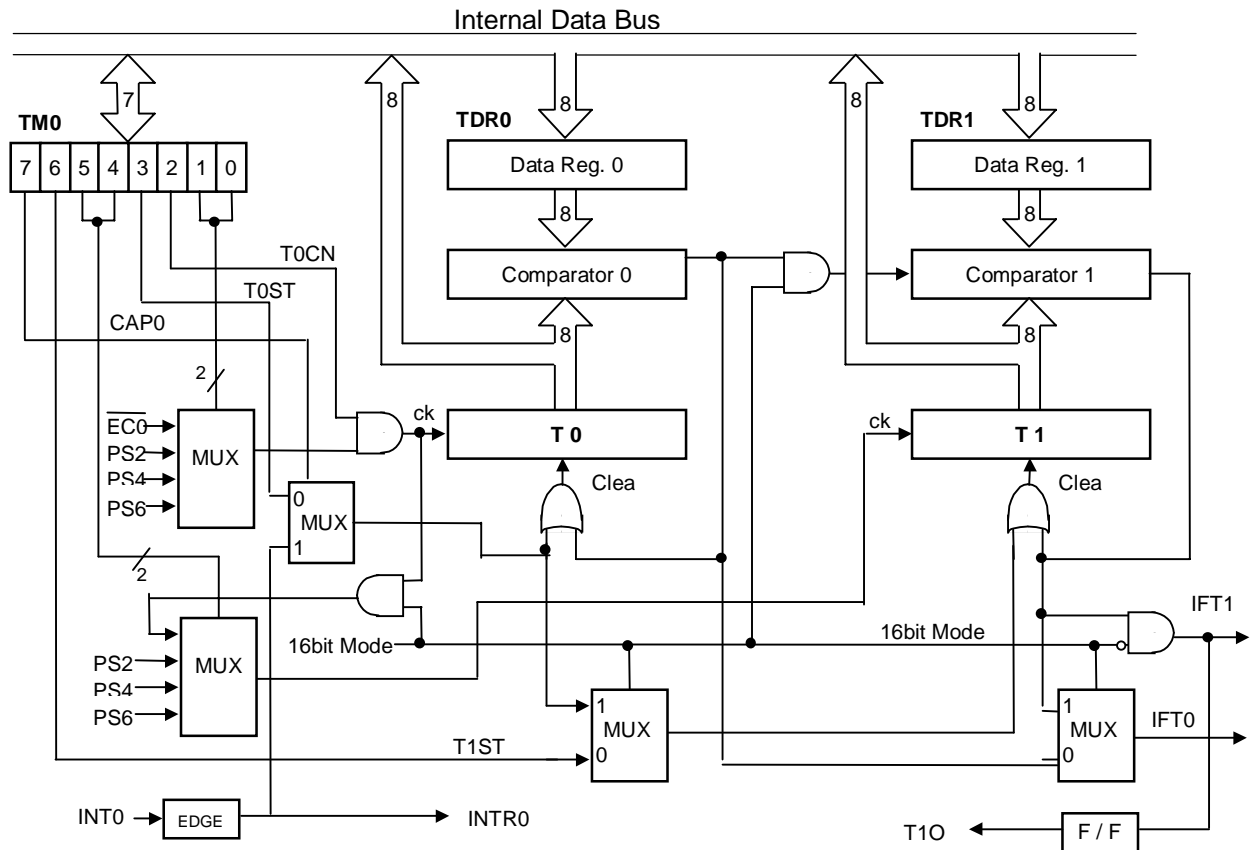
■ PORT R5 MODE REGISTER



2.6. TIMER

The GMS81508/16 has four multi-functional 8-bit binary timers(Timer0~Timer3).

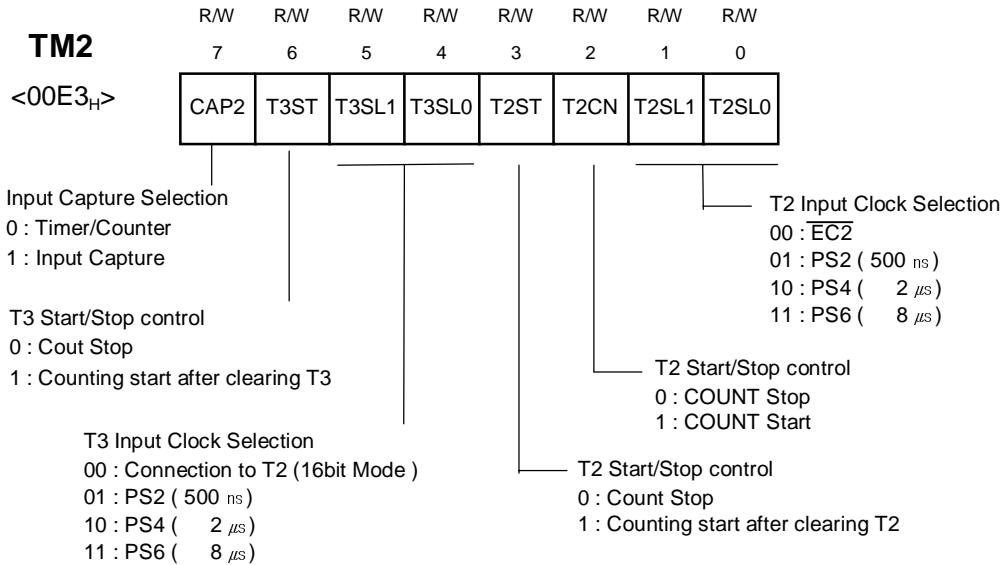
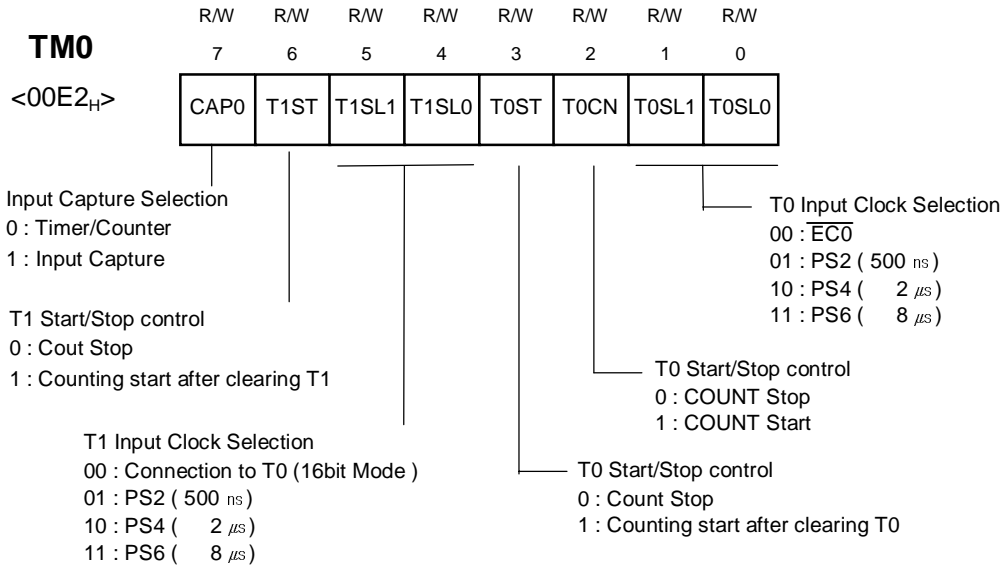
Timer0 (or Timer2) is can be used as a 16-bit timer/event counter with Timer1(or Timer3). The Timer0-1 and Timer2-3 have same functions and structures. So, We will explains about Timer0 and Timer1 only.



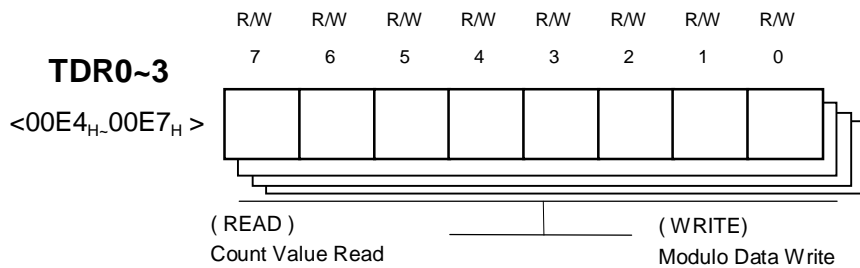
■ Operation Mode of Timer

| Timer0,Timer2 | Timer1,Timer3 |
|----------------------------------|----------------------------------|
| - 8-bit Interval Timer | - 8-bit Interval Timer |
| - 8-bit Event Counter | - 8-bit rectangular pulse output |
| - 8-bit input capture | |
| - 16-bit Interval Timer | |
| - 16-bit Event Counter | |
| - 8-bit rectangular pulse output | |

■ TIMER MODE REGISTER 0,2(TM0, TM2)

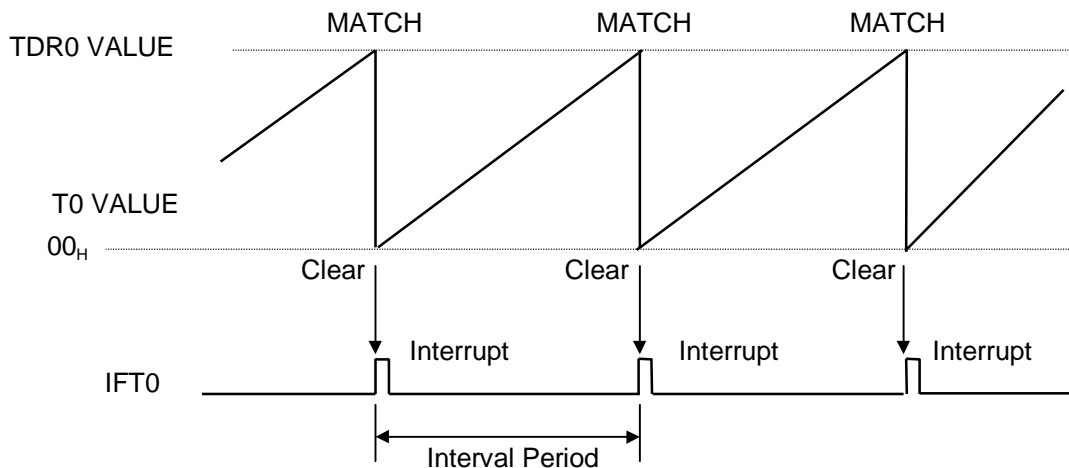


■ TIMER DATA REGISTER(TDR0 ~ TDR3)



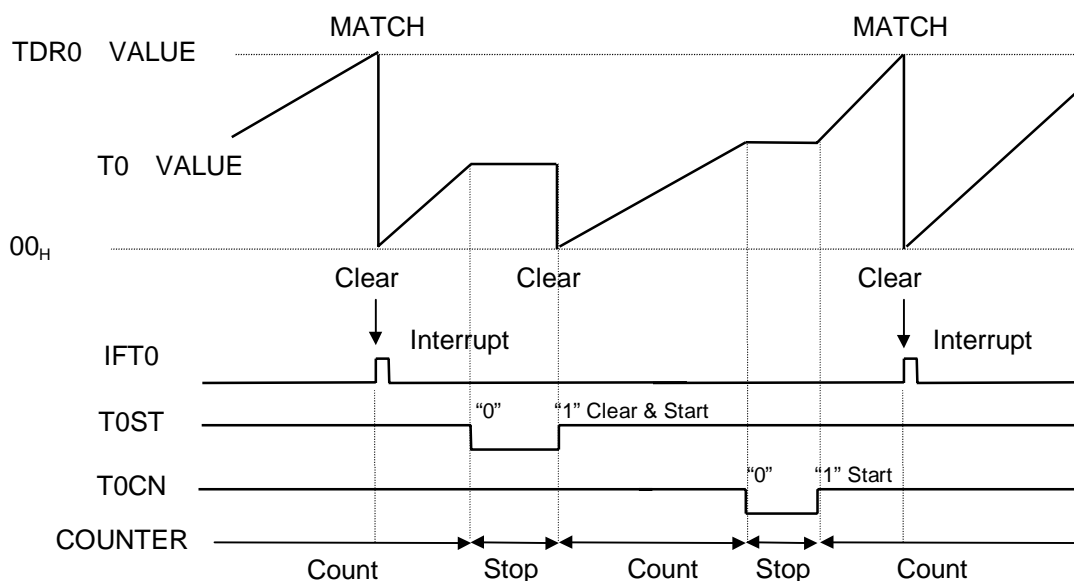
2.6.1. Control of Timer

T0 (T1) consists of 8-bit Binary Up-Counter. When the counting value of Timer0 , Timer1 and Timer0-1(16bit) become equal to the contents of Timer Data Register(TDR0,TDR1,TDR0-1) value, the counter is cleared to "00H" and restarts count-up operation. At this time, Interrupt request (IFT0 or IFT1) is generated.



Any of the PS2, PS4, PS6 or external clock can be selected as the clock source of T0 by bit1(T0SLI) and bit0(T0SL0) of TM0. Any 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. When the overflow of T0 is selected as input clock of T1, Timer0-1 operates as 16-bit timer. In this case, Timer0-1 only is controlled by T0ST,T0CN and the interrupt vector is Timer0 vector.

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 after timer clear. In order to enable count-up of timer, T0CN, T0ST and T1ST should become "1". In order to start count-up after clearing of counter, T0ST or T1ST should be set to "1" after set to "0" temporarily.



By read Timer Data Register(TDR0~3),The counting value of timer can be read at any time.

2.6.2. Interval Timer

The interrupt cycle is determined by the source clock of timer and the contents of TDR.

$$\text{Interrupt cycle} = \text{source clock} \times \text{the contents of TDR}$$

In order to write data to TDR, you have to stop timer. otherwise, TDR value is invalid.

◆ Maximum Interrupt Cycle according to source clock @ fex=8MHz

| | 8-bit TIMER Mode | | 16-bit TIMER Mode | |
|-------|---------------------|---------------|---------------------|-----------------|
| | source clock | max. count | source clock | max. count |
| T0,T2 | PS2 (0.5 μ S) | 128 μ S | PS2 (0.5 μ S) | 32,768 μ S |
| | PS4 (2 μ S) | 512 μ S | PS4 (2 μ S) | 131,072 μ S |
| | PS6 (8 μ S) | 2,048 μ S | PS6 (8 μ S) | 524,288 μ S |
| T1,T3 | PS2 (0.5 μ S) | 128 μ S | | |
| | PS4 (2 μ S) | 512 μ S | | |
| | PS6 (8 μ S) | 2,048 μ S | | |

2.6.3. Event Counter

The event counter operates in the same way as the interval timer except it counts the external event input from R44/EC0 and R45/EC1 port. it only counts at the falling edge of event input clock.

In order to input of external event clock, the relevant Port Mode Register(bit4,bit5 of PMR4) is set to "1". TDR value should be initialized to "FFH" because timer is cleared when it equals to TDR value, but if you want to use interrupt, TDR value should be written to "1H~FFH".

2.6.4. Pulse Output

A pulse width 50% cycle duty is output to the R46/T1 O or R47/T3 O port and reverse the output when timer interrupt is generated. This creates a pulse period which is two times that of the timer interrupt cycle. The output pulse period is determined by the source clock of timer and the contents of TDR.

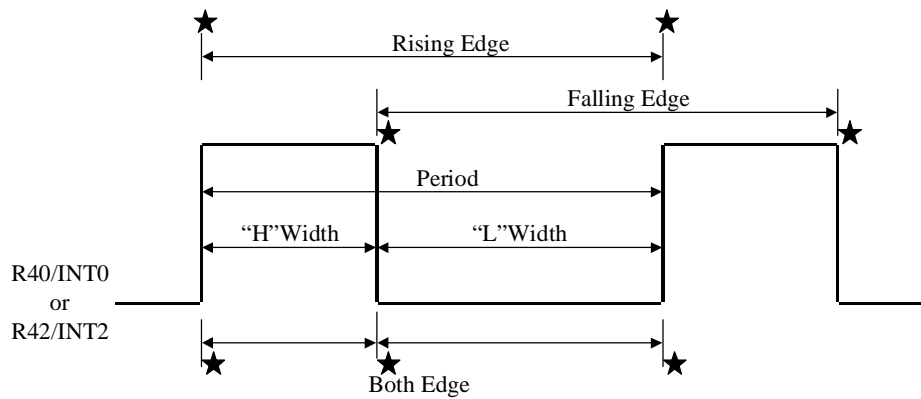
$$\text{output period} = \text{source clock}(\mu\text{S}) \times \text{the contents of TDR} \times 2$$

In order to output of pulse, the bit6,bit7 of PMR4 is set to "1".

2.6.5. Input Capture

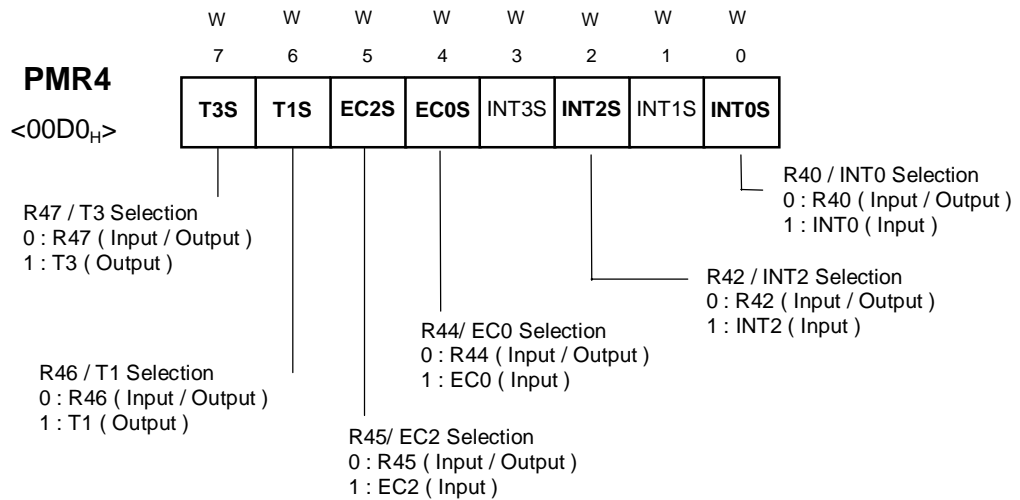
This function measures the period or width of pulse input from external INT. (R40/INT0, R42/INT2) port. The period of pulse is measured by selecting rising edge or falling edge of the interrupt edge select register(IEDS) and the width of pulse is measured by selecting both edge of IEDS.

The external interrupt is generated at the valid edge according to IEDS. At this time, The counting value of timer is loaded into TDR and counter is cleared and restarts count-up.



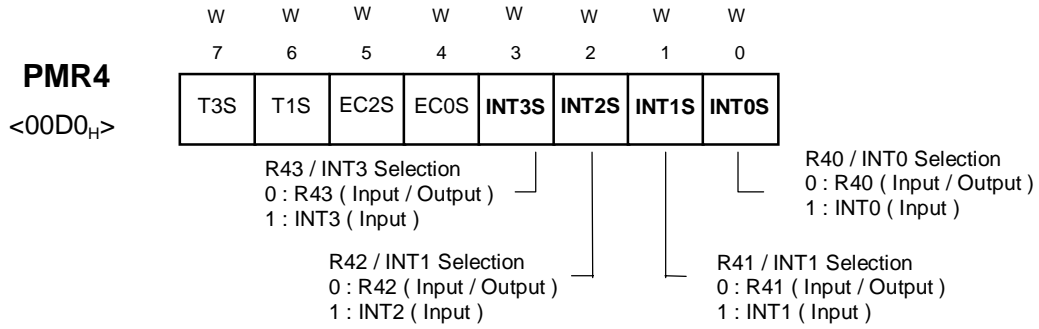
★ **Timer Operation**
the counting value of timer is latched
↓
timer is cleared to 00H
↓
timer restart count-up

■ **PORT R4 MODE REGISTER**

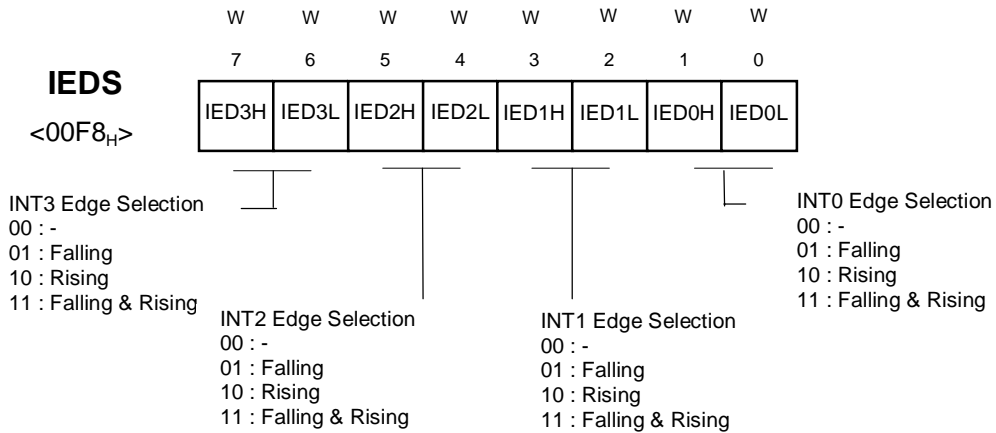


2.7. EXTERNAL INTERRUPT

An interrupt request is generated when a level-change from "H" to "L" or "L" to "H" of INT0,INT1,INT2,INT3 pin is detected. The edge of external interrupt is selected by interrupt edge selection register(IEDS) and ports(R40,R41,R42,R43) corresponding to INT0,INT1,INT4,INT3 are determined as a input port for external interrupt by bit0~3 of port4 mode register(PMR4).



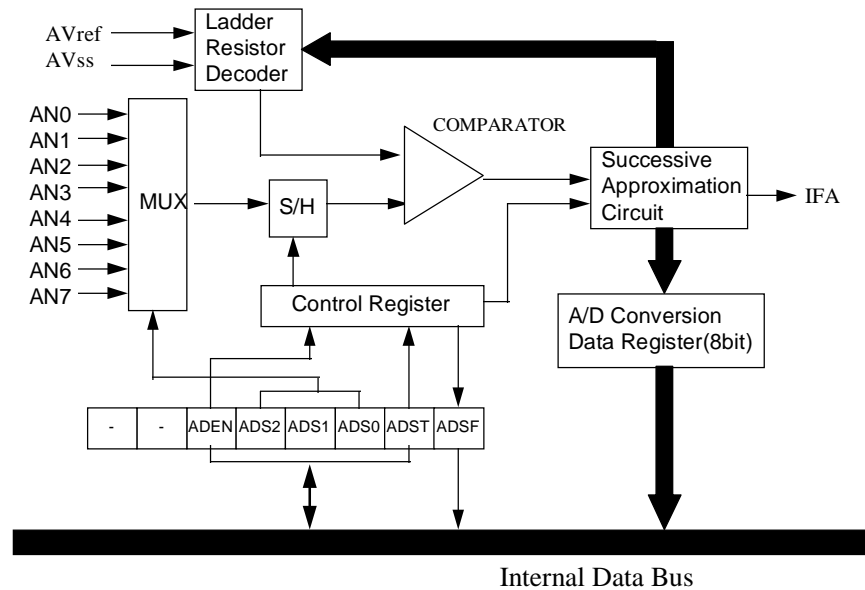
■ EXT. INTERRUPT EDGE SELECTION REGISTER



2.8. A/D CONVERTER

A/D Converter has an 8-bit resolution, and input is possible up to 8 channel.

A/D Converter consists of Analog Input Multiplexer, A/D convert Mode Register, Resistance Ladder, Sample and Holder, Successive Approximation Circuit and A/D Conversion Data Register.



2.8.1. Control of A/D Converter

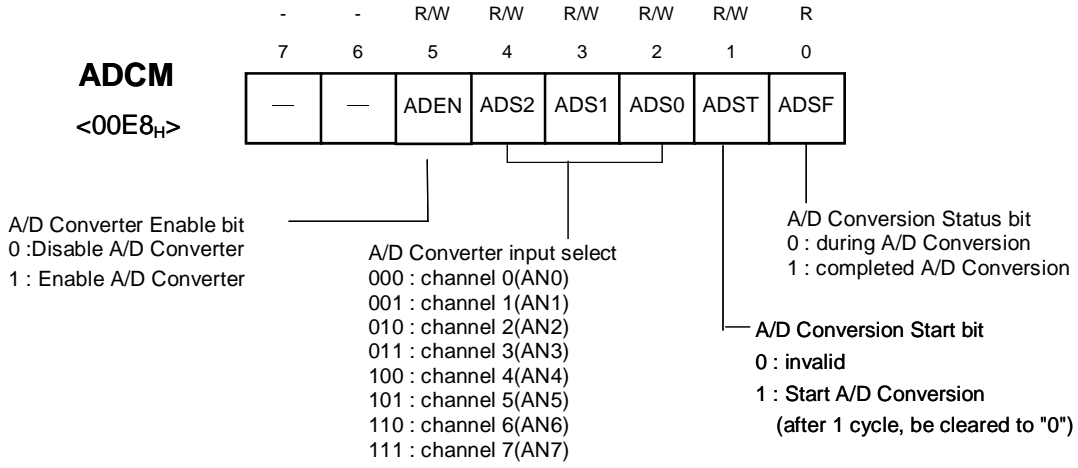
The analog input is selected by bit2~4 of A/D Converter Mode Register(ADCM). This bits chooses among AN0~AN7. The other analog pins which are not used not A/D conversion be used as normal port.

The A/D Conversion is started by setting A/D Conversion Start bit (ADST) to "1"(only for ADEN=1). After A/D Conversion is started, ADST is cleared by hardware. During A/D Conversion, when ADST is set to "1", A/D Conversion starts again from the beginning.

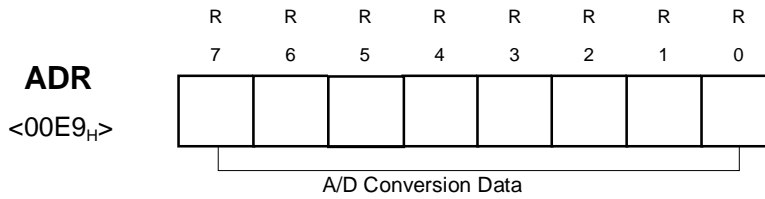
The analog input voltage and the reference voltage are compared and the result is stored in the A/D Converter Data Register(ADR) and ADSF(bit0 of ADCM) is set to "1". The A/D interrupt request is generated at the completion of A/D conversion.

The result of the conversion is obtained by reading out the A/D register(ADR).

■ A/D CONVERTER MODE REGISTER(ADCM)

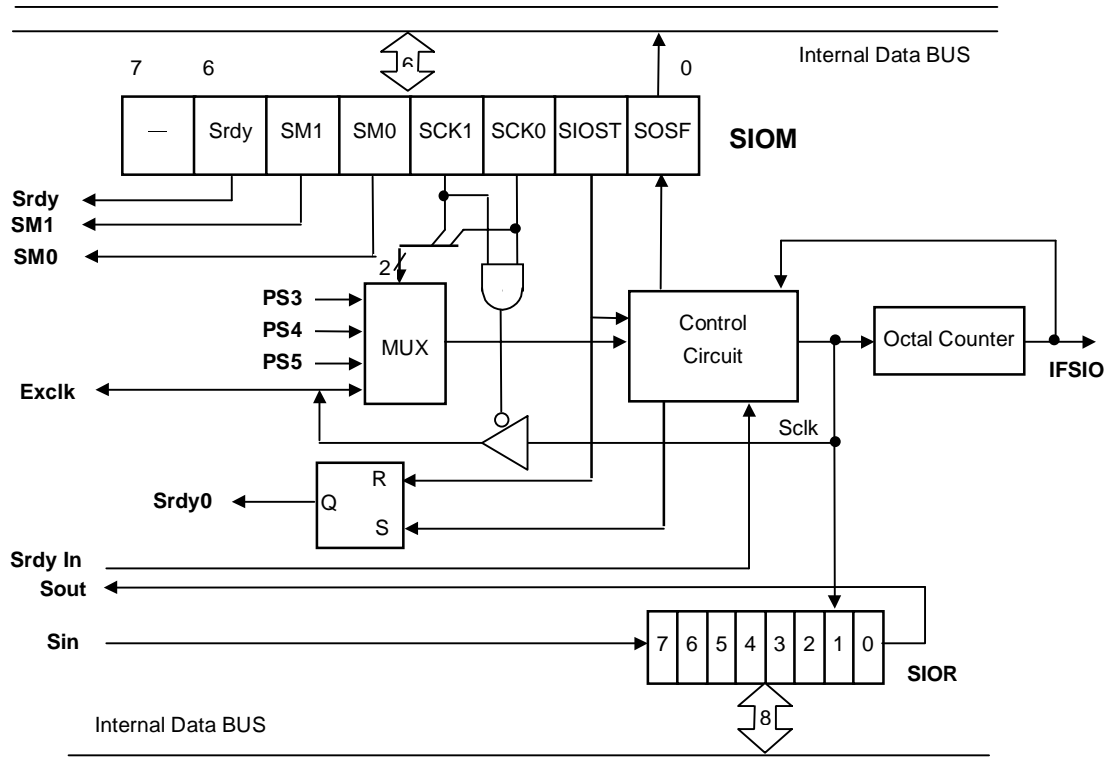


■ A/D CONVERTER DATA REGISTER(ADR)



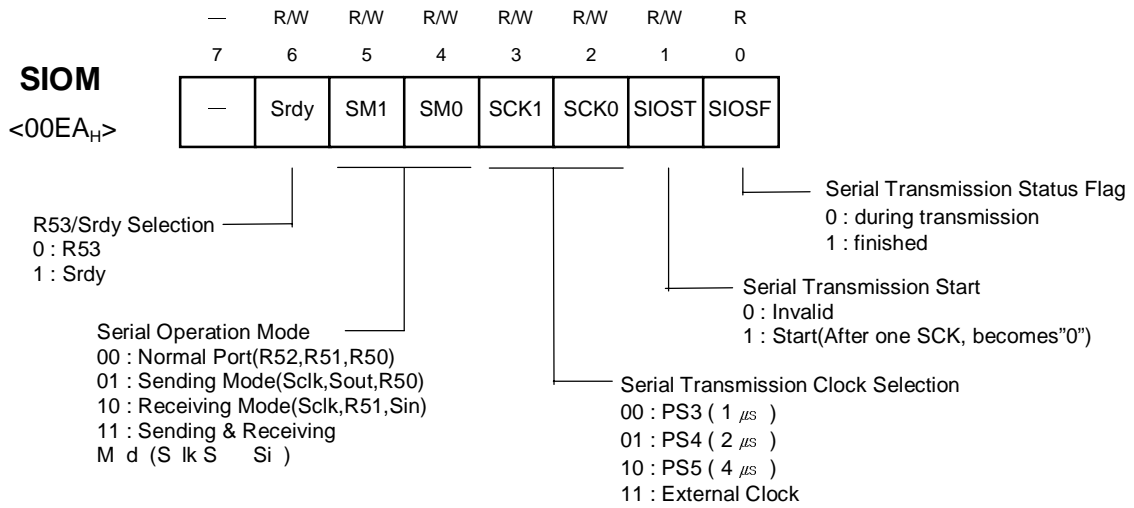
2.9. SERIAL I/O

The serial I/O is 8-bit clock synchronous type and consists of serial I/O register, serial I/O mode register, clock selection circuit octal counter and control circuit.



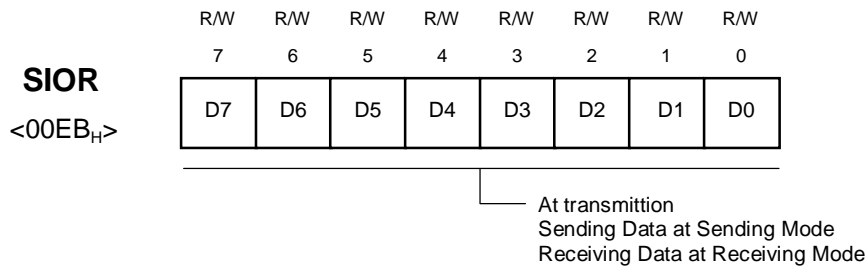
■ **Serial I/O Mode Register**

This register controls serial I/O function. According to SCK1 and SCK0, the internal clock or external clock can be selected.



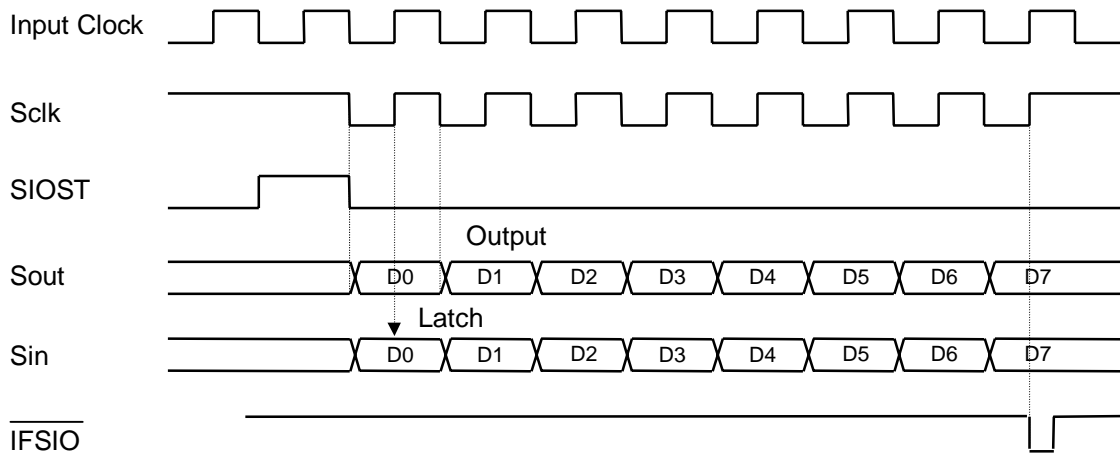
■ **Serial I/O Data Register**

The Serial I/O Data Register (**SIOR**) is a 8-bit shift register. First LSB is send or is received.



2.9.1. Data Transmission/Receiving Timing

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

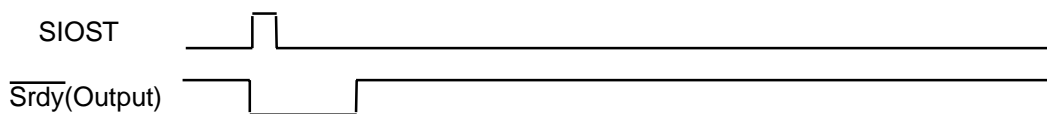


Timing Diagram of Serial I/O

2.9.2. The Serial I/O operation by Srdy pin

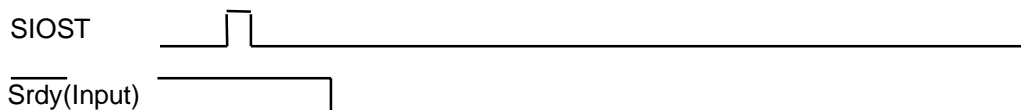
- transmission clock = external clock

The $\overline{\text{Srdy}}$ pin becomes "L" by SIOST = "1". This signal tells to the external system that this device is ready for serial transmission. The external system detects the "L" signal and starts transmission. The $\overline{\text{Srdy}}$ pin becomes "H" at the first rising edge of transmission clock.



- transmission clock = internal clock

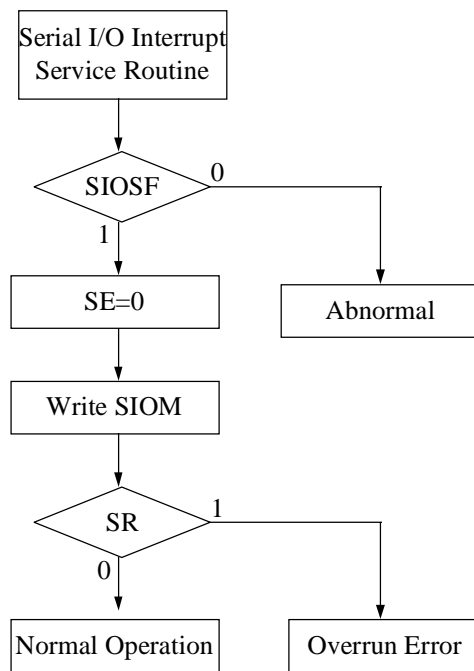
The I/O of $\overline{\text{Srdy}}$ pin is input mode. When the external system is ready for serial transmission, the "L" level is inputted at this pin. At this time this device starts serial transmission.



2.9.3. The method of Serial I/O

- ① Select transmission/receiving mode
<Notice> When external clock is used, the frequency should be less than 1MHz and recommended duty is 50%.
- ② In case of sending mode, write data to be send to SIOR.
- ③ Set SIOST to “1” to start serial transmission.
<Notice > If both transmission mode is selected and transmission is performed simultaneously it would be made error.
- ④ The SIO interrupt is generated at the completion of SIO and SIOSF is set to “1”. In SIO interrupt service routine, correct transmission should be tested.
- ⑤ In case of receiving mode, the received data is acquired by reading the SIOR.

2.9.4. The Method to Test Correct Transmission with S/W

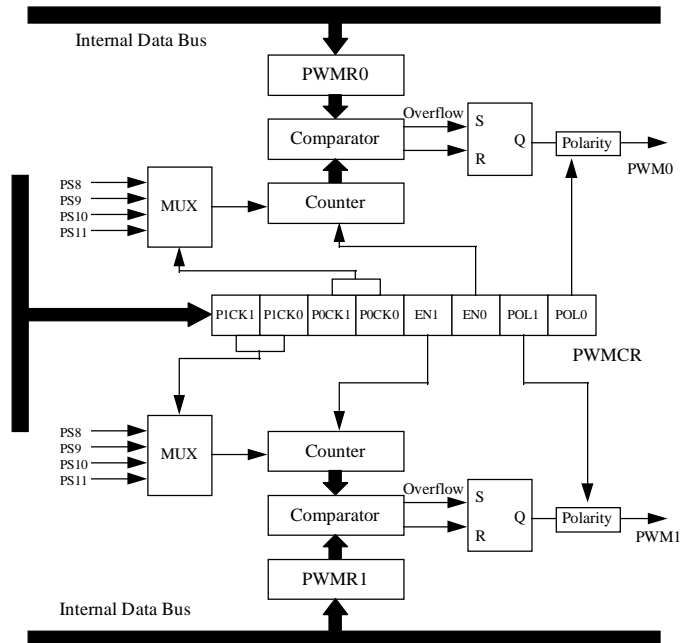


Serial Method to Test Transmission.

Note) **SE:** Interrupt Enable Regist Low **IENL (Bit3)**
SR : Interrupt Request Flag Regist Low **IRQL (Bit3)**

2.10. PWM

PWM(Pulse Width Modulation) has a 8-bit resolution and the PS8,PS9,PS10,PS11 of the prescaler can be selected as input clock PWM.



2.10.1. Controls of PWM

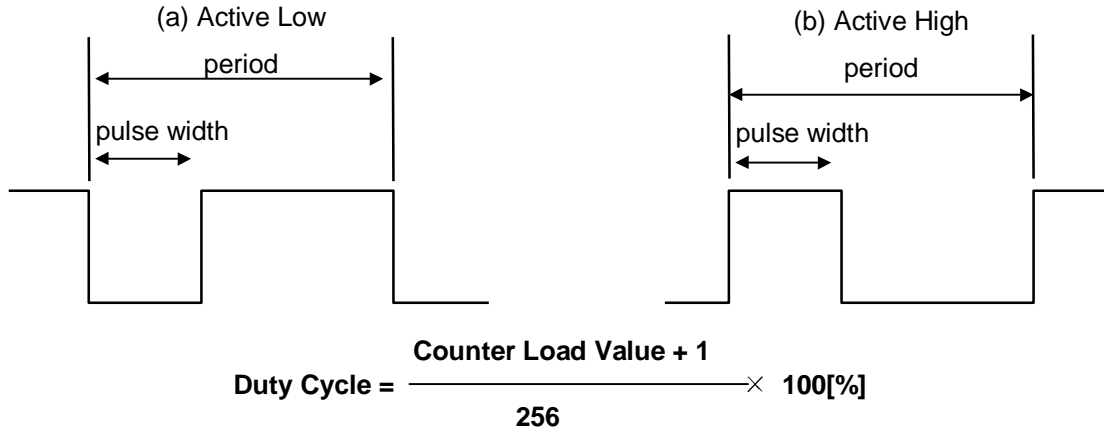
The input clock is selected by PWM Control Register (PWMCR), and the width of pulse is determined by the PWM Register (PWMR).

The pulse period according to input clock are as follows.

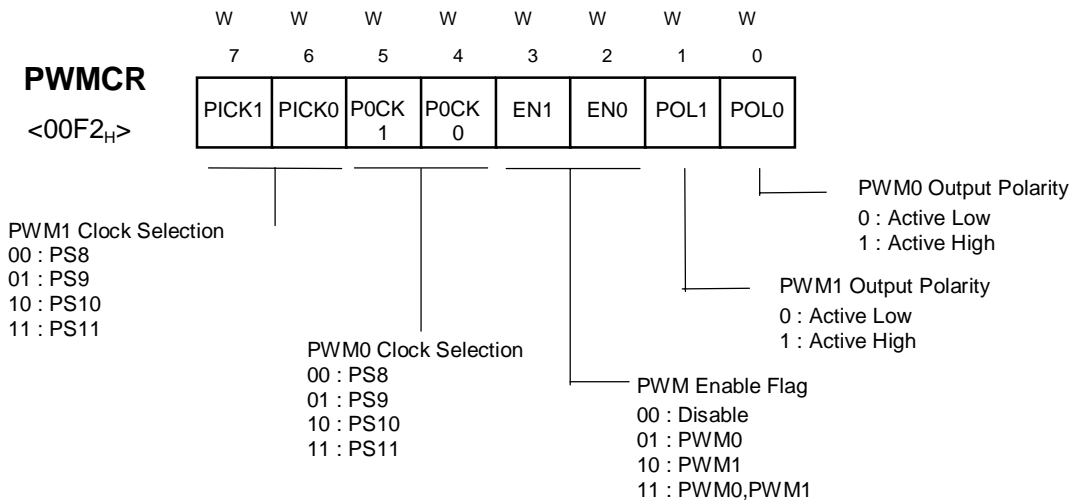
| Input clock | PWM Period |
|--------------------|----------------|
| PS8 (32 μ s) | 8,192 μ s |
| PS9 (64 μ s) | 16,384 μ s |
| PS10 (128 μ s) | 32,768 μ s |
| PS11 (256 μ s) | 65,536 μ s |

Bit2 (EN0) and bit3 (EN1) of PWM control Register (PWMCR) determine the operation channel of PWM. When EN0=0 and EN1=0, PWM does not executed. The EN0 and EN1 are Enable bit of PWM channel 0 and channel 1 respectively. When EN0=1, PWM channel0 executes. When EN1=1, PWM channel1 executes.

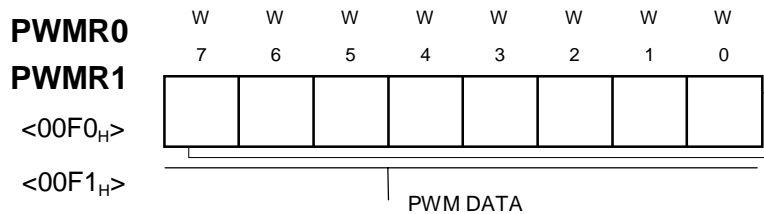
POL0 and POL1 are a polarity control bit of channel0 and channel1. When they are 0, LOW active. When 1, HIGH active. PWMCR becomes "00h" in reset state.



■ PWM CONTROL REGISTER

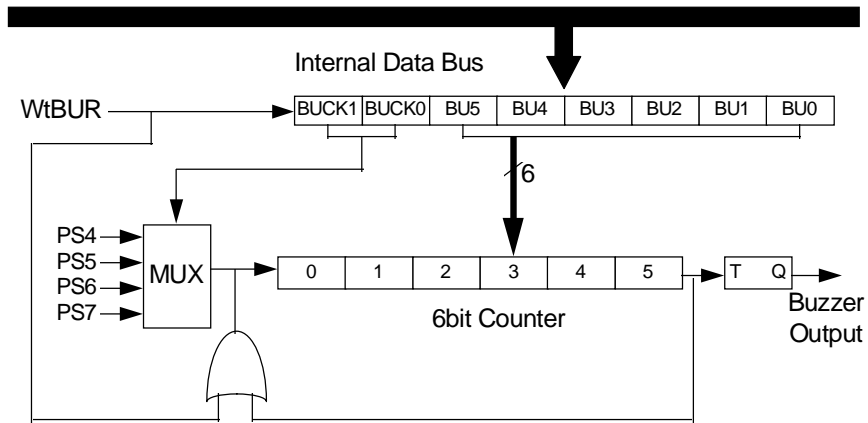


■ PWM DATA REGISTER

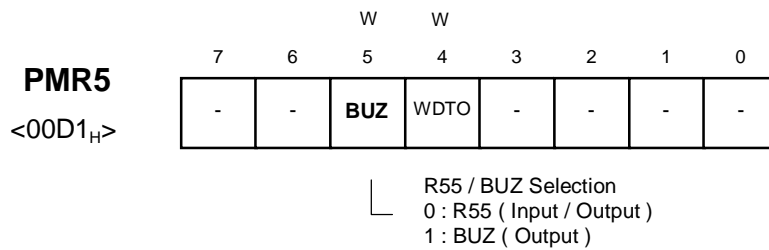


2.11. BUZZER DRIVER

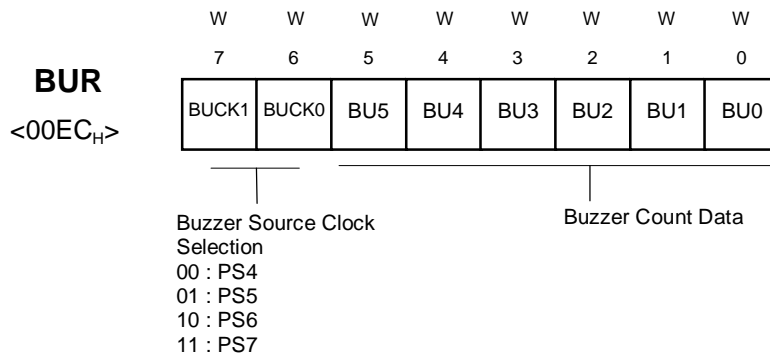
Buzzer driver consist of 6 bit binary counter, Buzzer Register(BUR), and selector of clock. The wide range frequency(500Hz~250KHz) can be generated using programmable counter. PORT R55 is assigned for output port of Buzzer Driver by setting bit5 of PMR5(\$00D1H) to "1".



■ PORT R5 MODE REGISTER



■ BUZZER DATA REGISTER



HYUNDAI MicroElectronics

2.11.1. Buzzer Driver Operation

The bit0-5 of Buzzer Register (BUR) determines output frequency for buzzer driving. The frequency is calculated as shown bellows.

$$N = \text{BUR data}$$

$$\text{freq.} = 1/(\text{source clock} \times N \times 2)$$

The bit6 and bit7 of Buzzer register (BUR) selects the source clock of the buzzer counter among PS4 (2us), PS5 (4us), PS6 (8us) and PS7 (16us).

The buzzer counter is cleared by Wt signal of BUR and starts the counting. also, It is cleared by counter overflow, and continues count-up to output the rectangular wave of duty 50%.

* Caution: don't use BUR register as 00H. (counter reset state)

◆ The output frequency of buzzer according to Buzzer Register bit5 - bit0 (fex = 8 MHz)

| REG. LOAD DEC | REG. LOAD HEX | OUTPUT FREQUENCY[KHz] | | | | REG. LOAD DEC | REG. LOAD HEX | OUTPUT FREQUENCY[KHz] | | | |
|---------------------|---------------------|-----------------------|--------------|--------------|---------------|---------------------|---------------------|-----------------------|--------------|--------------|---------------|
| | | PS4 (2us) | PS5 (4us) | PS6 (8us) | PS7 (16us) | | | PS4 (2us) | PS5 (4us) | PS6 (8us) | PS7 (16us) |
| 1 | 01 | 250 | 125 | 62.5 | 31.25 | 33 | 21 | 7.576 | 3.788 | 1.894 | 0.947 |
| 2 | 02 | 125 | 62.5 | 31.25 | 15.626 | 34 | 22 | 7.352 | 3.676 | 1.838 | 0.919 |
| 3 | 03 | 83.333 | 41.666 | 20.834 | 10.416 | 35 | 23 | 7.142 | 3.571 | 1.786 | 0.893 |
| 4 | 04 | 62.5 | 31.25 | 15.626 | 7.812 | 36 | 24 | 6.944 | 3.472 | 1.736 | 0.868 |
| 5 | 05 | 50 | 25 | 12.5 | 6.25 | 37 | 25 | 6.756 | 3.378 | 1.689 | 0.845 |
| 6 | 06 | 41.666 | 20.834 | 10.416 | 5.208 | 38 | 26 | 6.578 | 3.289 | 1.645 | 0.822 |
| 7 | 07 | 35.714 | 17.858 | 8.928 | 4.464 | 39 | 27 | 6.41 | 3.205 | 1.602 | 0.801 |
| 8 | 08 | 31.25 | 15.626 | 7.812 | 3.906 | 40 | 28 | 6.3 | 3.125 | 1.563 | 0.781 |
| 9 | 09 | 27.778 | 13.888 | 6.944 | 3.472 | 41 | 29 | 6.098 | 3.049 | 1.524 | 0.762 |
| 10 | 0A | 25 | 12.5 | 6.25 | 3.126 | 42 | 2A | 5.952 | 2.976 | 1.488 | 0.744 |
| 11 | 0B | 22.728 | 11.364 | 5.682 | 2.84 | 43 | 2B | 5.814 | 2.907 | 1.453 | 0.727 |
| 12 | 0C | 20.834 | 10.416 | 5.682 | 2.604 | 44 | 2C | 5.682 | 2.841 | 1.421 | 0.710 |
| 13 | 0D | 19.23 | 9.616 | 4.808 | 2.404 | 45 | 2D | 5.556 | 2.778 | 1.389 | 0.694 |
| 14 | 0E | 17.858 | 8.928 | 4.464 | 2.232 | 46 | 2E | 5.434 | 2.717 | 1.359 | 0.679 |
| 15 | 0F | 16.666 | 8.334 | 4.166 | 2.084 | 47 | 2F | 5.32 | 2.660 | 1.33 | 0.665 |
| 16 | 10 | 15.626 | 7.812 | 3.906 | 1.9541 | 48 | 30 | 5.208 | 2.604 | 1.302 | 0.651 |
| 17 | 11 | 14.706 | 7.352 | 3.676 | 1.838 | 49 | 31 | 5.102 | 2.551 | 1.276 | 0.638 |
| 18 | 12 | 13.888 | 6.944 | 3.472 | 1.736 | 50 | 32 | 5 | 2.5 | 1.25 | 0.625 |
| 19 | 13 | 13.158 | 6.579 | 3.288 | 1.644 | 51 | 33 | 4.902 | 2.451 | 1.225 | 0.613 |
| 20 | 14 | 12.5 | 6.25 | 3.124 | 1.562 | 52 | 34 | 4.808 | 2.404 | 1.202 | 0.601 |
| 21 | 15 | 11.904 | 5.952 | 2.976 | 1.488 | 53 | 35 | 4.716 | 2.358 | 1.179 | 0.590 |
| 22 | 16 | 11.364 | 5.682 | 2.840 | 1.420 | 54 | 36 | 4.63 | 2.315 | 1.157 | 0.579 |
| 23 | 17 | 10.87 | 5.434 | 2.718 | 1.358 | 55 | 37 | 4.546 | 2.273 | 1.136 | 0.568 |
| 24 | 18 | 10.416 | 5.208 | 2.604 | 1.302 | 56 | 38 | 4.464 | 2.232 | 1.116 | 0.558 |
| 25 | 19 | 10 | 5 | 2.5 | 1.250 | 57 | 39 | 4.386 | 2.193 | 1.096 | 0.548 |
| 26 | 1A | 9.616 | 4.808 | 2.404 | 1.202 | 58 | 3A | 4.31 | 2.155 | 1.078 | 0.539 |
| 27 | 1B | 9.26 | 4.630 | 2.314 | 1.158 | 59 | 3B | 4.238 | 2.119 | 1.059 | 0.530 |
| 28 | 1C | 8.928 | 4.464 | 2.232 | 1.116 | 60 | 3C | 4.166 | 2.083 | 1.042 | 0.521 |
| 29 | 1D | 8.62 | 4.310 | 2.156 | 1.078 | 61 | 3D | 4.098 | 2.049 | 1.025 | 0.512 |
| 30 | 1E | 8.334 | 4.166 | 2.084 | 1.042 | 62 | 3E | 4.032 | 2.016 | 1.008 | 0.504 |
| 31 | 1F | 8.064 | 4.032 | 2.016 | 1.008 | 63 | 3F | 3.968 | 1.984 | 0.992 | 0.496 |
| 32 | 20 | 7.812 | 3.906 | 1.954 | 0.976 | | | | | | |

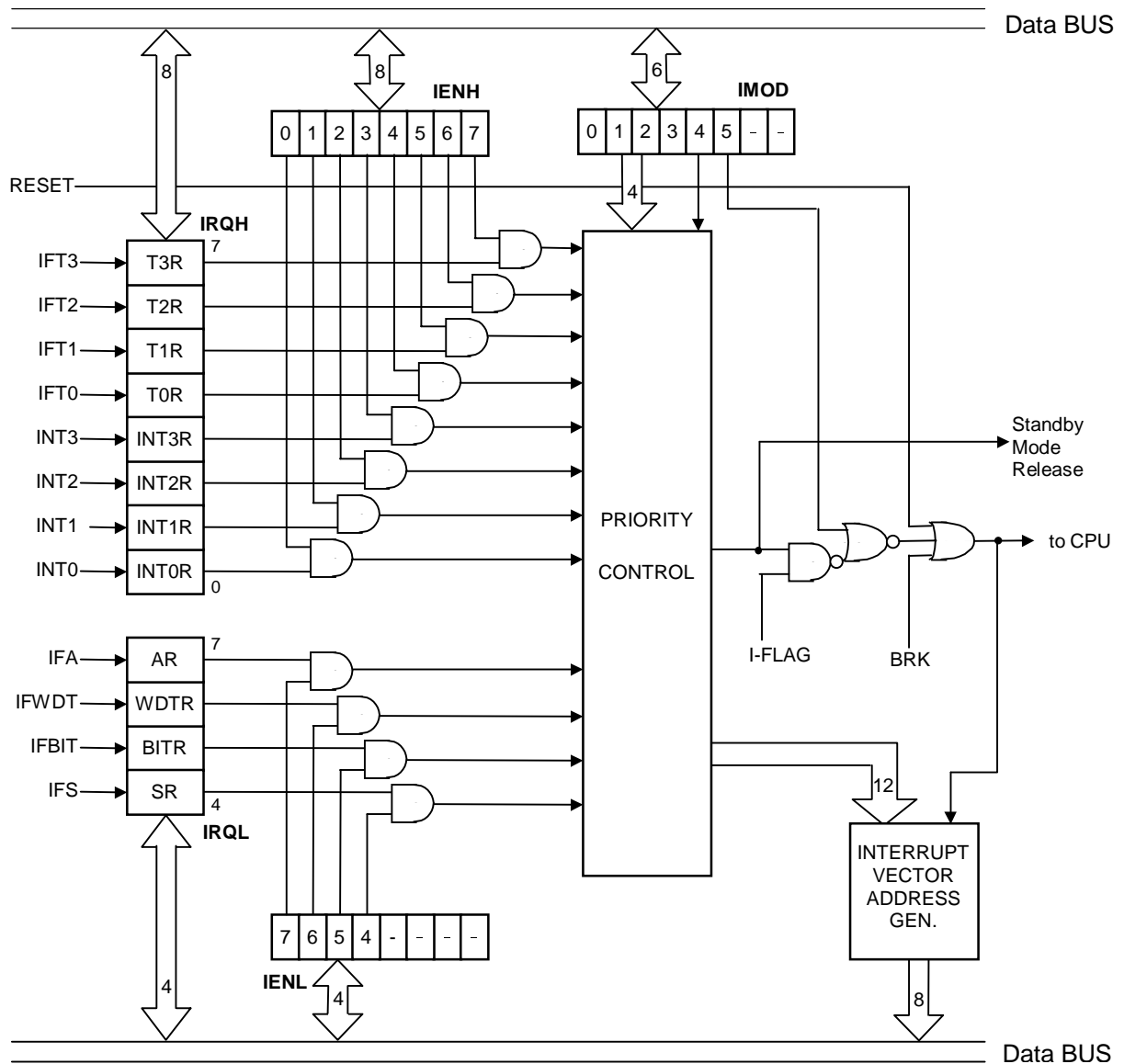
2.12. INTERRUPTS

The interrupts are usually used when the processing routine has the higher priority than on-going program and a routine must be executed at specific interval.

2.12.1. Interrupt Circuit Configuration and Kinds

GMS81508/16 Interrupt circuits consist of Interrupt Enable Register (IENH,IENL), Interrupt Request Register (IRQH,IRQL), priority circuit and selecting circuit.

The configuration of Interrupt circuit is shown in below.



■ **Interrupt Source**

The interrupts sources are external interrupt source(INT0, INT1,INT2,INT3), peripheral function source (T0,T1,T2,T3,B.I.T.,W.D.T.,SIO,A/DC) and software interrupt source(BRK).

After reset input(RESET), the program is executed from the address in reset vector table like general interrupts.

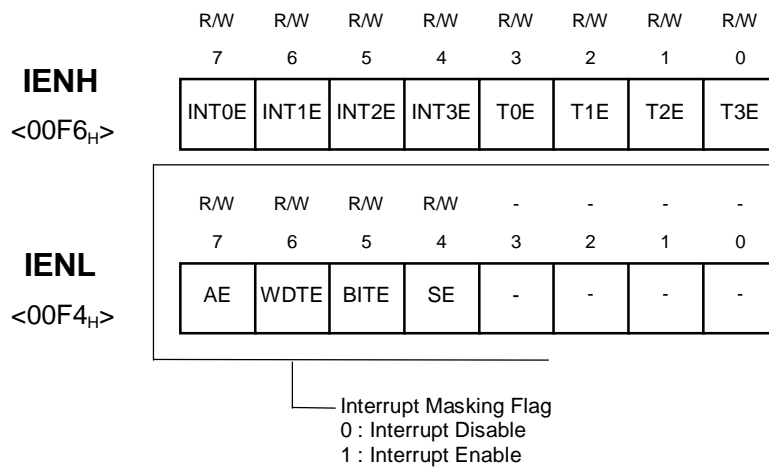
| Type | Mask | Priority | Interrupt Request Source | Vector H | Vector L |
|---------------|--------------|----------|----------------------------|-------------------|-------------------|
| H/W Interrupt | Non Maskable | 1 | RST Reset Pin | FFFF _H | FFFE _H |
| | | 2 | INT0R External Interrupt 0 | FFFB _H | FFFA _H |
| | | 3 | INT1R External Interrupt 1 | FFF9 _H | FFF8 _H |
| | | 4 | INT2R External Interrupt 2 | FFF7 _H | FFF6 _H |
| | | 5 | INT3R External Interrupt 3 | FFF5 _H | FFF4 _H |
| | | 6 | T0R Timer 0 | FFF3 _H | FFF2 _H |
| | | 7 | T1R Timer 1 | FFF1 _H | FFF0 _H |
| | | 8 | T2R Timer 2 | FFEF _H | FFEE _H |
| | | 9 | T3R Timer 3 | FFED _H | FFFC _H |
| | | 10 | AR A/D Converter | FFEB _H | FFEA _H |
| | | 11 | WDTR Watch Dog Timer | FFE9 _H | FFE8 _H |
| | | 12 | BITR Basic Interval Timer | FFE7 _H | FFE6 _H |
| | | 13 | SR Serial I/O | FFE5 _H | FFE4 _H |
| S/W Interrupt | Non Maskable | — | BRK Break Instruction | FFDF _H | FFDE _H |

2.12.2. Interrupt Control

The interrupts is controlled by the interrupt master enable flag I-Flag(3'rd bit of PSW), interrupt enable register(IENH,IENL), interrupt request register(IRQH,IRQL) except RESET and S/W interrupt.

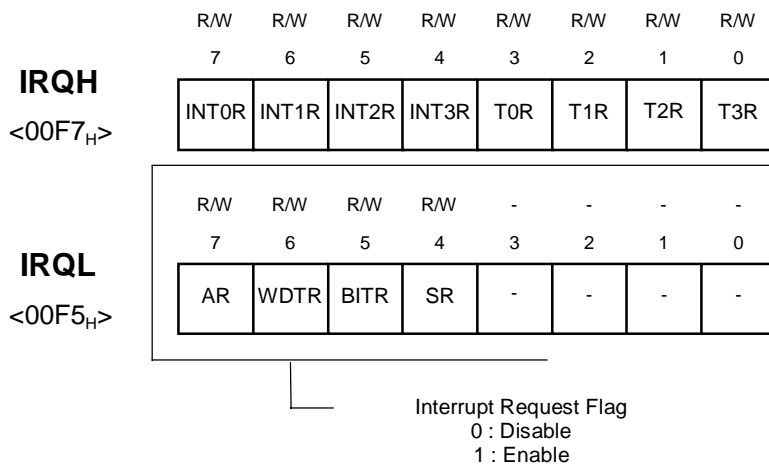
■ **Interrupt Enable Register (IENH, IENL)**

This register is composed of interrupt enable flags of each interrupt source, this flags determines whether an interrupt will be accepted or not. when enable flag is "0", an interrupt corresponding interrupt source is prohibited.



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

Whenever interrupt request is generated, the interrupt request flag is set. The request flag maintains '1' until interrupt is accepted. The accepted interrupt request flag is automatically cleared by interrupt process cycle. Interrupt Request Flag Register (**IRQH**, **IRQL**) is Read/ Write Register. So, it is possible to be checked and changed by program.



2.12.3. Interrupt Priority

When two or more interrupts requests are generated at the same sampling point, the interrupt having the higher priority is accepted. The interrupt priority is determined by H/W. however, multiple priority processing through software is possible by using interrupt control flags(IENH, IENL, I-flag) and interrupt mode register(IMOD).

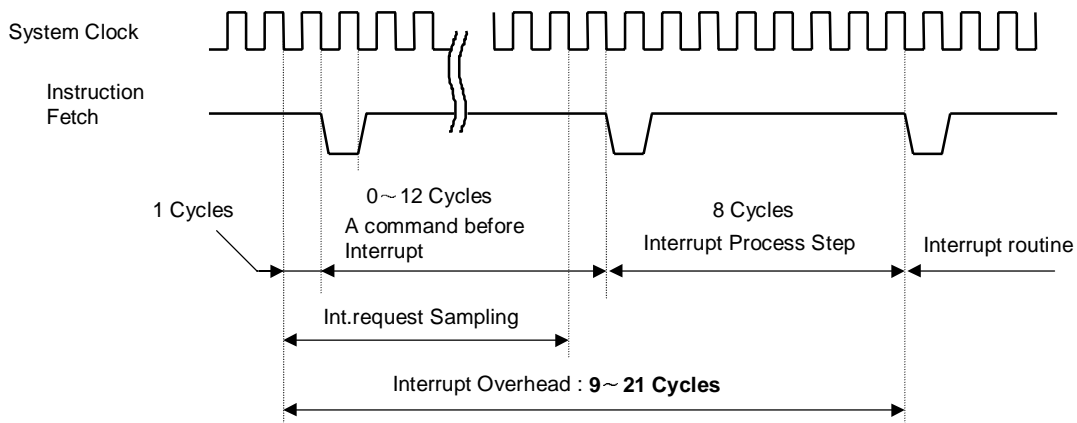
2.12.4. Interrupt Sequence

When interrupt is accepted, the on-going process is stopped and the interrupt service routine is executed. After the interrupt service routine is completed it is necessary to restore everything to the state before the interrupt occurred.

As soon as an interrupt is accepted, the contents of the program counter and the program status word are saved in the stack area. At the same time, the contents of the vector address corresponding to the accepted interrupt, which is in the interrupt vector table, enters into the program counter and interrupt service routine is executed.

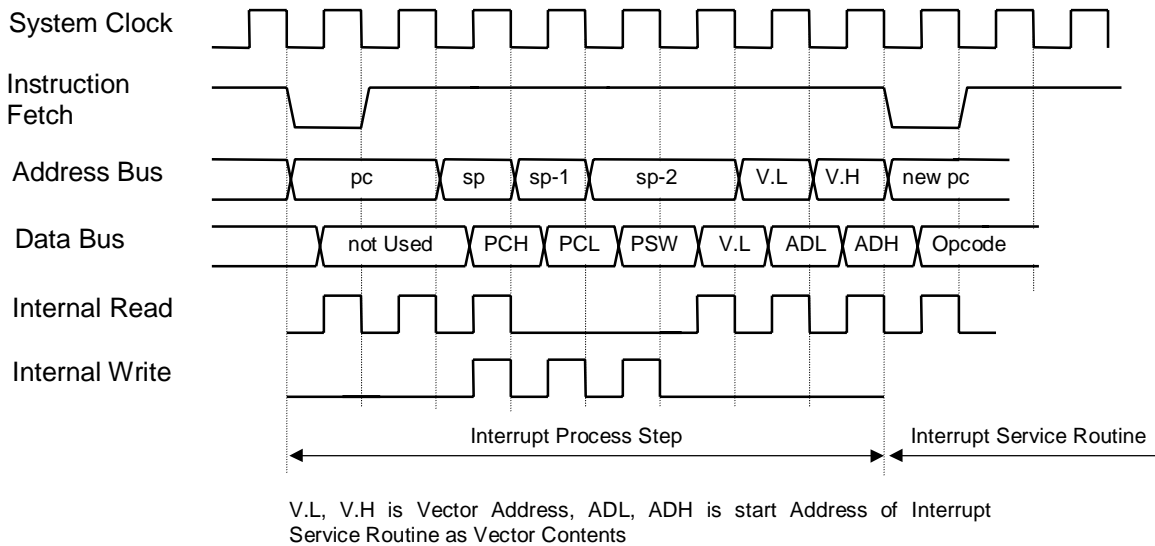
In the interrupt service routine, the corresponding interrupt request flag is cleared and interrupt master enable flag(I-flag) becomes "0", thereby another interrupts are not accepted before I-flag is set to "1" by program.

In order to execute the interrupt service routine, it is necessary to write the jump address(the first address of the interrupt service routine) in vector table corresponding to each interrupt.



◆ **Interrupt Accept Timing**

- The valid timing after executing Interrupt control Flag
 - ⓐ I-Flag is valid, after EI, DI executed
 - ⓑ IENH, IENL register is valid after next instruction

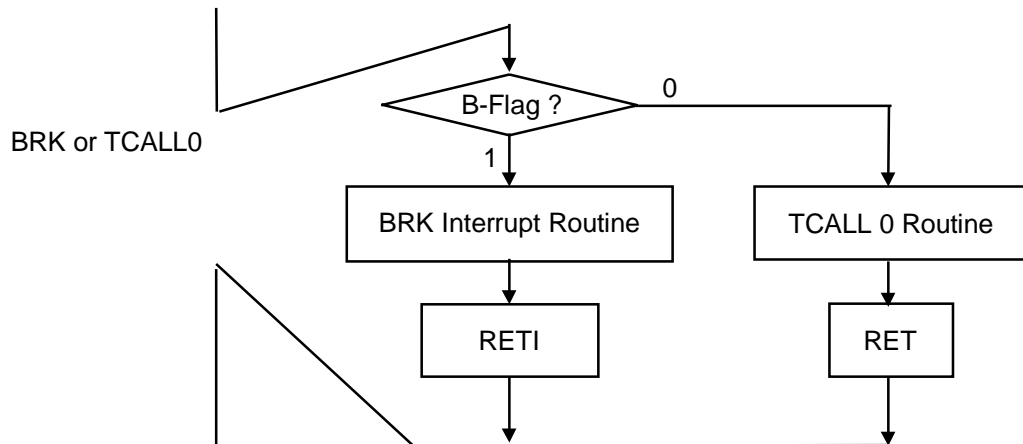


◆ **Interrupt Process Step Timing**

2.12.5. Software Interrupt

The interrupt is the lowest priority order software interrupt by BRK instruction. B-flag is set.

Interrupt vector of BRK instruction is shared with the vector of TCALL 0. Each processing step is determined by B-Flag as a below.



◆ **Execution of BRK/ TCALL0**

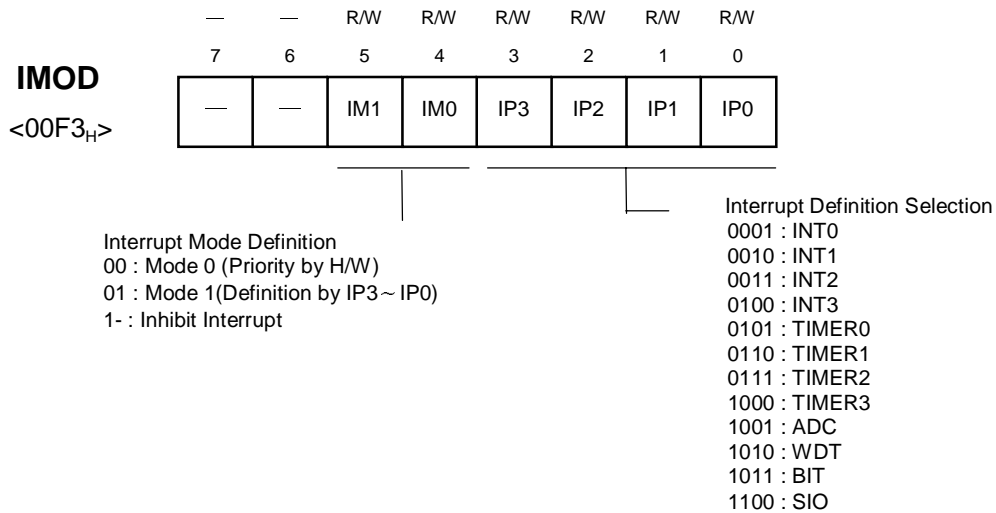
2.12.6. Multiple Interrupt

When an interrupt is accepted, and program flow goes to the interrupt service routine. The interrupt master enable flag(I-flag) is automatically cleared and other interrupts are inhibited. When interrupt service is completed by RETI instruction, I-flag is set automatically. If other interrupts are generated during interrupt service, The interrupt having higher priority is accepted when the previous interrupt service routine is completed.

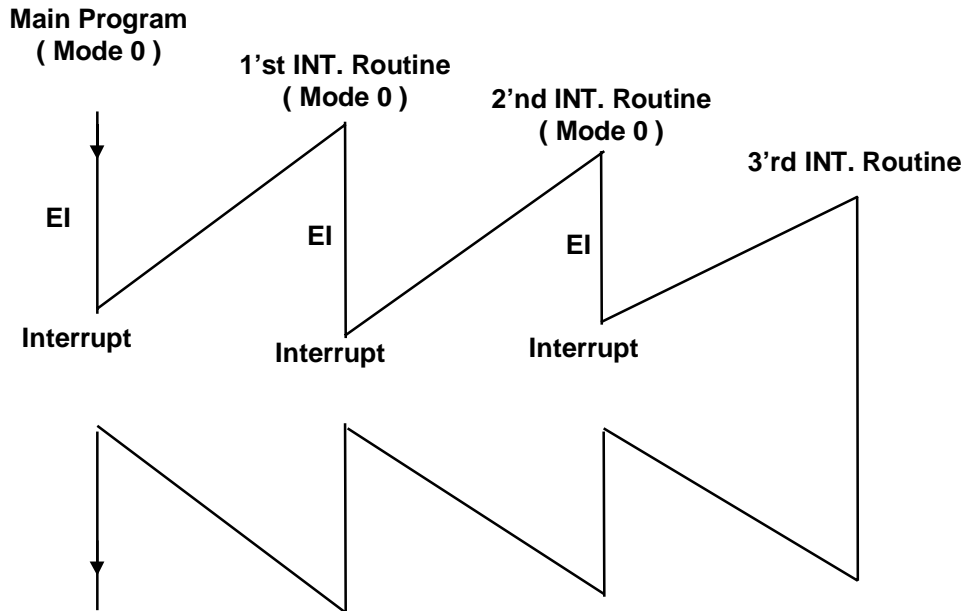
In order to multiple interrupts, I-flag must be cleared by EI instruction within the interrupt routine. Then, The higher priority interrupt is accepted among the interrupts that interrupt request flag is "1".

■ **Interrupt Mode Register (IMOD)**

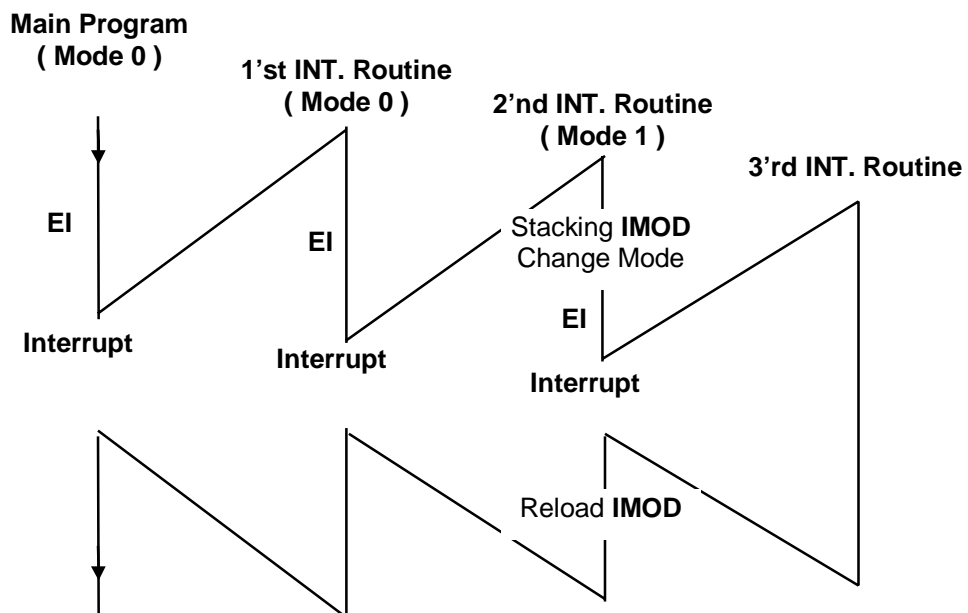
if IM1,IM0 is selected as "01", the interrupt selected by IP0~IP3 can be accepted and other interrupts are not accepted. Using this register, we can change the interrupt priority order by s/w.



When multiple interrupt is accepted, it is possible to change Interrupt Accept Mode.
 In case of multiple interrupt at hardware priority accept mode(Mode0)



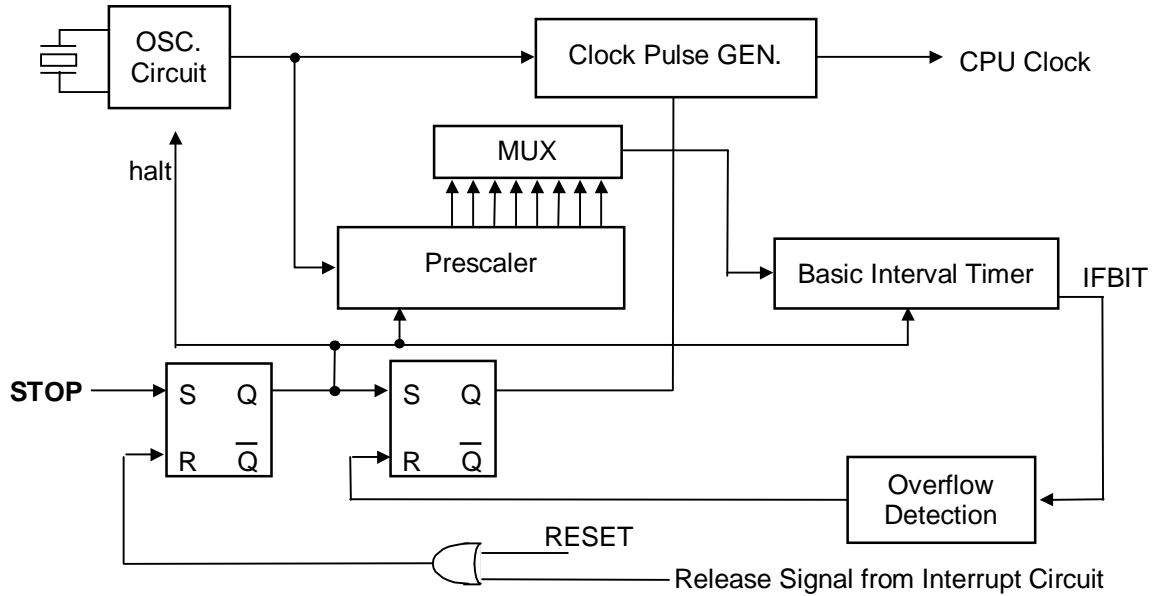
In case of multiple interrupts nest H/W priority accept mode (Mode0) and S/W selection accept mode(Mode1)



2.13. STANDBY FUNCTION

To save the consuming power of device, GMS81508/16 has STOP Mode.

In this mode, the execution of program is stopped. Stop Mode entered by STOP instruction.



◆ 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/DC,PWM, Buzzer Driver | × |
| Address Bus, Data Bus | Retain |
| \overline{Rd} , \overline{Wt} , $\overline{R/W}$ | Retain |
| \overline{HALT} , \overline{BRQ} , \overline{BAK} | Active |
| C | "L" level |
| \overline{SYNC} | "H" level |

2.13.1. 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 requiring much less power consumption. All register and RAM data are preserved.

Caution) NOP instruction have to be written more than 2 to next lines of STOP instruction.

2.13.2. STOP Mode Release

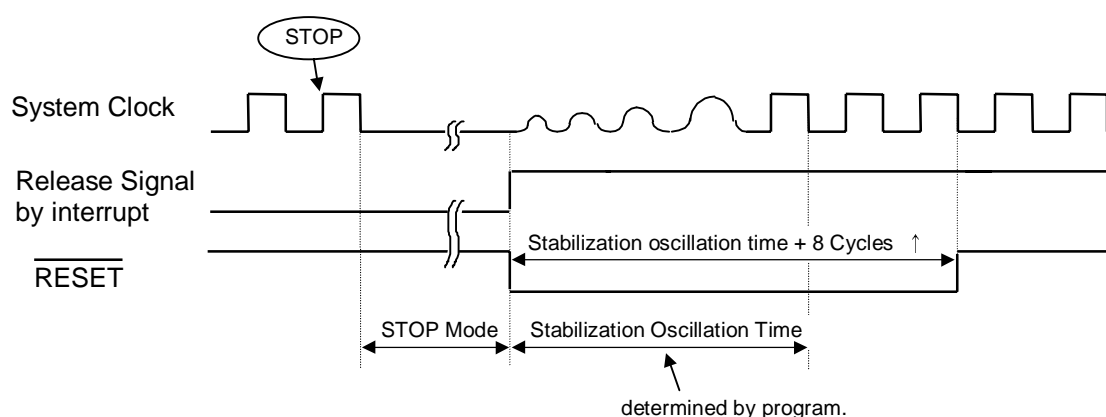
The release of STOP mode is done by reset input or interrupt. When there is a release signal of STOP mode, the instruction execution is started after stabilization oscillation time set by program. After releasing STOP mode, instruction execution is different by I-Flag(bit 2 of PSW).

If I-Flag = "1" entered Interrupt Service Routine,

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

◆ STOP Mode Release

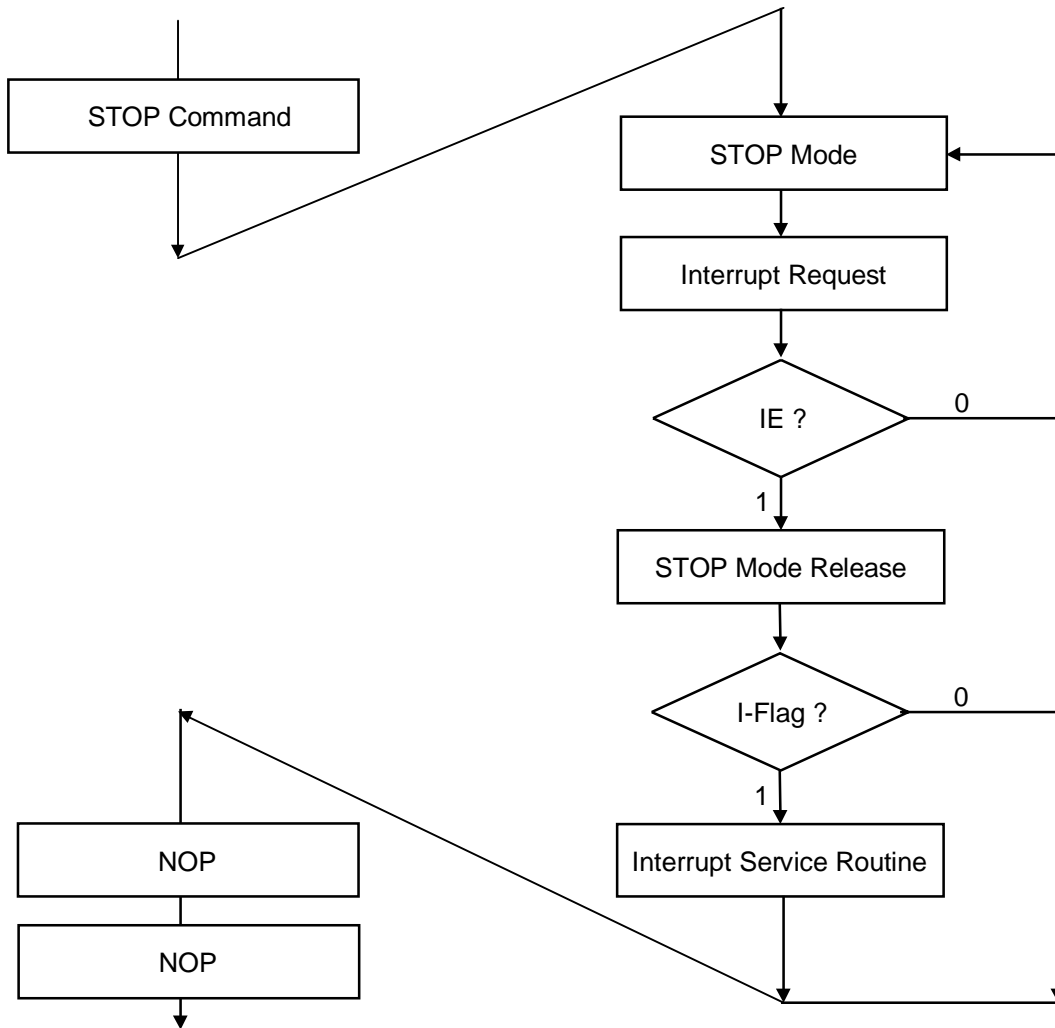
| Release Factor | Release Method |
|------------------------|--|
| RESET | By RESET pin=Low level, and Device is initialized. |
| INT0,INT1 INT2,INT3 | In the state of enable flag=1 corresponding to each interrupt at the edge. |
| Serial I/O | When Serial I/O is executed by external clock, STOP mode is released. |



◆ Release Timing of STOP Mode

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



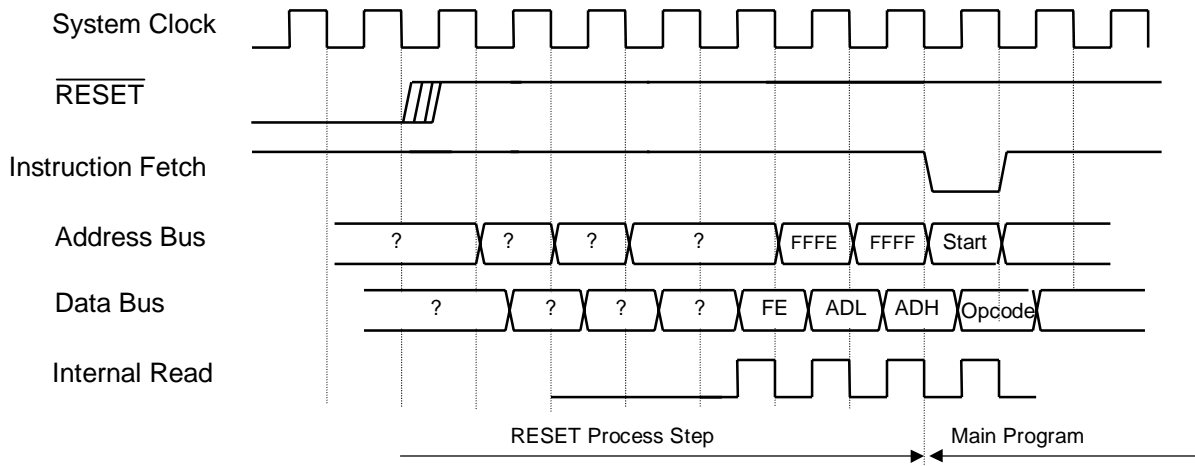
◆ STOP Mode Releasing Flow

2.14. RESET FUNCTION

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

$\overline{\text{RESET}}$ terminal is organized as schmitt input.

If initial value is undefined, it is needed initialize by a S/W.



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

◆ RESET Operation Timing

3. I/O PORTS

There are 7-ports(R0~R6) in this device. This ports are double-functional ports and the function can be selected by program.

The direction of ports is determined by Port Direction Register.(1=output, 0=input) The data that is written on the programmed output pin is stored in the port data register and is transferred to the output pin. When data is input to the programmed pin. data is read not from output pin but from port data register. therefore, previously output data can be read correctly regardless or the logical level of the pin due to output loading.

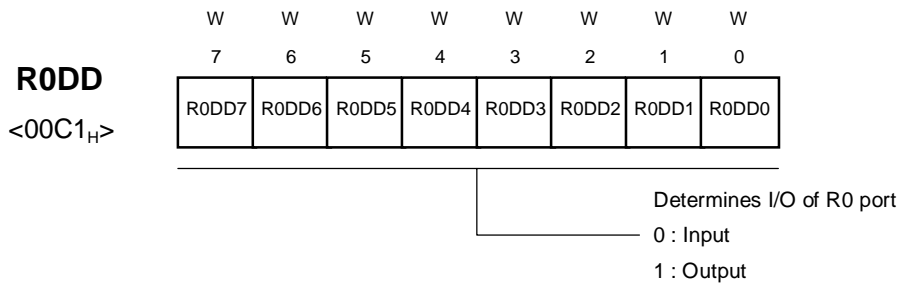
Because the programmed input pin is floating, the value of the pin can be read correctly. When data is written to the programmed input pin, it is written only to the port data register and the pin remains floating.

3.1. R0 PORT

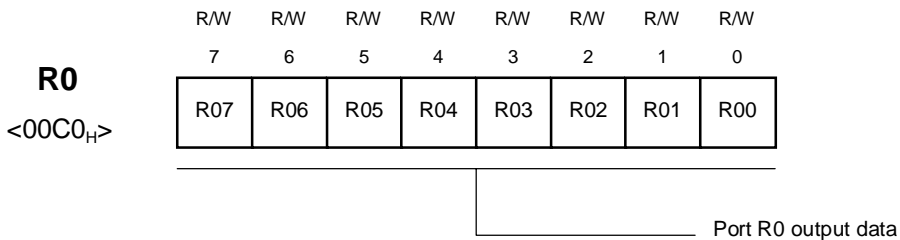
R0 Port is composed of 8-bit programmable I/O pin.

| 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



■ R0 PORT DATA REGISTER



■ Pin Function According to Operation Modes

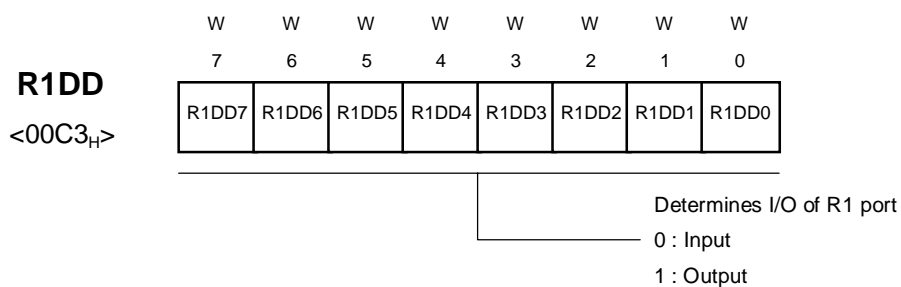
| PIN | Single Chip Mode | | Microprocessor Mode | |
|--------|------------------|-----------------------|---------------------|--|
| R00/D0 | I/O | Programmable I/O Port | I/O | Data I/O Port from/to External Memory for CPU. |
| R01/D1 | I/O | | I/O | |
| R02/D2 | I/O | | I/O | |
| R03/D3 | I/O | | I/O | |
| R04/D4 | I/O | | I/O | |
| R05/D5 | I/O | | I/O | |
| R06/D6 | I/O | | I/O | |
| R07/D7 | I/O | | I/O | |

3.2. R1 PORT

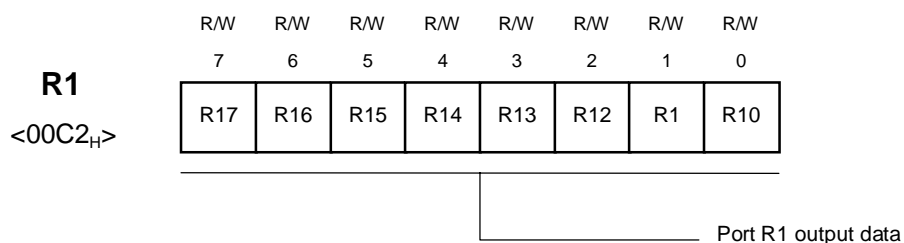
R1 Port is composed of 8-bit programmable I/O pin.

| Register Name | Symbol | R/W | Address | Initial Value |
|---------------------------|--------|-----|-------------------|-----------------|
| R0 I/O Direction Register | R1DD | W | 00C3 _H | 0000 0000 |
| R0 PORT Data Register | R1 | R/W | 00C2 _H | Not initialized |

■ R1 PORT I/O DIRECTION REGISTER



■ R1 PORT DATA REGISTER



■ Pin Function According to Operation Modes

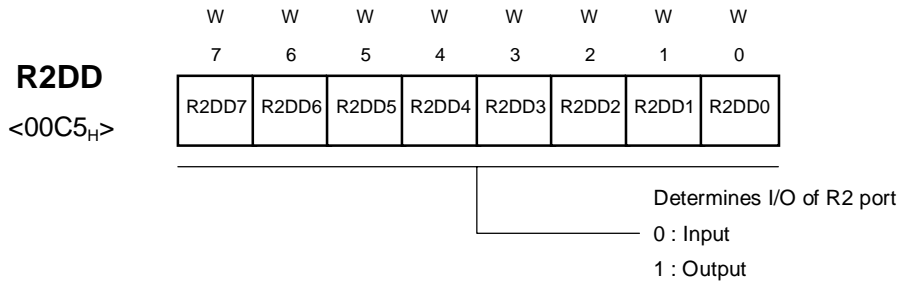
| PIN | Single Chip Mode | | Microprocessor Mode | |
|--------|------------------|-----------------------|---------------------|--|
| R10/A0 | I/O | Programmable I/O Port | O | low 8bit address of External Memory for CPU. |
| R11/A1 | I/O | | O | |
| R12/A2 | I/O | | O | |
| R13/A3 | I/O | | O | |
| R14/A4 | I/O | | O | |
| R15/A5 | I/O | | O | |
| R16/A6 | I/O | | O | |
| R17/A7 | I/O | | O | |

3.3. R2 PORT

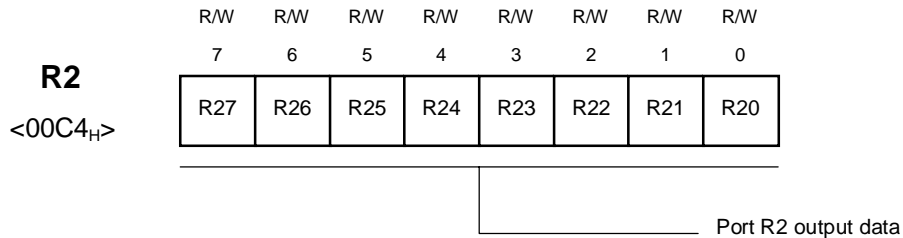
R2 Port is composed of 8-bit programmable I/O pin.

| 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 |

■ R2 PORT I/O DIRECTION REGISTER



■ R2 PORT DATA REGISTER



■ Pin Function According to Operation Modes

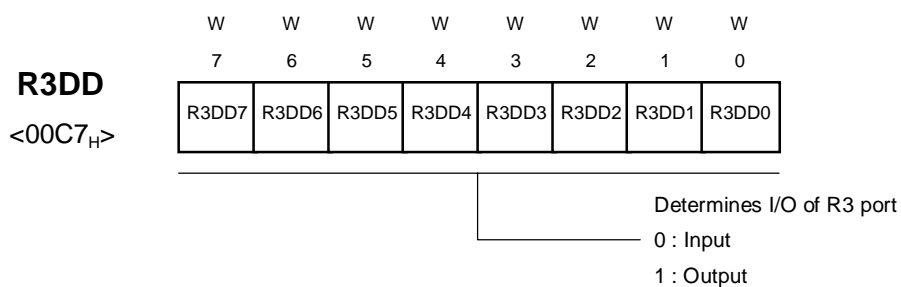
| PIN | Single Chip Mode | | Microprocessor Mode | |
|---------|------------------|-----------------------|---------------------|--|
| R20/A8 | I/O | Programmable I/O Port | O | upper 8bit address of External Memory for CPU. |
| R21/A9 | I/O | | O | |
| R22/A10 | I/O | | O | |
| R23/A11 | I/O | | O | |
| R24/A12 | I/O | | O | |
| R25/A13 | I/O | | O | |
| R26/A14 | I/O | | O | |
| R27/A15 | I/O | | O | |

3.4. R3 PORT

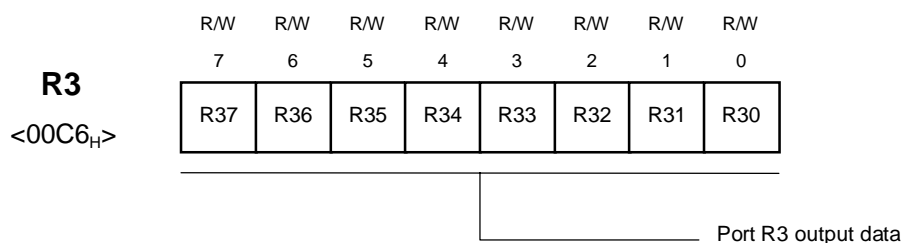
R3 Port is composed of 8-bit programmable I/O pin.

| 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 |

■ R3 PORT I/O DIRECTION REGISTER



■ R3 PORT DATA REGISTER



■ Pin Function According to Operation Modes

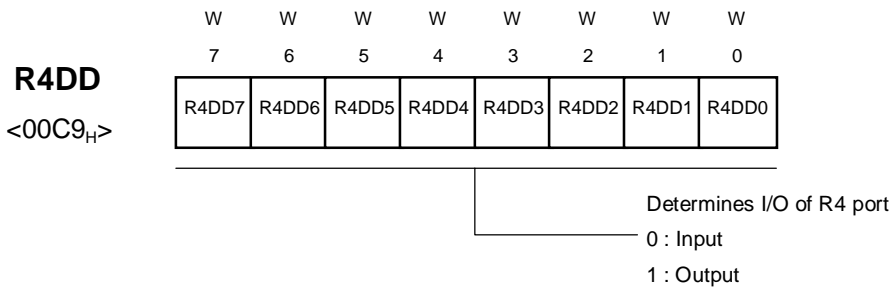
| PIN | Single Chip Mode | | Microprocessor Mode | |
|-----|------------------|-----------------------|---------------------|--|
| R30 | I/O | Programmable I/O Port | O | \overline{Rd} : external memory read strobe |
| R31 | I/O | | O | \overline{Wt} : external memory write strobe |
| R32 | I/O | | O | $\overline{R/W}$:Read/Write cycle output pin of CPU |
| R33 | I/O | | O | C : timing signal output pin |
| R34 | I/O | | O | \overline{SYNC} : op code fetch output pin of CPU |
| R35 | I/O | | O | \overline{BRK} : bus acknowledge output pin of CPU |
| R36 | I/O | | I | \overline{BRQ} : bus request input pin of CUP |
| R37 | I/O | | I | \overline{HALT} : CPU halt input pin |

3.5. R4 PORT

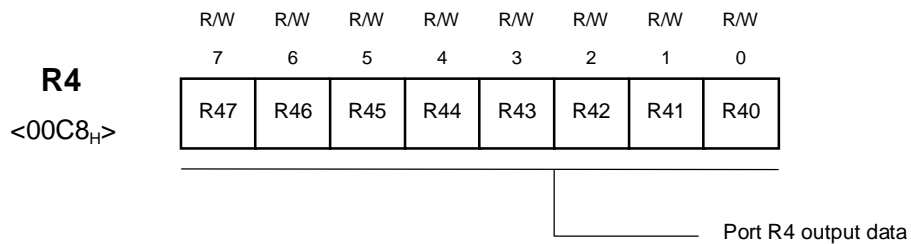
R4 Port is composed of 8bit programmable I/O port and this port are double functional pin.

| Register Name | Symbol | R/W | Address | Initial Value |
|--------------------------------|--------|-----|-------------------|-----------------|
| R4 I/O Direction Register | R4DD | W | 00C9 _H | 0000 0000 |
| R4 Port Data Register | R4 | R/W | 00C8 _H | Not initialized |
| Port R4 Mode Register | PMR4 | W | 00D0 _H | 0000 0000 |
| Interrupt Edge Select Register | IEDS | R/W | 00F8 _H | 0000 0000 |

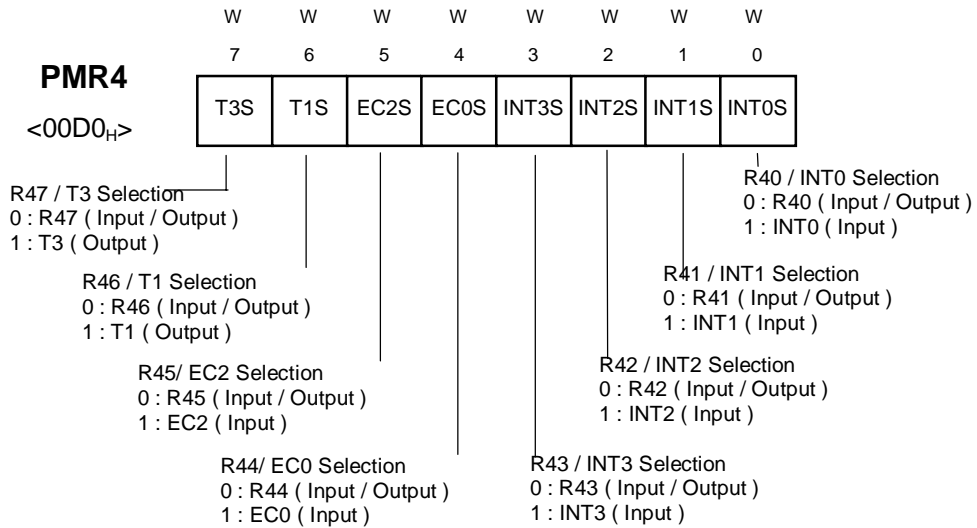
■ R4 PORT I/O DIRECTION REGISTER



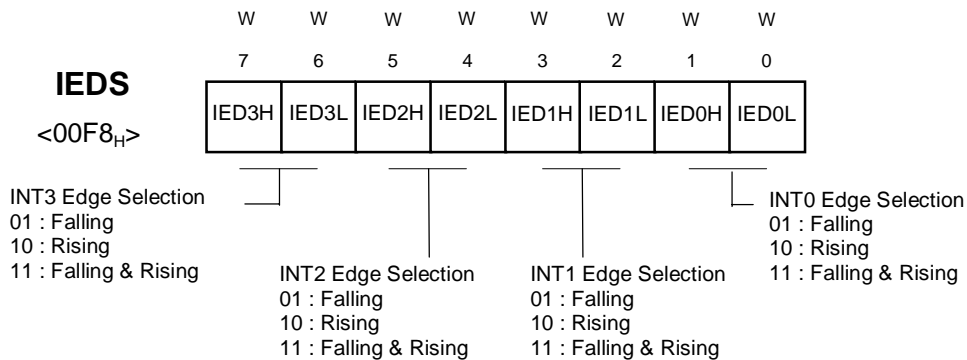
■ R4 PORT DATA REGISTER



■ PORT R4 MODE REGISTER



■ INTERRUPT EDGE SELECTION REGISTER

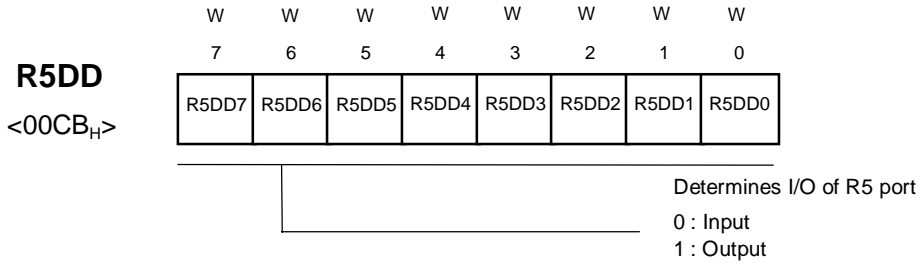


3.6. R5 PORT

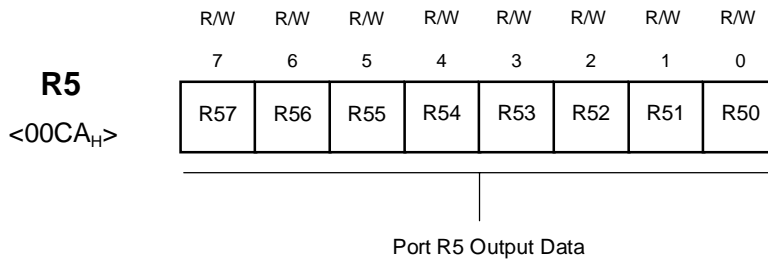
R5 Port is composed of 8-bit programmable I/O port. R54,R55 is double functional pin.

| Register Name | Symbol | R/W | Address | Initial Value |
|---------------------------|--------|-----|-------------------|-----------------|
| R5 I/O Direction Register | R5DD | W | 00CB _H | 0000 0000 |
| R5 Port Data Register | R5 | R/W | 00CA _H | Not initialized |
| R5 Port Mode Register | PMR5 | W | 00D1 _H | --00 ---- |

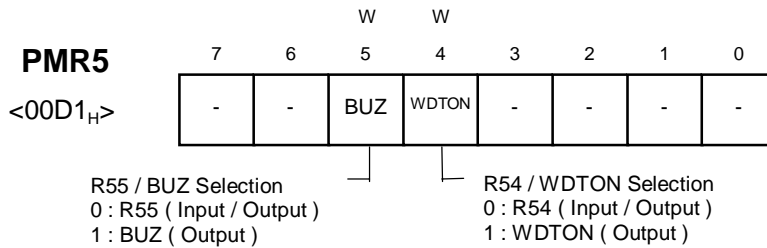
■ R5 PORT I/O DIRECTION REGISTER



■ R5 PORT DATA REGISTER



■ PORT R5 MODE REGISTER

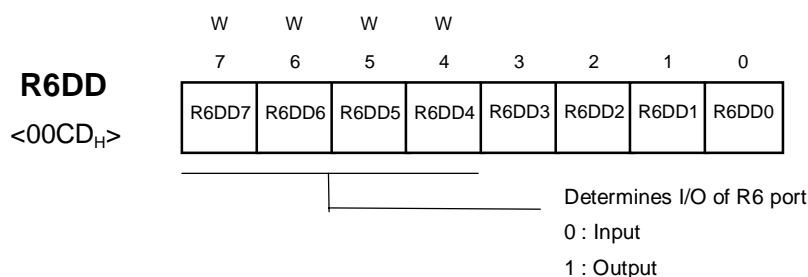


3.7. R6 PORT

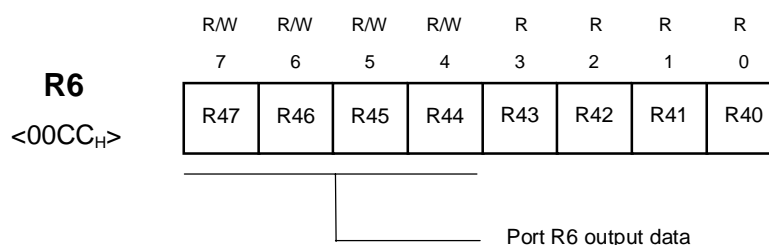
R6 Port consists of 4-bit Programmable I/O ports and 4-bit input only ports and this port can be used as a analog input port for A/D conversion by program.

| Register Name | Symbol | R/W | Address | Initial Value |
|-----------------------------|--------|-----|-------------------|-----------------|
| R6 I/O Direction Register | R6DD | W | 00CD _H | 0000 ---- |
| R6 Port Data Register | R6 | R/W | 00CC _H | Not initialized |
| A/D Converter Mode Register | ADCM | W | 00E8 _H | --00 0001 |

■ R6 PORT I/O DIRECTION REGISTER



■ R6 PORT DATA REGISTER



On the initial RESET, R60 can't be used digital input port, because this port is selected as an analog input port by ADCM register. To use this port as a digital I/O port, change the value of lower 4 bits of ADCM(address 0E8H). On the other hand, R6 port, all eight pins can not be used as digital I/O port simultaneously. At least one pin is used as an analog input.

■ UNUSED PORTS

All unused ports should be set properly that current flow through port doesn't exist.

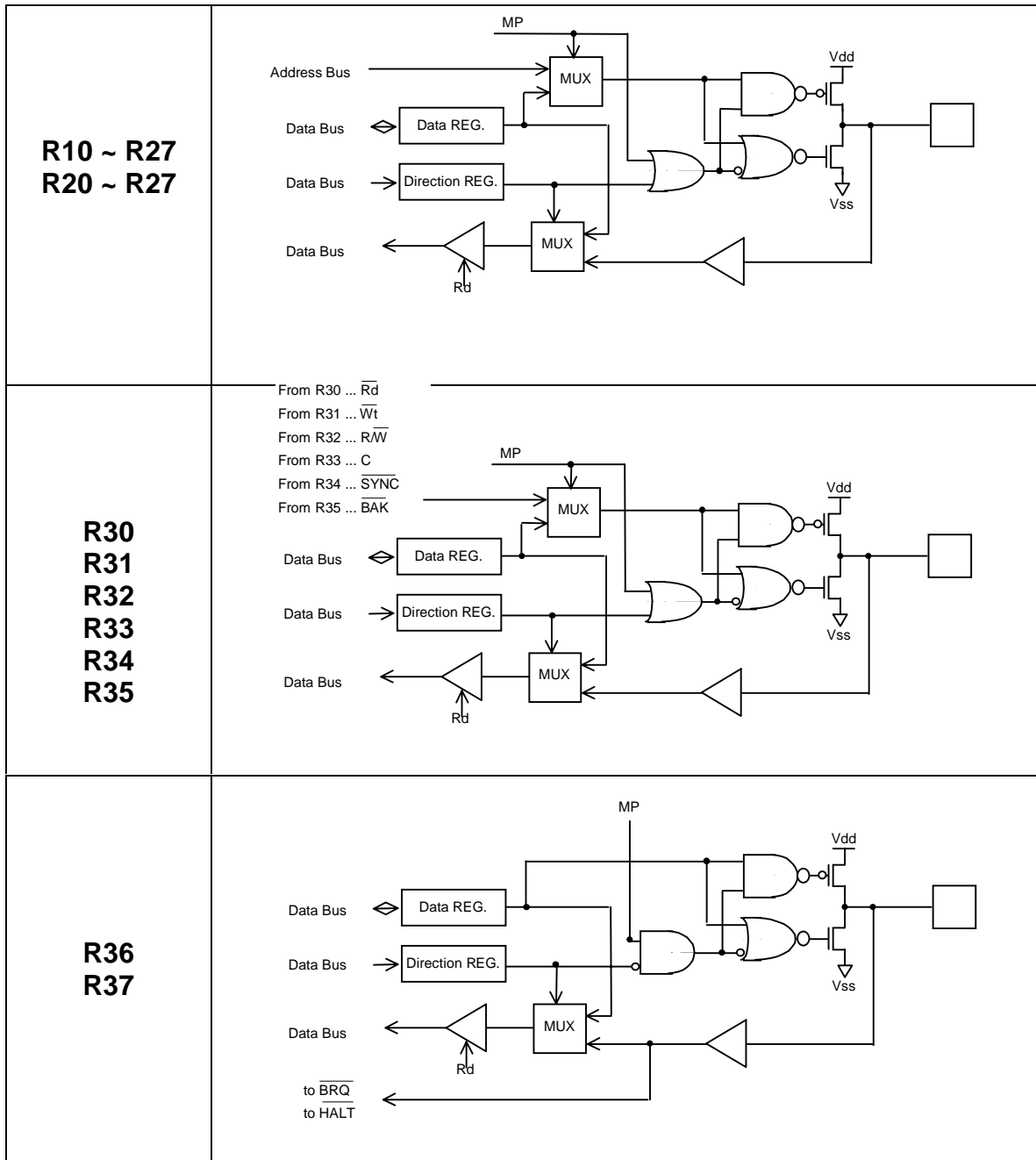
First consider the setting the port as an input mode. Be sure that there is no current flow after considering its relationship with external circuit. In input mode, the pin impedance viewing from external MCU is very high that the current doesn't flow.

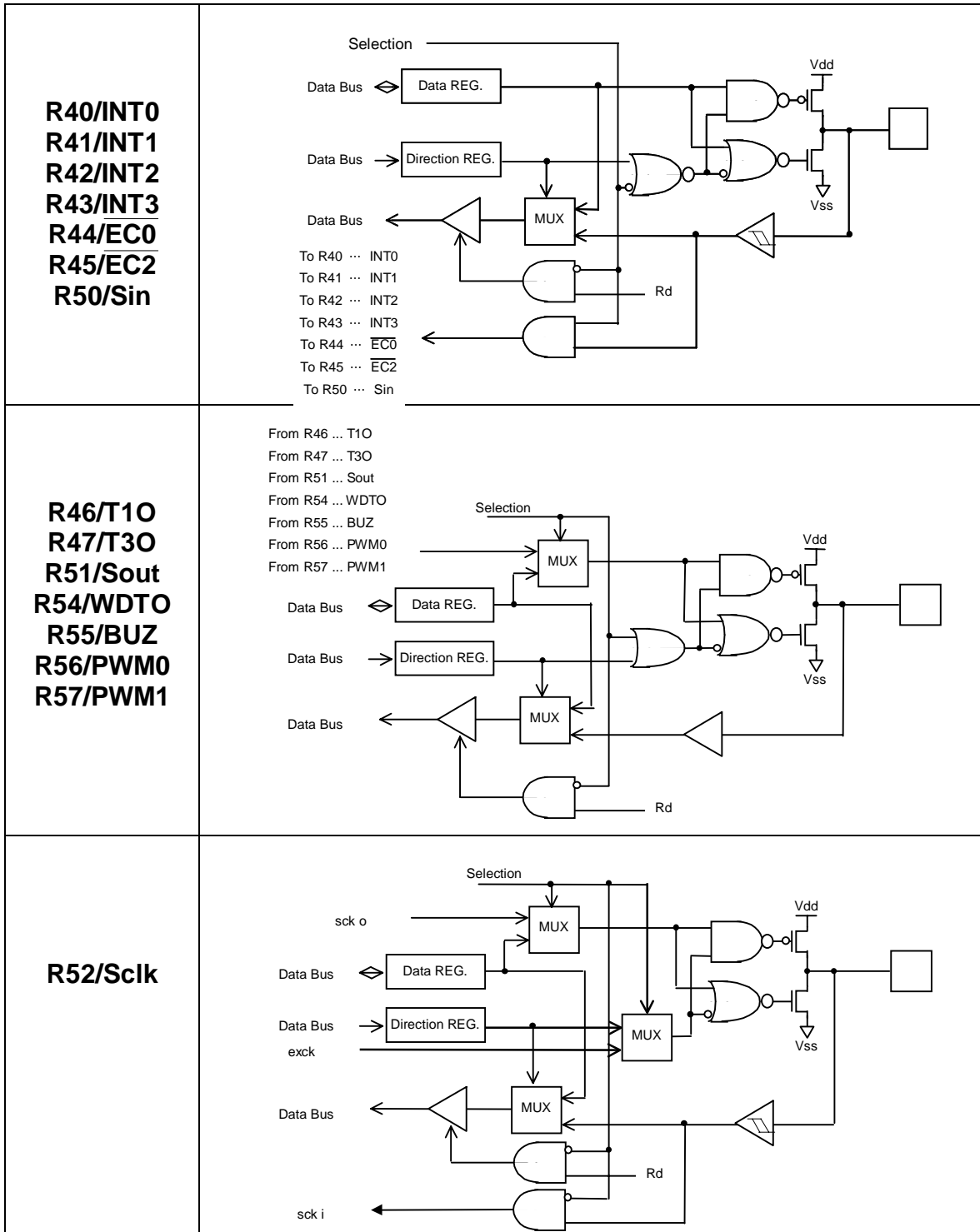
But input voltage level should be V_{SS} or V_{DD} . Be careful that if unspecified voltage, i.e. if unfirmed level voltage is applied to input pin, there can be little current (max. 1mA at 2V) flow.

If it is not appropriate to set as an input mode, then set to output mode considering there is no current flow. Setting to High or Low is decided considering its relationship with external circuit. For example, if there is external pull-up resistor then it is set to output mode, i.e. to High, and if there is external pull-down register, it is set to low.

3.8. TERMINAL TYPES

| PIN | TERMINAL TYPE |
|-----------------------------------|---------------|
| <p>Xin Xout</p> | |
| <p>RESET</p> | |
| <p>MP</p> | |
| <p>R00 ~ R07</p> | |





4. ELECTRICAL CHARACTERISTICS

4.1. ABSOLUTE MAXIMUM RATINGS

| Parameter | Symbol | Unit | Ratings |
|---------------------|--------|------|----------------|
| Supply Voltage | Vdd | V | -0.3 ~ 7.0 |
| Input Voltage | Vi | V | -0.3 ~ Vdd+0.3 |
| Storage Temperature | Tstg | °C | -40 ~ 125 |

4.2. RECOMMENDED OPERATING CONDITIONS

| Parameter | Symbol | Unit | Specifications | | |
|-----------------------|--------|------|----------------|------|------|
| | | | Min. | Typ. | Max. |
| Supply Voltage | Vdd | V | 4.5 | | 5.5 |
| Operating Frequency | fXin | MHz | 1 | | 8 |
| Operating Temperature | Topr | °C | -20 | | 85 |

4.3. A/D CONVERTER CHARACTERISTICS

(Vdd = 5V ± 10%, Vss = 0V , , f (Xin) = 8 MHz)

| Parameter | Pin | Symbol | Unit | SPECIFICATION | | | ETC |
|---------------------------------|---------|--------|------|---------------|------|------|-----|
| | | | | Min. | Typ. | Max. | |
| Analog Input Range | AN0~AN7 | VAIN | V | Vss | | Vref | |
| Accuracy | | | LSB | | | ± 3 | |
| Conversion Time | | Tconv | μs | | | 20 | |
| Analog Power Supply Input Range | AVref | Vref | V | | | Vdd | |

4.4. DC CHARACTERISTICS

(Vdd = 5.0V ± 10%, Vss = 0V, Ta = -20 ~ 85°C, f (Xin) = 8 MHz)

| Parameter | Symbol | Pin | Test Condition | Unit | Specifications | | |
|---------------------------|-----------|---|-----------------|------|----------------|------|---------|
| | | | | | Min. | Typ. | Max. |
| "H" Input voltage | Vih | $\overline{\text{RESET}}, \text{R4, R5, R6}$ | | V | 0.8Vdd | | Vdd |
| | | R0, R1, R2, R3 | | | 0.7Vdd | | Vdd |
| | | Xin | | | 0.9Vdd | | Vdd |
| "L" Input voltage | Vil | $\overline{\text{RESET}}, \text{R4, R5, R6}$ | | V | 0 | | 0.12Vdd |
| | | R0, R1, R2, R3 | | | 0 | | 0.3Vdd |
| | | Xin | | | 0 | | 0.1Vdd |
| "H" Input Leakage Current | lih | all input pins | Vi = Vdd | μA | -5 | | 5 |
| "L" Input Leakage Current | lil | all input pins | Vi = Vss | μA | -5 | | 5 |
| "H" output Voltage | Voh | R0, R1, R2, R3, R4, R5 | Ioh = -2mA | V | Vdd-1 | | |
| "L" output Voltage | | R0, R1, R2, R3, R4, R5 | Iol = 5mA | V | | | 1.0 |
| Power Current | Operating | Idd | all input = Vss | mA | | 20 | 40 |
| | STOP | Istop | | | μA | | 20 |
| Hysteresis | VT+ ~ VT- | $\overline{\text{RESET}},$ EC2, EC0, Sin, Sclk, INT0~3 | | V | 0.3 | | 0.8 |
| | | | | | 0.3 | | 0.8 |
| RAM Data Retention | Vram | Vdd | at clock stop | V | 2.0 | | |

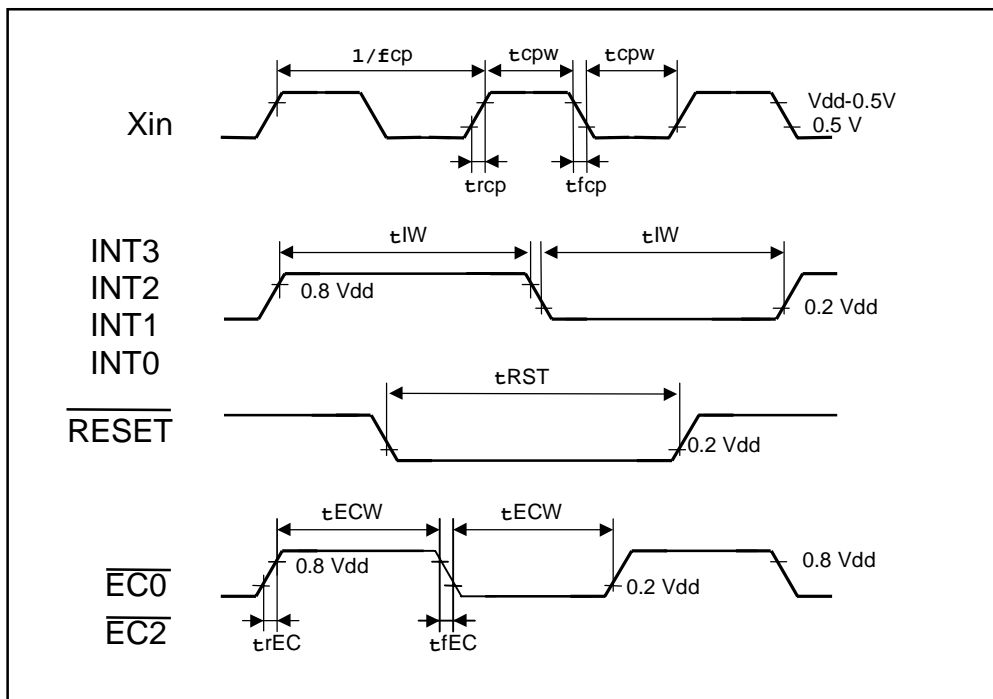
4.5. AC CHARACTERISTICS

4.5.1. Input Conditions

(Vdd = 5.0V±10%, Vss = 0V, Ta = -20 ~ 85°C, f (Xin) = 8 MHz)

| Parameter | Pin | Symbol | Unit | SPECIFICATION | | | ETC |
|---------------------------------|-----------|------------|------|---------------|------|------|-----|
| | | | | MIN. | TYP. | MAX. | |
| Operating Frequency | Xin | fcp | MHz | 1 | - | 8 | |
| System Clock | | tsys | ns | 500 | - | 250 | |
| Oscillation Stabilization Time | Xin, Xout | tst | ms | | | 20 | |
| External Clock Pulse Width | Xin | tcpw | ns | 100 | | | |
| External Clock Transition Time | Xin | trcp, tfcp | ns | | | 20 | |
| Interrupt Pulse Width | INT0~INT3 | tiw | tsys | 2 | | | |
| RESET Input "L" Width | RESET | trst | tsys | 8 | | | |
| Event Counter Input Pulse Width | EC0, EC2 | tecw | tsys | 2 | | | |
| Event Counter Transition Time | EC0, EC2 | trEC, tfEC | ns | | | 20 | |

■ Timing Chart

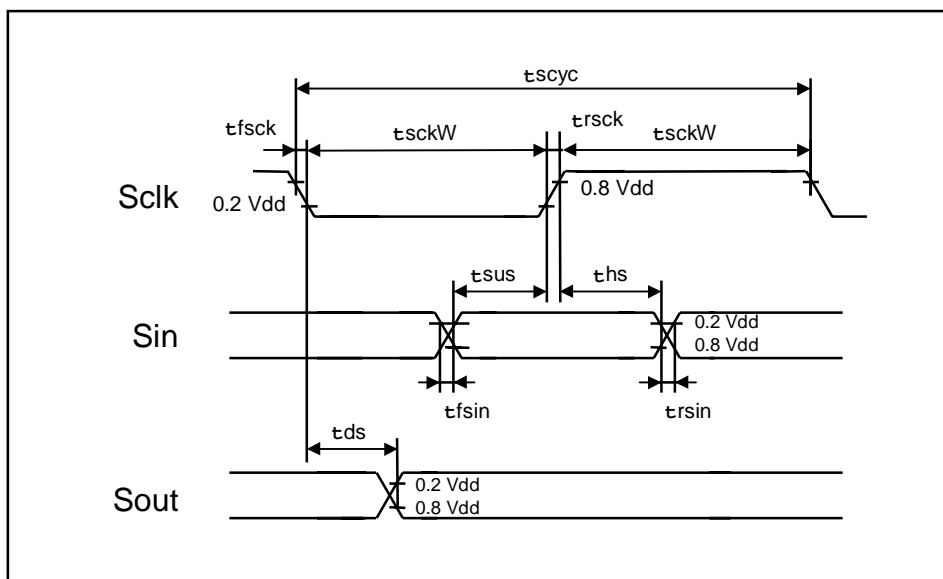


4.5.2. Serial Transfer

($V_{dd} = 5.0V \pm 10\%$, $V_{ss} = 0V$, $T_a = -20 \sim 85^\circ C$, $f(Xin) = 8 \text{ MHz}$)

| Parametet | Pin | Symbol | Unit | SPECIFICATION | | | etc |
|--|--------------------------|---------------|------|----------------|------|-------------|-----|
| | | | | MIN. | TYP. | MAX. | |
| Serial Input Clock Pulse | $\overline{\text{Sclk}}$ | tscyc | ns | $2t_{sys}+200$ | - | 8 | |
| Serial Input Clock Pulse Width | $\overline{\text{Sclk}}$ | tsckw | ns | $t_{sys}+70$ | - | 8 | |
| Serial Input Clock Pulse Transition Time | $\overline{\text{Sclk}}$ | tfscck,trscck | ns | | - | 30 | |
| Sin Input Pulse Transition Time | Sin | tfsin,trsin | ns | | - | 30 | |
| Sin Input Setup time(External Sclk) | Sin | tsus | ns | 100 | - | | |
| Sin Input Setup time(Internal Sclk) | Sin | tsus | ns | 200 | - | | |
| Sin Input Hold Time | Sin | ths | ns | $t_{sys}+70$ | - | | |
| Serial Output Clock Cycle Time | $\overline{\text{Sclk}}$ | tscyc | ns | $4t_{sys}$ | - | $16t_{sys}$ | |
| Serial Output Clock Transition Time | $\overline{\text{Sclk}}$ | tsckw | ns | $2t_{sys}-30$ | - | | |
| Serial Output Clock Transition Time | $\overline{\text{Sclk}}$ | tfscck,trscck | ns | | - | 30 | |
| Serial Output Delay Time | Sout | trec, tfec | ns | | | 100 | |

■ Serial I/O Timing Chart

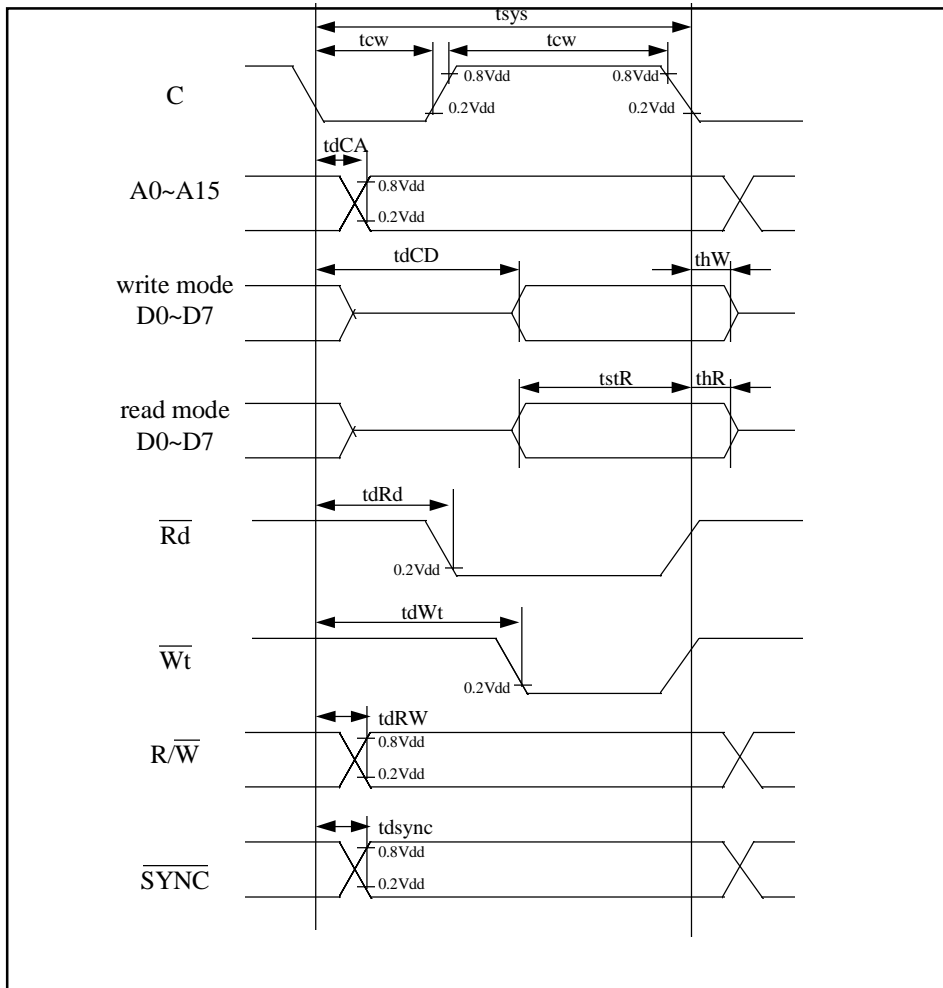


4.5.3. Microprocessor Mode I/O Timing

(Vdd = 5.0V±10%, Vss = 0V, Ta = -20 ~ 85°C, f (Xin) = 8 MHz)

| Parameter | Pin | Symbol | Unit | SPECIFICATION | | | etc |
|----------------------------|-------------------|--------|------|---------------|------|------|-----|
| | | | | MIN. | TYP. | MAX. | |
| Control Clock Output Width | C | tCL | ns | 90 | - | | |
| Address Output Delay Time | A0 ~ A15 | tdCA | ns | | - | 80 | |
| Data Output Delay Time | D0 ~ D7 | tdCD | ns | | - | 180 | |
| Data Output Hold Time | D0 ~ D7 | thw | ns | | - | 20 | |
| Data Input Setup Time | D0 ~ D7 | tsuR | ns | 80 | - | | |
| Data Input Hold Time | D0 ~ D7 | thR | ns | 15 | - | | |
| Rd Output Delay Time | \overline{Rd} | tdRd | tsys | | - | 90 | |
| Wt Output Delay Time | \overline{Wt} | tdWt | tsys | | - | 130 | |
| R/W Output Delay Time | R/W | tdRW | tsys | | - | 50 | |
| sync Output Delay Time | \overline{SYNC} | tdsync | tsys | | - | 50 | |

■ Timing Chart

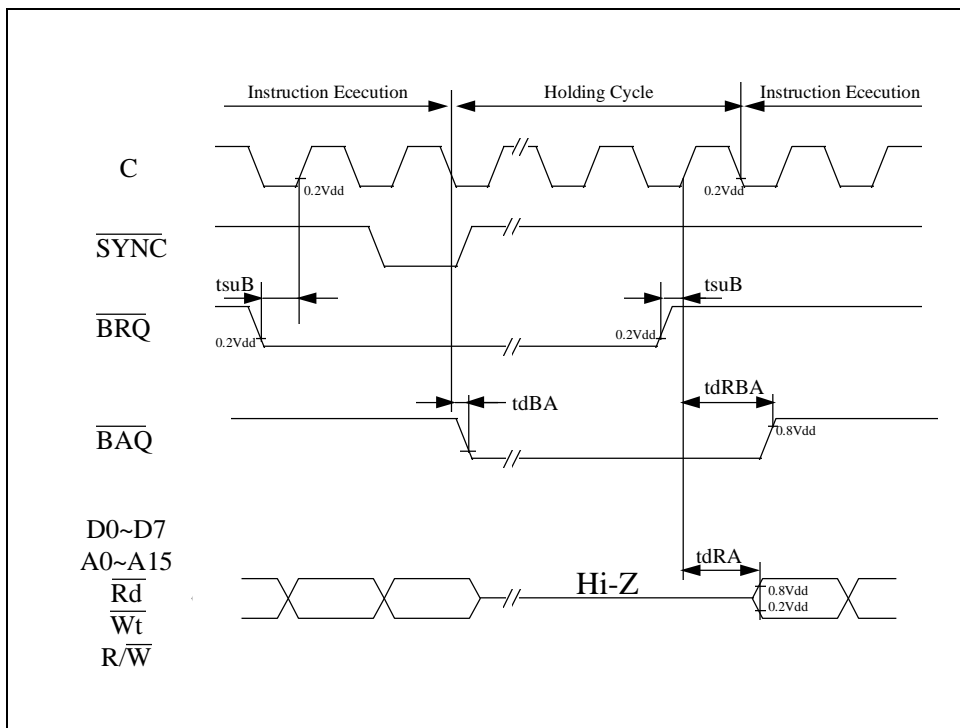


4.5.4. Bus Holding Timing

(Vdd = 5.0V±10%, Vss = 0V , Ta = -20 ~ 85°C , f (Xin) = 8 MHz)

| Parameter | Pin | Symbol | Unit | SPECIFICATION | | | etc |
|--|--|--------|------|---------------|------|------|-----|
| | | | | MIN. | TYP. | MAX. | |
| BRQ Setup Time | $\overline{\text{BRQ}}$ | tSUB | tsys | 100 | - | | |
| BAK Delay Time | $\overline{\text{BAK}}$ | tdBA | tsys | | - | 50 | |
| BAK Release Delay Time | $\overline{\text{BAK}}$ | tdRBA | tsys | | - | 220 | |
| Bus(Address,Data) Control Release Delay Time | D0 ~ D7 A0 ~ A15 $\overline{\text{Rd}}$, $\overline{\text{Wt}}$, $\overline{\text{R/W}}$ | tdRA | tsys | | - | 210 | |

■ Timing Chart



5. INSTRUCTION SET

1. ARITHMETIC/ LOGIC OPERATION

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------------|---------|---------|----------|--|------------------|
| 1 | ADC #imm | 04 | 2 | 2 | Add with carry. $A \leftarrow (A) + (M) + C$ | NV--H-ZC |
| 2 | ADC dp | 05 | 2 | 3 | | |
| 3 | ADC dp + X | 06 | 2 | 4 | | |
| 4 | ADC !abs | 07 | 3 | 4 | | |
| 5 | ADC !abs + Y | 15 | 3 | 5 | | |
| 6 | ADC [dp + X] | 16 | 2 | 6 | | |
| 7 | ADC [dp] + Y | 17 | 2 | 6 | | |
| 8 | ADC { X } | 14 | 1 | 3 | | |
| 9 | AND #imm | 84 | 2 | 2 | Logical AND $A \leftarrow (A) \wedge (M)$ | N-----Z- |
| 10 | AND dp | 85 | 2 | 3 | | |
| 11 | AND dp + X | 86 | 2 | 4 | | |
| 12 | AND !abs | 87 | 3 | 4 | | |
| 13 | AND !abs + Y | 95 | 3 | 5 | | |
| 14 | AND [dp + X] | 96 | 2 | 6 | | |
| 15 | AND [dp] + Y | 97 | 2 | 6 | | |
| 16 | AND { X } | 94 | 1 | 3 | | |
| 17 | ASL A | 08 | 1 | 2 | Arithmetic shift left $ \begin{array}{cccccccc} C & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \boxed{} & \leftarrow & \leftarrow & \leftarrow & \leftarrow & \leftarrow & \leftarrow & \leftarrow & \leftarrow "0" \end{array} $ | N-----ZC |
| 18 | ASL dp | 09 | 2 | 4 | | |
| 19 | ASL dp + X | 19 | 2 | 5 | | |
| 20 | ASL !abs | 18 | 3 | 5 | | |
| 21 | CMP #imm | 44 | 2 | 2 | | |
| 22 | CMP dp | 45 | 2 | 3 | | |
| 23 | CMP dp + X | 46 | 2 | 4 | | |
| 24 | CMP !abs | 47 | 3 | 4 | | |
| 25 | CMP !abs + Y | 55 | 3 | 5 | | |
| 26 | CMP [dp + X] | 56 | 2 | 6 | | |
| 27 | CMP [dp] + Y | 57 | 2 | 6 | | |
| 28 | CMP { X } | 54 | 1 | 3 | | |
| 29 | CMPX #imm | 5E | 2 | 2 | Compare X contents with memory contents $(X) - (M)$ | N-----ZC |
| 30 | CMPX dp | 6C | 2 | 3 | | |
| 31 | CMPX !abs | 7C | 3 | 4 | | |
| 32 | CMPY #imm | 7E | 2 | 2 | Compare Y contents with memory contents $(Y) - (M)$ | N-----ZC |
| 33 | CMPY dp | 8C | 2 | 3 | | |
| 34 | CMPY !abs | 9C | 3 | 4 | | |
| 35 | COM dp | 2C | 2 | 4 | | |
| 36 | DAA | DF | 1 | 3 | Decimal adjust for addition | N-----ZC |
| 37 | DAS | CF | 1 | 3 | Decimal adjust for subtraction | N-----ZC |

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------------|---------|---------|----------|---------------------------------------|------------------|
| 38 | DEC A | A8 | 1 | 2 | Decrement | N-----Z- |
| 39 | DEC dp | A9 | 2 | 4 | $M \leftarrow (M) - 1$ | N-----Z- |
| 40 | DEC dp + X | B9 | 2 | 5 | | N-----Z- |
| 41 | DEC !abs | B8 | 3 | 5 | | N-----Z- |
| 42 | DEC X | AF | 1 | 2 | | N-----Z- |
| 43 | DEC Y | BE | 1 | 2 | | N-----Z- |
| 44 | DIV | 9B | 1 | 12 | Divide : YA / XQ: A, R: Y | NV--H-Z- |
| 45 | EOR #imm | A4 | 2 | 2 | Exclusive OR | |
| 46 | EOR dp | A5 | 2 | 3 | $A \leftarrow (A) \oplus (M)$ | |
| 47 | EOR dp + X | A6 | 2 | 4 | | N-----Z- |
| 48 | EOR !abs | A7 | 3 | 4 | | N-----Z- |
| 49 | EOR !abs + Y | B5 | 3 | 5 | | N-----Z- |
| 50 | EOR [dp + X] | B6 | 2 | 6 | | N-----Z- |
| 51 | EOR [dp] + Y | B7 | 2 | 6 | | N-----Z- |
| 52 | EOR { X } | B4 | 1 | 3 | | N-----Z- |
| 53 | INC A | 88 | 1 | 2 | Increment | N-----ZC |
| 54 | INC dp | 89 | 2 | 4 | $M \leftarrow (M) + 1$ | N-----Z- |
| 55 | INC dp + X | 99 | 2 | 5 | | N-----Z- |
| 56 | INC !abs | 98 | 3 | 5 | | N-----Z- |
| 57 | INC X | 8F | 1 | 2 | | N-----Z- |
| 58 | INC Y | 9E | 1 | 2 | | N-----Z- |
| 59 | LSR A | 48 | 1 | 2 | Logical shift right | |
| 60 | LSR dp | 49 | 2 | 4 | | N-----ZC |
| 61 | LSR dp + X | 59 | 2 | 5 | | N-----ZC |
| 62 | LSR !abs | 58 | 3 | 5 | | N-----ZC |
| 63 | MUL | 5B | 1 | 9 | Multiply : $YA \leftarrow Y \times A$ | N-----Z- |
| 64 | OR #imm | 64 | 2 | 2 | Logical OR | |
| 65 | OR dp | 65 | 2 | 3 | $A \leftarrow (A) \vee (M)$ | |
| 66 | OR dp + X | 66 | 2 | 4 | | N-----Z- |
| 67 | OR !abs | 67 | 3 | 4 | | N-----Z- |
| 68 | OR !abs + Y | 75 | 3 | 5 | | N-----Z- |
| 69 | OR [dp + X] | 76 | 2 | 6 | | N-----Z- |
| 70 | OR [dp] + Y | 77 | 2 | 6 | | N-----Z- |
| 71 | OR { X } | 74 | 1 | 3 | | N-----Z- |
| 72 | ROL A | 28 | 1 | 2 | Rotate left through carry | |
| 73 | ROL dp | 29 | 2 | 4 | | N-----ZC |
| 74 | ROL dp + X | 39 | 2 | 5 | | N-----ZC |
| 75 | ROL !abs | 38 | 3 | 5 | | N-----ZC |
| 76 | ROR A | 68 | 1 | 2 | Rotate right through carry | |
| 77 | ROR dp | 69 | 2 | 4 | | N-----ZC |
| 78 | ROR dp + X | 79 | 2 | 5 | | N-----ZC |
| 79 | ROR !abs | 78 | 3 | 5 | | N-----ZC |

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------------|---------|---------|----------|--|------------------|
| 80 | SBC #imm | 24 | 2 | 2 | Subtract with carry $A \leftarrow (A) - (M) - \overline{(C)}$ | NV--HZC |
| 81 | SBC dp | 25 | 2 | 3 | | |
| 82 | SBC dp + X | 26 | 2 | 4 | | |
| 83 | SBC !abs | 27 | 3 | 4 | | |
| 84 | SBC !abs + Y | 35 | 3 | 5 | | |
| 85 | SBC [dp + X] | 36 | 2 | 6 | | |
| 86 | SBC [dp] + Y | 37 | 2 | 6 | | |
| 87 | SBC { X } | 34 | 1 | 3 | | |
| 88 | TST dp | 4C | 2 | 3 | Test memory contents for negative or zero (dp) - 00 _H | N-----Z- |
| 89 | XCN | CE | 1 | 5 | Exchange nibbles within the accumulator $A_7-A_4 \leftrightarrow A_3-A_0$ | N-----Z- |

2. REGISTER / MEMORY OPERATION

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC | |
|-----|----------------|---------|---------|----------|---|------------------|---|
| 1 | LDA #imm | C4 | 2 | 2 | Load accumulator $A \leftarrow (M)$ | N-----Z- | |
| 2 | LDA dp | C5 | 2 | 3 | | | |
| 3 | LDA dp + X | C6 | 2 | 4 | | | |
| 4 | LDA !abs | C7 | 3 | 4 | | | |
| 5 | LDA !abs + Y | D5 | 3 | 5 | | | |
| 6 | LDA [dp + X] | D6 | 2 | 6 | | | |
| 7 | LDA [dp] + Y | D7 | 2 | 6 | | | |
| 8 | LDA { X } | D4 | 1 | 3 | | | |
| 9 | LDA { X }+ | DB | 1 | 4 | X- register auto-increment : $A \leftarrow (M), X \leftarrow X + 1$ | | |
| 10 | LDM dp,#imm | E4 | 3 | 5 | Load memory with immediate data : $(M) \leftarrow imm$ | ----- | |
| 11 | LDX #imm | 1E | 2 | 2 | Load X-register $X \leftarrow (M)$ | N-----Z- | |
| 12 | LDX dp | CC | 2 | 3 | | | |
| 13 | LDX dp + Y | CD | 2 | 4 | | | |
| 14 | LDX !abs | DC | 3 | 4 | | | |
| 15 | LDY #imm | 3E | 2 | 2 | | | Load Y-register $Y \leftarrow (M)$ |
| 16 | LDY dp | C9 | 2 | 3 | | | |
| 17 | LDY dp + X | D9 | 2 | 4 | | | |
| 18 | LDY !abs | D8 | 3 | 4 | | | |
| 19 | STA dp | E5 | 2 | 3 | Store accumulator contents in memoy $(M) \leftarrow A$ | ----- | |
| 20 | STA dp + X | E6 | 2 | 4 | | | |
| 21 | STA !abs | E7 | 3 | 4 | | | |
| 22 | STA !abs + Y | F5 | 3 | 5 | | | |
| 23 | STA [dp + X] | F6 | 2 | 6 | | | |
| 24 | STA [dp] + Y | F7 | 2 | 6 | | | |
| 25 | STA { X } | F4 | 1 | 3 | | | |
| 26 | STA { X }+ | FB | 1 | 4 | | | X- register auto-increment : $(M) \leftarrow A, X \leftarrow X + 1$ |

| | | OP | BYTE | CYCLE | | FLAG |
|--|--|----|------|-------|--|------|
|--|--|----|------|-------|--|------|

| NO. | MNEMONIC | CODE | NO | NO | OPERATION | NVGBHIZC |
|-----|------------|------|----|----|--|----------|
| 27 | STX dp | EC | 2 | 4 | Store X-register contents in memoy | |
| 28 | STX dp + Y | ED | 2 | 5 | (M) ← X | ----- |
| 29 | STX !abs | FC | 3 | 5 | | |
| 30 | STY dp | E9 | 2 | 4 | Store Y-register contents in memoy | |
| 31 | STY dp + X | F9 | 2 | 5 | (M) ← Y | ----- |
| 32 | STY !abs | F8 | 3 | 5 | | |
| 33 | TAX | E8 | 1 | 2 | Transfer accumulator contents to X-register : X ← A | N-----Z- |
| 34 | TAY | 9F | 1 | 2 | Transfer accumulator contents to Y-register : Y ← A | N-----Z- |
| 35 | TSPX | AE | 1 | 2 | Transfer stack-pointer contents to X-register : X ← sp | N-----Z- |
| 36 | TXA | C8 | 1 | 2 | Transfer X-register contents to accumulator: A ← X | N-----Z- |
| 37 | TXSP | 8E | 1 | 2 | Transfer X-register contents to stack-pointer: sp ← X | N-----Z- |
| 38 | TYA | BF | 1 | 2 | Transfer Y-register contents to accumulator: A ← Y | N-----Z- |
| 39 | XAX | EE | 1 | 4 | Exchange X-register contents with accumulator :X↔ A | ----- |
| 40 | XAY | DE | 1 | 4 | Exchange Y-register contents with accumulator :Y↔ A | ----- |
| 41 | XMA dp | BC | 2 | 5 | Exchange memory contents with accumulator | |
| 42 | XMA dp+X | AD | 2 | 6 | (M) ↔ A | N-----Z- |
| 43 | XMA {X} | BB | 1 | 5 | | |
| 44 | XYX | FE | 1 | 4 | Exchange X-register contents with Y-register : X↔ Y | ----- |

3. 16-BIT OPERATION

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------|---------|---------|----------|--|---------------|
| 1 | ADDW dp | 1D | 2 | 5 | 16-Bits add without carry YA ← (YA) + (dp + 1) (dp) | NV--H-ZC |
| 2 | CMPW dp | 5D | 2 | 4 | Compare YA contents with memory pair contents : (YA) - (dp+1)(dp) | N-----ZC |
| 3 | DECW dp | BD | 2 | 6 | Decrement memory pair (dp+1)(dp) ← (dp+1) (dp) - 1 | N-----Z- |
| 4 | INCW dp | 9D | 2 | 6 | Increment memory pair (dp+1) (dp) ← (dp+1) (dp) + 1 | N-----Z- |
| 5 | LDYA dp | 7D | 2 | 5 | Load YA YA ← (dp + 1) (dp) | N-----Z- |
| 6 | STYA dp | DD | 2 | 5 | Store YA (dp + 1) (dp) ← YA | ----- |
| 7 | SUBW dp | 3D | 2 | 5 | 16-Bits substract without carry YA ← (YA) - (dp + 1) (dp) | NV--H-ZC |

4. BIT MANIPULATION

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|-------------|---------|---------|----------|--|---------------|
| 1 | AND1 M.bit | 8B | 3 | 4 | Bit AND C-flag : $C \leftarrow (C) \wedge (M.bit)$ | -----C |
| 2 | AND1B M.bit | 8B | 3 | 4 | Bit AND C-flag and NOT : $C \leftarrow (C) \wedge \overline{(M.bit)}$ | -----C |
| 3 | BIT dp | 0C | 2 | 4 | Bit test A with memory : | MM----Z- |
| 4 | BIT !abs | 1C | 3 | 5 | $Z \leftarrow (A) \wedge (M), N \leftarrow (M_7), V \leftarrow (M_6)$ | |
| 5 | CLR1 dp.bit | y1 | 2 | 4 | Clear bit : $(M.bit) \leftarrow "0"$ | ----- |
| 6 | CLRA1 A.bit | 2B | 2 | 2 | Clear A bit : $(A.bit) \leftarrow "0"$ | ----- |
| 7 | CLRC | 20 | 1 | 2 | Clear C-flag : $C \leftarrow "0"$ | -----0 |
| 8 | CLRG | 40 | 1 | 2 | Clear G-flag : $G \leftarrow "0"$ | --0----- |
| 9 | CLR V | 80 | 1 | 2 | Clear V-flag : $V \leftarrow "0"$ | -0--0--- |
| 10 | EOR1 M.bit | AB | 3 | 5 | Bit exclusive-OR C-flag : $C \leftarrow (C) \oplus (M.bit)$ | -----C |
| 11 | EOR1B M.bit | AB | 3 | 5 | Bit exclusive-OR C-flag and NOT : $C \leftarrow (C) \oplus \overline{(M.bit)}$ | -----C |
| 12 | LDC M.bit | CB | 3 | 4 | Load C-flag : $C \leftarrow (M.bit)$ | -----C |
| 13 | LDCB M.bit | CB | 3 | 4 | Load C-flag with NOT : $C \leftarrow \overline{(M.bit)}$ | -----C |
| 14 | NOT1 M.bit | 4B | 3 | 5 | Bit complement : $(M.bit) \leftarrow \overline{(M.bit)}$ | ----- |
| 15 | OR1 M.bit | 6B | 3 | 5 | Bit OR C-flag : $C \leftarrow (C) \vee (M.bit)$ | -----C |
| 16 | OR1B M.bit | 6B | 3 | 5 | Bit OR C-flag and NOT : $C \leftarrow (C) \vee \overline{(M.bit)}$ | -----C |
| 17 | SET1 dp.bit | x1 | 2 | 4 | Set bit : $(M.bit) \leftarrow "1"$ | ----- |
| 18 | SETA1 A.bit | 0B | 2 | 2 | Set A bit : $(A.bit) \leftarrow "1"$ | ----- |
| 19 | SETC | A0 | 1 | 2 | Set C-flag : $C \leftarrow "1"$ | -----1 |
| 20 | SETG | C0 | 1 | 2 | Set G-flag : $G \leftarrow "1"$ | --1----- |
| 21 | STC M.bit | EB | 3 | 6 | Store C-flag : $(M.bit) \leftarrow C$ | ----- |
| 22 | TCLR1 !abs | 5C | 3 | 6 | Test and clear bits with A : $A \leftarrow (M), (M) \leftarrow (M) \wedge \overline{(A)}$ | N-----Z- |
| 23 | TSET1 !abs | 3C | 3 | 6 | Test and set bits with A : $A \leftarrow (M), (M) \leftarrow (M) \vee (A)$ | N-----Z- |

5. BRANCH / JUMP OPERATION

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------------|---------|---------|----------|---|---------------|
| 1 | BBC A.bit,rel | y2 | 2 | 4/6 | Branch if bit clear : | ----- |
| 2 | BBC dp.bit,rel | y3 | 3 | 5/7 | if $(bit) = 0$, then $pc \leftarrow (pc) + rel$ | |
| 3 | BBS A.bit,rel | x2 | 2 | 4/6 | Branch if bit set : | ----- |
| 4 | BBS dp.bit,rel | x3 | 3 | 5/7 | if $(bit) = 1$, then $pc \leftarrow (pc) + rel$ | |
| 5 | BCC rel | 50 | 2 | 2/4 | Branch if carry bit clear if $(C) = 0$, then $pc \leftarrow (pc) + rel$ | ----- |
| 6 | BCS rel | D0 | 2 | 2/4 | Branch if carry bit set if $(C) = 1$, then $pc \leftarrow (pc) + rel$ | ----- |
| 7 | BEQ rel | D0 | 2 | 2/4 | Branch if equal if $(Z) = 1$, then $pc \leftarrow (pc) + rel$ | ----- |

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|---------------|------------|------------|-------------|--|------------------|
| 8 | BMI rel | 90 | 2 | 2/4 | Branch if minus if (N) = 1, then $pc \leftarrow (pc) + rel$ | ----- |
| 9 | BNE rel | 70 | 2 | 2/4 | Branch if not equal if (Z) = 0, then | ----- |
| 10 | BPL rel | 10 | 2 | 2/4 | Branch if minus if (N) = 0, then $pc \leftarrow (pc) + rel$ | ----- |
| 11 | BRA rel | 2F | 2 | 4 | Branch always $pc \leftarrow (pc) + rel$ | ----- |
| 12 | BVC rel | 30 | 2 | 2/4 | Branch if overflow bit clear if (V) = 0, then $pc \leftarrow (pc) + rel$ | ----- |
| 13 | BVS rel | B0 | 2 | 2/4 | Branch if overflow bit set if (V) = 1, then $pc \leftarrow (pc) + rel$ | ----- |
| 14 | CALL !abs | 3B | 3 | 8 | Subroutine call | ----- |
| 15 | CALL [dp] | 5F | 2 | 8 | $M(sp) \leftarrow (pc_H), sp \leftarrow sp - 1, M(sp) \leftarrow (pc_L), sp \leftarrow sp - 1,$ if !abs, $pc \leftarrow abs$; if [dp], $pc_L \leftarrow (dp), pc_H \leftarrow (dp+1)$. | ----- |
| 16 | CBNE dp,rel | FD | 3 | 5/7 | Compare and branch if not equal : | ----- |
| 17 | CBNE dp+X,rel | 8D | 3 | 6/8 | if (A) \neq (M), then $pc \leftarrow (pc) + rel$. | ----- |
| 18 | DBNE dp,rel | AC | 3 | 5/7 | Decrement and branch if not equal : | ----- |
| 19 | DBNE Y,rel | 7B | 2 | 4/6 | if (M) \neq 0, then $pc \leftarrow (pc) + rel$. | ----- |
| 20 | JMP !abs | 1B | 3 | 3 | Unconditional jump | ----- |
| 21 | JMP [!abs] | 1F | 3 | 5 | $pc \leftarrow$ jump address | ----- |
| 22 | JMP [dp] | 3F | 2 | 4 | | ----- |
| 23 | PCALL upage | 4F | 2 | 6 | U-page call $M(sp) \leftarrow (pc_H), sp \leftarrow sp - 1, M(sp) \leftarrow (pc_L),$ $sp \leftarrow sp - 1, pc_L \leftarrow (upage), pc_H \leftarrow "0FF_H"$. | ----- |
| 24 | TCALL n | nA | 1 | 8 | Table call : $(sp) \leftarrow (pc_H), sp \leftarrow sp - 1,$ $M(sp) \leftarrow (pc_L), sp \leftarrow sp - 1,$ $pc_L \leftarrow (Table\ vector\ L), pc_H \leftarrow (Table\ vector\ H)$ | ----- |

6. CONTROL OPERATION & etc.

| NO. | MNEMONIC | OP CODE | BYTE NO | CYCLE NO | OPERATION | FLAG NVGBHIZC |
|-----|----------|---------|---------|----------|--|---------------|
| 1 | BRK | 0F | 1 | 8 | Software interrupt : $B \leftarrow "1"$, $M(sp) \leftarrow (pc_H)$, $sp \leftarrow sp - 1$, $M(s) \leftarrow (pc_L)$, $sp \leftarrow sp - 1$, $M(sp) \leftarrow (PSW)$, $sp \leftarrow sp - 1$, $pc_L \leftarrow (OFFDE_H)$, $pc_H \leftarrow (OFFDF_H)$. | ---1-0-- |
| 2 | DI | 60 | 1 | 3 | Disable interrupts : $I \leftarrow "0"$ | -----0-- |
| 3 | EI | E0 | 1 | 3 | Enable interrupts : $I \leftarrow "1"$ | -----1-- |
| 4 | NOP | FF | 1 | 2 | No operation | ----- |
| 5 | POP A | 0D | 1 | 4 | $sp \leftarrow sp + 1$, $A \leftarrow M(sp)$ | (restored) |
| 6 | POP X | 2D | 1 | 4 | $sp \leftarrow sp + 1$, $X \leftarrow M(sp)$ | |
| 7 | POP Y | 4D | 1 | 4 | $sp \leftarrow sp + 1$, $Y \leftarrow M(sp)$ | |
| 8 | POP PSW | 6D | 1 | 4 | $sp \leftarrow sp + 1$, $PSW \leftarrow M(sp)$ | |
| 9 | PUSH A | 0E | 1 | 4 | $M(sp) \leftarrow A$, $sp \leftarrow sp - 1$ | ----- |
| 10 | PUSH X | 2E | 1 | 4 | $M(sp) \leftarrow X$, $sp \leftarrow sp - 1$ | |
| 11 | PUSH Y | 4E | 1 | 4 | $M(sp) \leftarrow Y$, $sp \leftarrow sp - 1$ | |
| 12 | PUSH PSW | 6E | 1 | 4 | $M(sp) \leftarrow PSW$, $sp \leftarrow sp - 1$ | |
| 13 | RET | 6F | 1 | 5 | Return from subroutine $sp \leftarrow sp + 1$, $pc_L \leftarrow M(sp)$, $sp \leftarrow sp + 1$, $pc_H \leftarrow M(sp)$ | ----- |
| 14 | RETI | 7F | 1 | 6 | Return from interrupt $sp \leftarrow sp + 1$, $PSW \leftarrow M(sp)$, $sp \leftarrow sp + 1$, $pc_L \leftarrow M(sp)$, $sp \leftarrow sp + 1$, $pc_H \leftarrow M(sp)$ | (restored) |
| 15 | STOP | 00 | 1 | 3 | Stop mode (halt CPU, stop oscillator) | ----- |

6. GMS81516AT (OTP) PROGRAMMING

The GMS81516AT is one-time PROM (OTP) microcontroller with 16K bytes electrically programmable read only memory for the GMS81508/16 system evaluation, first production and fast mass production.

To programming the OTP device, user can have two way. One is using the universal programmer which is support HME microcontrollers, other is using the general EPROM programmer.

1. Using the Universal programmer

Third party universal programmer support to program the GMS81516AT microcontrollers and lists are shown as below.

Manufacturer: **Advantech**
Web site: <http://www.aec.com.tw>
Programmer: LabTool-48

Manufacturer: **Hi-Lo systems**
Web site: <http://www.hilosystems.com.tw>
Programmer: ALL-11, GANG-08

Socket adapters are supported by third party programmer manufacturer.

2. Using the general EPROM(27C256) programmer

The programming algorithm is similar with the standard EPROM 27C256. It gives some convenience that user can use standard EPROM programmer. **Make sure that 1ms programming pulse must be used, it generally called "Intelligent Mode"**. Do not use 100us programming pulse mode, "Quick Pulse Mode".

When user use general EPROM programmer, socket adapter is essentially required. It convert pin to fit the pin of general 27C256 EPROM.

Three type socket adapters are provided according to package variation as below table.

| Socket Adapter | Package Type |
|----------------|-----------------------|
| OA815A-64SD | 64 pin SDIP |
| OA815A-64QF-10 | 64 pin LQFP (10 x 10) |
| OA815A-64QF | 64 pin QFP (14 x 20) |

With these socket adapters, the GMS81516AT can easily be programmed and verified using Intel 27C256 EPROM mode on general-purpose PROM programmer.

In assembler and file type, two files are generated after compiling. One is "*.HEX", another is "*.OTP". The "*.HEX" file is used for emulation in circuit emulator (CHOICE-Dr™ or CHOICE-Jr™) and "*.OTP" file is used for programming to the OTP device.

Programming Procedure

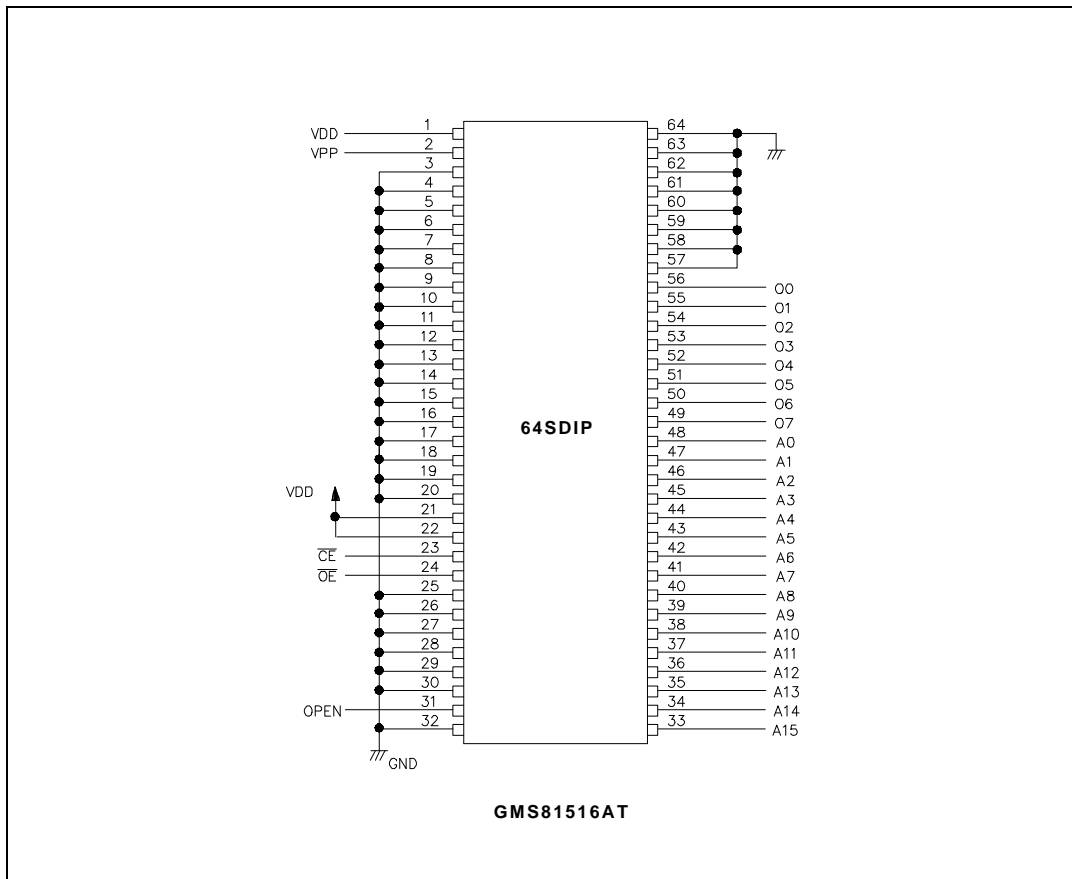
1. Select the EPROM device and manufacturer on EPROM programmer (Intel 27C256).
 2. Select the programming algorithm as an Intelligent mode (apply 1ms writing pulse), not a Quick pulse mode.
 3. Load the file (*.OTP) to the programmer.
 4. Set the programming address range as below table.
- | Address | Set Value |
|----------------------|-----------|
| Buffer start address | 4000H |
| Buffer end address | 7FFFH |
| Device start address | 4000H |
5. Mount the socket adapter with the GMS81516AT on the PROM programmer.
 6. Start the PROM programmer to programming/verifying.

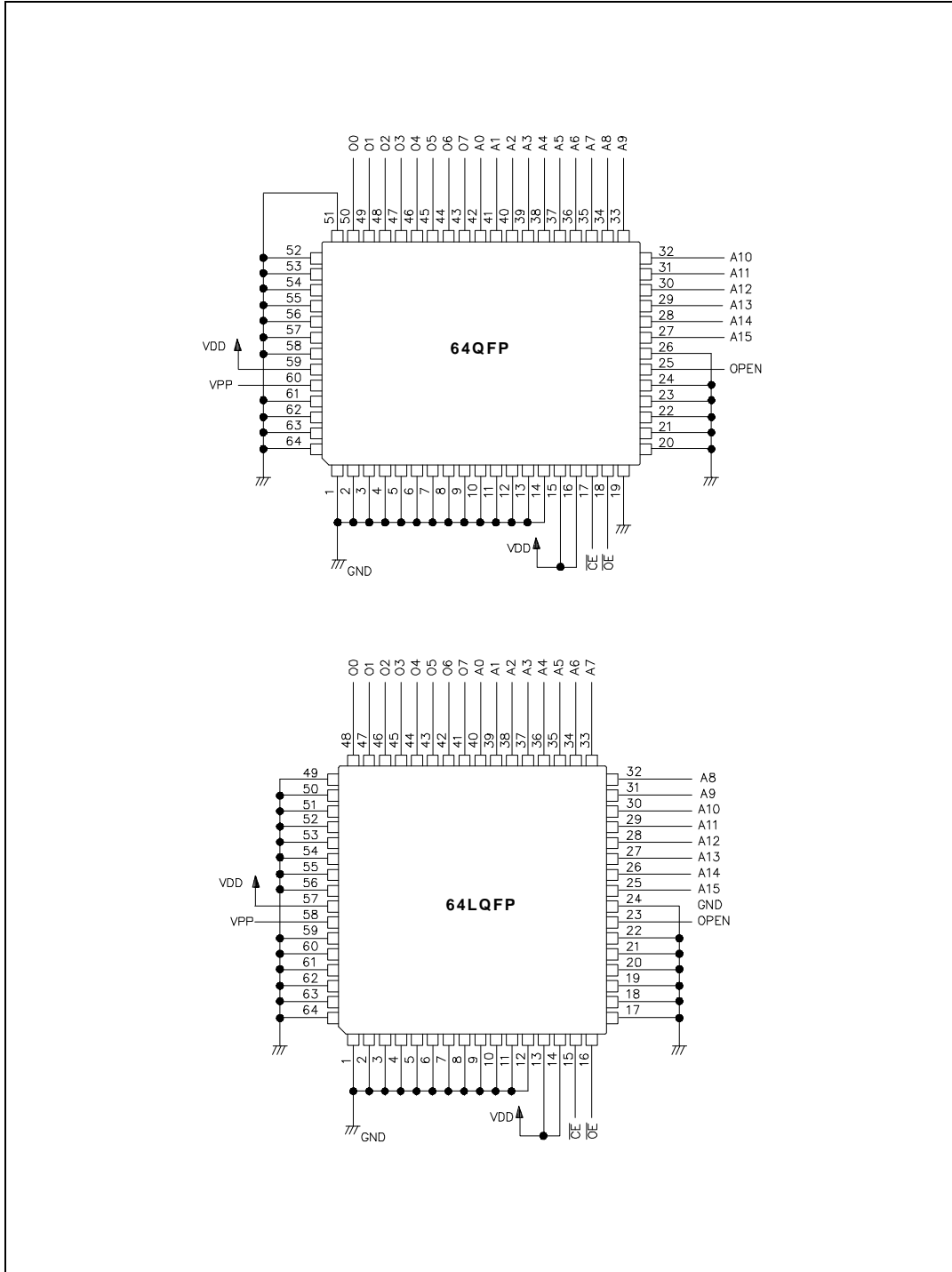
**GMS81516AT PROGRAMMING
MANUAL**

DEVICE OVERVIEW

The GMS81516AT is a high-performance CMOS 8-bit microcontroller with 16K bytes of EPROM. The device is one of GMS800 family. The HME GMS81516AT is a powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. The GMS81516AT provides the following standard features: 16K bytes of EPROM, 448 bytes of RAM, 56 I/O lines, 16-bit or 8-bit timer/counter, a precision analog to digital converter, PWM, on-chip oscillator and clock circuitry.

PIN CONFIGURATION





64SDIP Package for GMS81516AT

| Pin No. | MCU Mode | OTP Mode | Pin No. | MCU Mode | OTP Mode |
|---------|-----------------------------------|--------------------------|---------|----------|----------|
| 1 | V _{DD} - | V _{DD} - | 33 | R27 I/O | A15 I |
| 2 | MP I | V _{PP} - | 34 | R26 I/O | A14 I |
| 3 | AVSS I | (1) I | 35 | R25 I/O | A13 I |
| 4 | AVREF I | (1) I | 36 | R24 I/O | A12 I |
| 5 | R67/AN7 I/O | (1) I | 37 | R23 I/O | A11 I |
| 6 | R66/AN6 I/O | (1) I | 38 | R22 I/O | A10 I |
| 7 | R65/AN5 I/O | (1) I | 39 | R21 I/O | A9 I |
| 8 | R64/AN4 I/O | (1) I | 40 | R20 I/O | A8 I |
| 9 | R63/AN3 I | (1) I | 41 | R17 I/O | A7 I |
| 10 | R62/AN2 I | (1) I | 42 | R16 I/O | A6 I |
| 11 | R61/AN1 I | (1) I | 43 | R15 I/O | A5 I |
| 12 | R60/AN0 I | (1) I | 44 | R14 I/O | A4 I |
| 13 | R57/PWM1 I/O | (1) I | 45 | R13 I/O | A3 I |
| 14 | R56/PWM0 I/O | (1) I | 46 | R12 I/O | A2 I |
| 15 | R55/BUZ I/O | (1) I | 47 | R11 I/O | A1 I |
| 16 | R54/WDTO I/O | (1) I | 48 | R10 I/O | A0 I |
| 17 | R53/ $\overline{\text{SRDY}}$ I/O | (1) I | 49 | R07 I/O | O7 I/O |
| 18 | R52/SCLK I/O | (1) I | 50 | R06 I/O | O6 I/O |
| 19 | R51/SOUT I/O | (1) I | 51 | R05 I/O | O5 I/O |
| 20 | R50/SIN I/O | (1) I | 52 | R04 I/O | O4 I/O |
| 21 | R47/T3O I/O | (2) I | 53 | R03 I/O | O3 I/O |
| 22 | R46/T1O I/O | (2) I | 54 | R02 I/O | O2 I/O |
| 23 | R45/ $\overline{\text{EC2}}$ I/O | $\overline{\text{CE}}$ I | 55 | R01 I/O | O1 I/O |
| 24 | R44/ $\overline{\text{EC0}}$ I/O | $\overline{\text{OE}}$ I | 56 | R00 I/O | O0 I/O |
| 25 | R43/INT3 I/O | (1) I | 57 | R37 I/O | (1) I |
| 26 | R42/INT2 I/O | (1) I | 58 | R36 I/O | (1) I |
| 27 | R41/INT1 I/O | (1) I | 59 | R35 I/O | (1) I |
| 28 | R40/INT0 I/O | (1) I | 60 | R34 I/O | (1) I |
| 29 | RESET I | (1) I | 61 | R33 I/O | (1) I |
| 30 | X _{IN} I | (1) I | 62 | R32 I/O | (1) I |
| 31 | X _{OUT} O | (3) O | 63 | R31 I/O | (1) I |
| 32 | V _{SS} - | V _{SS} - | 64 | R30 I/O | (1) I |

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

64QFP Package for GMS81516AT

| Pin No. | MCU Mode | OTP Mode | Pin No. | MCU Mode | OTP Mode |
|---------|--------------|----------|---------|-------------|----------|
| 1 | R65/AN5 I/O | (1) I | 33 | R21 I/O | A9 I |
| 2 | R64/AN4 I/O | (1) I | 34 | R20 I/O | A8 I |
| 3 | R63/AN3 I | (1) I | 35 | R17 I/O | A7 I |
| 4 | R62/AN2 I | (1) I | 36 | R16 I/O | A6 I |
| 5 | R61/AN1 I | (1) I | 37 | R15 I/O | A5 I |
| 6 | R60/AN0 I | (1) I | 38 | R14 I/O | A4 I |
| 7 | R57/PWM1 I/O | (1) I | 39 | R13 I/O | A3 I |
| 8 | R56/PWM0 I/O | (1) I | 40 | R12 I/O | A2 I |
| 9 | R55/BUZ I/O | (1) I | 41 | R11 I/O | A1 I |
| 10 | R54/WDTO I/O | (1) I | 42 | R10 I/O | A0 I |
| 11 | R53/SRDY I/O | (1) I | 43 | R07 I/O | O7 I/O |
| 12 | R52/SCLK I/O | (1) I | 44 | R06 I/O | O6 I/O |
| 13 | R51/SOUT I/O | (1) I | 45 | R05 I/O | O5 I/O |
| 14 | R50/SIN I/O | (1) I | 46 | R04 I/O | O4 I/O |
| 15 | R47/T3O I/O | (2) I | 47 | R03 I/O | O3 I/O |
| 16 | R46/T1O I/O | (2) I | 48 | R02 I/O | O2 I/O |
| 17 | R45/EC2 I/O | CE I | 49 | R01 I/O | O1 I/O |
| 18 | R44/EC0 I/O | OE I | 50 | R00 I/O | O0 I/O |
| 19 | R43/INT3 I/O | (1) I | 51 | R37 I/O | (1) I |
| 20 | R42/INT2 I/O | (1) I | 52 | R36 I/O | (1) I |
| 21 | R41/INT1 I/O | (1) I | 53 | R35 I/O | (1) I |
| 22 | R40/INT0 I/O | (1) I | 54 | R34 I/O | (1) I |
| 23 | RESET I | (1) I | 55 | R33 I/O | (1) I |
| 24 | XIN I | (1) I | 56 | R32 I/O | (1) I |
| 25 | XOUT O | (3) O | 57 | R31 I/O | (1) I |
| 26 | VSS - | VSS - | 58 | R30 I/O | (1) I |
| 27 | R27 I/O | A15 I | 59 | VDD - | VDD - |
| 28 | R26 I/O | A14 I | 60 | MP I | VPP - |
| 29 | R25 I/O | A13 I | 61 | AVSS I | (1) I |
| 30 | R24 I/O | A12 I | 62 | AVREF I | (1) I |
| 31 | R23 I/O | A11 I | 63 | R67/AN7 I/O | (1) I |
| 32 | R22 I/O | A10 I | 64 | R66/AN6 I/O | (1) I |

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

64LQFP Package for GMS81516AT

| Pin No. | MCU Mode | OTP Mode | Pin No. | MCU Mode | OTP Mode |
|---------|-----------------------------------|--------------------------|---------|---------------------|-------------------|
| 1 | R63/AN3 I | (1) I | 33 | R17 I/O | A7 I |
| 2 | R62/AN2 I | (1) I | 34 | R16 I/O | A6 I |
| 3 | R61/AN1 I | (1) I | 35 | R15 I/O | A5 I |
| 4 | R60/AN0 I | (1) I | 36 | R14 I/O | A4 I |
| 5 | R57/PWM1 I/O | (1) I | 37 | R13 I/O | A3 I |
| 6 | R56/PWM0 I/O | (1) I | 38 | R12 I/O | A2 I |
| 7 | R55/BUZ I/O | (1) I | 39 | R11 I/O | A1 I |
| 8 | R54/WDTO I/O | (1) I | 40 | R10 I/O | A0 I |
| 9 | R53/ $\overline{\text{SRDY}}$ I/O | (1) I | 41 | R07 I/O | O7 I/O |
| 10 | R52/SCLK I/O | (1) I | 42 | R06 I/O | O6 I/O |
| 11 | R51/SOUT I/O | (1) I | 43 | R05 I/O | O5 I/O |
| 12 | R50/SIN I/O | (1) I | 44 | R04 I/O | O4 I/O |
| 13 | R47/T3O I/O | (2) I | 45 | R03 I/O | O3 I/O |
| 14 | R46/T1O I/O | (2) I | 46 | R02 I/O | O2 I/O |
| 15 | R45/ $\overline{\text{EC2}}$ I/O | $\overline{\text{CE}}$ I | 47 | R01 I/O | O1 I/O |
| 16 | R44/ $\overline{\text{EC0}}$ I/O | $\overline{\text{OE}}$ I | 48 | R00 I/O | O0 I/O |
| 17 | R43/INT3 I/O | (1) I | 49 | R37 I/O | (1) I |
| 18 | R42/INT2 I/O | (1) I | 50 | R36 I/O | (1) I |
| 19 | R41/INT1 I/O | (1) I | 51 | R35 I/O | (1) I |
| 20 | R40/INT0 I/O | (1) I | 52 | R34 I/O | (1) I |
| 21 | $\overline{\text{RESET}}$ I | (1) I | 53 | R33 I/O | (1) I |
| 22 | X _{IN} I | (1) I | 54 | R32 I/O | (1) I |
| 23 | X _{OUT} O | (3) O | 55 | R31 I/O | (1) I |
| 24 | V _{SS} - | V _{SS} - | 56 | R30 I/O | (1) I |
| 25 | R27 I/O | A15 I | 57 | V _{DD} - | V _{DD} - |
| 26 | R26 I/O | A14 I | 58 | MP I | V _{PP} - |
| 27 | R25 I/O | A13 I | 59 | AV _{SS} I | (1) I |
| 29 | R24 I/O | A12 I | 60 | AV _{REF} I | (1) I |
| 29 | R23 I/O | A11 I | 61 | R67/AN7 I/O | (1) I |
| 30 | R22 I/O | A10 I | 62 | R66/AN6 I/O | (1) I |
| 31 | R21 I/O | A9 I | 63 | R65/AN5 I/O | (1) I |
| 32 | R20 I/O | A8 I | 64 | R64/AN4 I/O | (1) I |

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

PIN FUNCTION (OTP Mode)**V_{PP} (Program Voltage)**

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

 $\overline{\text{CE}}$ (Chip Enable)

$\overline{\text{CE}}$ is the input for programming and verifying internal EPROM.

 $\overline{\text{OE}}$ (Output Enable)

$\overline{\text{OE}}$ is the input of data output control signal for verify.

A₀~A₁₅ (Address Bus)

A₀~A₁₅ are address input pins for internal EPROM.

O₀~O₇ (EPROM Data Bus)

These are data bus for internal EPROM.

PROGRAMMING

The GMS81516AT has address A₀~A₁₅ pins. Therefore, the programmer just program the data (from 4000_H to 7FFF_H) into the GMS81516AT OTP device, during addresses A₁₄, A₁₅ must be pulled to a logic high. When the programmer write the data from 4000_H to 7FFF_H, consequently, the data actually will be written into addresses C000_H to FFFF_H of the OTP device.

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

Ex) "Motorola S1" format;

S0080000574154434880

S1244000E1FF3BFF04A13F8F06E101711B821B1BE01D1B3B191BF6181BF01C1BFF081BFF0AEO

S12440211BF5091BFF0B1BFF3F1B003E1B003D1B003C1BFF3B1B003A1BFF391BFF381BFF353D

:

:

S1057FF2983FB2

S1057FFEFF3F3F

S9030000FC

2. Down load above data into programmer from PC.

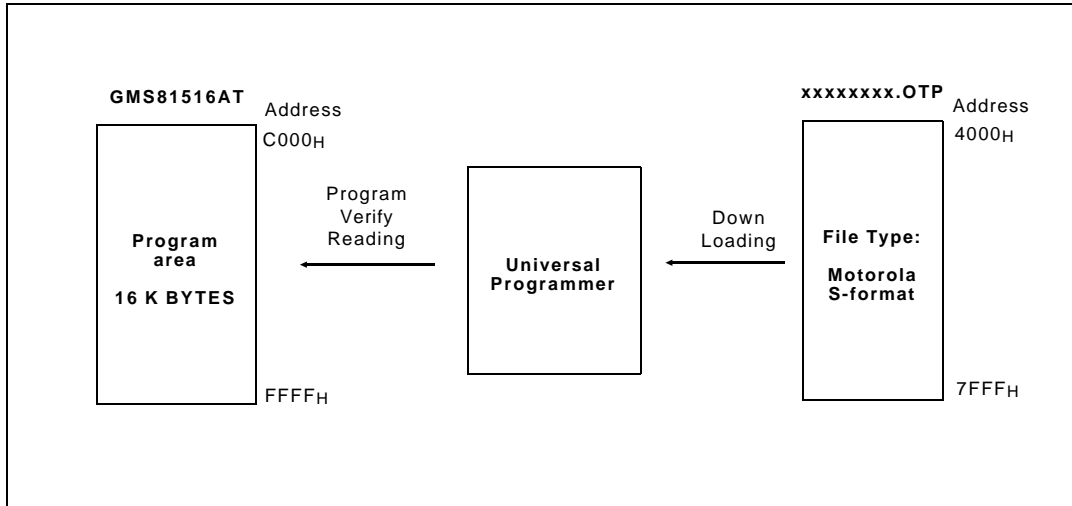
3. Programming the data from address 4000_H to 7FFF_H into the OTP MCU, the data must be turned over respectively, and then record the data. When read the data, it also must be turned over.

Ex) 00(00000000)→FF(11111111), 76(01110110)→89(10001001), FF(11111111)→00(00000000) etc.

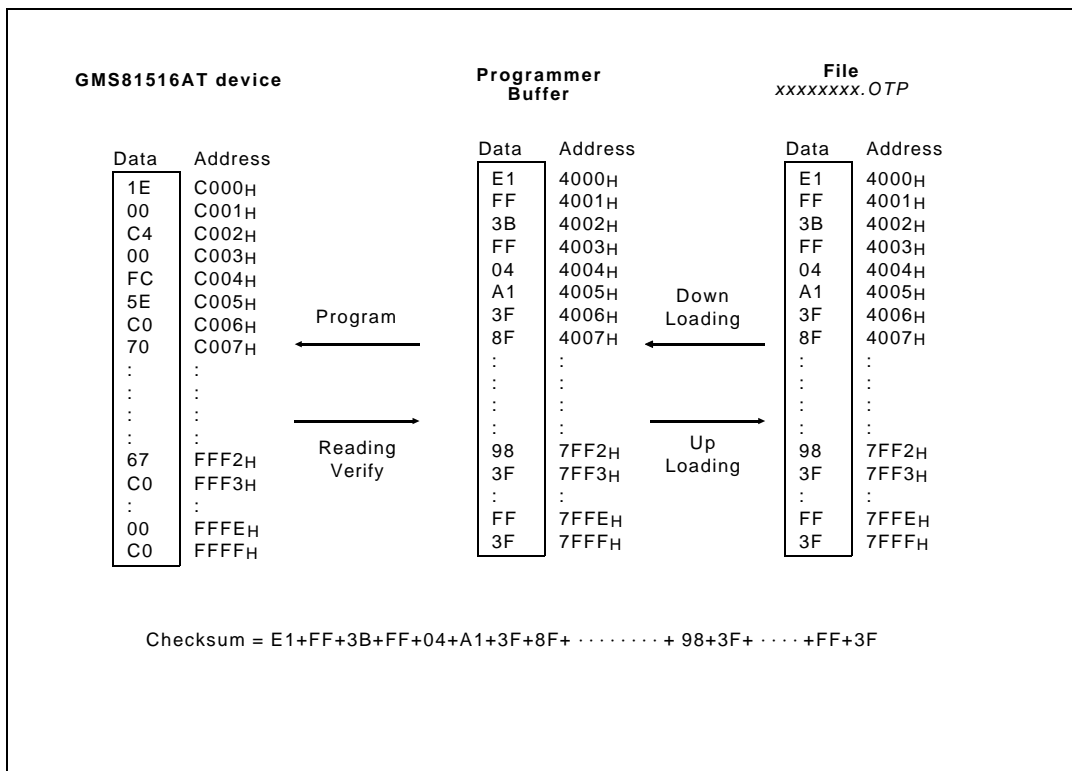
4. Of course, the check sum is result of the sum of whole data from address 4000_H to 7FFF_H in the file (not reverse data of OTP MCU).

* When GMS81516AT shipped, the blank data of GMS81516AT is initially 00_H (not FF_H).



Programming Flow



Programming Example



DEVICE OPERATION MODE(T_A = 25°C ± 5°C)

| Mode | \overline{CE} | \overline{OE} | A0-A15 | V _{PP} | V _{DD} | 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.

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 | |

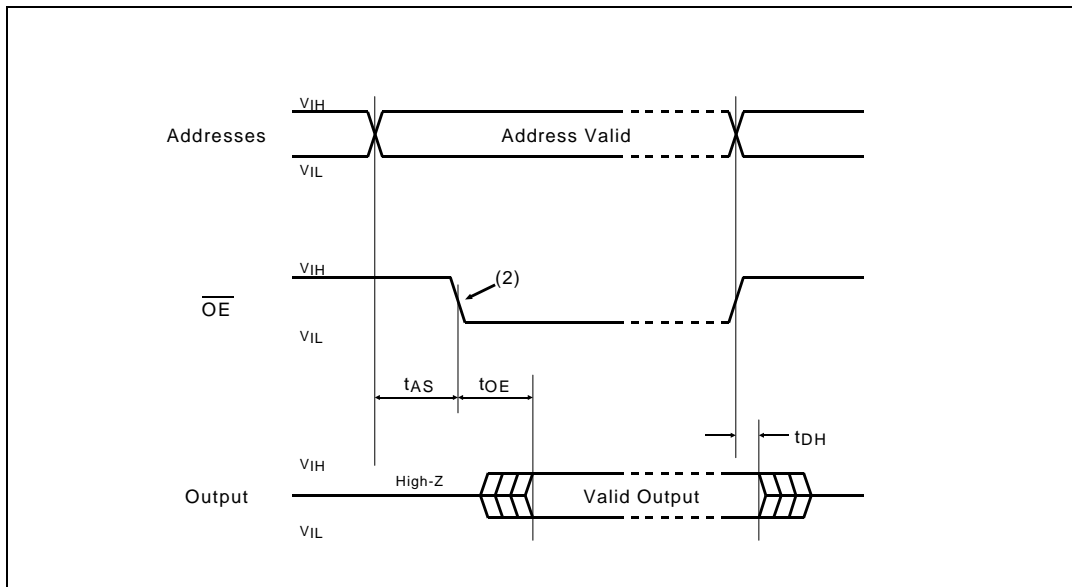
NOTES:

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 O₀ to O₇ 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 |

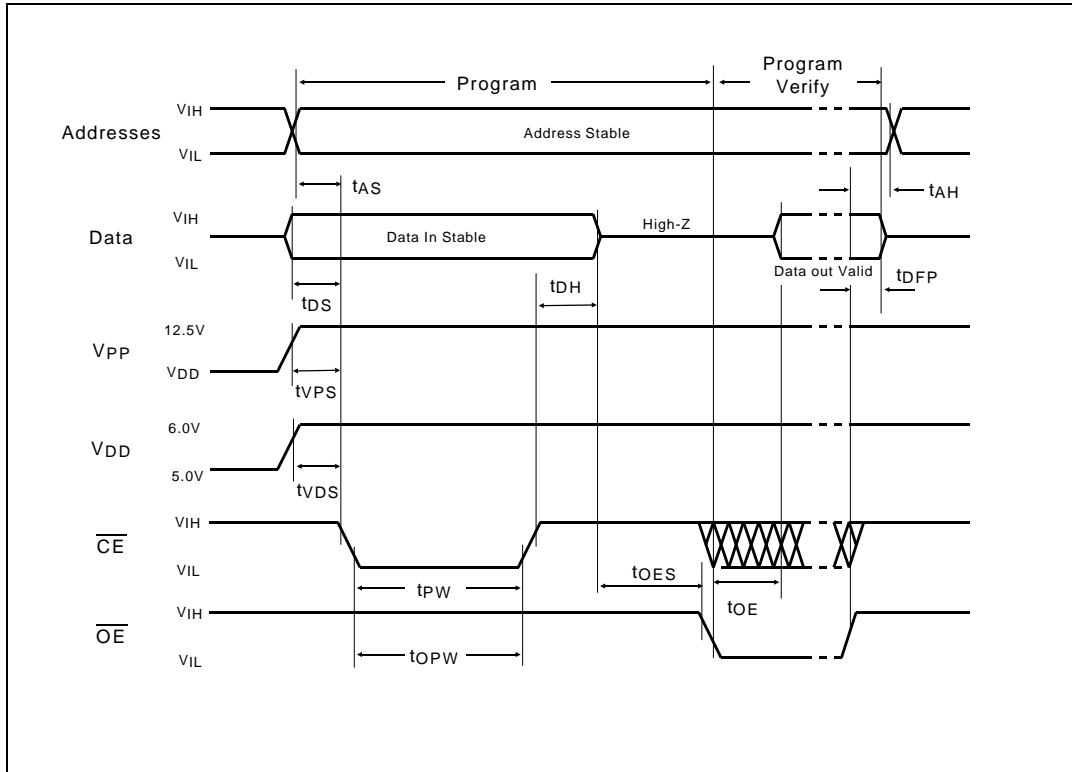
READING WAVEFORMS



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 $V_{DD}=5.0V$

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:

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

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} | $\overline{\text{OE}}$ 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} | $\overline{\text{CE}}$ 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:

- V_{DD} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
- 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. Refer to page 13.

Intelligent Programming Algorithm

