Table of Contents

GMS81C3004

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH LCD DRIVER

1. OVERVIEW

1.1 Description

The GMS81C3004 is an advanced CMOS 8-bit microcontroller with 4K bytes of ROM. The device is one of GMS800 family. The LG Semicon GMS81C3004 is a powerful microcontroller which provides a highly flexible and cost effective solution to many LCD applications such as controller with LCD and toys. The GMS81C3004 provides the following standard features: 4K bytes of ROM, 256 bytes of RAM, 8-bit timer/counter, on-chip oscillator and clock circuitry. In addition, the GMS81C3004 supports power saving modes to reduce power consumption.

Device name	ROM Size	RAM Size	Package
GMS81C3004	4K bytes	256 bytes	80QFP or DIE

1.2 Features

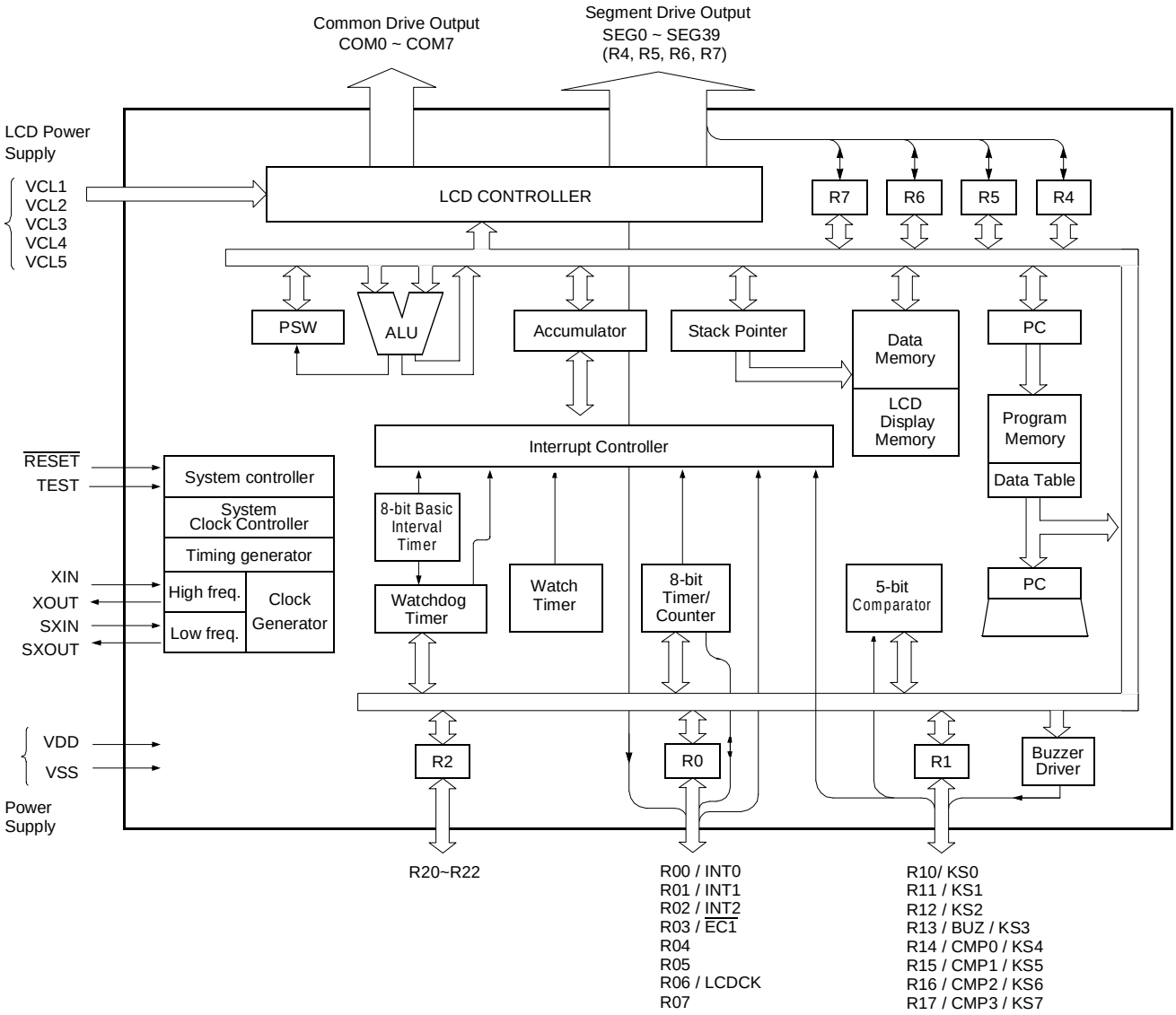
- **4K Bytes On-chip Program Memory**
- **256 Bytes of On-chip Data RAM (Included 64 bytes stack memory)**
- **Dot Matrix LCD Driver**
  - **Max. 320 dots (40 seg. x 8 com.)**
  - **40 bytes of Display RAM**
- **Instruction Cycle Time:**
  - **0.5us, 1.9us, 3.8us, 15.2us at 4.19MHz**
  - **61us, 244us, 488us, 1.95ms at 32.768KHz**
- **51 Programmable I/O pins (Included 32 LCD pins)**
- **2.2V to 5.5V Wide Operating Range**
- **Dual Clock Operation (4.19MHz, 32kHz)**
- **One 8-bit Basic Interval Timer**
- **Key Scan**
- **One 8-bit Timer/ Counter**
- **Watch Timer**
- **Watchdog timer**
- **Eight Interrupt sources**
  - **External input: 3**
  - **Keyscan input: 1**
  - **Timer: 4**
- **Buzzer Driving port**
  - **500Hz ~ 130kHz**
- **4-channel 5-bit On-chip Comparator**
- **Power Down Mode**
  - **STOP mode**
  - **SLEEP mode**

1.3 Development Tools

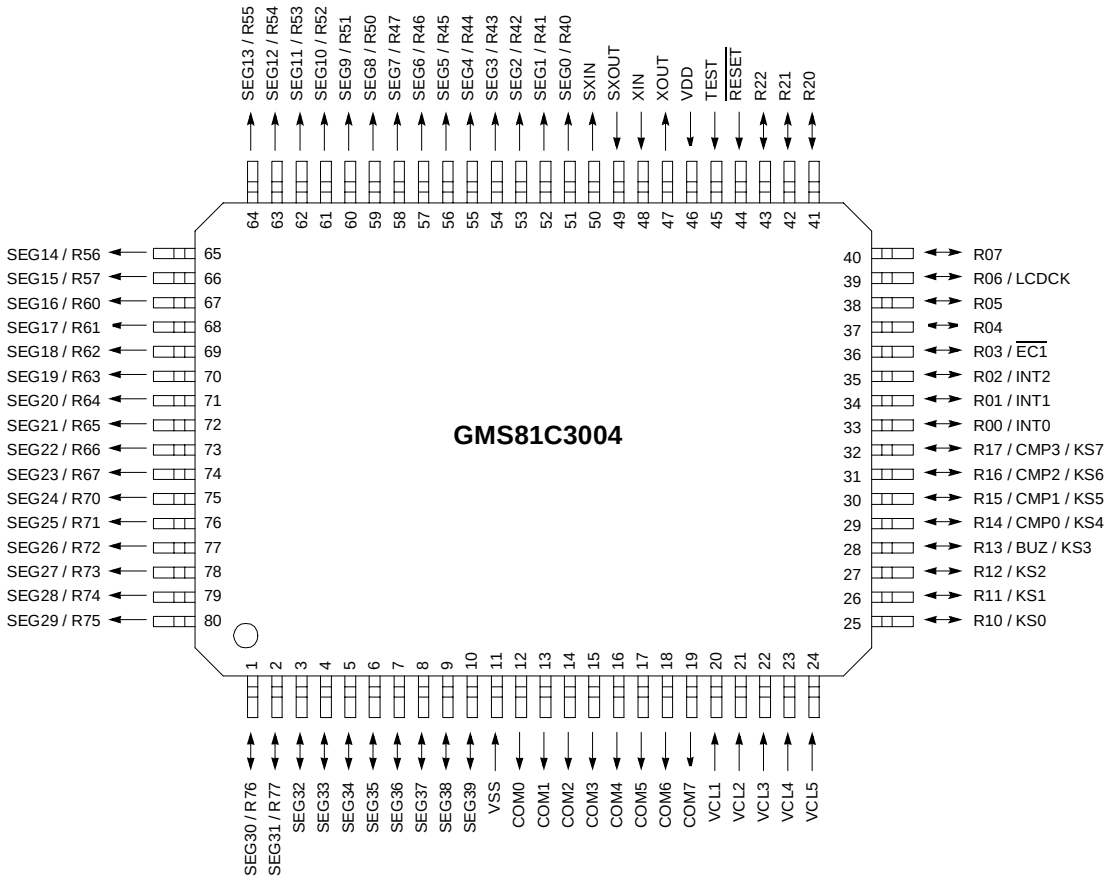
The GMS81C3004 is supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Dr^{TM}.

In Circuit Emulators	CHOICE-Dr. (with EVA81C)
LCD Simulator	Under development
Assembler	LGS Macro Assembler

2. BLOCK DIAGRAM



3. PIN ASSIGNMENT



4. PACKAGE DIAGRAM

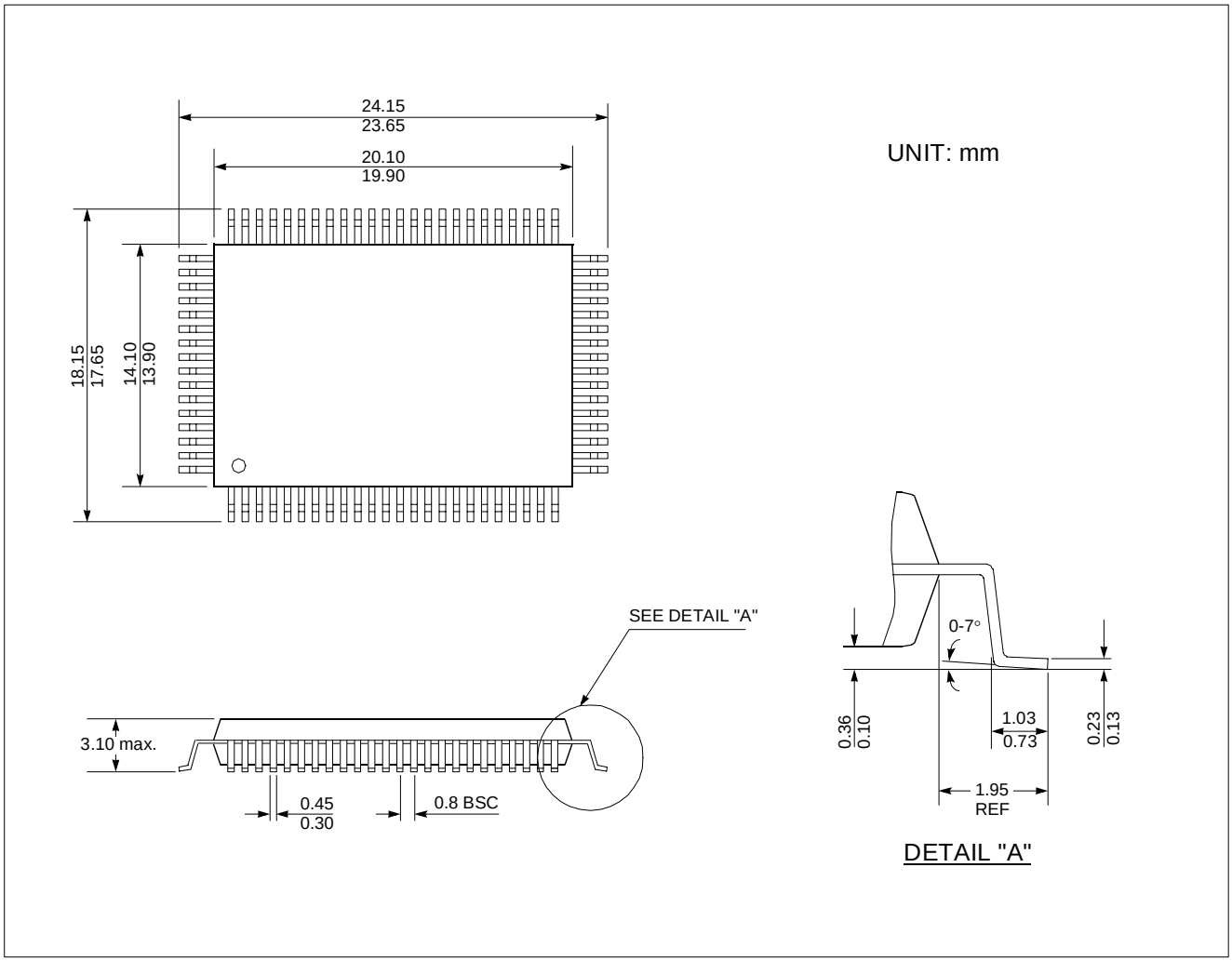


Figure 4-1 Package Diagram

5. PIN FUNCTION

V_{DD}: Supply voltage.

V_{SS}: Circuit ground.

TEST: Used for shipping inspection of the IC. For normal operation, it should be connected to V_{SS}.

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

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

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

SX_{IN}: Input to the internal sub system clock operating circuit.

SX_{OUT}: Output from the inverting subsystem oscillator amplifier.

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

In addition, R0 serves the functions of the various following special features.

Port pin	Alternate function
R00	INT0 (External interrupt 0)
R01	INT1 (External interrupt 1)
R02	INT2 (External interrupt 2)
R03	Event counter input
R06	LCD clock output

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

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

Port pin	Alternate function
R10	KS0 (Key scan input 0)
R11	KS1 (Key scan input 1)
R12	KS2 (Key scan input 2)
R13	BUZ / KS3 (Buzzer output or Key scan input 3)
R14	CMP0 / KS4 (Comparator input or Key scan input 4)
R15	CMP1 / KS5 (Comparator input or Key scan input 5)
R16	CMP2 / KS6 (Comparator input or Key scan input 6)
R17	CMP3 / KS7 (Comparator input or Key scan input 7)

R20~R22: R2 is a 3-bit CMOS bidirectional I/O port. Each pins 1 or 0 written to the their Port Direction Register can be used as outputs or inputs.

R40~R47, R50~57, R60~R67, R70~R77:
R4, R5, R6, R7 are four 8-bit CMOS bidirectional I/O port. Each pins 1 or 0 written to the their Port Direction Register can be used as outputs or inputs.

Ports is multiplexed with SEG0~SEG31 respectively.

Port pin	Alternate function
SEG0~SEG7	R40~R47
SEG8~SEG15	R50~R57
SEG16~SEG23	R60~R67
SEG24~SEG31	R70~R77

After the reset of the MCU, port is initialized as a segment output port.

SEG0~SEG39: Segment signal output pins for the LCD display. See "15. LCD DRIVER" on page 55 for details.

COM0~COM7: Common signal output pins for the LCD display. See "15. LCD DRIVER" on page 55 for details.

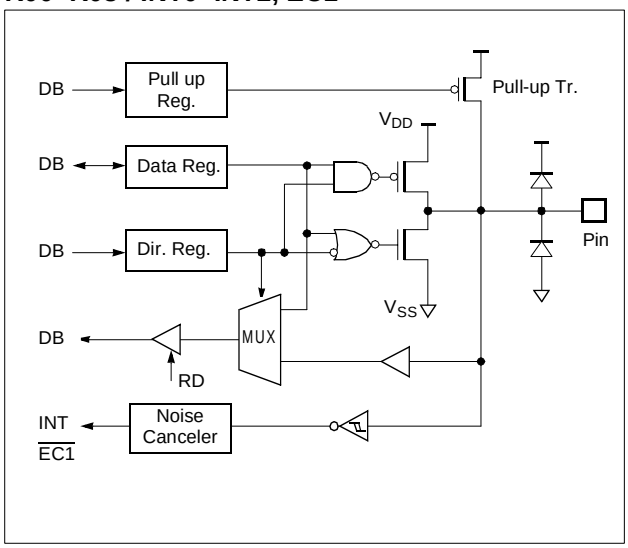
V_{CL1}~V_{CL5}: Power supply pins for the LCD driver. Since the LCD driving resistors are provided internally, no lines should be connected to these pins. The voltage on each pin is V_{DD}> V_{CL1}> V_{CL2}> V_{CL3}> V_{CL4}> V_{CL5}> V_{SS}. For details, Refer to Section "15.".

PIN NAME	Pin No.	In/Out	Function	
V_{DD}	46	-	Supply voltage	
V_{SS}	11	-	Circuit ground	
TEST	45	I	For test purposes. Should connect it to GND for normal operation.	
$\overline{RESET}$	44	I	Reset signal input	
VCL1~VCL5	20~24	-	LCD power supply	
X_{IN}	48	I	Main oscillation input	
X_{OUT}	47	O	Main oscillation output	
SX_{IN}	50	I	Sub oscillation input	
SX_{OUT}	49	O	Sub oscillation output	
R00 (INT0)	33	I/O (Input)	8-bit general I/O ports	External interrupt 0 input
R01 (INT1)	34	I/O (Input)		External interrupt 1 input
R02 (INT2)	35	I/O (Input)		External interrupt 2 input
R03 (EC1)	36	I/O (Input)		External counter input
R04	37	I/O		-
R05	38	I/O		-
R06 (LCDCK)	39	I/O (Output)		LCD clock output
R07	40	I/O		-
R10 (KS0)	25	I/O (Input)	8-bit general I/O ports	Key scan input
R11 (KS1)	26	I/O (Input)		
R12 (KS2)	27	I/O (Input)		
R13 (BUZ/KS3)	28	I/O (Output/Input)		Buzzer output or key scan input
R14~R17 (CMP0~CMP3/ KS4~KS7)	29~32	I/O (Input/Input)		Comparator input 0~3 or key scan input 4~7
R20~R22	41,42, 43	I/O	3-bit general I/O ports	-
SEG0~SEG7 (R40~R47)	51~58	Output (I/O)	Segment signal output ports	8-bit general I/O ports
SEG8~SEG15 (R50~R57)	59~66	Output (I/O)		8-bit general I/O ports
SEG16~SEG23 (R60~R67)	67~74	Output (I/O)		8-bit general I/O ports
SEG24~SEG31 (R70~R77)	1,2, 75~80	Output (I/O)		8-bit general I/O ports
SEG32~SEG39	3~10	O	Segment signal output ports	
COM0~COM7	12~19	O	Common signal output ports	

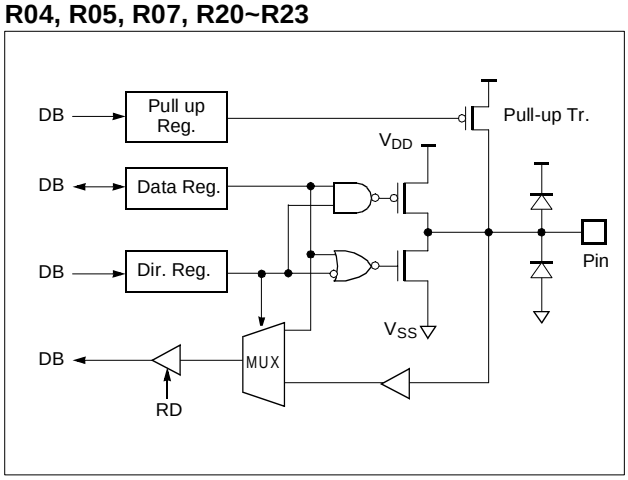
Table 5-1 Port Function Description

6. PORT STRUCTURES

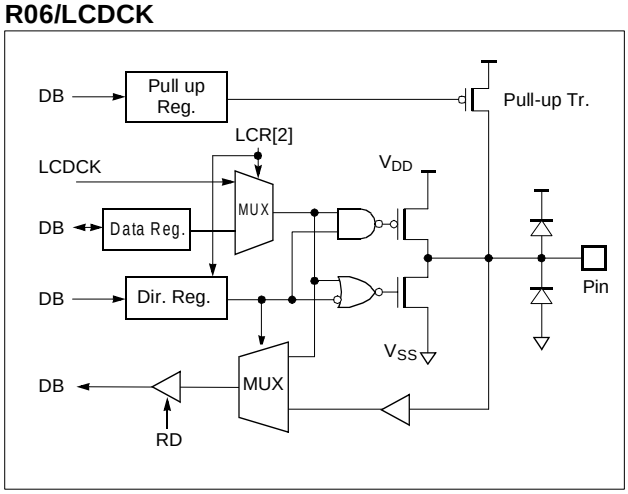
R00~R03 / INT0~INT2, $\overline{EC1}$



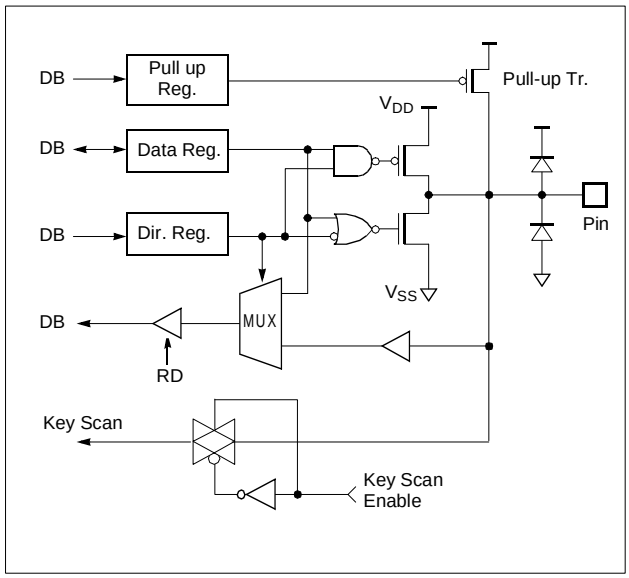
R04, R05, R07, R20~R23



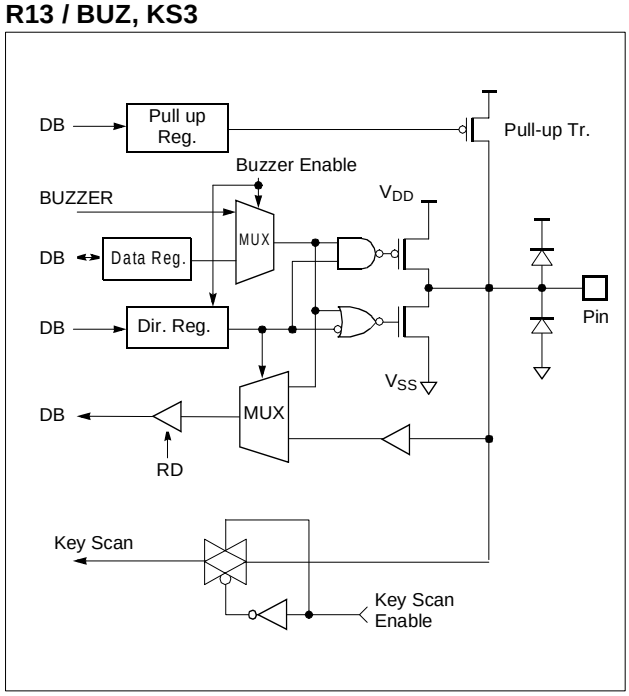
R06/LCDCK



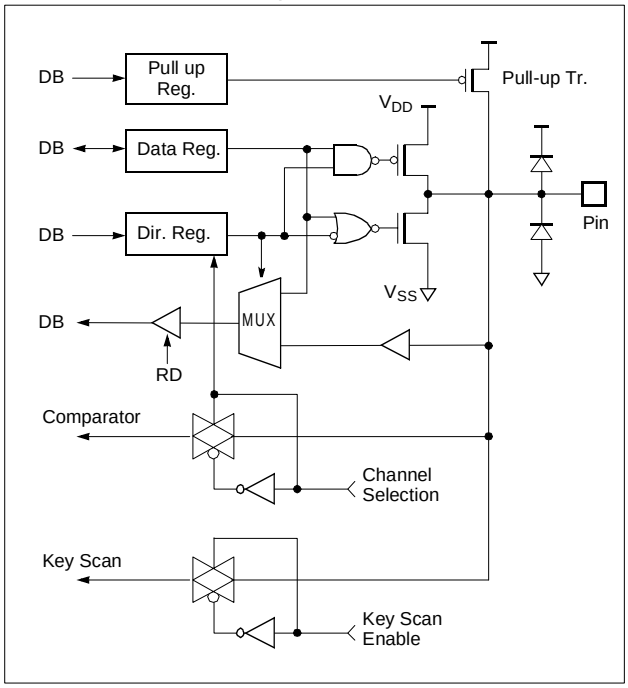
R10~R12 / KS0~KS2



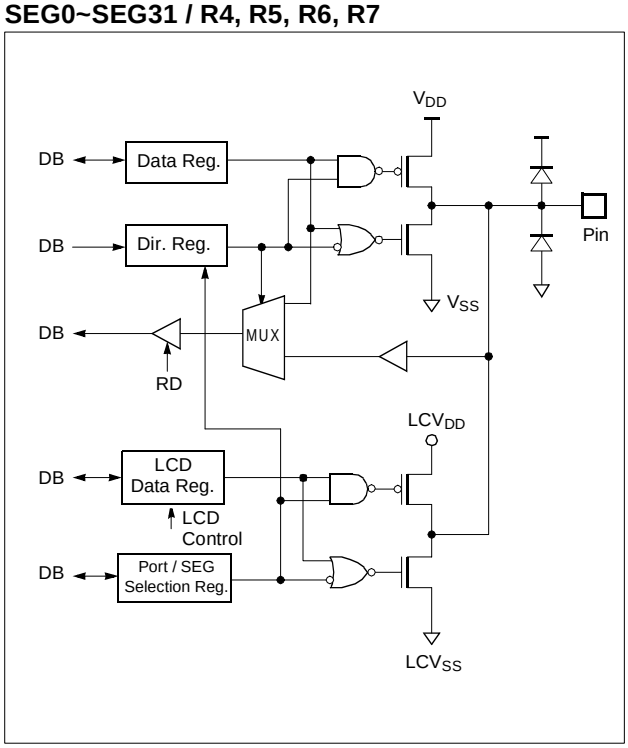
R13 / BUZ, KS3



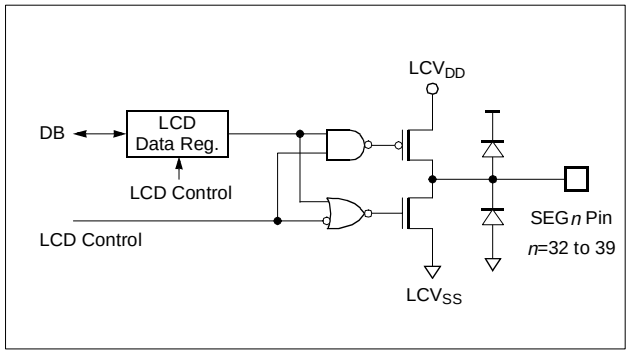
R14~R17 / CIN0~CIN3, KS4~KS7



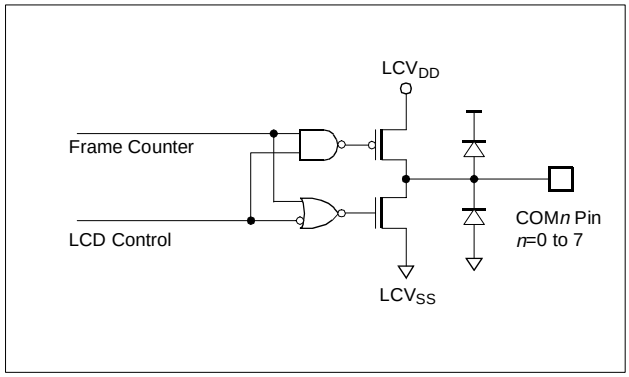
SEG0~SEG31 / R4, R5, R6, R7



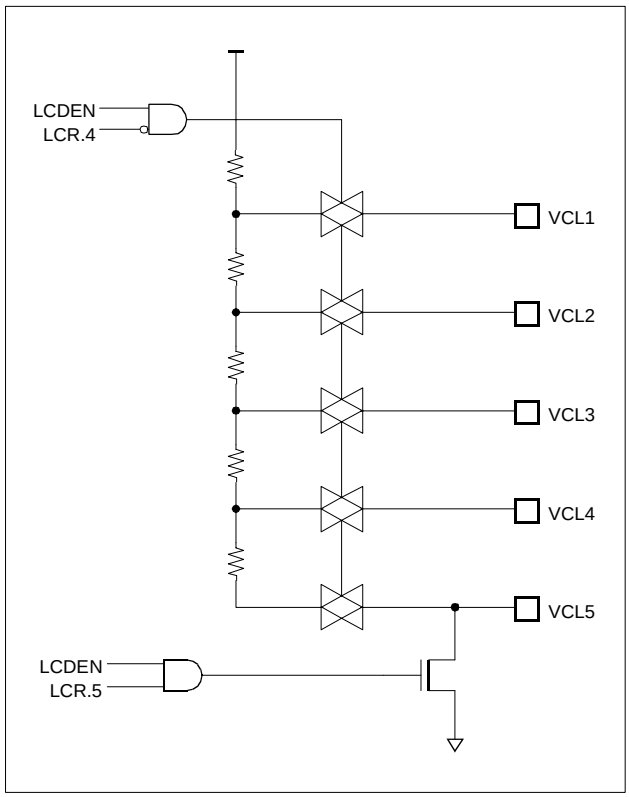
SEG32 ~ SEG39



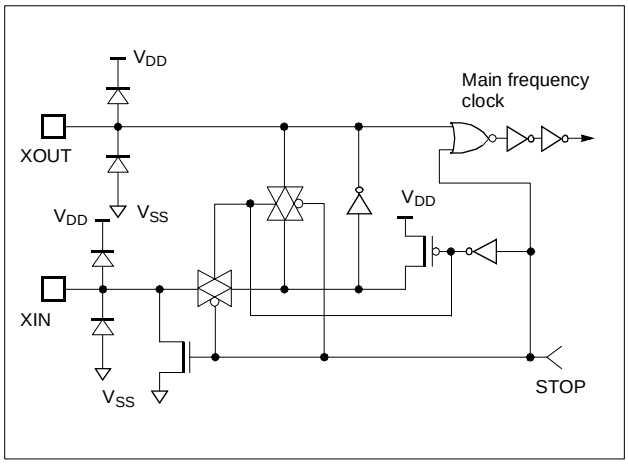
COM0 ~ COM7



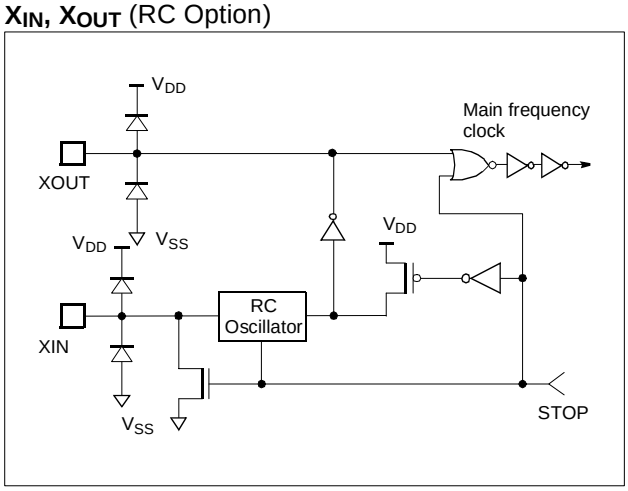
VCL1 ~ VCL5



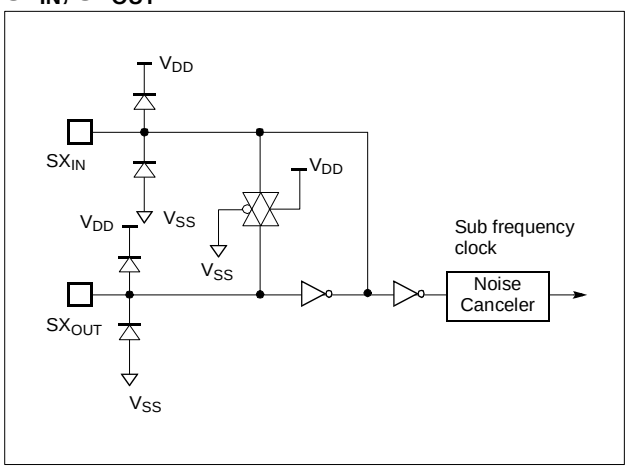
X_{IN}, X_{OUT} (Crystal or Ceramic resonator Option)



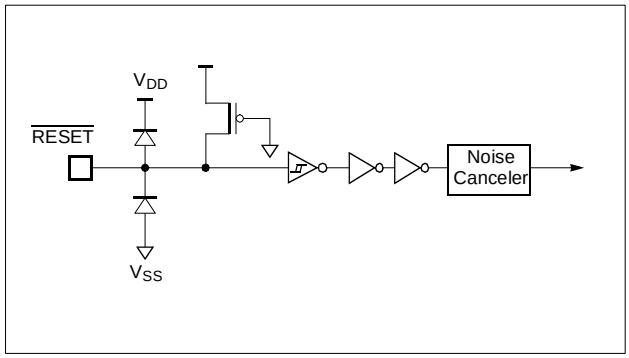
X_{IN}, X_{OUT} (RC Option)



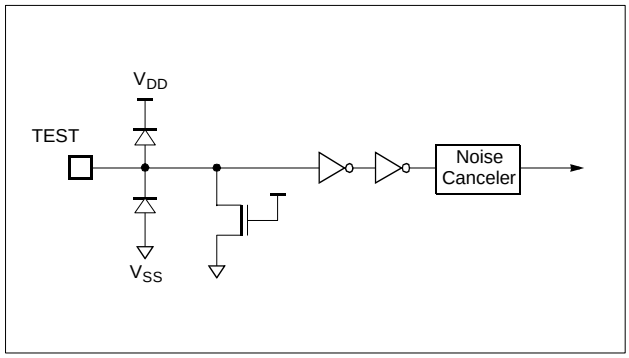
SX_{IN}, SX_{OUT}



RESET



TEST



7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings

Supply voltage ..... -0.3 to +6.0 V

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

Voltage on any pin with respect to Ground (V_{SS}) ..... -0.3 to V_{DD}+0.3

Maximum current out of V_{SS} pin ..... 100 mA

Maximum current into V_{DD} pin ..... 80 mA

Maximum current sunk by (I_{OL} per I/O Pin) ..... 20 mA

Maximum output current sourced by (I_{OH} per I/O Pin) ..... 8 mA

Maximum current (ΣI_{OL}) ..... 80 mA

Maximum current (ΣI_{OH}) ..... 50 mA

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

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Specifications		Unit
			Min.	Max.	
Supply Voltage	V_{DD}	f_{XIN}=4.19MHz f_{SXIN}=32.768kHz	2.2	5.5	V
Operating Frequency	f_{XIN}	V_{DD}=2.2~5.5V	1	4.5	MHz
Sub Operating Frequency	f_{SXIN}	V_{DD}=2.2~5.5V	32	35	kHz
Operating Temperature	T_{OPR}		-20	85	°C

7.3 DC Electrical Characteristics

(T_A=-20~85°C, V_{DD}=2.2~5.5V),

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH1}	All input pins except X_{IN} and SX_{IN}	0.8 V_{DD}	-	V_{DD}	V
	V_{IH2}	X_{IN} and SX_{IN}	V_{DD}-0.5	-	V_{DD}	V
Input Low Voltage	V_{IL1}	All input pins except X_{IN} and SX_{IN}	-	-	0.2 V_{DD}	V
	V_{IL2}	X_{IN} and SX_{IN}	-	-	0.4	V
Output High Voltage	V_{OH}	V_{DD}=2.2 ~ 5.5V, I_{OH1}=-500μA R0,R1,R2,R4,R5,R6,R7	0.8 V_{DD}	-	-	V
Output Low Voltage	V_{OL}	V_{DD}=2.2 ~ 5.5V, I_{OL1}=500μA R0,R1,R2,R4,R5,R6,R7	-	-	0.1 V_{DD}	V
Input High Leakage Current	I_{IH1}	V_{IN}=V_{DD} , All input pins except X_{IN}, SX_{IN}	-	-	3	μA
	I_{IH2}	V_{IN}=V_{DD}, X_{IN}, SX_{IN}	-	-	20	μA
Input Low Leakage Current	I_{IL1}	V_{IN}=V_{DD} , All input pins except X_{IN}, SX_{IN}	-	-	-3	μA
	I_{IL2}	V_{IN}=V_{DD}, X_{IN}, SX_{IN}	-	-	-20	μA

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Output High Leakage Current	I_{OHL}	V_O= V_{DD}, All output pins	-	-	3	μA
Output Low Leakage Current	I_{OLL}	V_O=0V , All output pins	-	-	-3	μA
Pull-up Resistor[1]	R_{PORT}	V_{IN}=0V, V_{DD}=3V±10%, R0, R1, R2	50	100	200	kΩ
	R_{RESET}	V_{IN}=0V, V_{DD}=3V±10%, $\overline{RESET}$	30	60	120	
LCD Voltage Dividing Resistor	R_{LCD}	V_{DD}=2.7 ~ 5.5V	50	70	90	kΩ
Voltage Drop |V_{DD}-COM*n*| , *n*=0~7	V_{DC}	V_{DD}=2.7 ~ 5.5V -15μA per common pin	-	-	120	mV
Voltage Drop |V_{DD}-SEG*n*| , *n*=0~39	V_{DS}	V_{DD}=2.7 ~ 5.5V -15μA per segment pin	-	-	120	mV
V_{CL1} Output Voltage	V_{CL1}	V_{DD}=2.7 ~ 5.5V 1/4 bias (V_{CL2}=V_{CL3})	0.75V_{DD} -0.2	0.75V_{DD}	0.75V_{DD} +0.2	V
V_{CL2} Output Voltage	V_{CL2}		0.5V_{DD}- 0.2	0.5V_{DD}	0.5V_{DD}+ 0.2	
V_{CL3} Output Voltage	V_{CL3}		0.5V_{DD}- 0.2	0.5V_{DD}	0.5V_{DD}+ 0.2	
V_{CL4} Output Voltage	V_{CL4}		0.25V_{DD} -0.2	0.25V_{DD}	0.25V_{DD} +0.2	
V_{CL5} Output Voltage	V_{CL5}		-0.2	0	+0.2	
Supply Current[1]	I_{DD1}	**Main clock mode[2]** V_{DD}=3V±10% 4.19MHz Crystal Oscillator, C_{L1}=C_{L2}=30pF	-	1.4	3.0	mA
	I_{DD2}	**Sleep mode[3]** V_{DD}=3V±10% 4.19MHz Crystal Oscillator, C_{L1}=C_{L2}=30pF	-	0.6	1.0	mA
	I_{DD3}	**Sub clock mode[4]** V_{DD}=3V±10% S_{XIN}=32kHz	-	10	30	μA
	I_{DD4}	**Sleep mode** V_{DD}=3V±10% S_{XIN}=32kHz	-	6	10	μA
	I_{DD5}	**Stop mode[5]** V_{DD}=5V±10% S_{XIN}=0V	-	1.3	10	μA

1. The Data for 5V operation, refer to "7.6. Typical Characteristics" on page 14.
2. This mode set System Clock Mode Register(SCMR) to xxxx0000 that is f_{XIN}/2
3. This mode set SCMR to xxxx0000 (f_{XIN}/2)
4. Main-frequency clock stops and the sub-frequency clock is operates.

† Supply current in the following circuits are not included; on-chip pull-up resistors, internal LCD voltage dividing resistors, comparator voltage divide resistor, LVD circuit and output port drive currents.

5. Main-frequency clock stops and sub-frequency clock in not used

7.4 A/D Comparator Characteristics

(T_A=-20~85°C, V_{DD}=5.0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Analog Input Voltage Range	V_{AIN}	CMP0~CMP3	V_{SS}	-	V_{DD}	V
Accuracy	N_{FS}	-	-	-	±1	LSB

7.5 AC Characteristics

(T_A=-20~+85°C, V_{DD}=5V±10%, V_{SS}=0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Operating Frequency	f_{MAIN}	X_{IN}	1	-	4.5	MHz
	f_{SUB}	SX_{IN}	32	-	35	kHz
External Clock Pulse Width	t_{MCPW}	X_{IN}	80	-	500	nS
	t_{SCPW}	SX_{IN}	5	-	15	µS
External Clock Transition Time	t_{MRCP},t_{MFCP}	X_{IN}	-	-	20	nS
	t_{SRCP},t_{SFCP}	SX_{IN}	-	-	20	nS
Oscillation Stabilizing Time	t_{ST}	X_{IN}, X_{OUT}	-	-	20	mS
Interrupt Pulse Width	t_{IW}	INT0, INT1, INT2	2	-	-	t_{SYS}[1]
$\overline{\text{RESET}}$ Input Width	t_{RST}	$\overline{\text{RESET}}$	8	-	-	t_{SYS}[1]
Event Counter Input Pulse Width	t_{ECW}	$\overline{\text{EC1}}$	2	-	-	t_{SYS}[1]
Event Counter Transition Time	t_{REC},t_{FEC}	$\overline{\text{EC1}}$	-	-	20	nS

1. t_{SYS} is one of $2/f_{MAIN}$ or $8/f_{MAIN}$ or $16/f_{MAIN}$ or $64/f_{MAIN}$ in main clock operation mode,
   t_{SYS} is one of $2/f_{SUB}$ or $8/f_{SUB}$ or $16/f_{SUB}$ or $64/f_{SUB}$ in sub clock operation mode.

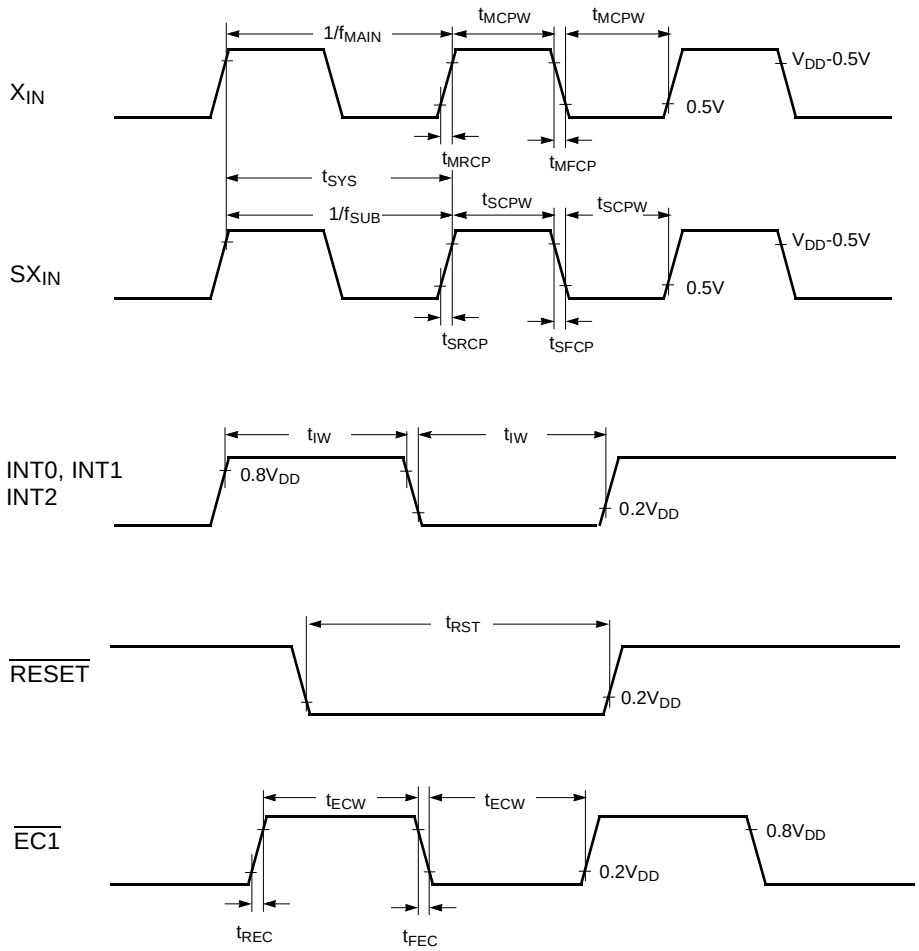


Figure 7-1 Timing Chart

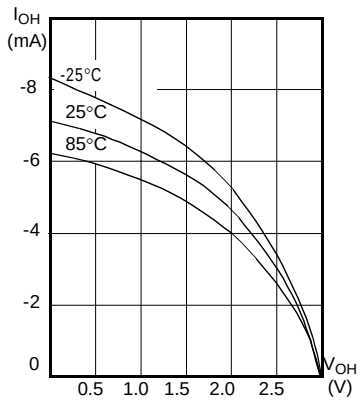
7.6 Typical Characteristics

This graphs and tables provided in this section are for design guidance only and are not tested or guranteed.

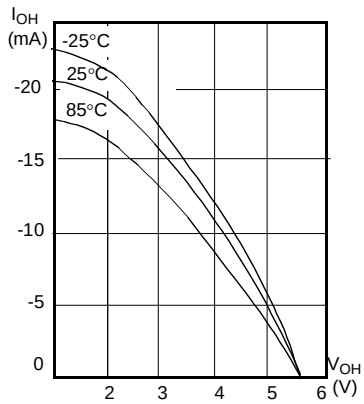
In some graphs or tables the data presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for imformation only and divices are guranteed to operate properly only within the specified range.

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

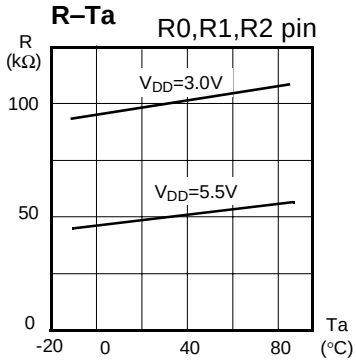
I_{OH}–V_{OH}, V_{DD}=3.0V



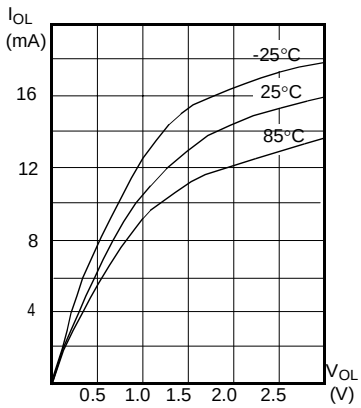
I_{OH}–V_{OH}, V_{DD}=5.5V



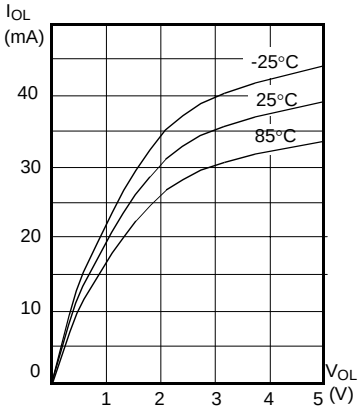
R–Ta R0,R1,R2 pin



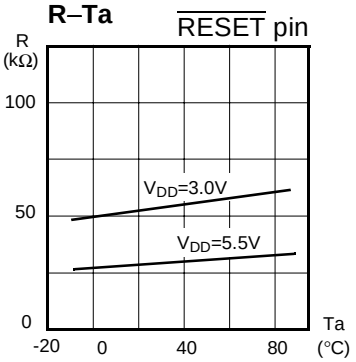
I_{OL}–V_{OL}, V_{DD}=3.0V



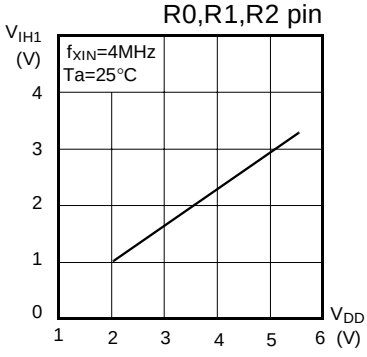
I_{OL}–V_{OL}, V_{DD}=5.5V



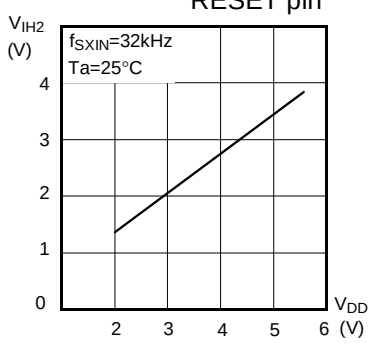
R–Ta $\overline{\text{RESET}}$ pin



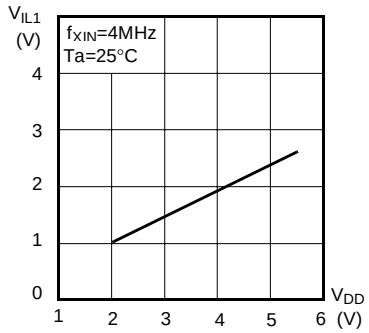
V_{DD}–V_{IH1} R0,R1,R2 pin



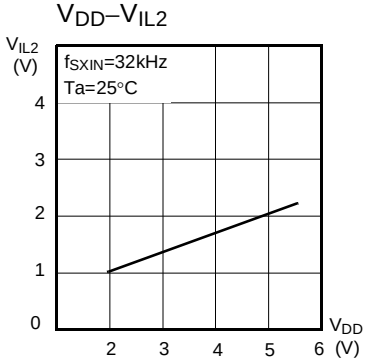
V_{DD}–V_{IH2} $\overline{\text{RESET}}$ pin



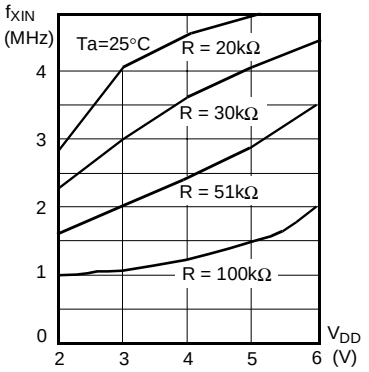
V_{DD}–V_{IL1}



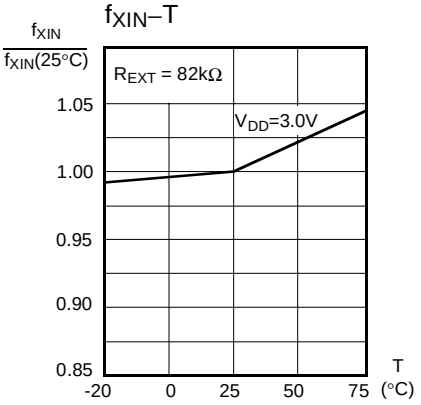
V_{DD}–V_{IL2}



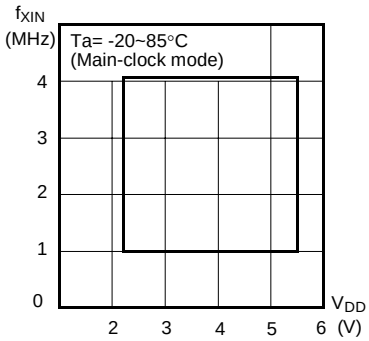
f_{XIN}–V_{DD}



f_{XIN}–T

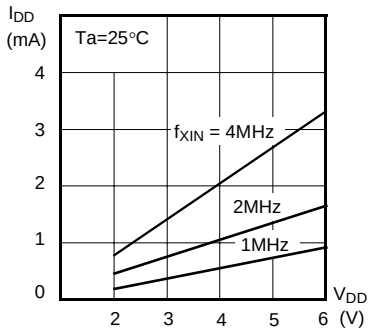


Operating Area



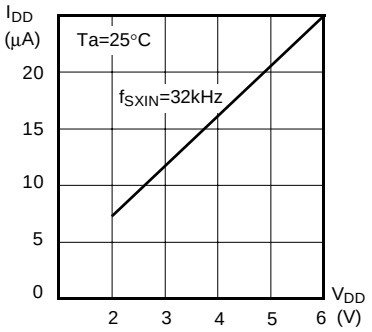
Normal Operation

I_{DD1}–V_{DD}



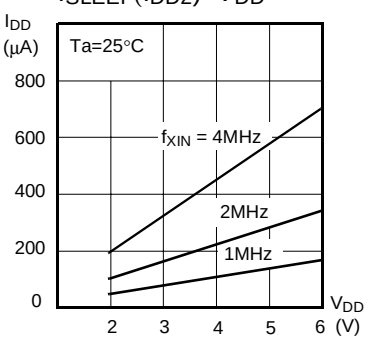
Normal Operation

I_{DD3}–V_{DD}



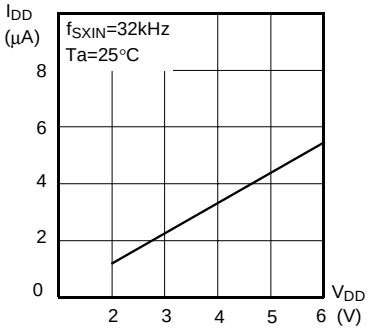
Sleep Mode

$I_{SLEEP}(I_{DD2})$–V_{DD}



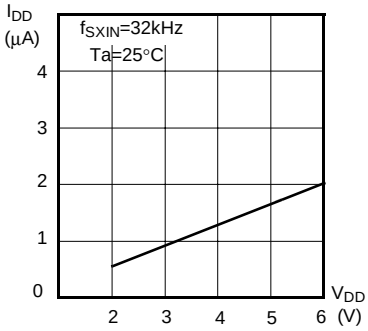
Sleep Mode

$I_{SLEEP}(I_{DD4})$–V_{DD}



Stop Mode

$I_{STOP}(I_{DD5})$–V_{DD}



8. MEMORY ORGANIZATION

The GMS81C3004 has separate address spaces for Program memory, Data Memory and Display memory. Program memory can only be read, not written to. It can be up to 4K bytes of Program memory. Data memory can be read and written to up to 256 bytes including the stack area. Display memory has prepared 40 bytes for LCD.

8.1 Registers

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

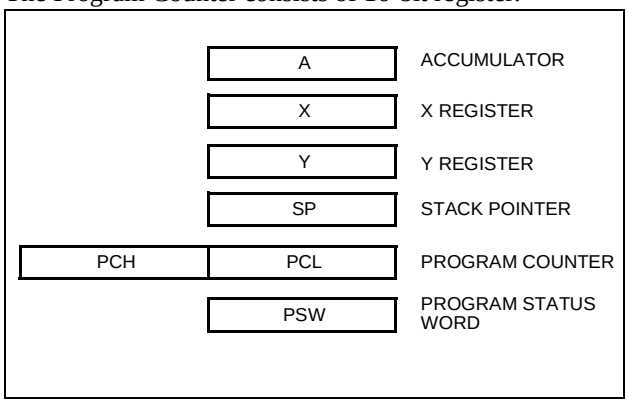


Figure 8-1 Configuration of Registers

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

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

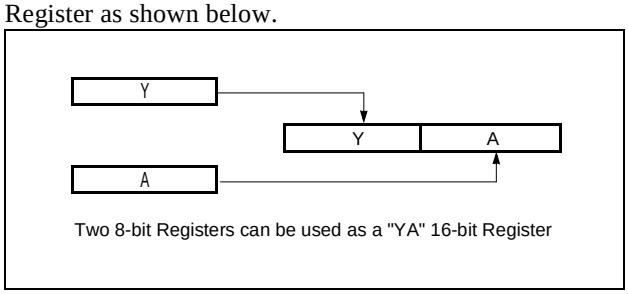


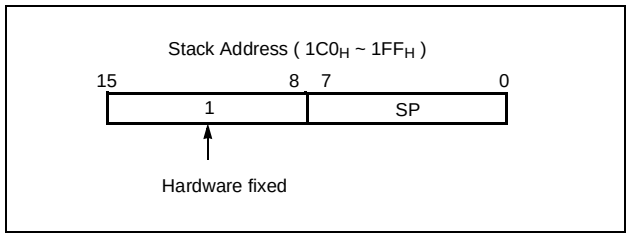
Figure 8-2 Configuration of YA 16-bit Register

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

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

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

The stack can be located at any position within $1C0_H$ to $1FF_H$ of the internal data memory. The SP is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of "FF_H" is used.



Caution:

The Stack Pointer must be initialized by software because its value is undefined after RESET.

Example: To initialize the SP

```
LDX    #0FFH
TXSP                ; SP ← FFH
```

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

Program Status Word: The Program Status Word (PSW) contains several bits that reflect the current state of the CPU. The PSW is described in Figure 8-3 . It contains the Negative flag, the Overflow flag, the Break flag the Half Carry (for BCD operation), the Interrupt enable flag, the Zero flag, and the Carry flag.

[Carry flag C]

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

[Zero flag Z]

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

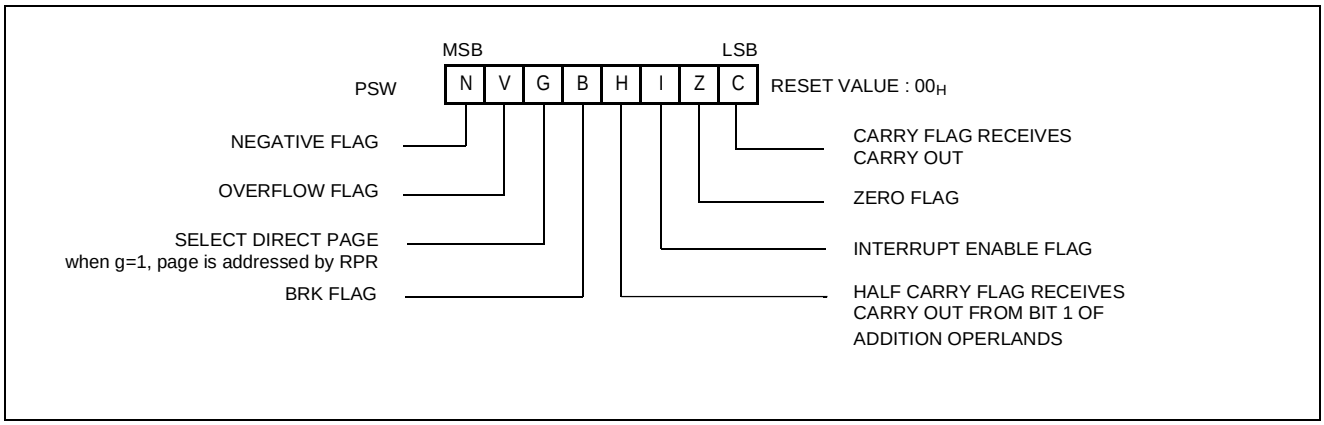


Figure 8-3 PSW (Program Status Word) Register

[Interrupt disable flag I]

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

[Half carry flag H]

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

[Break flag B]

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

[Direct page flag G]

This flag assigns RAM page for direct addressing mode. In the direct addressing mode, addressing area is from zero page 00_H to $0FF_H$ when this flag is "0". If it is set to "1", addressing area is assigned by RPR register (address $0F8_H$). It is set by SETG instruction and cleared by CLRG.

[Overflow flag V]

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

[Negative flag N]

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

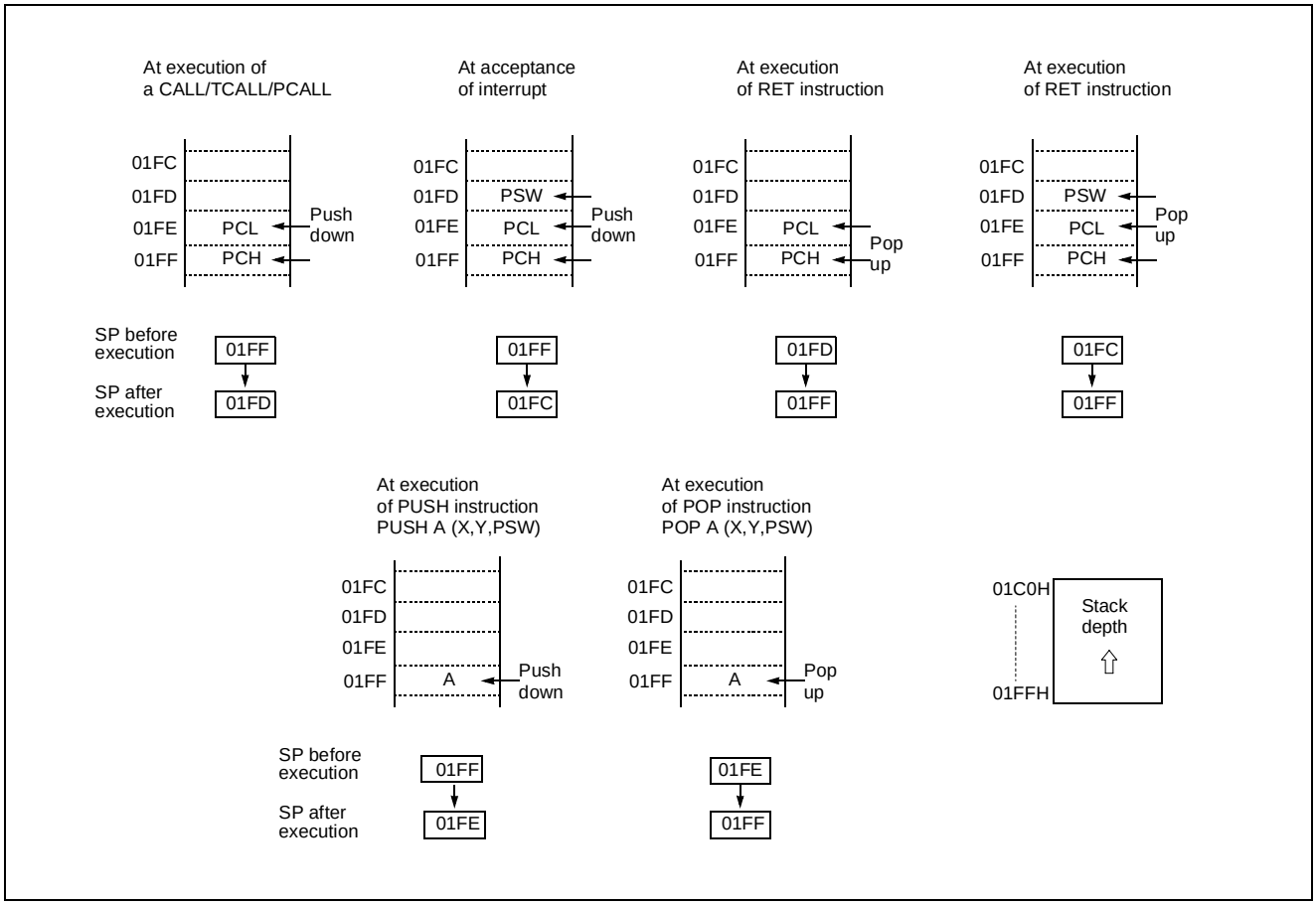


Figure 8-4 Stack Operation

8.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 4K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

Figure 8-5 , shows a map of Program Memory. After reset, the CPU begins execution from reset vector which is stored in address $FFFE_H$ and $FFFF_H$ as shown in Figure 8-6 .

As shown in Figure 8-5 , each area is assigned a fixed location in Program Memory. Program Memory area contains the user program.

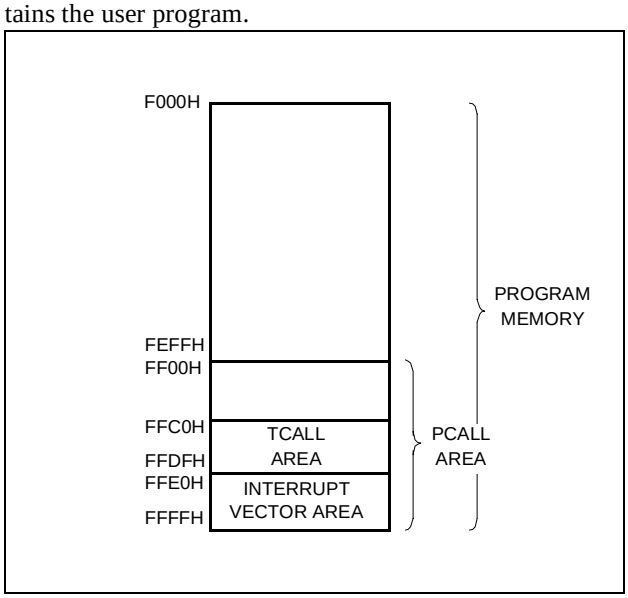


Figure 8-5 Program Memory Map

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

Table Call (TCALL) causes the CPU to jump to each TCALL address, where it commences the execution of the service routine. The Table Call service area spaces 2-byte for every TCALL: $0FFC0_H$ for TCALL15, $0FFC2_H$ for TCALL14, etc., as shown in Figure 8-7 .

Example: Usage of TCALL

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

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

Any area from $0FF00_H$ to $0FFFF_H$, if it is not going to be used, its service location is available as general purpose Program Memory.

Address	Vector Area Memory
$0FFE0_H$	-
E2	-
E4	-
E6	Key Scan Interrupt Vector Area
E8	Watch Timer Interrupt Vector Area
EA	-
EC	-
EE	-
F0	Watchdog Timer Interrupt Vector Area
F2	External Interrupt 2 Vector Area
F4	Timer/Counter 1 Interrupt Vector Area
F6	-
F8	External Interrupt 1 Vector Area
FA	External Interrupt 0 Vector Area
FC	Basic Interval Timer Interrupt Vector Area
FE	RESET Vector Area

NOTE:
"-" means reserved area.

Figure 8-6 Interrupt Vector Area

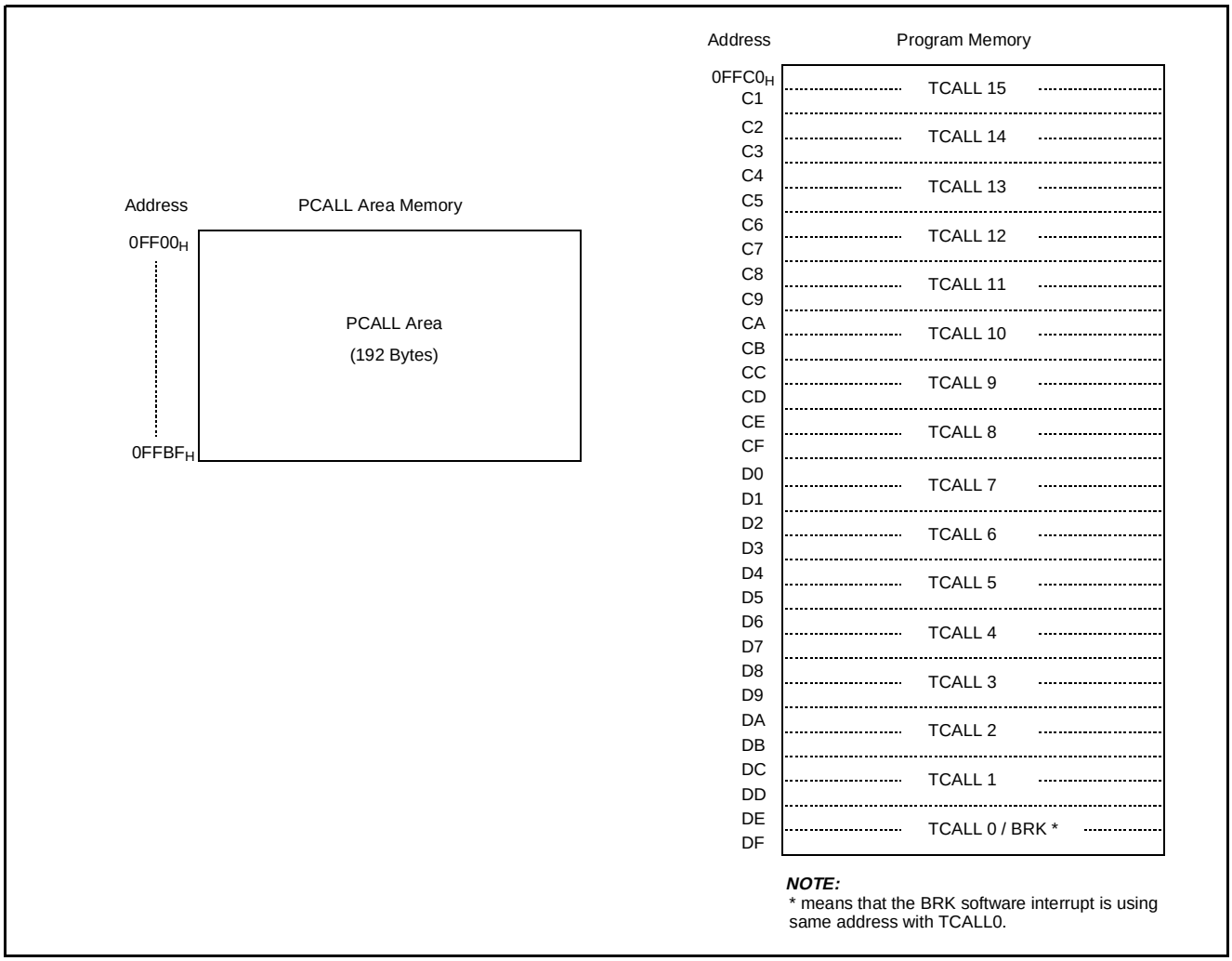
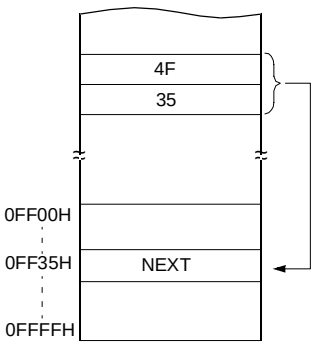


Figure 8-7 PCALL and TCALL Memory Area

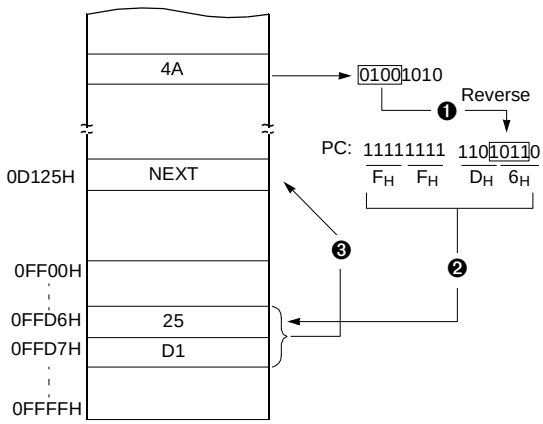
PCALL→ rel

4F35 PCALL 35H



TCALL→ n

4A TCALL 4



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

```
          ORG     0FFE0H

          DW      NOT_USED
          DW      NOT_USED
          DW      NOT_USED
          DW      KEY_INT                 ; Key Scan
          DW      WT_INT                  ; Watch Timer
          DW      NOT_USED
          DW      NOT_USED
          DW      NOT_USED
          DW      WDT_INT                 ; Watch Dog Timer
          DW      INT2                    ; Int.2
          DW      TMR1_INT                ; Timer-1
          DW      NOT_USED
          DW      INT1                    ; Int.1
          DW      INT0                    ; Int.0
          DW      BIT_INT                 ; BIT
          DW      RESET                   ; Reset


          ORG     0F000H

;*****************************************
;             MAIN     PROGRAM             *
;*****************************************
;
;
RESET:    DI                              ;Disable All Interrupts
          CLRG
          LDX     #0
RAM_CLR:  LDA     #0                      ;RAM Clear(!0000H->!00BFH)
          STA     {X}+
          CMPX    #0C0H
          BNE     RAM_CLR
;
          LDX     #0FFH                   ;Stack Pointer Initialize
          TXSP
;
          CALL    LCD_CLR                 ;Clear LCD display memory
;
          LDM     R0, #0                  ;Normal Port 0
          LDM     R0DD,#1000_0010B        ;Normal Port Direction
          LDM     PUR0,#1000_0010B        ;Pull Up Selection Set
          LDM     PMR0,#0000_0001B        ;R0 port / int
          :
          :
          LDM     PCOR,#1                 ;Enable Peripheral clock
          :
          :
```

8.3 Data Memory

Figure 8-8 shows the internal Data Memory space available. Data Memory is divided into four groups, a user RAM, control registers, Stack, and LCD memory.

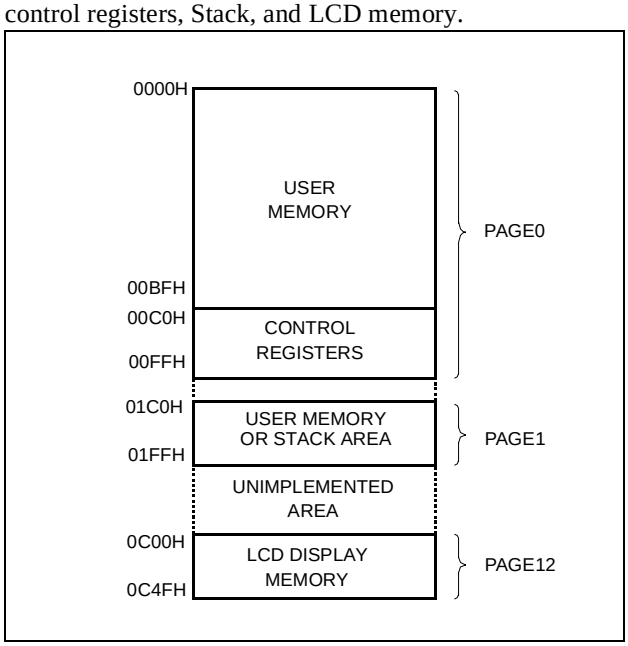


Figure 8-8 Data Memory Map

User Memory

The GMS81C3004 has 256 × 8 bits for the user memory (RAM).

Control Registers

The control registers are used by the CPU and Peripheral function blocks for controlling the desired operation of the device. Therefore these registers contain control and status bits for the interrupt system, the timer/ counters, analog to digital converters and I/O ports. The control registers are in address range of $0C0_H$ to $0FF_H$.

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

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

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

Example; To write at CKCTLR

```
LDM    CKCTLR,#09H ;Divide ratio ÷8
```

Address	Symbol	R/W	RESET Value	Addressing mode
0C0H	R0	R/W	Undefined	byte, bit[1]
0C1H	R1	R/W	Undefined	byte, bit
0C2H	R2	R/W	Undefined	byte, bit
0C4H	R4	R/W	Undefined	byte, bit
0C5H	R5	R/W	Undefined	byte, bit
0C6H	R6	R/W	Undefined	byte, bit
0C7H	R7	R/W	Undefined	byte, bit
0C8H	R0DD	W	00000000	byte[2]
0C9H	R1DD	W	00000000	byte
0CAH	R2DD	W	-----000	byte
0CCH	R4DD	W	00000000	byte
0CDH	R5DD	W	00000000	byte
0CEH	R6DD	W	00000000	byte
0CFH	R7DD	W	00000000	byte
0D4H	PUR0	W	00000000	byte
0D5H	PUR1	W	00000000	byte
0D6H	PUR2	W	-----000	byte
0D8H	IESR	W	--000000	byte
0D9H	PMR0	W	----0000	byte
0DAH	IENL	R/W	--00---0	byte, bit
0DBH	IENH	R/W	--00-000	byte, bit
0DCH	IRQL	R/W	--00---0	byte, bit
0DDH	IRQH	R/W	--00-000	byte, bit
0DEH	SLMR	W	-------0	byte
0DFH	WDTR	W	00111111	byte
0E4H	TM1	R/W	---00000	byte, bit
0E5H	T1	R	00000000	byte, bit
0E5H	TDR1	W	Undefined	byte
0ECH	CMR	W	00-00000	byte
0EDH	CSR	W	0-----00	byte
0F0H	WTMR	W	----0000	byte
0F1H	LCR	R/W	0-00-000	byte, bit
0F3H	LPMR	R/W	00000000	byte, bit
0F4H	KSCR	R/W	00000000	byte, bit
0F5H	KDTR	R	00000000	byte, bit
0F6H	PMR1	R/W	-------0	byte, bit
0F7H	BUR	W	11111111	byte
0F8H	RPR	R/W	----0000	byte, bit
0F9H	BITR	R	Undefined	byte, bit
0F9H	CKCTLR	W	----0111	byte
0FAH	SCMR	R/W	----0000	byte, bit
0FBH	PCOR	W	-------0	byte
0FEH	LVDR	R/W	---10000	byte, bit

Table 8-1 Control Registers

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

Stack Area

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

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

The save/restore locations in the stack are determined by the stack pointed (SP). The SP is automatically decreased after the saving, and increased before the restoring. This means the value of the SP indicates the stack location number for the next save. Refer to Figure 8-4 on page 18.

LCD Display Memory

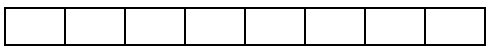
LCD display data area is handled in LCD section.

See "15.4 LCD Display Memory" on page 59.

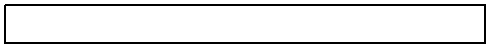
Address	Symbol	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Power-on Reset value
0C0H	R0									xxxx xxxx
0C1H	R1									xxxx xxxx
0C2H	R2	-	-	-	-	-				---- -xxx
0C4H	R4									xxxx xxxx
0C5H	R5									xxxx xxxx
0C6H	R6									xxxx xxxx
0C7H	R7									xxxx xxxx
0C8H	R0DD									0000 0000
0C9H	R1DD									0000 0000
0CAH	R2DD	-	-	-	-	-				---- -000
0CCH	R4DD									0000 0000
0CDH	R5DD									0000 0000
0CEH	R6DD									0000 0000
0CFH	R7DD									0000 0000
0D4H	PUR0									0000 0000
0D5H	PUR1									0000 0000
0D6H	PUR2	-	-	-	-	-				---- -000
0D8H	IESR	-	-							--00 0000
0D9H	PMR0	-	-	-	-					---- 0000
0DAH	IENL	-	-	KSEN	WTEN	-	-	-	WDTEN	--00 ---0
0DBH	IENH	-	-	INT2EN	T1EN	-	INT1EN	INT0EN	BITEN	--00 -000

Address	Symbol	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Power-on Reset value
0DCH	IRQL	-	-	KSIF	WTIF	-	-	-	WDTIF	--00 ---0
0DDH	IRQH	-	-	INT2IF	T1IF	-	INT1IF	INT0IF	BITIF	--00 -000
0DEH	SLMR	-	-	-	-	-	-	-		---- ---0
0DFH	WDTR									0011 1111
0E4H	TM1	-	-	-						---0 0000
0E5H [1]	T1									0000 0000
0E5H [1]	TDR1									Undefined
0ECH	CMR			-						00-0 0000
0EDH	CSR		-	-	-	-	-			0--- --00
0F0H	WTMR	-	-	-	-					---- 0000
0F1H	LCR		-			-				0-00 -000
0F3H	LPMR									0000 0000
0F4H	KSCR									0000 0000
0F5H	KDTR									0000 0000
0F6H	PMR1	-	-	-	-	-	-	-	R13/BUZ	---- ---0
0F7H	BUR									1111 1111
0F8H	RPR	-	-	-	-					---- 0000
0F9H [2]	BITR									Undefined
0F9H [2]	CKCTLR	-	-	-	-					---- 0111
0FAH	SCMR	-	-	-	-					---- 0000
0FBH	PCOR	-	-	-	-	-	-	-		---- ---0
0FEH	LVDR	-	-	-						---1 0000

1. The register T1 and TMR1 are located at same address. Address 0E5H is read as T1, and written to TMR1.
2. The register BITR and CKCTLR are located at same address. Address 0F9H is read as BITR, and written to CKCTLR.



SFR bit and byte addressable



SFR not bit addressable

- : this bit location is reserved

8.4 Addressing Mode

The GMS81C3004 uses six addressing modes;

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

(1) Register Addressing

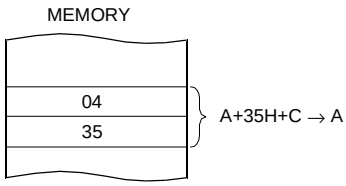
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

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

Example:

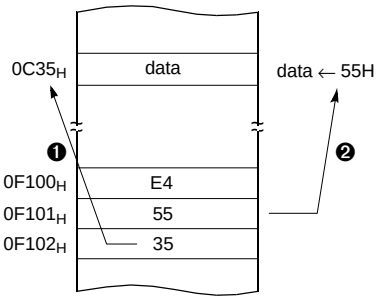
```
0435    ADC    #35H
```



When G-flag is 1, then RAM address is difined by 16-bit address which is composed of 8-bit RAM paging register (RPR) and 8-bit immediate data.

Example: G=1, RPR=0CH

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

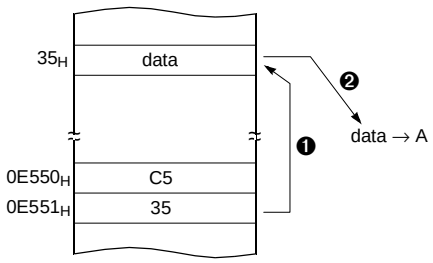


(3) Direct Page Addressing → dp

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

Example; G=0

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



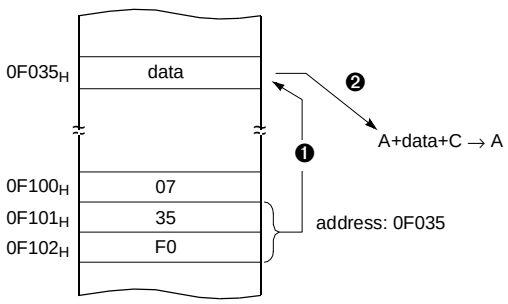
(4) Absolute Addressing → !abs

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

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

Example;

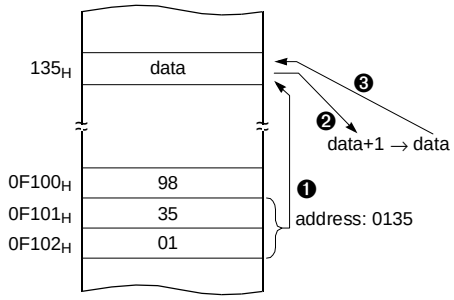
```
0735F0  ADC    !0F035H      ;A ←ROM[0F035H]
```



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

Example; Addressing accesses the address 0135_H regardless of G-flag and RPR.

```
983501   INC    !0135H        ;A ←ROM[135H]
```



(5) Indexed Addressing

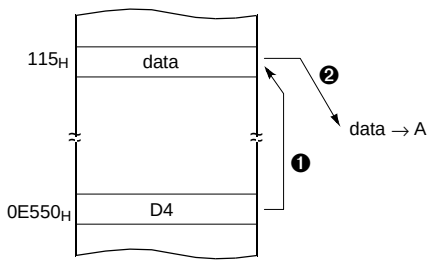
X indexed direct page (no offset) → {X}

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

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

Example; $X=15_H$, G=1, $RPR=01_H$

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



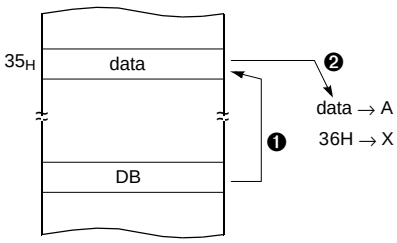
X indexed direct page, auto increment→ {X}+

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

LDA, STA

Example; G=0, $X=35_H$

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



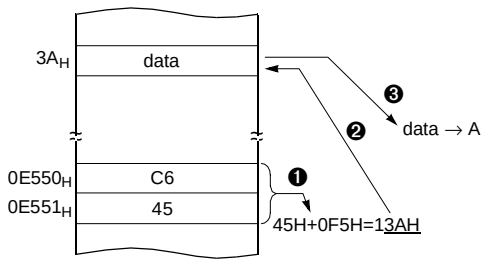
X indexed direct page (8 bit offset) → dp+X

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

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

Example; G=0, $X=0F5_H$

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



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

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

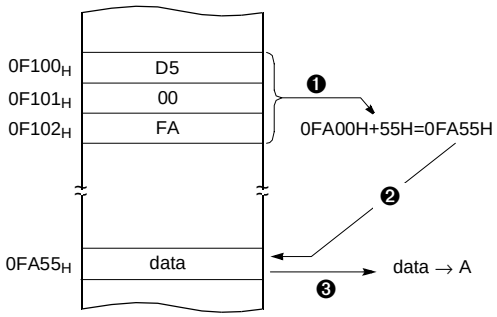
This is same with above (2). Use Y register instead of X.

Y indexed absolute → !abs+Y

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

Example; Y=55_H

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



(6) Indirect Addressing

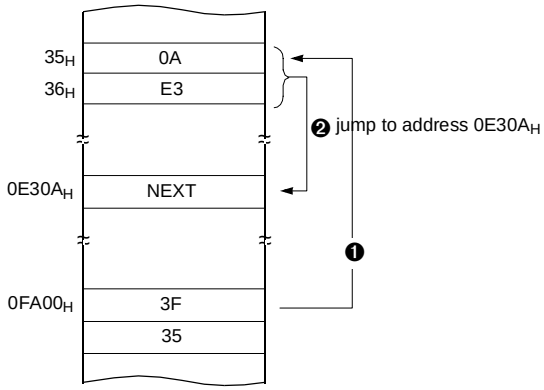
Direct page indirect → [dp]

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

JMP, CALL

Example; G=0

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



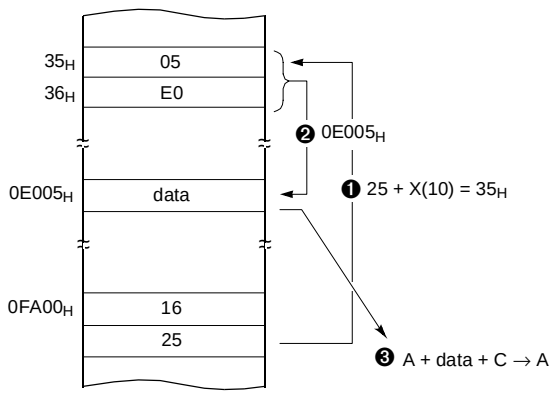
X indexed indirect → [dp+X]

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

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

Example; G=0, X=10_H

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



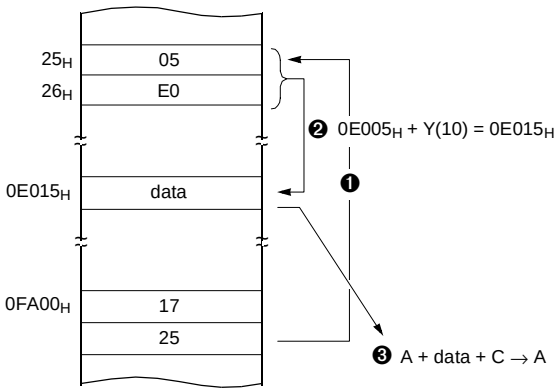
Y indexed indirect → [dp]+Y

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

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

Example; G=0, Y=10_H

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



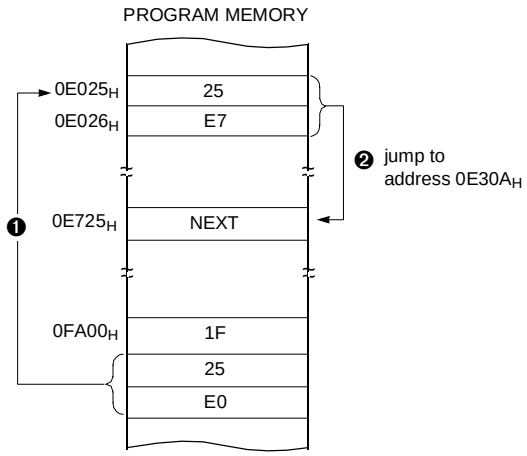
Absolute indirect → [!abs]

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

JMP

Example; G=0

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



9. I/O PORTS

The GMS81C3004 has seven ports (R0, R1, R2, R4, R5, R6, and R7), and LCD segment port (SEG0~SEG39), and LCD common port (COM0~COM7).

These ports pins may be multiplexed with an alternate function for the peripheral features on the device. In general, in a initial reset state, R0,R1,R2 ports are used as a general purpose input port and R4, R5, R6 and R7 ports are used as LCD segment drive output port.

9.1 Registers for Port

Port Data Registers

The Port Data Registers in I/O buffer in each seven ports (R0,R1,R2,R4,R5,R6,R7) are represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to data register" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read data register" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to "read data register" signal from the CPU. Some instructions that read a port activating the "read register" signal, and others activating the "read pin" signal

Port Direction Registers

All pins have data direction registers which can define these ports as output or input. A "1" in the port direction register configure the corresponding port pin as output. Conversely, write "0" to the corresponding bit to specify it as input pin. For example, to use the even numbered bit of R0 as output ports and the odd numbered bits as input ports, write "55_H" to address $0C8_H$ (R0 port direction register) during initial setting as shown in Figure 9-1 .

All the port direction registers in the GMS81C3004 have 0 written to them by reset function. On the other hand, its initial status is input.

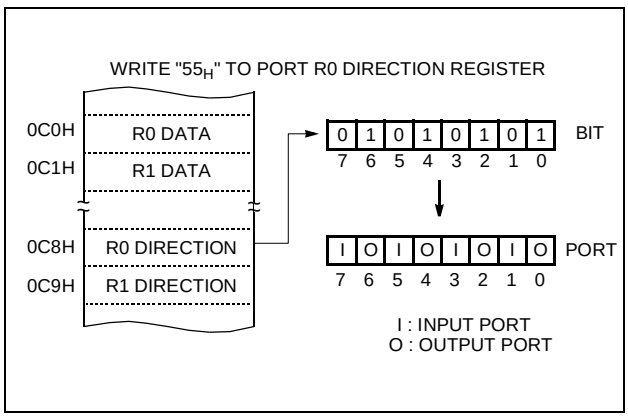


Figure 9-1 Example of port I/O assignment

Pull-up Control Registers

The R0, R1, and R2 ports have internal pull-up resistors. Figure 9-2 shows a functional diagram of a typical pull-up port. It is connected or disconnected by Pull-up Control register (PUR*n*). The value of that resistor is typically 100kΩ. Refer to DC characteristics for more details.

When a port is used as key input, input logic is firmly either low or high, therefore external pull-down or pull-up resisters are required practically. The GMS81C3004 has internal pull-up, it can be logic high by pull-up that can be able to configure either connect or disconnect individually by pull-up control registers PUR*n*.

When ports are configured as inputs and pull-up resistor is selected by software, they are pulled to high. If port is configured as an output, pull-up is disabled automatically regardless of setting of PUR*n*.

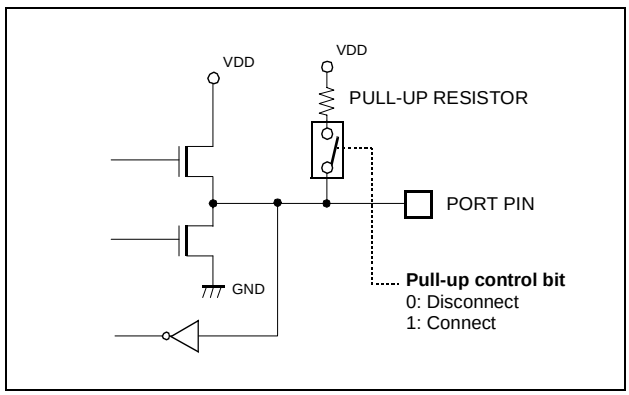


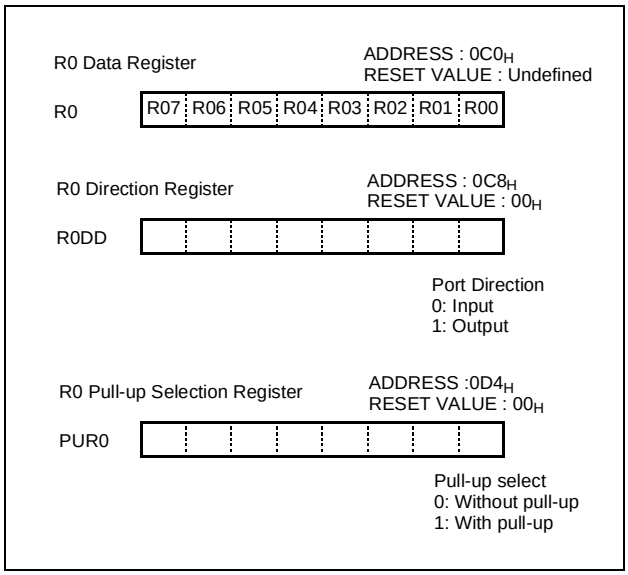
Figure 9-2 Pull-up Port Structure

9.2 I/O Ports Configuration

R0 Ports

R0 is an 8-bit CMOS bidirectional I/O port (address $0C0_H$). Each I/O pin can independently used as an input or an output through the R0DD register (address $0C8_H$).

R0 has internal pull-ups that is independently connected or disconnected by PUR0. The control registers for R0 are shown below.



In addition, Port R0 is multiplexed with various special features. The control register PMR0 (address 0D9H) controls the selection of alternate function. After reset, this value is "0", port may be used as normal I/O port.
To use alternate function such as External Interrupt rather than normal I/O, write "1" in the corresponding bit of PMR0.

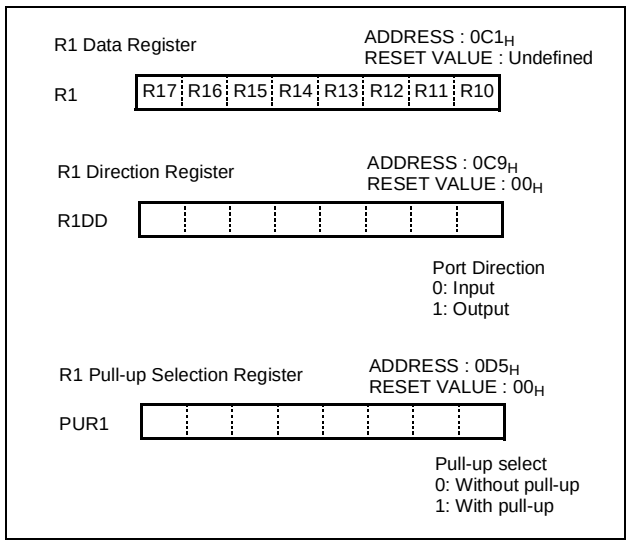
Port Pin	Alternate Function
R00	INT0 (External Interrupt 0)
R01	INT1 (External Interrupt 1)
R02	$\overline{INT2}$ (External Interrupt 2)
R03	$\overline{EC1}$ (External count input to Timer/Counter 1)
R06	LCDCK (LCD clock output)

R1 Ports

R1 is an 8-bit CMOS bidirectional I/O port (address $0C1_H$). Each I/O pin can independently used as an input or an output through the R1DD register (address $0C9_H$).

R1 has internal pull-ups that is independently connected or disconnected by register PUR1. If the key scan function is used, these pin can input the key switch signal without external pull-up registers. For more details refer to "14.. KEY SCAN" on page 53.

The control registers for R1 are shown below.



Port R1 is multiplexed with various special features.The control registers controls the selection of alternate function. After reset, this value is "0", port may be used as normal I/O port. The way to select alternate function such as comparator input or buzzer will be shown in each peripheral section.

In addition, R1 port is used as key scan function which operate with normal input port.

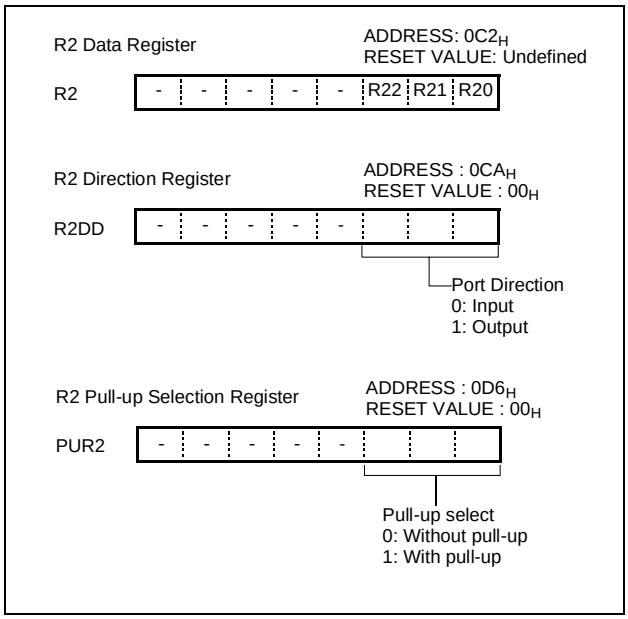
Port Pin	Alternate Function
R10	KS0
R11	KS1
R12	KS2
R13	KS3/BUZ (Buzzer frequency output)
R14	KS4/CMP0 (Comparator input 0)
R15	KS5/CMP1 (Comparator input 1)
R16	KS6/CMP2 (Comparator input 2)
R17	KS7/CMP3 (Comparator input 3)

Input or output is configured automatically by each function register (CSR, PMR1, KSCR) regardless of R1DD.

R2 Port

R2 is an 3-bit CMOS bidirectional I/O port (address $0C2_H$). Each I/O pin can independently used as an input or an output through the R2DD register (address $0CA_H$).

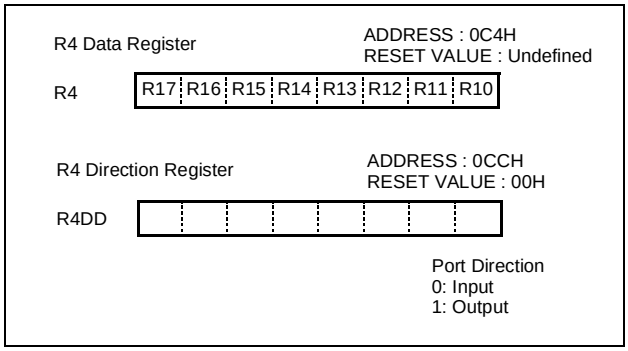
R2 has internal pull-ups that is independently connected or disconnected by PUR2 (address $0D6_H$). The control registers for R2 are shown as below.



R4 Port / SEG0 ~ SEG7

R4 is an 8-bit CMOS bidirectional I/O port (address $0C4_H$). Each I/O pin can independently used as an input or an output through the R4DD register (address $0CC_H$).

R4 has difference that it doesn't have internal pull-ups and is shared with LCD segment ports.



On the initial reset, R4 is configured as LCD segment output ports regardless of Direction Register R4DD. The LCD Port Mode Register (LPMR) should be properly set to be used as normal I/O.

Example: To use as I/O ports

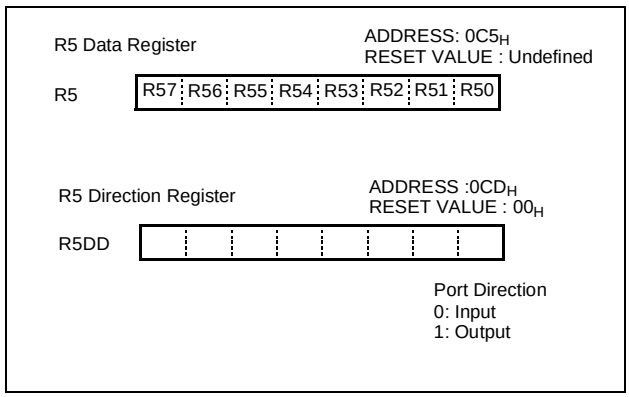
```
        :
        :
LDM     LPMR,#xxxx_xx11B
        :
        :
        :
```

x: Don't care

R5 Port / SEG8 ~ SEG15

R5 is an 8-bit CMOS bidirectional I/O port (address $0C5_H$). Each I/O pin can independently used as an input or an output through the R5DD register (address $0CD_H$).

R5 is shared with LCD segment ports.



On the initial reset, R5 is configured as LCD segment output port regardless of Direction Register R5DD. The LCD Port Mode Register (LPMR) should be set properly to be used as normal I/O. Refer to example below.

Example: To use as an I/O port

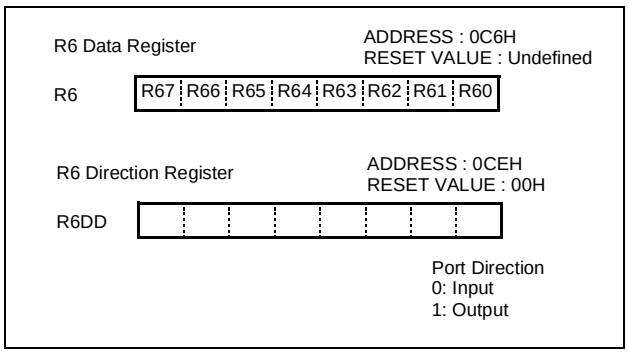
```
        :
        :
LDM     LPMR,#xxxx_11xxB
        :
        :
        :
```

x: Don't care

R6 Port / SEG16 ~ SEG23

R6 is an 8-bit CMOS bidirectional I/O port (address $0C6_H$). Each I/O pin can independently used as an input or an output through the R6DD register (address $0CE_H$).

R6 is shared with LCD segment ports.



After reset, R6 is initialized as LCD segment output ports regardless of Direction Register R6DD. The LCD Port Mode Register (LPMR) should be set properly to use as normal I/O. Refer to example below.

Example: To use as an I/O port

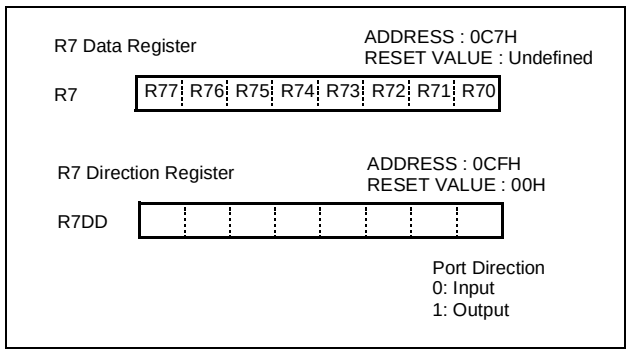
```
LDM    LPMR,#xx11_xxxxB
```

x: Don't care

R7 Port / SEG24 ~ SEG31

R7 is an 8-bit CMOS bidirectional I/O port (address $0C7_H$). Each I/O pin can independently used as an input or an output through the R7DD register (address $0CF_H$).

R7 is shared with LCD segment ports.



After reset, R7 is initialized as LCD segment output ports regardless of Direction Register R7DD. The LCD Port Mode Register (LPMR) should be set properly to use as normal I/O. Refer to example below.

Example: To use as an I/O port

```
LDM    LPMR,#11xx_xxxxB
```

x: Don't care

SEG0~SEG39

Segment signal output pins for the LCD display.

COM0~COM7

Common signal output pins for the LCD display.

10. CLOCK GENERATOR

As shown in Figure 10-1 , the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains two oscillators: a main-frequency clock oscillator and a sub-frequency clock oscillator. Power consumption can be reduced by switching them to the low power operation frequency clock can be easily obtained by attaching a resonator between the X_{IN} and X_{OUT} pin and the SX_{IN} and SX_{OUT} pin, respectively. The system clock can also be obtained from the external oscillator.

The clock generator produces the system clocks forming clock pulse, which are supplied to the CPU and the peripheral hardware. The internal system clock can be selected by bit2, and bit3 of the system clock mode register, SCMR.

CPU clock	Instruction cycle time	
	f_{MAIN} = 4.19MHz	f_{SUB} = 32.768kHz
÷ 2	0.48 us	61 us
÷ 8	1.90 us	244 us
÷ 16	3.80 us	488 us
÷ 64	15.30 us	1953 us

The registers are shown in Figure 10-2 .

To the peripheral block, the clock among the not-divided original clocks, divided by 2, 4,..., up to 1024 can be provided. Peripheral clock is enabled or disabled by bit 0 of the peripheral clock enable register (ENPCK).

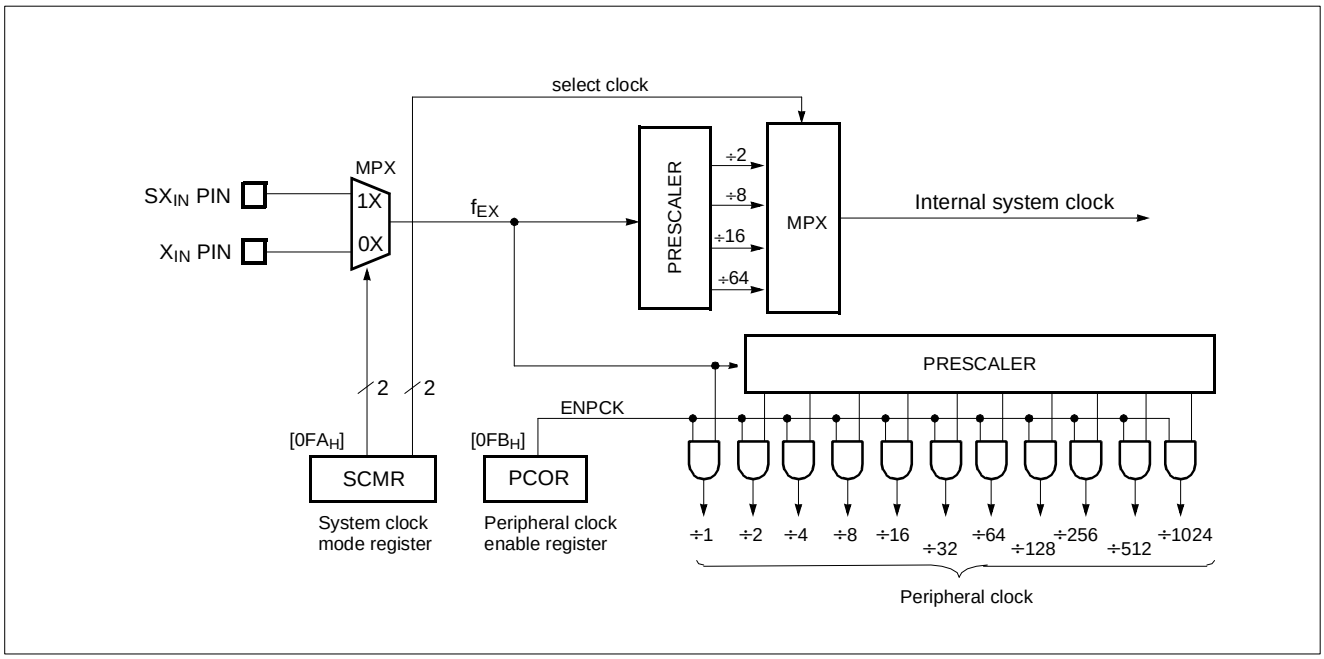


Figure 10-1 Block Diagram of Clock Generator

Note: *On the initial reset, all peripherals are stopped because peripheral clock is not supplied to each function block. Therefore, Peripheral Clock Enable Register, PCOR must be written to "1" in software initial part. Then, timer and other functions may be operated by provided clock.*

Example; PCOR setting and Basic Interval Timer

```
PCOR    EQU   0FBH
CKCTLR  EQU   0F9H
IENL    EQU   0DAH
IENH    EQU   0DBH
BITEN   EQU   0,IENH

        LDM   PCOR,#1
        LDM   CKCTLR,#0CH
        SET1  BITEN
        EI
```

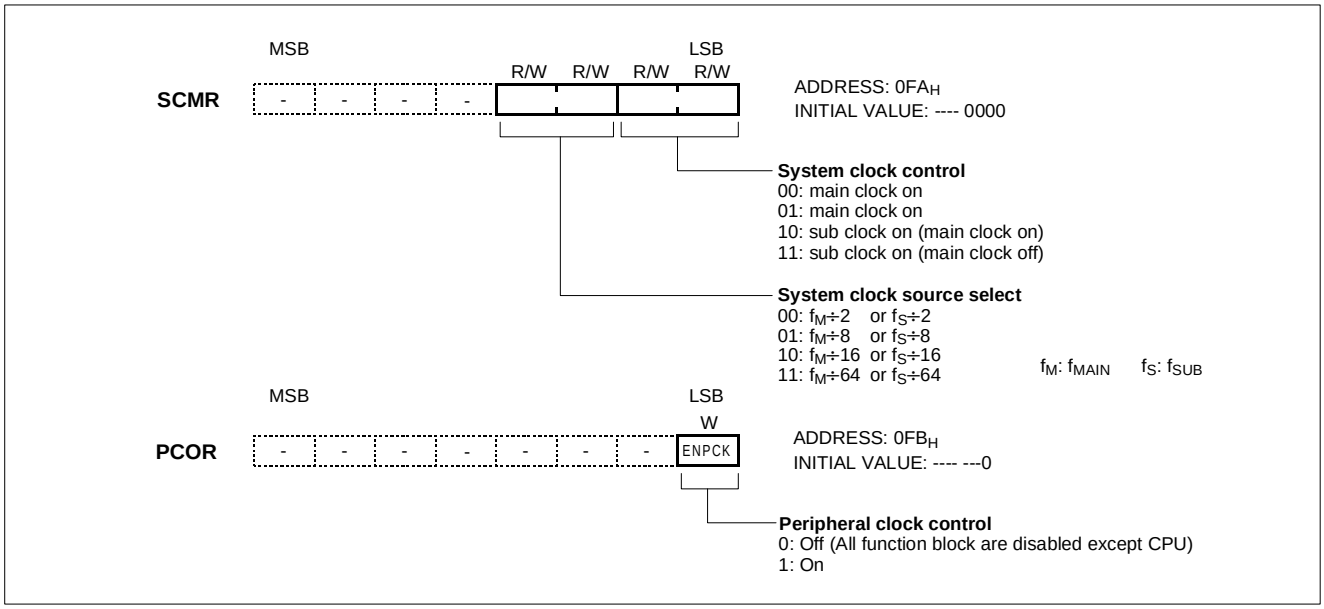


Figure 10-2 SCMR, PCOR: System Clock Control Registers

10.1 Operation Mode

The system clock controller starts or stops the main-frequency clock oscillator and switches between the sub frequency clock. The operating mode is generally divided into the main-clock mode and the sub-clock mode, which are controlled by System clock mode register (SCMR). Figure 10-3 shows the operating mode transition diagram.

System clock control is performed by the system clock mode register, SCMR. During reset, this register is initialized to "0" so that the main-clock operating mode is selected.

Main-clock operating mode

This mode is fast-frequency operating mode.
The CPU and the peripheral hardwares are operated on the high-frequency clock. At reset release, this mode is invoked.

Sub-clock operating mode

This mode is low-frequency operating mode
In this mode, the high-frequency clock oscillation is stops to operate the CPU and the peripheral hardware on the low-frequency clock, thereby reducing power consumption

SLEEP mode

In this mode, the CPU clock stops while peripherals and the oscillation source continue to operate normally.

STOP mode

In this mode, the system operations are all stopped, holding the internal states valid immediately before the stop at the low power consumption level.

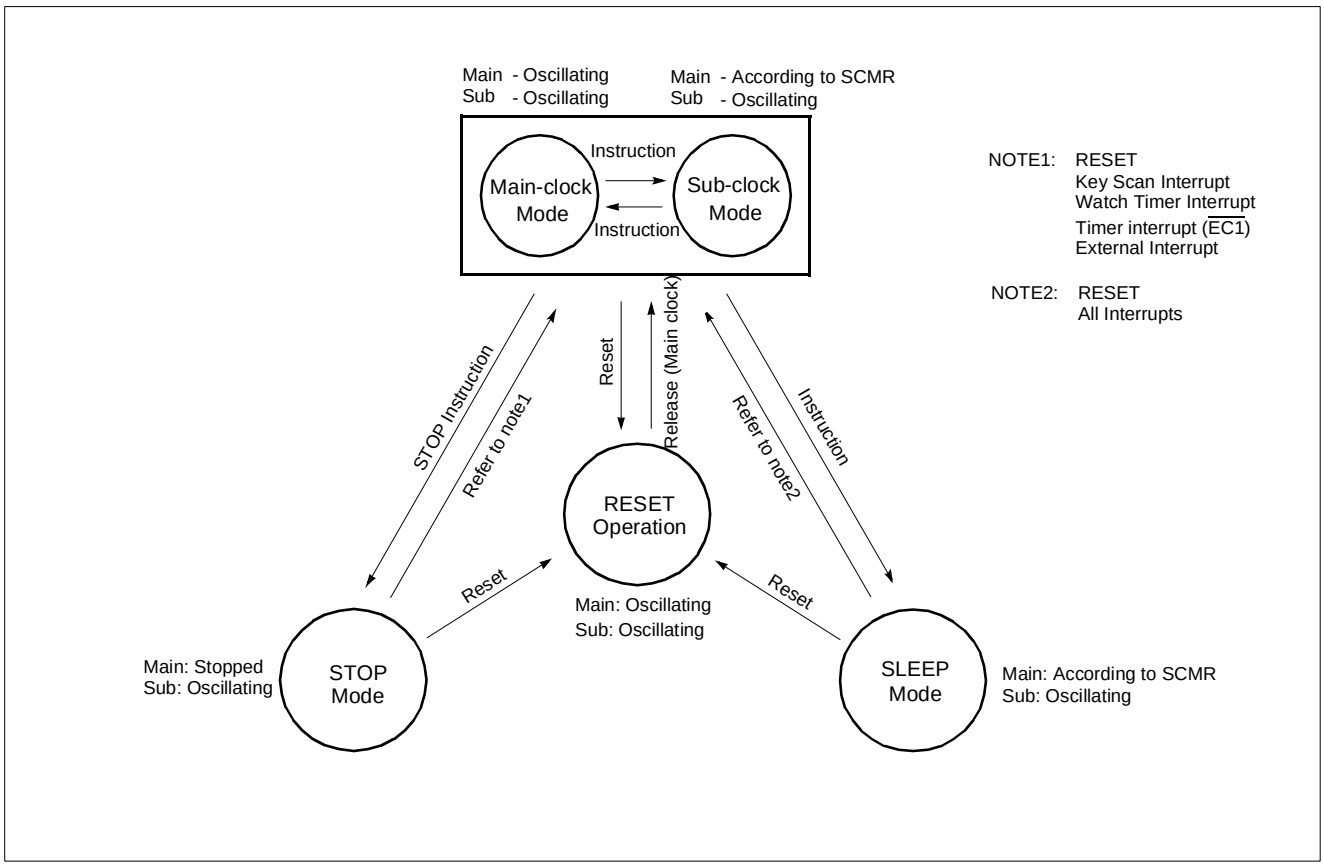


Figure 10-3 Operating Mode

10.2 Operation Mode Switching

In the Main-clock operation mode, only the high-frequency clock oscillator is used.

In the Sub-clock operation mode, the high-frequency clock oscillation stops, enabling the low power voltage operation or the low power consumption operation. Instruction execution does not stop when the operation speed switching is performed. However, some peripheral hardware capabilities may be affected. For details, refer to the description of the relevant operation.

The following describes the switching between the Main-clock and the Sub-clock operations. During reset, the system clock mode register is initialized at the Main-clock mode. It must be set to the Sub-clock operation for the low-power consumption mode.

Switching from main clock operation to sub-clock operation

First, write "10B" into lower 2 bits of SCMR to switch the main system clock to the sub-frequency clock.
Next, write "11B" to turn off main frequency oscillation.

Example:

```
        :
        :
        :
MOV     SCMR,#2       ;Switch to sub mode
MOV     SCMR,#3       ;Turn off main clock
        :
        :
```

Returning from Sub clock operation to main clock operation

First, write "10B" into lower 2 bits of the SCMR to turn on the main-frequency oscillation, when the stabilization (warm-up) has been taken by the software delay routine. Sub clock operation mode can also be released by setting the $\overline{\text{RESET}}$ pin to low, which immediately performs the reset operation. After reset, the GMS81C3004 is placed in main frequency operation mode.

Example:

```
        :
        :
        :
MOV     SCMR,#2       ;Turn on main-clock
CALL    DLY           ;Wait until stable
MOV     SCMR,#0       ;Move to main mode
        :
        :
        :


;20ms software delay
DLY:    LDY    #0
DLP0:   LDA    #0
DLP1:   NOP
        INC    A
        BCC    DLP1
        INC    Y
        CMPY   #20
        BCC    DLP0
        RET
```

Shifting from the Normal operation to the SLEEP mode

By setting bit 0 of SMR, the CPU clock stops and the SLEEP mode is invoked. The CPU stops while other peripherals are operate normally.

The way of release from this mode is RESET and all available interrupts.

For more detail, See "18.1 SLEEP Mode" on page 68

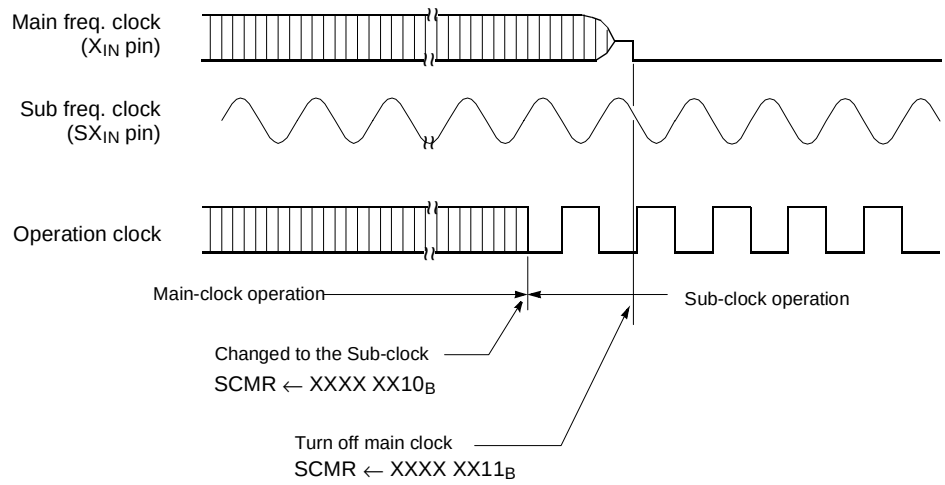
Shifting from the Normal operation to the STOP mode

By executing STOP instruction, the main-frequency clock oscillation stops and the STOP mode is invoked. But sub-frequency clock oscillation is operated continuously.
After the STOP operation is released by reset, the operation mode is changed to Main-clock mode.

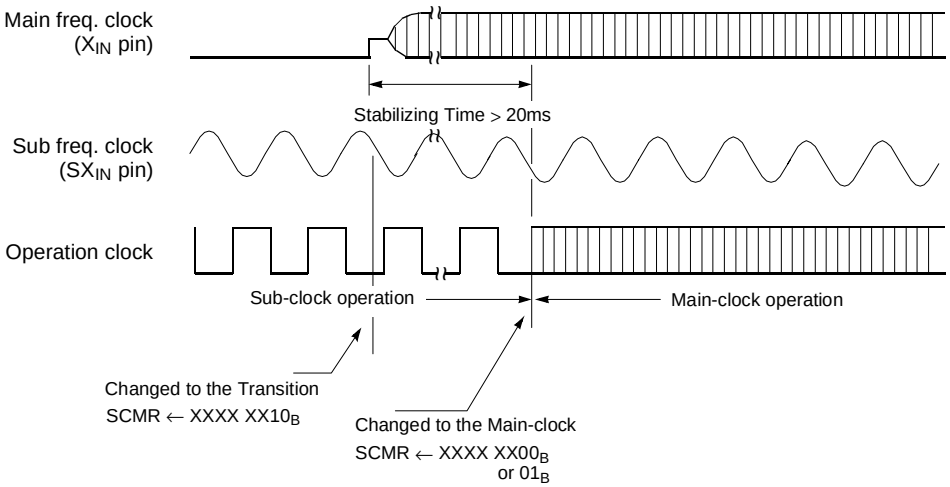
The methods of release are RESET, Key scan interrupt, Watch Timer interrupt, Timer/Event counter1 (EC1 pin), and External Interrupt.

For more details, see "18.2. STOP Mode" on page 69.

Note: *In the STOP and SLOW operating modes, the power consumed by the oscillator and the internal hardware is reduced. However, the power for the pin interface (depending on external circuitry and program) is not directly associated with the low-power consumption operation. This must be considered in system design as well as interface circuit design.*



(a) Main clock mode → Sub clock mode



(b) Sub clock → Main clock

Figure 10-4 System Clock Switching Timing

11. TIMER

11.1 Basic Interval Timer

The GMS81C3004 has one 8-bit Basic Interval Timer that is free-run and can not stop. Block diagram is shown in Figure 11-1 .

The Basic Interval Timer generates the time base for key scanning, watchdog timer counting, and etc. It also provides a Basic interval timer interrupt (BITIF). As the count overflow from FF_H to 00_H, this overflow causes the interrupt to be generated. The Basic Interval Timer is controlled by the clock control register (CKCTLR) shown in Figure 11-2 .

Source clock can be selected by lower 3 bits of CKCTLR.

BITR and CKCTLR are located at same address, and address $0F9_H$ is read as a BITR, and written to CKCTLR..

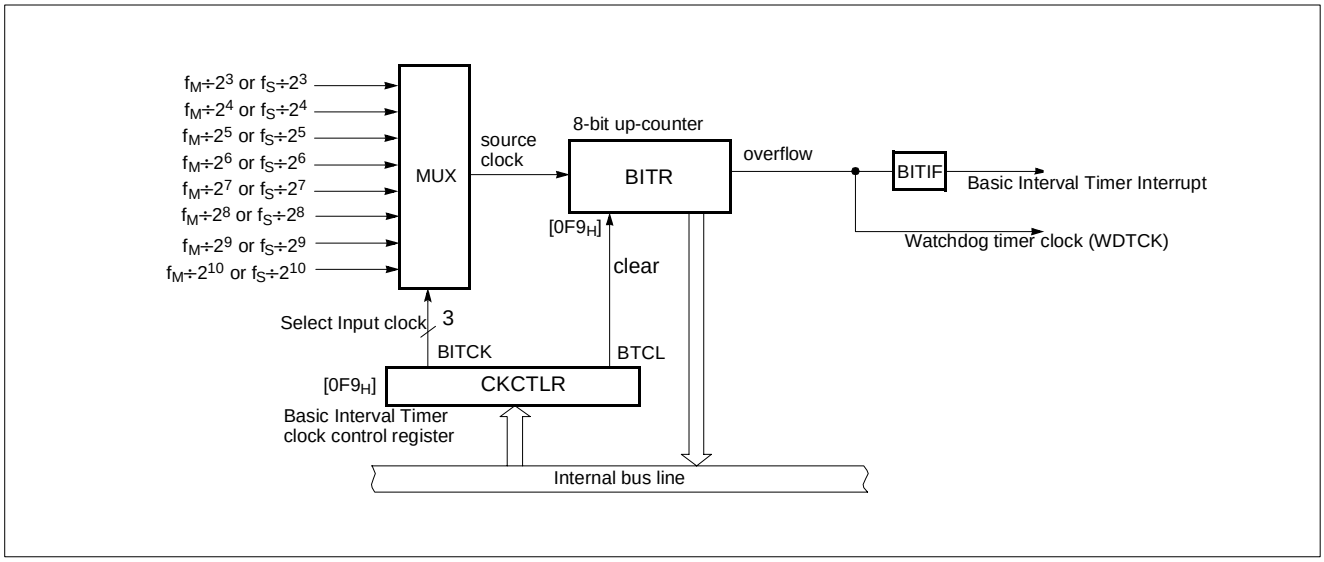


Figure 11-1 Block Diagram of Basic Interval Timer

CKCTLR [2:0]	Source clock		Interrupt (overflow) Period	
	SCMR[1:0]= 00 or 01	SCMR[1:0]= 10 or 11	At f_{MAIN}=4.19MHz	At f_{SUB}=32.768kHz
000	$f_M÷2^3$	$f_S÷2^3$	0.488 ms	62.5 ms
001	$f_M÷2^4$	$f_S÷2^4$	0.976	125.0
010	$f_M÷2^5$	$f_S÷2^5$	1.953	250.0
011	$f_M÷2^6$	$f_S÷2^6$	3.906	500.0
100	$f_M÷2^7$	$f_S÷2^7$	7.812	1000.0
101	$f_M÷2^8$	$f_S÷2^8$	15.625	2000.0
110	$f_M÷2^9$	$f_S÷2^9$	31.250	4000.0
111	$f_M÷2^{10}$	$f_S÷2^{10}$	62.500	5000.0

Table 11-1 Basic Interval Timer Interrupt Time

f_{MAIN}: main clock frequency (ex: 4.19MHz)
f_{SUB}: sub clock frequency (ex: 32.768kHz)

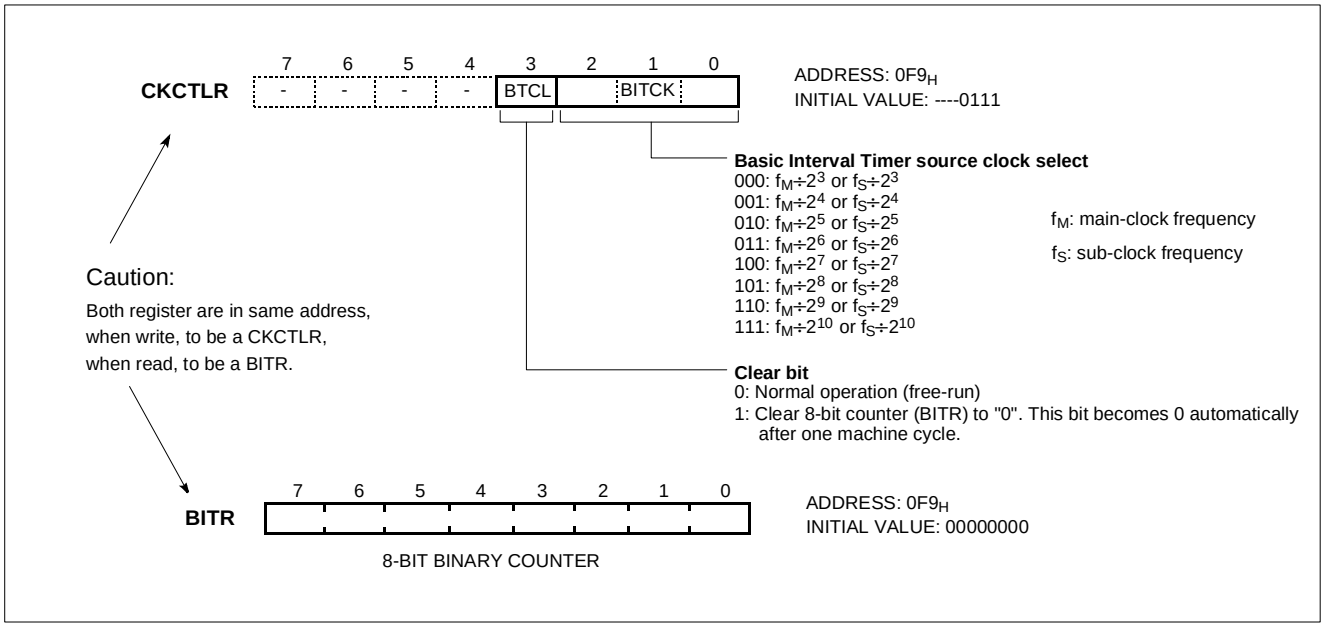


Figure 11-2 BITR: Basic Interval Timer Mode Register

11.2 Timer/Event Counter 1

Timer/Event Counter 1 consists of prescaler, multiplexer, 8-bit compare data register, 8-bit count register, Control register, and Comparator as shown in Figure 11-3 .

The timer/counter 1 has two operating modes. One is the timer mode which is operated by internal clock, other is event counter mode which is operated by external clock from pin $\overline{EC1}$.

The contents of TDR1 are compared with the contents of up-counter T1. If a match is found, a timer/counter 1 interrupt (T1IF) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

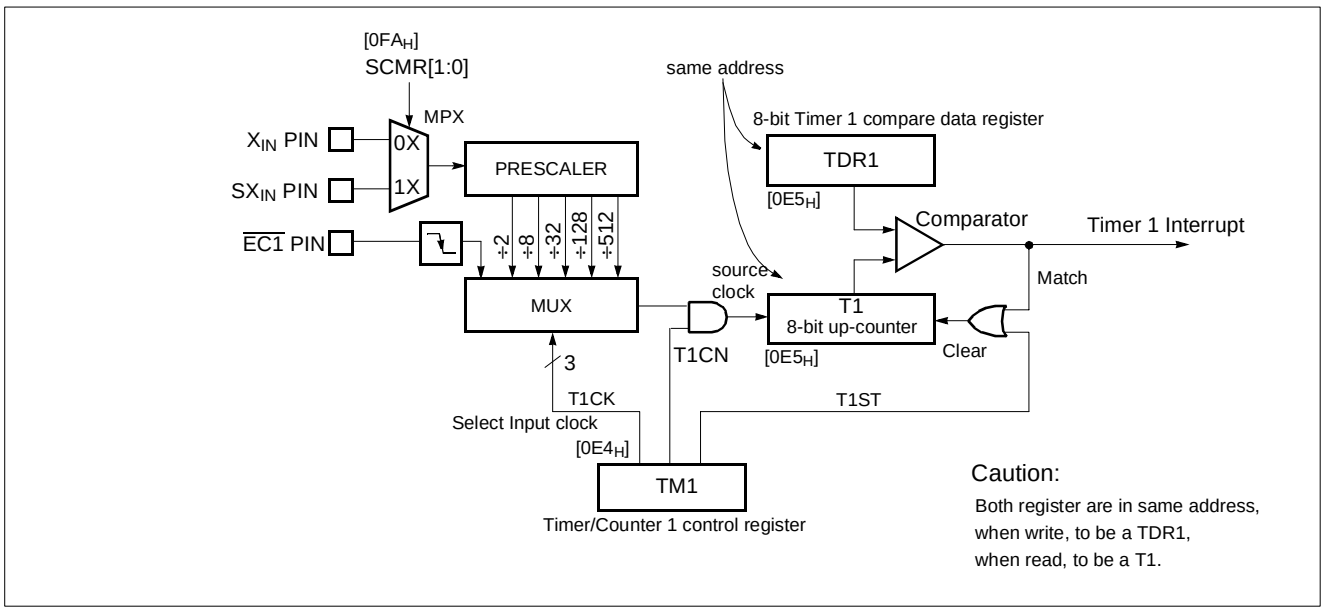
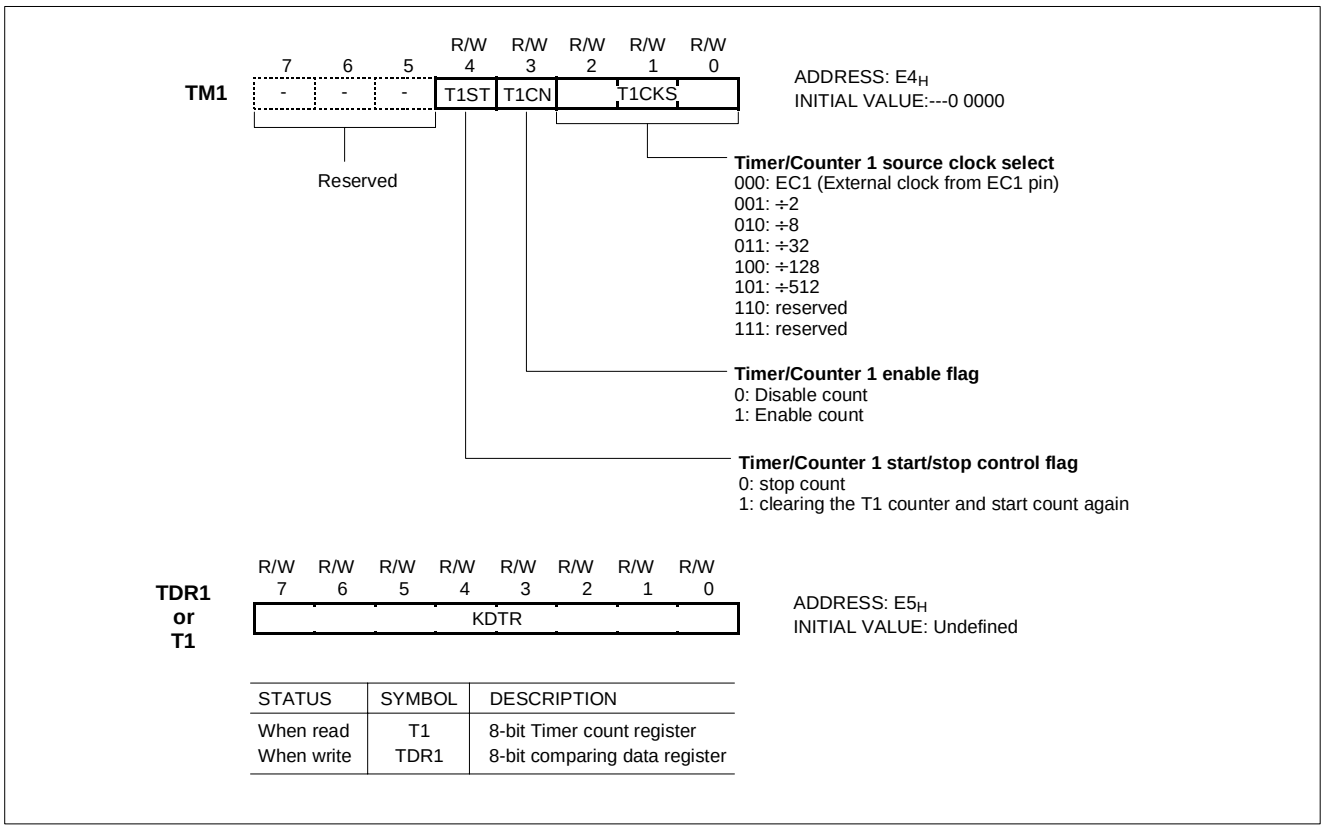


Figure 11-3 Block Diagram of Timer/Event Counter

Note: *The content of TDR1 must be initialized (by software) with the value between 1_H and $0FF_H$,not to 0.*



STATUS	SYMBOL	DESCRIPTION
When read	T1	8-bit Timer count register
When write	TDR1	8-bit comparing data register

Figure 11-4 Timer Mode Register and TDR1, T1 Registers

Timer Mode

In the timer mode, the internal clock is used for counting up. Thus, you can think of it as counting internal clock input. The contents of TDR1 are compared with the contents of up-counter, T1. If match is found, a timer 1 interrupt (T1IF) is generated and the up-counter is cleared to 0. Counting up is resumed after the up-counter is cleared.

As the value of TDR1 is changeable by software, time interval is set as you want.

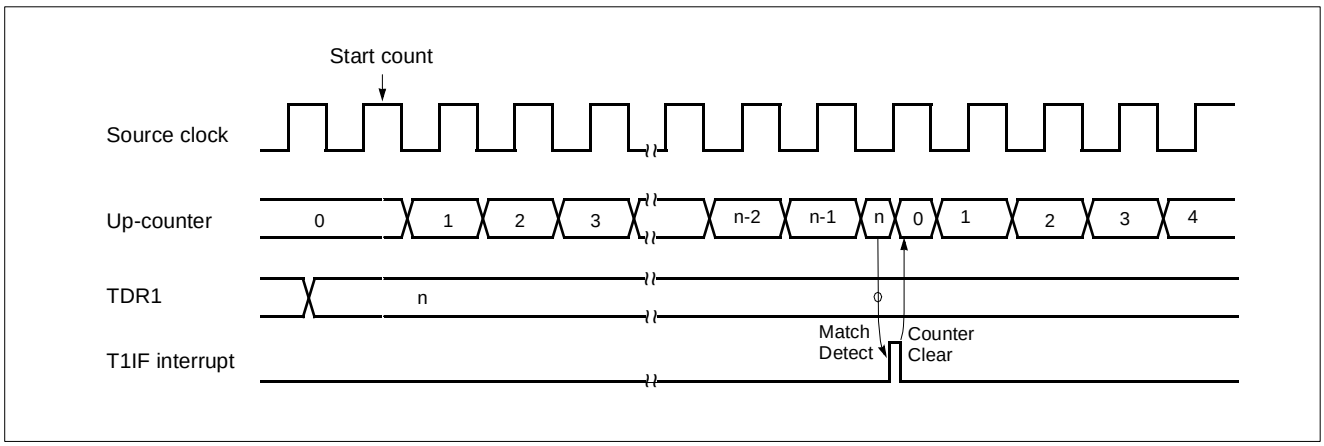


Figure 11-5 Timer Mode Timing Chart

Value of TM[2:0]	Clock Source		Resolution		Maximum Time Setting	
	SCMR[1:0]= 00 or 01	SCMR[1:0]=10 or 11	At f_{MAIN}=4.19MHz	At f_{SUB}=32.768kHz	At f_{MAIN}=4.19MHz	At f_{SUB}=32.768kHz
000	f_{EC1}	f_{EC1}	$1/f_{EC1}$ s	$1/f_{EC1}$ s	$1/f_{EC1}$ x 256 s	$1/f_{EC1}$ x 256 s
001	$f_M \div 2$	$f_S \div 2$	0.476 us	61.03 us	122.1 us	15.6 ms
010	$f_M \div 2^3$	$f_S \div 2^3$	1.907 us	244.14 us	488.3 us	62.5 ms
011	$f_M \div 2^5$	$f_S \div 2^5$	7.629 us	976.56 us	1953.1 us	250.0 ms
100	$f_M \div 2^7$	$f_S \div 2^7$	30.517 us	3906.25 us	7812.5 us	1000.0 ms
101	$f_M \div 2^9$	$f_S \div 2^9$	122.070 us	15625.00 us	31250.0 us	4000.0 ms
110	Invalid	-	-	-	-	-
111	Invalid	-	-	-	-	-

Table 11-2 Timer/Counter 1 Source clock Interrupt Time

f_M: main-clock frequency, f_S: sub-clock frequency, f_{EC1}: external event from $\overline{EC1}$ pin frequency

Event counter Mode

In this mode, counting up is started by an external trigger. This trigger means falling edge of the $\overline{EC1}$ pin input. Source clock is used as an internal clock selected with TM1. The contents of TDR1 are compared with the contents of the up-counter. If a match is found, an T1IF interrupt is generated, and the counter is cleared to "0". The counter is restarted by the falling edge of the $\overline{EC1}$ pin input.

The maximum frequency applied to the $\overline{EC1}$ pin is f_{MAIN}/2 [Hz] in main clock mode, and f_{SUB}/2[Hz] is sub clock mode.

In order to use event counter function, the bit EC1S of the Port Mode Register PMR0(address $0D9_H$) is required to be set to "1".

After reset, the value of TDR1 is undefined, it should be initialized to between 1_H~FF_H, not to "0".

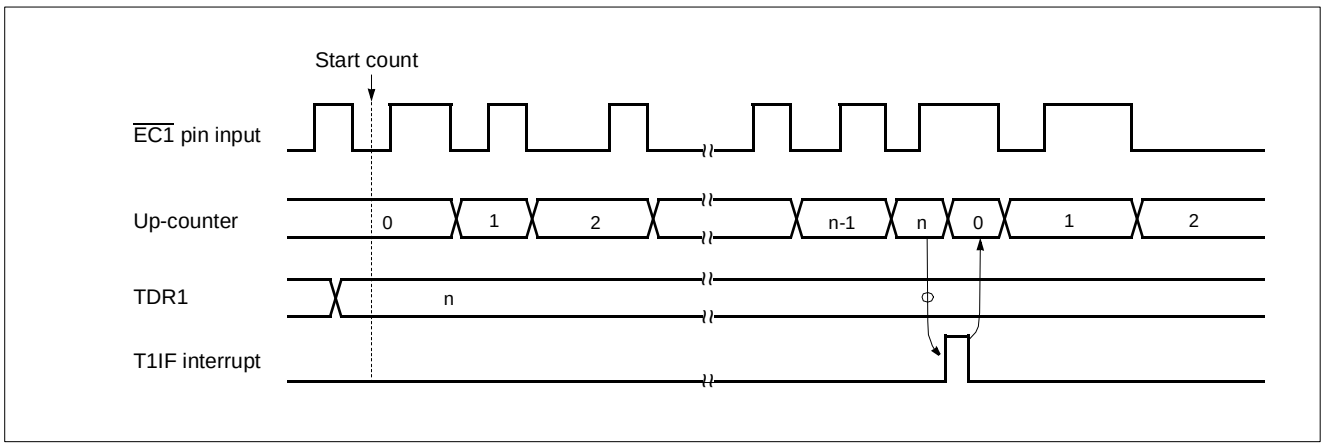


Figure 11-6 Event Counter Mode Timing Chart

The interval period of Timer is calculated as below equation.

$$Period = \frac{1}{f_{XIN}} \times Prescaler \text{ ratio} \times TDR$$

Example:

Every 1ms interrupt request flag is generated at 4MHz

```
:
LDM     PCOR,#1         ; Enable Peri. Clock
LDM     TM1,#1BH        ; divide by 8
LDM     TDR1,#125       ; 8us x 125= 1ms
SET1    T1E             ; Enable Timer 1 Int.
EI                      ; Enable Master Int.
```

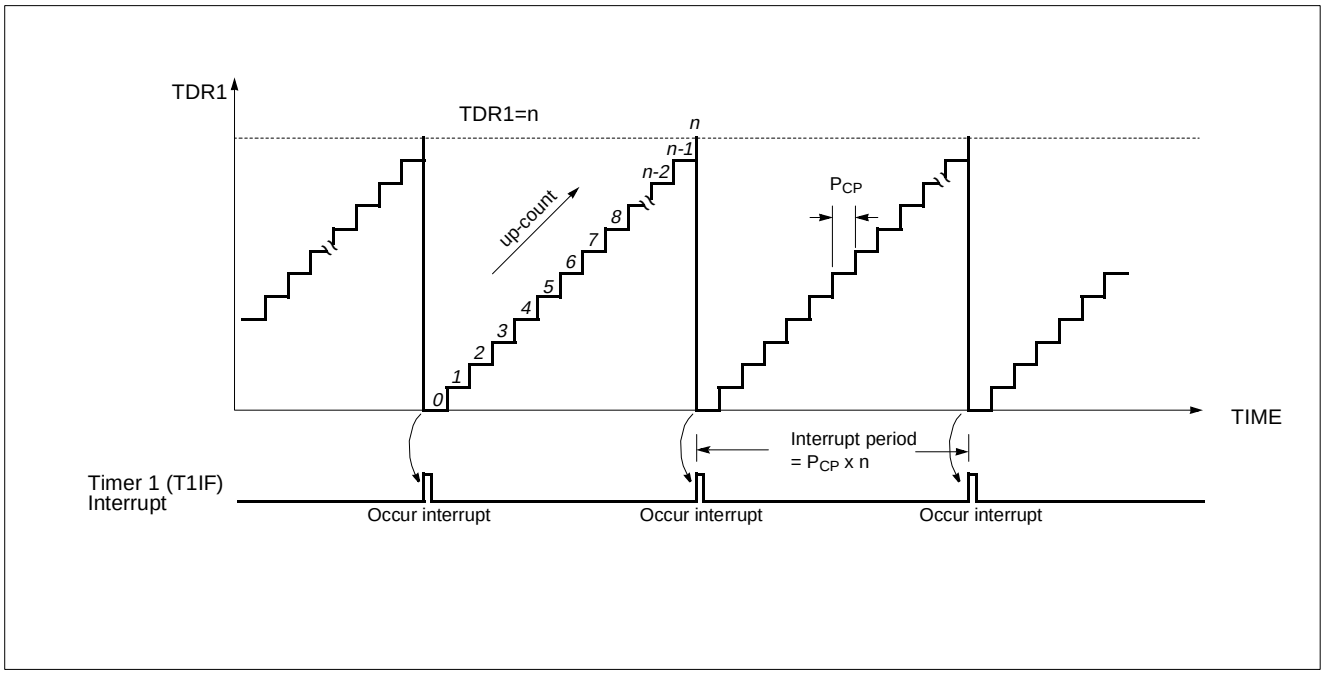


Figure 11-7 Count Example of Timer / Event counter

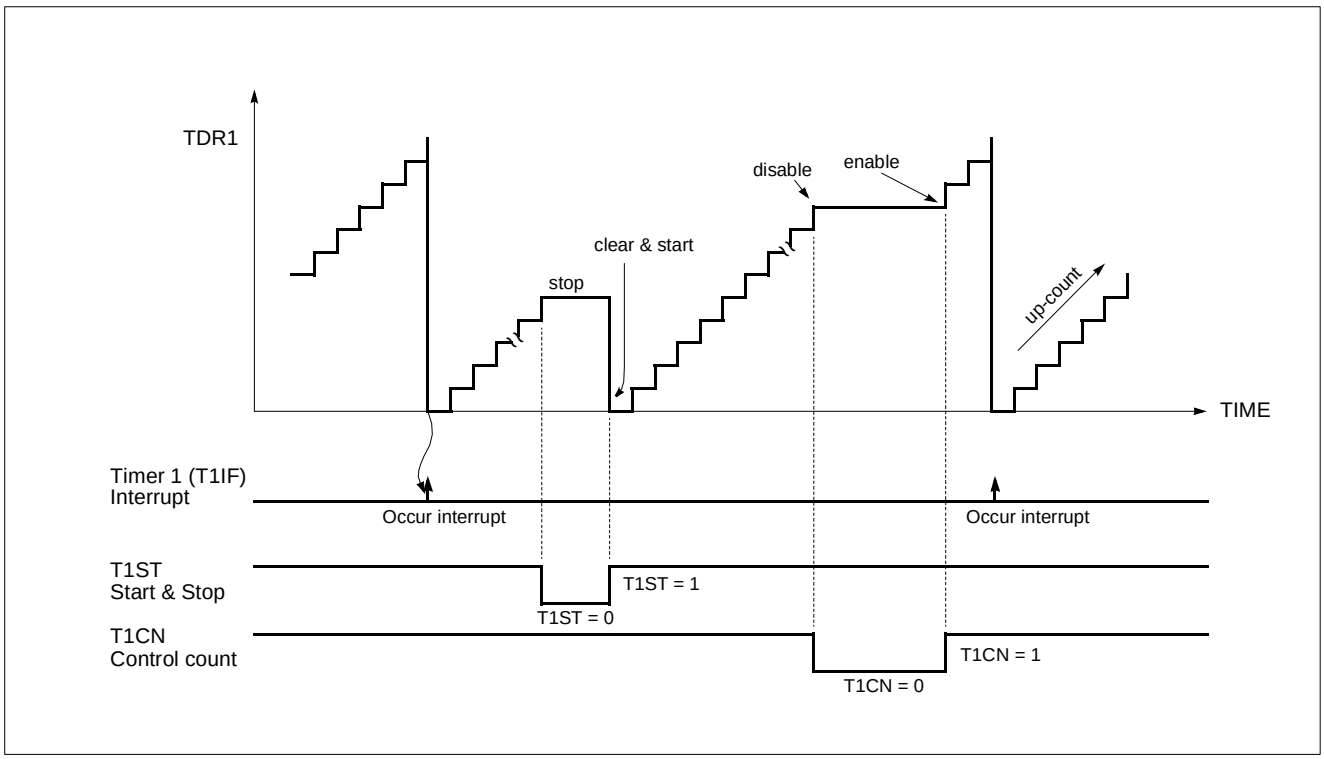


Figure 11-8 Count Operation of Timer / Event counter

11.3 Watch Timer

The watch timer consists of the clock selector, 14-bit binary counter and Watch timer mode register. It is a multi-purpose timer. It is generally used for watch design. Since Sub-frequency keeps running in spite of Stop mode, Watch Timer continues its operation.

Bit 3 of WTMR enables or stops counter, and bit 2 and bit 1 determine the clock source between main or sub frequency. Because in Stop Mode, main-frequency clock stops, clock source should be sub-frequency clock.

In case that circuit uses 4.19MHz and Sub-frequency is 32.768kHz, bit 0 of WTMR may choose either 2Hz or 256Hz.

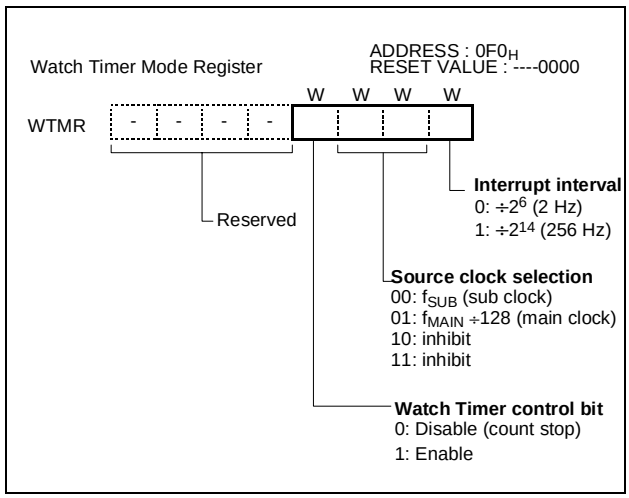


Figure 11-9 Watch Timer Mode Register

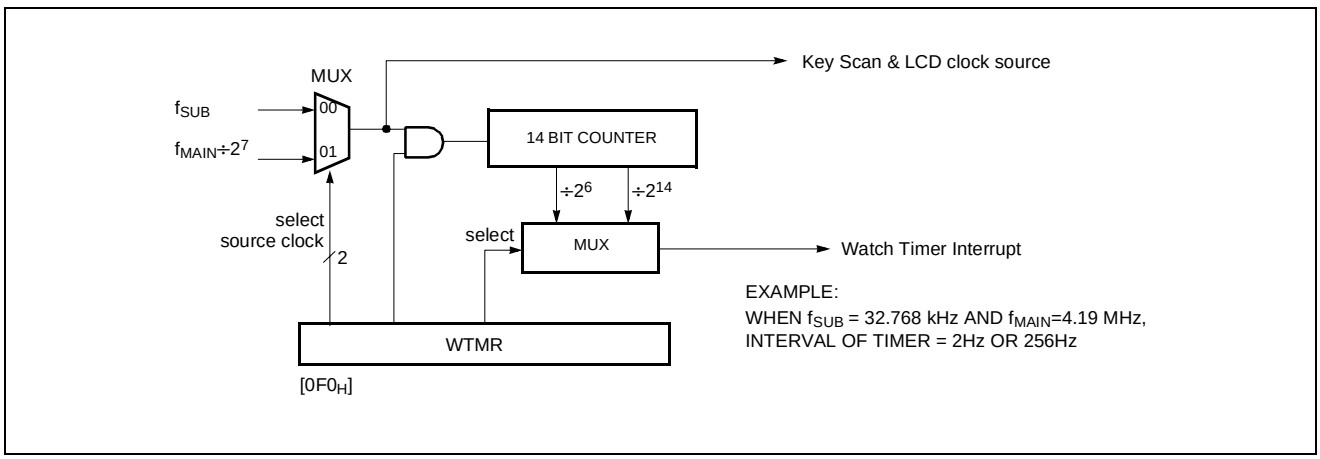


Figure 11-10 Watch Timer Block Diagram

Usage of Watch Timer in STOP mode

When system is off and watch should keep working, follow the steps below.

1. It determines the mode to perform between main mode and sub mode when released from Stop mode. and is set to Sub-frequency operation mode.
2. Enters in STOP mode.
3. After released by 0.5 second watch timer interrupt, count up 1 second and refreshes LCD Display. When the performing count up and refresh the LCD, the CPU operates either in main frequency mode or sub frequency mode.
4. Enters in STOP mode again.
5. Repeats 3 and 4.

As mentioned above, by releasing every 0.5 sec., power consumption can be reduced consideravably.

12. COMPARATOR

The A/D comparator circuit is shown in Figure 12-1 .

The A/D comparator circuit consists of the switch tree, ladder resistor, comparator and control register CMR, CSR (address $0EC_H$, $0ED_H$). The CSR register select normal port or analog input. The bit 7 of CSR has 1's written to them, port can be configured as comparator ports, and in that state can be used as analog input. The lower 2 bits of CMR control which port applied into comparator input. As analog inputs, unselected port can be used digital input (normal input) as shown in Table 12-2.

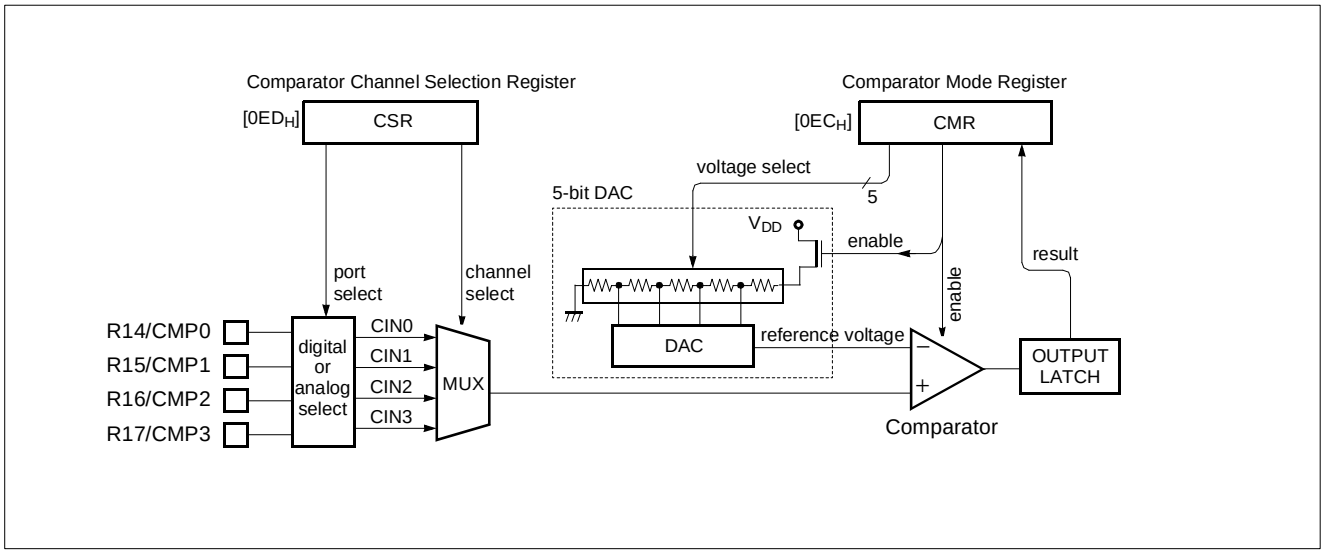


Figure 12-1 Block Diagram of Comparator circuit

Control

The comparator module has four analog inputs for the GMS81C3004.

The Comparator Register, that is the comparator register CMR and CSR are shown in Figure 12-2 .

Lower 5 bits of CMR can select voltage as 1/64 V_{DD} step internal reference voltage, based on the setting of bits 0 to bit 5. The comparator result between the analog input voltage and the internal reference voltage is stored in bit 6 of CMR.

Bit 6 of CMR	Description
0	input voltage < reference voltage
1	input voltage > reference voltage

The CMR can be read or tested by byte manipulation instruction, not bit manipulation.

Example:

```
        :
        LDA    CMR
        BBC    A.6,GOTO3
        :
        :

GOTO3:
        :
        :
        :
```

16 machine cycle (8μs at 4MHz) is required for comparison the result of comparison is stored in the bit 6 of Comparator Mode Register CMR (address $0EC_H$). The bit 7 is comparator enable bit. When comparator is enabled, the current consumption of comparator is typically 0.95mA (to be defined after).

00000: $V_{DD}/64$
00001: $3 \cdot V_{DD}/64$
00010: $5 \cdot V_{DD}/64$
00011: $7 \cdot V_{DD}/64$
00100: $9 \cdot V_{DD}/64$
00101: $11 \cdot V_{DD}/64$
00110: $13 \cdot V_{DD}/64$
00111: $15 \cdot V_{DD}/64$
:
:
:
:
11000: $49 \cdot V_{DD}/64$
11001: $51 \cdot V_{DD}/64$
11010: $53 \cdot V_{DD}/64$
11011: $55 \cdot V_{DD}/64$
11100: $57 \cdot V_{DD}/64$
11101: $59 \cdot V_{DD}/64$
11110: $61 \cdot V_{DD}/64$
11111: $63 \cdot V_{DD}/64$

Table 12-1 Setting the Reference voltage

Select analog input pin by using bit 1 and bit 0 of the Channel Selection Register CSR (address $0ED_H$).

The port pins can be configured as analog inputs or as digital I/O by setting the CSR. Refer to Table 12-2.

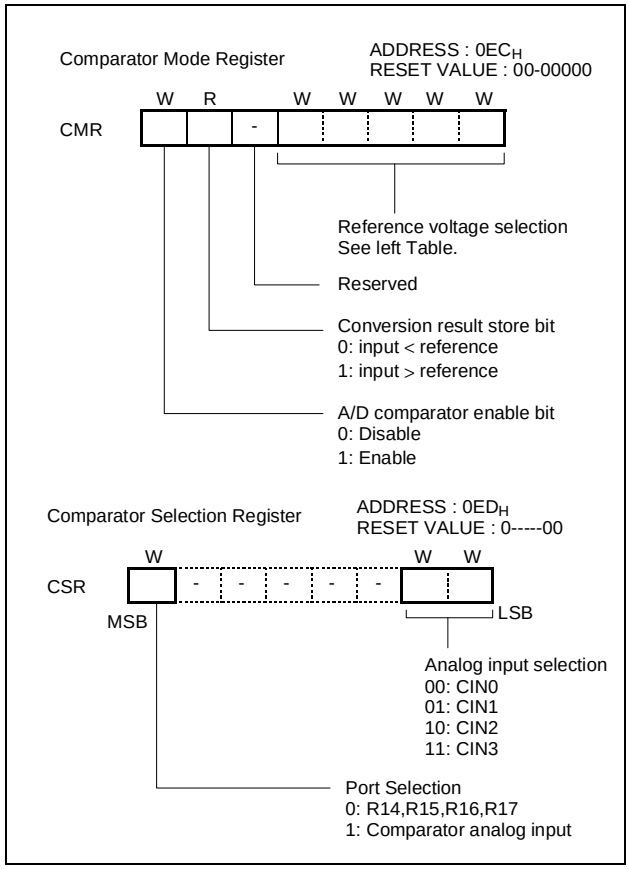


Figure 12-2 Comparator Registers

CSR[7]	CSR[1:0]	CHANNEL	Remarks
0	XX	-	R14,R15,R16,R17
1	00	CIN0	R15,R16,R17 can be used as digital input
1	01	CIN1	R14,R16,R17 can be used as digital input
1	10	CIN2	R14,R15,R17 can be used as digital input
1	11	CIN3	R14,R15,R16 can be used as digital input

Table 12-2 Pin Configuration of Analog input

13. INTERRUPTS

The GMS81C3004 interrupt circuits consist of Interrupt enable register (IENH, IENL), Interrupt request flags of IRQH, IRQL, Priority circuit, and Master enable flag ("I" flag of PSW). 9 interrupt sources are provided. The configuration of interrupt circuit is shown in Figure 13-1 .

Below table shows the Interrupt priority

Reset/Interrupt	Symbol	Priority
Hardware Reset	RESET	-
Basic Interval Timer	BIT	1
External Interrupt 0	INT0	2
External Interrupt 1	INT1	3
Timer/Counter 1	Timer 1	4
External Interrupt 2	INT2	5
Watchdog Timer	WDT	6
Watch Timer	WT	7
Key Scan interrupt	KS	8

The External Interrupts INT0, INT1, INT2 each can be transition-activated (1-to-0 or 0-to-1 transition).
The flags that actually generate these interrupts are bit INT0F, INT1F and INT2F in register IRQH. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.

The Timer 1 Interrupts are generated by T1IF which is set by a match in their respective timer/counter register.

The Basic Interval Timer Interrupt is generated by BITIF which is set by an overflow in the timer register.

The interrupts are controlled by the interrupt master enable flag I-flag (bit 2 of PSW), the interrupt enable register (IENH, IENL), and the interrupt request flags (in IRQH and IRQL) except Power-on reset and software BRK interrupt.

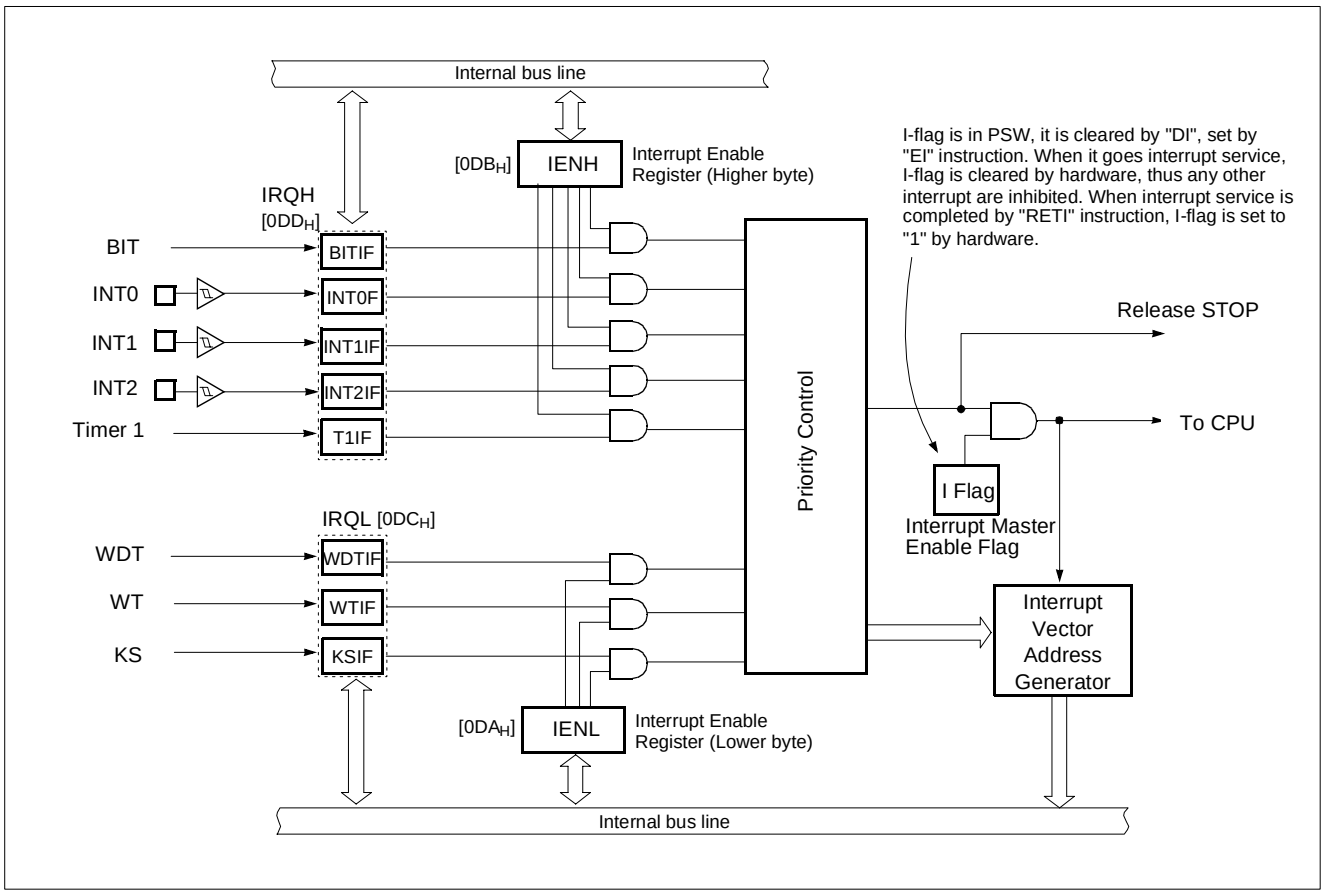


Figure 13-1 Block Diagram of Interrupt

Interrupt enable registers are shown in Figure 13-3 . These registers are composed of interrupt enable flags of each interrupt source and these flags determines whether an interrupt will be accepted or not. When enable flag is "0", a corresponding interrupt source is prohibited. Note that PSW contains also a master enable bit, I-flag, which disables all interrupts at once.

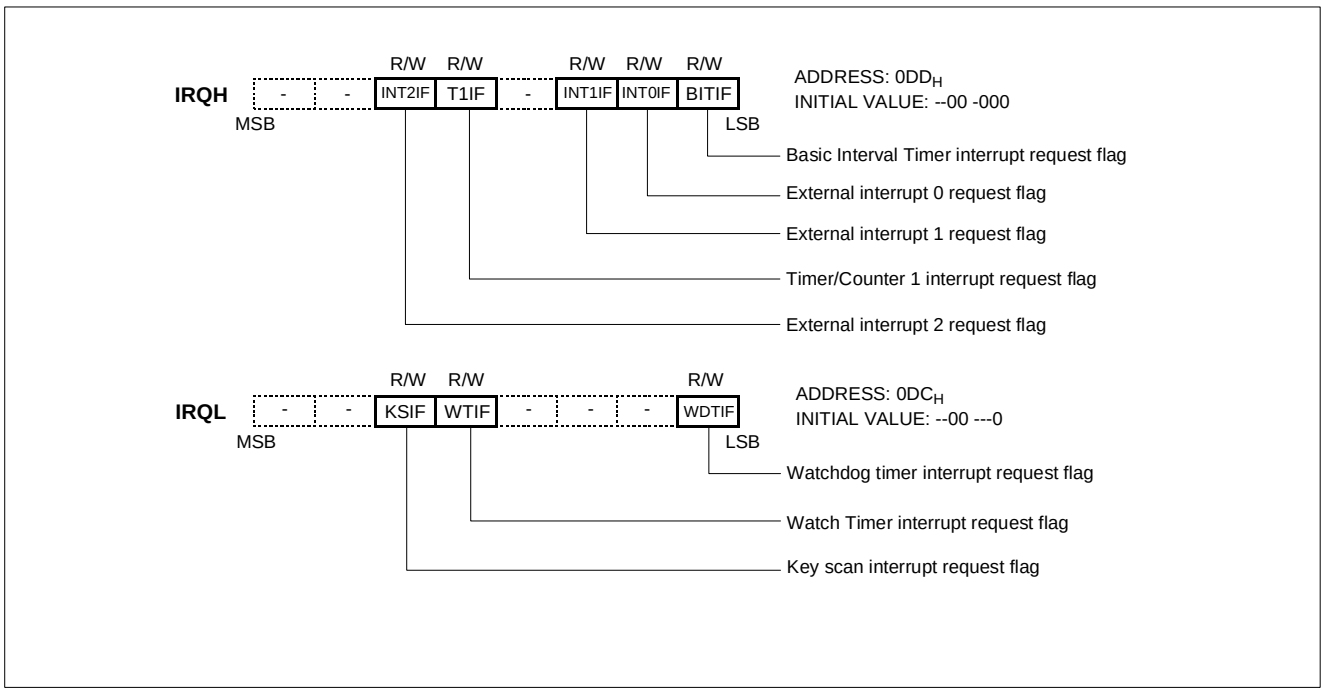


Figure 13-2 Interrupt Request Flag

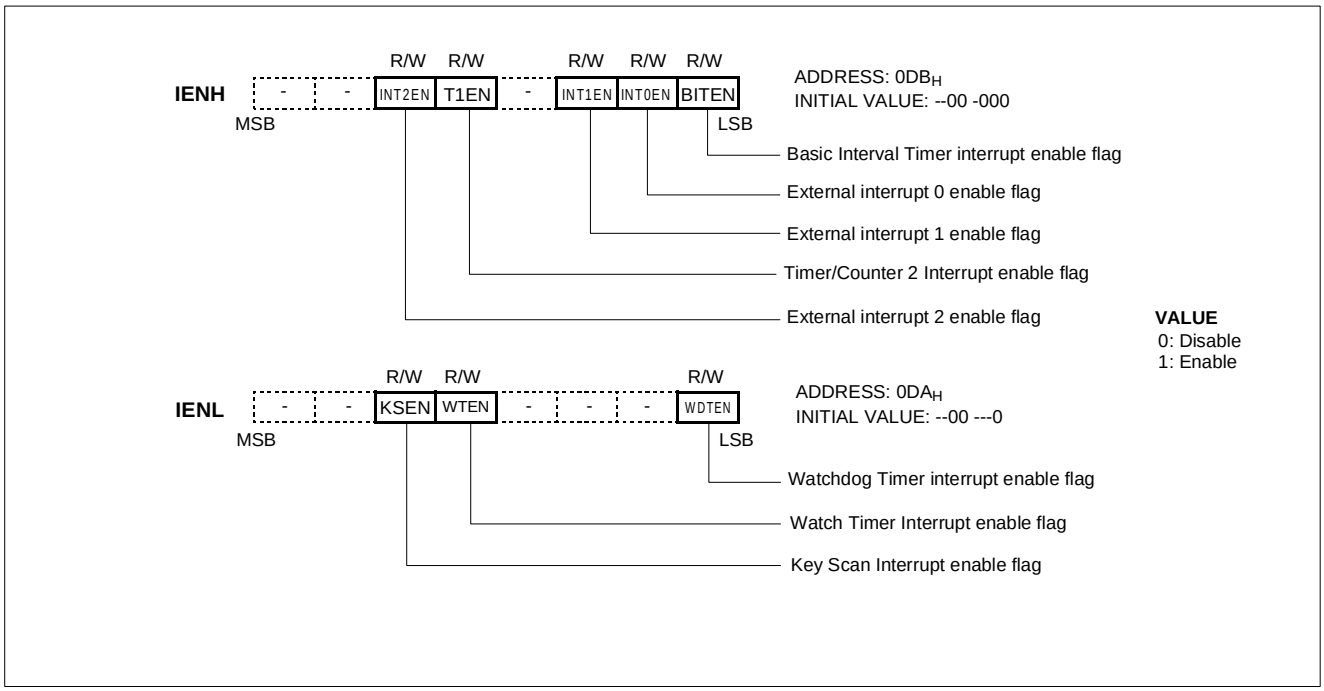


Figure 13-3 Interrupt Enable Flag

13.1 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 8 f_{OSC} (2 µs at f_{MAIN}=4.19MHz) after the completion of the current instruction execution. The interrupt service task is terminated upon execution of an interrupt return instruction [RETI].

Interrupt acceptance

1. The interrupt master enable flag (I-flag) is cleared to "0" to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.
2. Interrupt request flag for the interrupt source accepted is cleared to "0".
3. The contents of the program counter (return address) and the program status word are saved (pushed) onto the stack area. The stack pointer decreases 3 times.
4. The entry address of the interrupt service program is read from the vector table address and the entry address is loaded to the program counter.
5. The instruction stored at the entry address of the interrupt service program is executed.

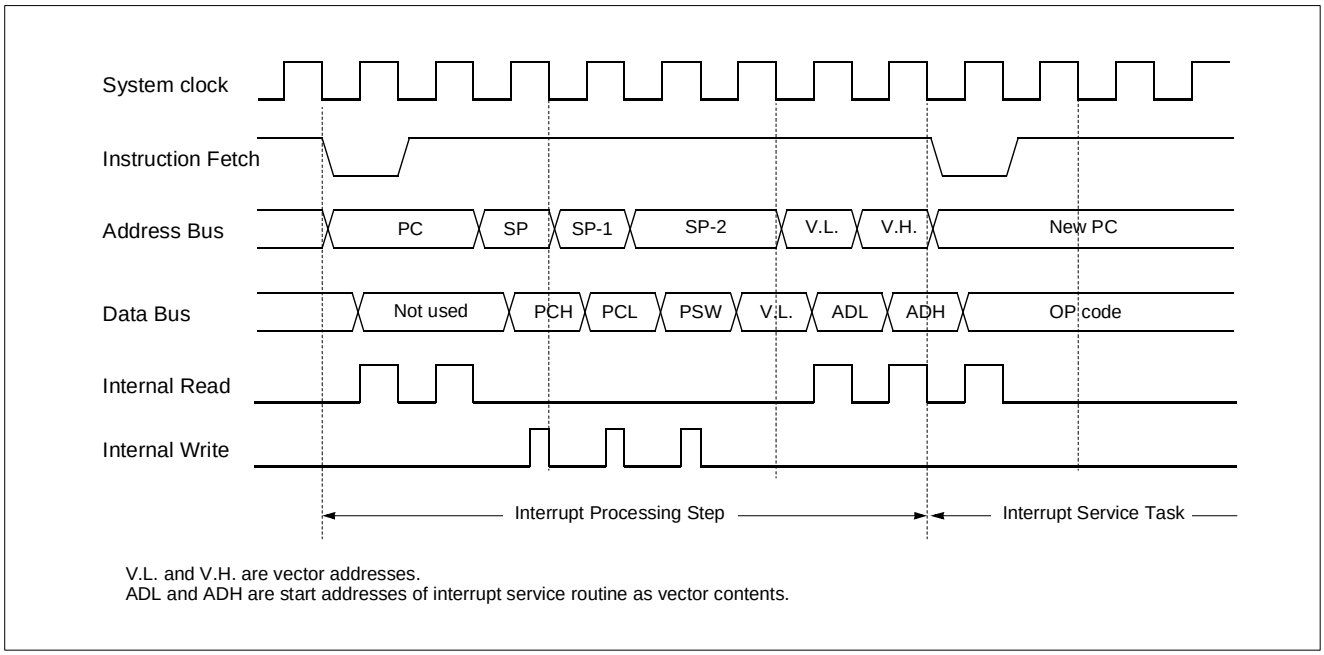
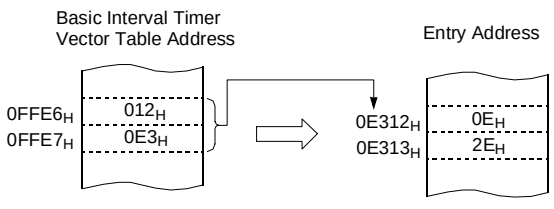


Figure 13-4 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



Correspondence between vector table address for BIT interrupt and the entry address of the interrupt service program.

A interrupt request is not accepted until the I-flag is set to "1" even if a requested interrupt has higher priority than that of the current interrupt being serviced.

When nested interrupt service is required, the I-flag should be set to "1" by “EI” instruction in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

Saving/Restoring General-purpose Register

During interrupt acceptance processing, the program counter and the program status word are automatically saved on the stack, but accumulator and other registers are not saved itself. These registers are saved by the software if necessary. Also, when multiple interrupt services are nested, it is necessary to avoid using the same data memory

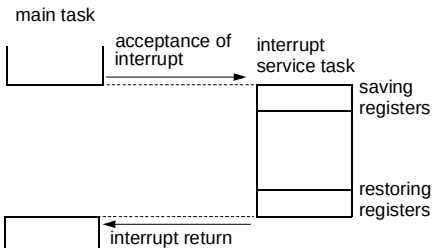
area for saving registers.

The following method is used to save/restore the general-purpose registers.

Example: Register save using push and pop instructions

```
INTxx:  PUSH    A       ;SAVE ACC.
        PUSH    X       ;SAVE X REG.
        LDA     RPR     ;SAVE RPR
        PUSH    A
        interrupt processing
        POP     A
        STA     PRP     ;RESTORE RPR
        POP     X       ;RESTORE X REG.
        POP     A       ;RESTORE ACC.
        RETI            ;RETURN
```

General-purpose register save/restore using push and pop instructions;



13.2 BRK Interrupt

Software interrupt can be invoked by BRK instruction, which has the lowest priority order.

Interrupt vector address of BRK is shared with the vector of TCALL 0 (Refer to Program Memory Section). When BRK interrupt is generated, B-flag of PSW is set to distinguish BRK from TCALL 0.

Each processing step is determined by B-flag as shown in Figure 13-5 .

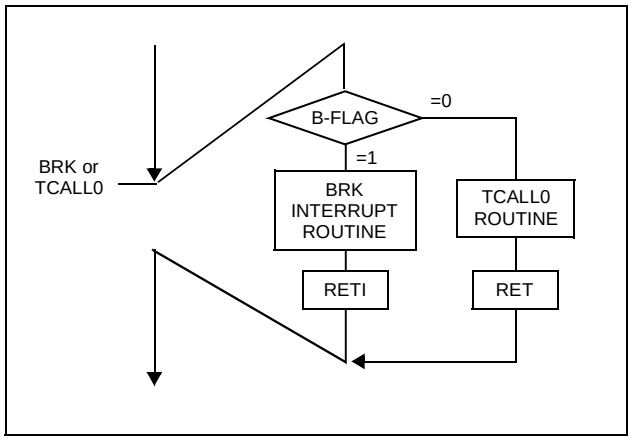


Figure 13-5 Execution of BRK/TCALL0

13.3 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the interrupt are received at the same time simultaneously, an internal polling sequence determines by hardware which request is serviced.

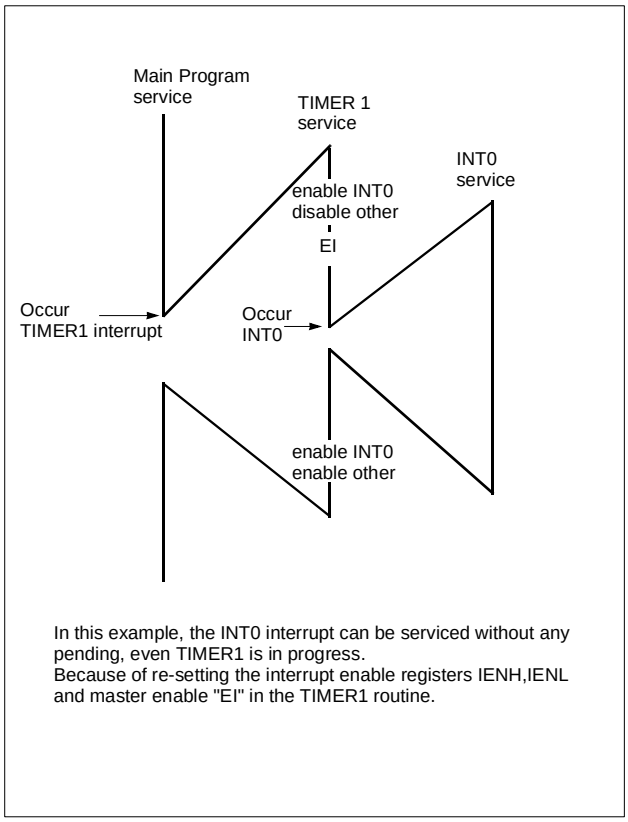


Figure 13-6 Execution of Multi Interrupt

However, multiple processing through software for special features is possible. Generally when an interrupt is accepted, the I-flag is cleared to disable any further interrupt. But as user sets I-flag in interrupt routine, some further interrupt can be serviced even if certain interrupt is in progress.

Example: Even though Timer1 interrupt is in progress, INT0 interrupt serviced without any suspend.

```
TIMER1:  PUSH   A
         PUSH   X
         PUSH   Y
         LDM    IENH,#2      ;Enable INT0 only
         LDM    IENL,#0      ;Disable other
         EI                  ;Enable Interrupt
         :
         :
         :

         :
         :
         :
         LDM    IENH,#37H    ;Enable all interrupts
         LDM    IENL,#31H
         POP    Y
         POP    X
         POP    A
         RETI
```

13.4 External Interrupt

The external interrupt on INT0, INT1 and INT2 pins are edge triggered depending on the edge selection register IESR (address $0D8_H$) as shown in Figure 13-7 .

The edge detection of external interrupt has three transition activated mode: rising edge, falling edge, and both edge.

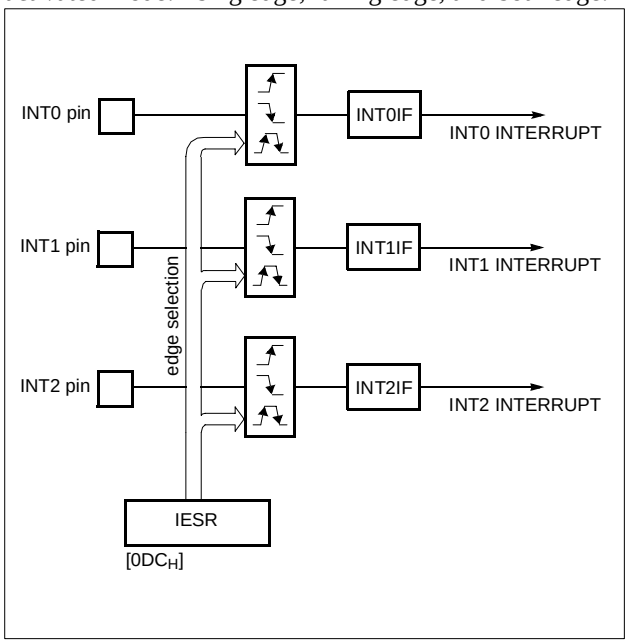


Figure 13-7 External Interrupt Block Diagram

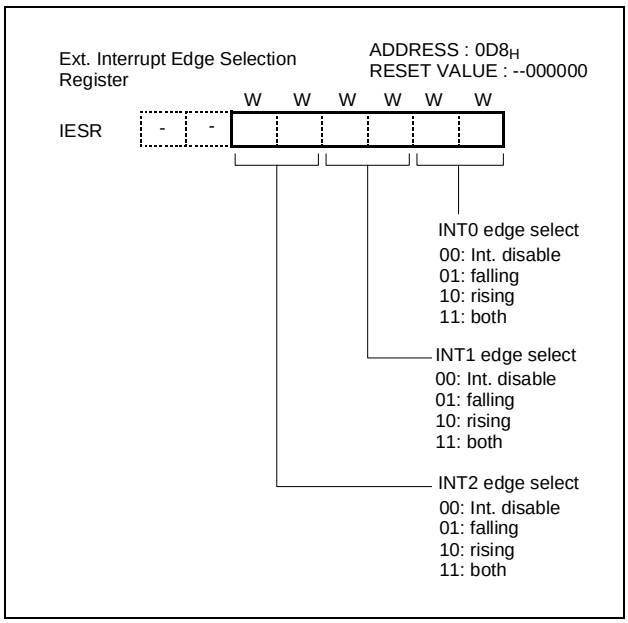


Figure 13-8 External Interrupt Edge Selection Register

INT0, INT1 and INT2 are multiplexed with general I/O ports (R10~R12). To use external interrupt pin, the bit of R0 port mode register PMR0 should be set to "1" correspondingly.

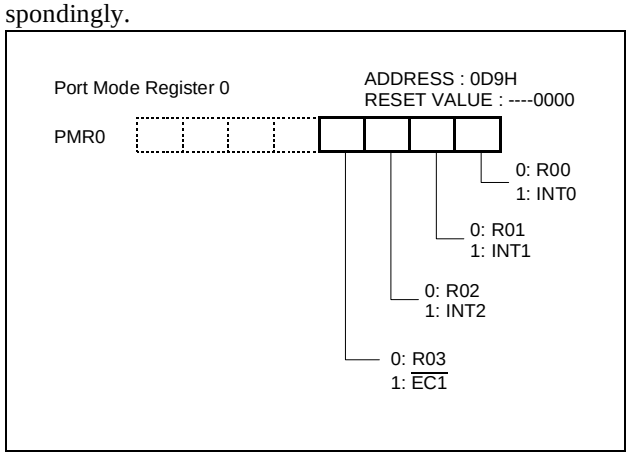


Figure 13-9 PMR0: R0 Port Mode Register

Example: To use as an INT0 and INT2

```
          :
          :
;**** Set port as an input port R00,R02
          LDM    R0DD,#1111_1010B
          ;
;**** Set port as an interrupt port
          LDM    PMR0,#05H
          ;
;**** Set Falling-edge Detection
          LDM    IESR,#0001_0001B
          :
          :
          :
```

Response Time

The INT0, INT1 and INT2 edge are latched into INT1IF, INT1IF and INT2IF at every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The DIV itself takes twelve cycles. Thus, a minimum of twelve complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine.

shows interrupt response timings.

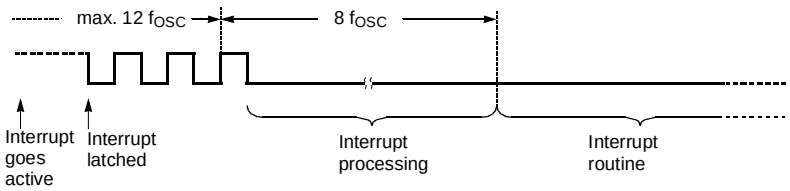


Figure 13-10 Interrupt Response Timing Diagram

14. KEY SCAN

The key-scan block consists of Port selection Multiplexer, Interrupt controller, 4-bit binary counter and Key scan control register, and Key data register.

When the key scan interrupt is used, key scan register KSCR (address $0F4_H$) should be set properly as shown in Figure 14-2 .

Key Scan matrix is configured by 8 inputs (KS0~KS7) and 16 outputs (SEG16~SEG31). Number of key inputs are defined by the key scan control register (KSCR[6:5]). Output signal that are strobe are fixed as SEG16 to SEG31.

If key scan is detected at any one or more of these pins, the KSIF request flag is set to "1". This generates an interrupt request. It also can be used in the way of release from STOP mode.

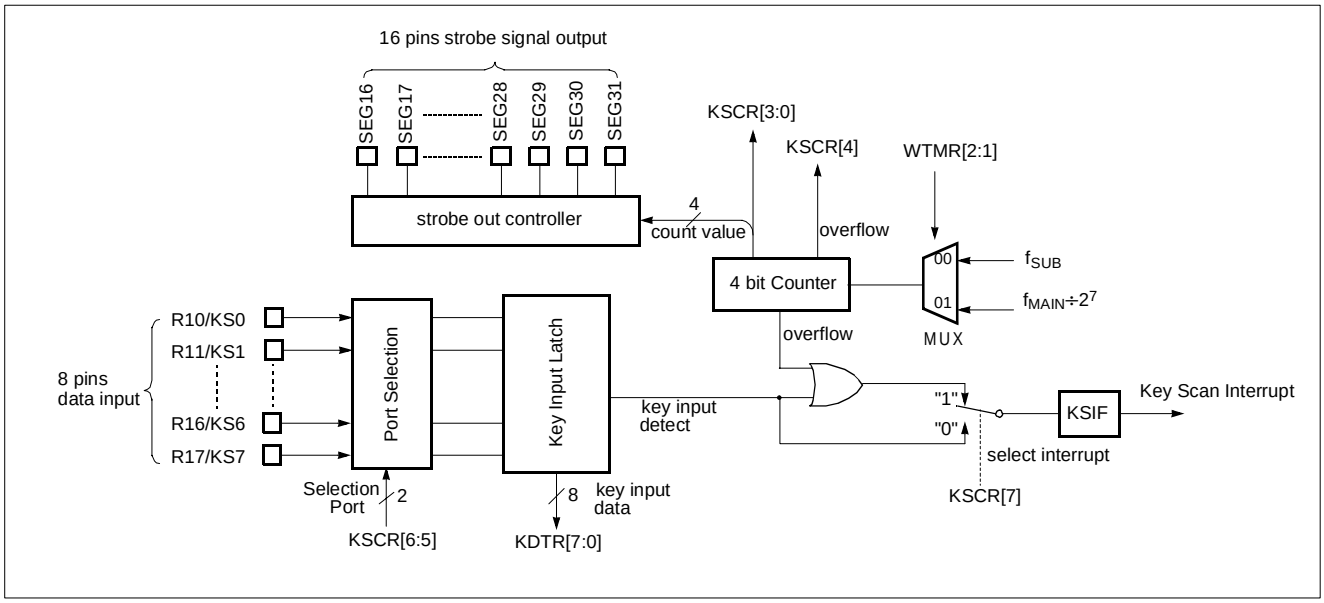


Figure 14-1 Key Scan Interrupt Block Diagram

Strobe output signals are generated according to 4-bit count value. The relation between Key scan register value and strobe signal is shown as below table.

Counter value	Strobe Pin
0	SEG16
1	SEG17
2	SEG18
3	SEG19
4	SEG20
5	SEG21
6	SEG22
7	SEG23
8	SEG24
9	SEG25
10	SEG26
11	SEG27
12	SEG28
13	SEG29
14	SEG30
15	SEG31

Once key scan interrupt occurs, key scan interrupt is disabled. To accept next interrupt, the KDTR has to be read by software. Otherwise, key-scan is not enabled.

At every 8th clock, one strobe output is generated. There are 16 pins in all, therefore total key scan time is 3900µs (244µs x 16). Refer to Figure 14-3 .

Example: The registers should be defined properly to use key scan input function.

```
        :
        :
LDM     PUR1,#0       ;For disabling pull-ups
LDM     KSCR,#0E0H    ;For using 8 inputs
LDM     IENL,#20H     ;Enable Keyscan
EI
        :
        :
```

Note: *When R1 is used as key scan port, there should be no pull-up. PUR1 should be written to '0' in order not to operate pull-up. Otherwise, VCLn voltage is changed and may occur flicker in LCD panel display.*

Usage of Key Scan

1. Clear bit 7 of the KSCR, interrupt activate on Key Scan.
2. Specify bit 5 and bit 6 of the KSCR properly by selecting what port you want as key scan input.
3. Enable key scan interrupt.
4. When interrupt occurs, store the 4-bit counter value (lower 4-bit of KSCR) and key input data (KDTR) into user RAM area.
5. When the next interrupt occurs, compare the 4-bit counter values of KSCR and KDTR with RAM value stored before.
6. In case that these 2 values are not equal:
   If KDTR value is different, it means 2 keys are pressed successively. And if KSCR value is different, it means more than 2 keys are pressed simultaneously.

   In case that these 2 values are equal;
   If the number of bit '0' in KSCR values is over 2, more than 2keys are pressed.
   And if the number of bit "0" is one, it indicates the key input pin of bit "0"and seg pin number of strobe point as Counter Value.
   Therefore, it is possible to distinguish which key is pressed.

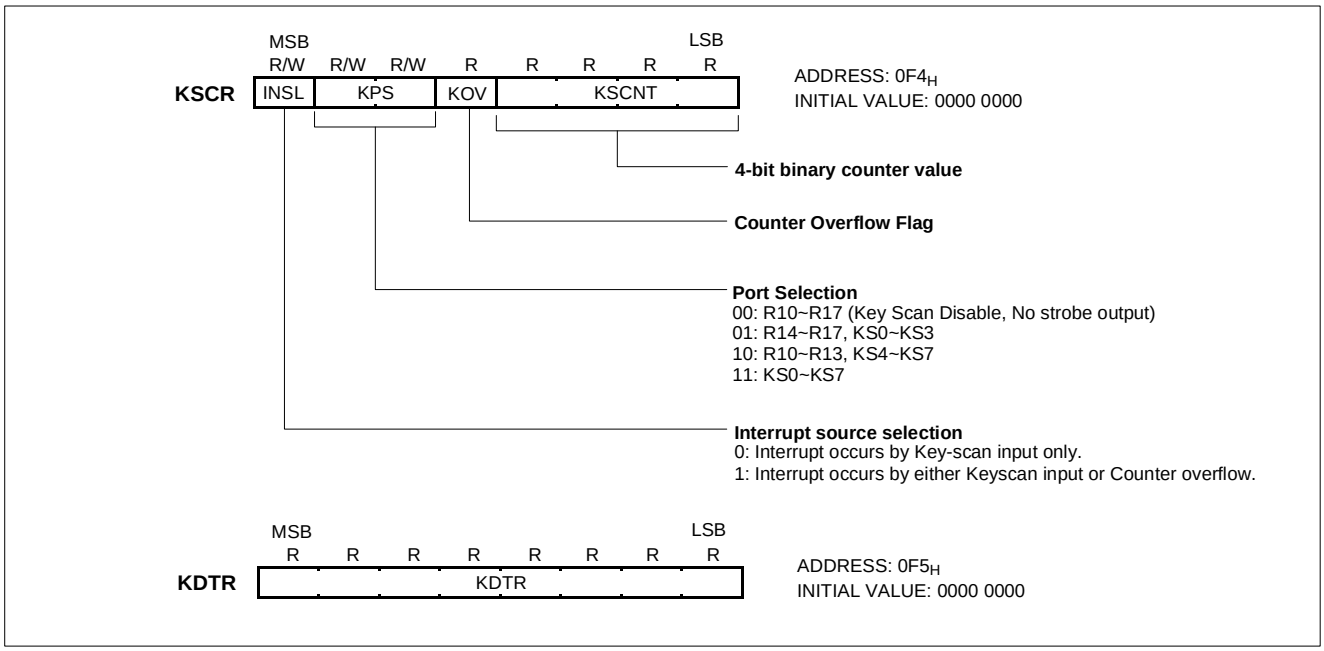


Figure 14-2 Key Scan Registers

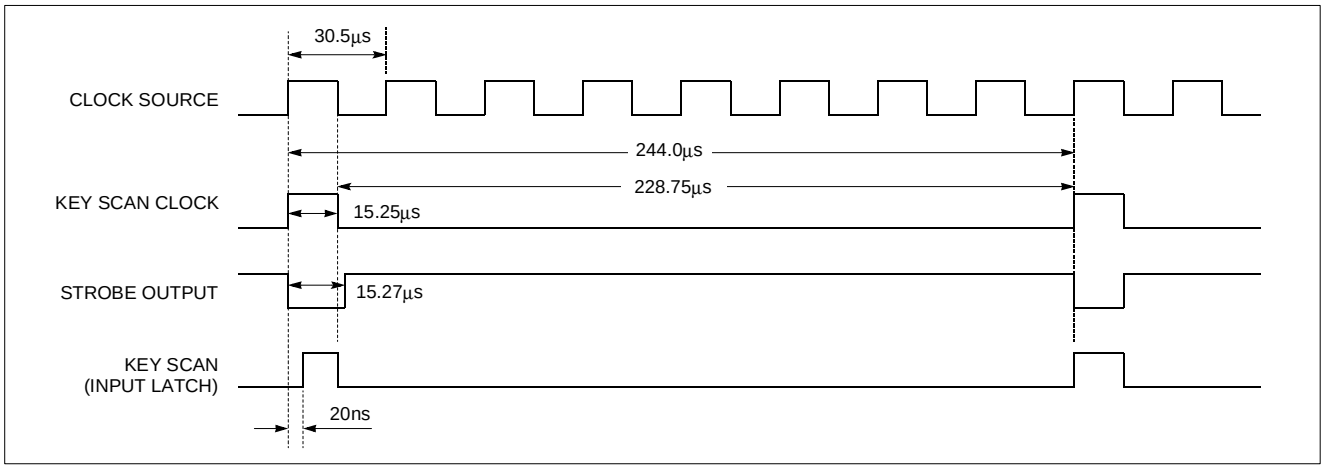


Figure 14-3 Key Scan Timing

15. LCD DRIVER

The GMS81C3004 has the circuit that directly drives the liquid crystal display (LCD) and its control circuit.

The GMS81C3004 has the following pins connected with LCD.

① Segment output port 40 pins (SEG0-SEG39)
② Common output port 8 pins (COM0-COM7)

In addition, VCL*n* pin is provided as the drive power pin.

The devices that can be directly driven are shown below.

1/8 duty (1/4 Bias) LCD.....Max. 320 Segment
By short between pin VCL2 and VCL3, 1/4 bias is used in GMS81C3004.

15.1 Configuration of LCD driver

Figure 15-1 shows the configuration of the LCD driver.

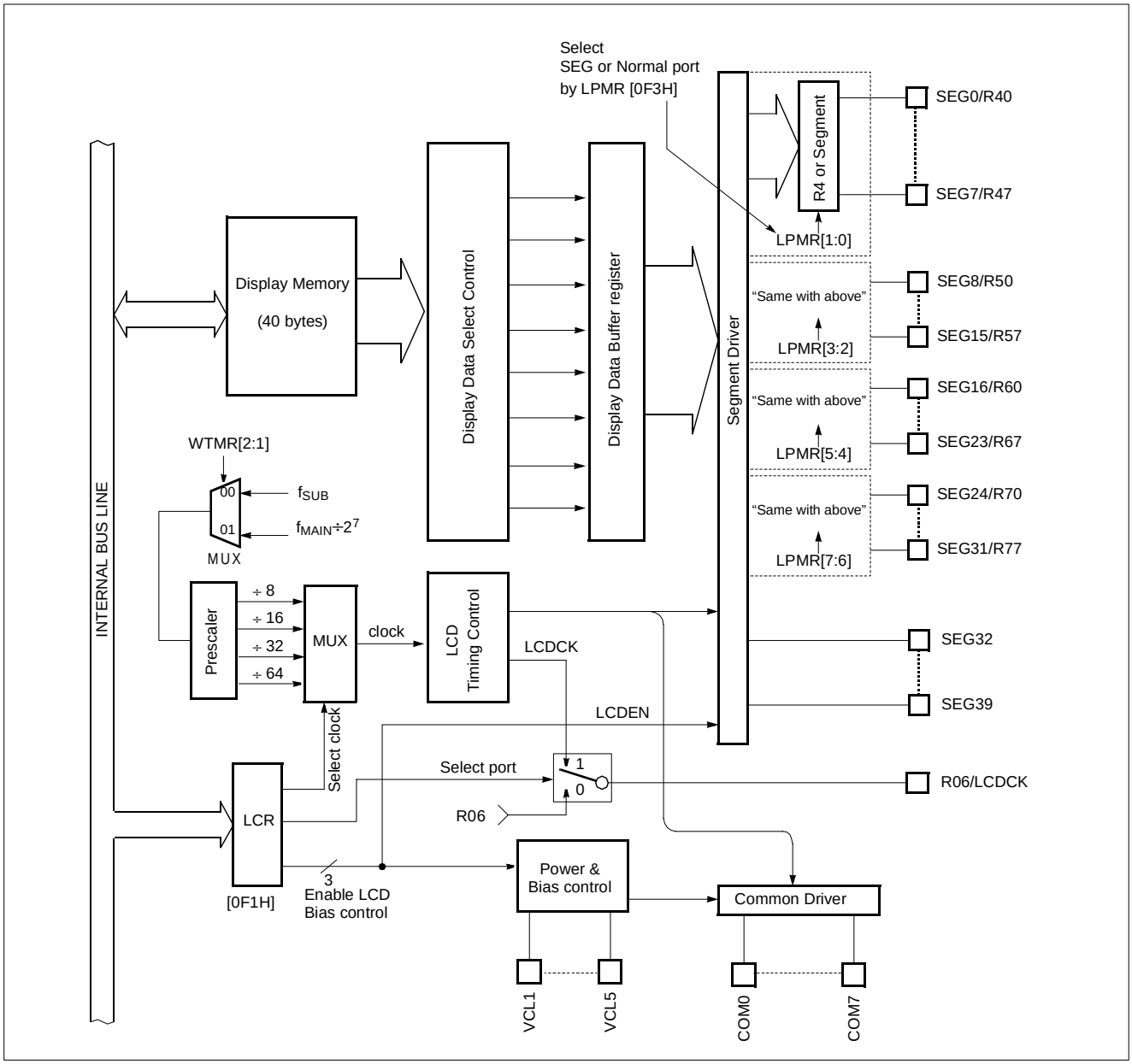
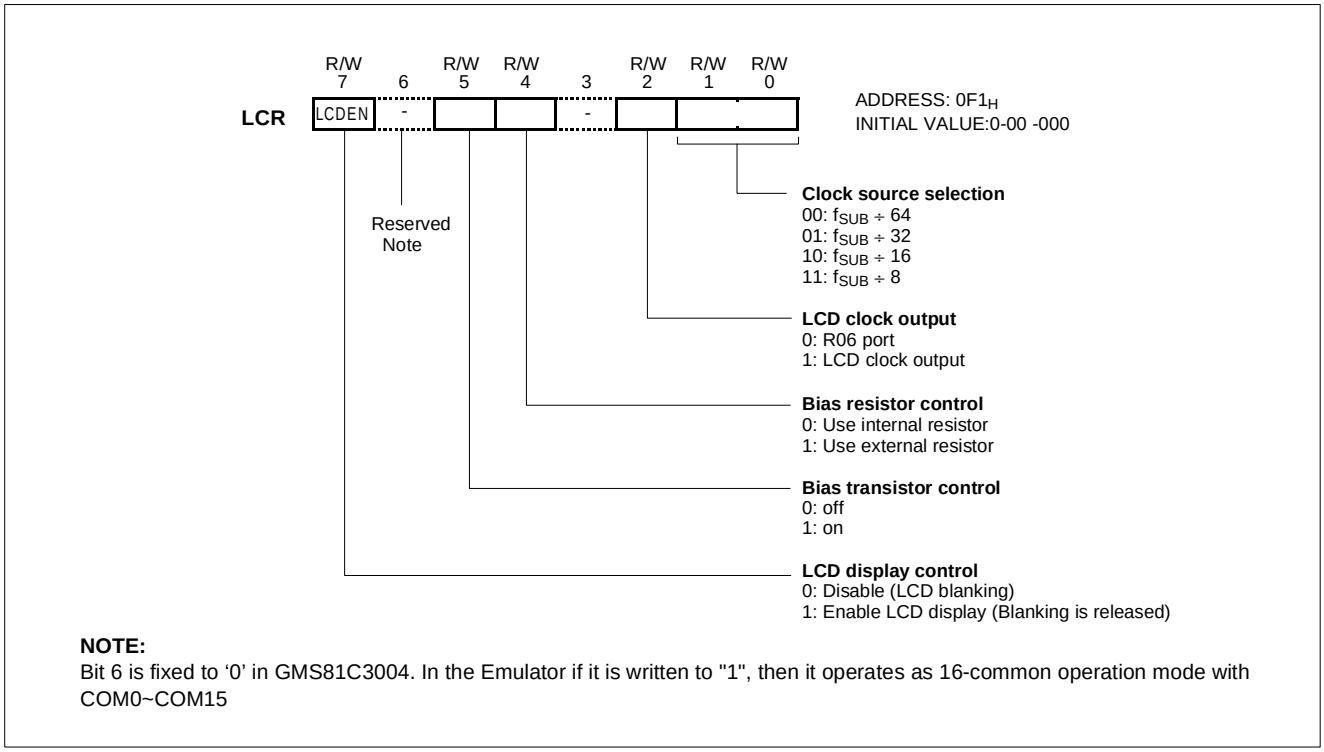


Figure 15-1 LCD Driver Block Diagram

15.2 Control of LCD Driver Circuit

The LCD driver is controlled by the LCD control register, LCR. Further, when the LCD is accessed, the most significant bit of the LCR must be cleared to "0" (Blanking).



NOTE:
Bit 6 is fixed to '0' in GMS81C3004. In the Emulator if it is written to "1", then it operates as 16-common operation mode with COM0~COM15

Figure 15-2 LCD Control Register

Selecting Frame Frequency

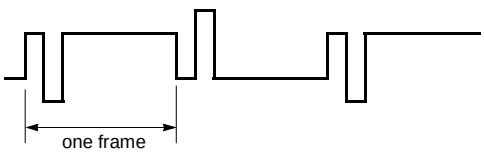
Frame frequency is set to the base frequency as shown in the following Table 15-1.

LCR[1:0]	LCD clock	Frame Frequency (Hz) (When f_{SUB} = 32.768 kHz)
00	$f_{SUB} \div 64$	64
01	$f_{SUB} \div 32$	128
10	$f_{SUB} \div 16$	256
11	$f_{SUB} \div 8$	512

Table 15-1 Setting of LCD Frame Frequency

The LCR[1:0] determines the frequency of COM signal scanning of each segment output. This is also referred to as the LCD clock signal pin, LCDCK. Since LCDCK is generated by dividing the watch timer clock(fW), the watch timer must be enabled when the LCD display is turned on. RESET clears the LCD control register LCR values to logic zero. When Bit 2 of LCR is '1', this clock outputs to R06 pin.

The LCD display can continue to operate during SLEEP and STOP modes if a sub-frequency clock is used as system clock source.



Display On/Off

Blanking is applied by setting LCDEN (bit 7 of LCR) to "0" and turns off the LCD by outputting the non light operation level to the COM pin. When setting Frame frequency or changing operating mode, LCD display should be off before operation, to prevent display flickering.

15.3 Bias Resistor

To operate LCD, built-in Bias resistor dividing V_{DD} to V_{SS} section into several stages generates necessary voltage.

Bit 5 of LCR switches Transistor supplying voltage to serially connected Bias resistor. If it is '1', it turns on, and if it is '0', it turns off. When the system needs adjusting the contrast of LCD, the bit 5 of LCR should be clear to "0" always. Then power is supplied through the external resistor as shown in Figure 15-3 .

Note: *Since the GMS81C3004 using 1/8 duty, so recommend to use 1/4 Bias. To use 1/4 Bias, VCL 2 pin and VCL 3 pin should be shorted externally as shown in Figure 15-3 (a). Furthermore, in case that user wants to use the specific voltage instead of voltage of internal Bias resistor, external Bias resistor can be used as shown in Figure 15-3 (b).*
To use external Bias resistor, Bit 4 and Bit 5 of LCR should be set.

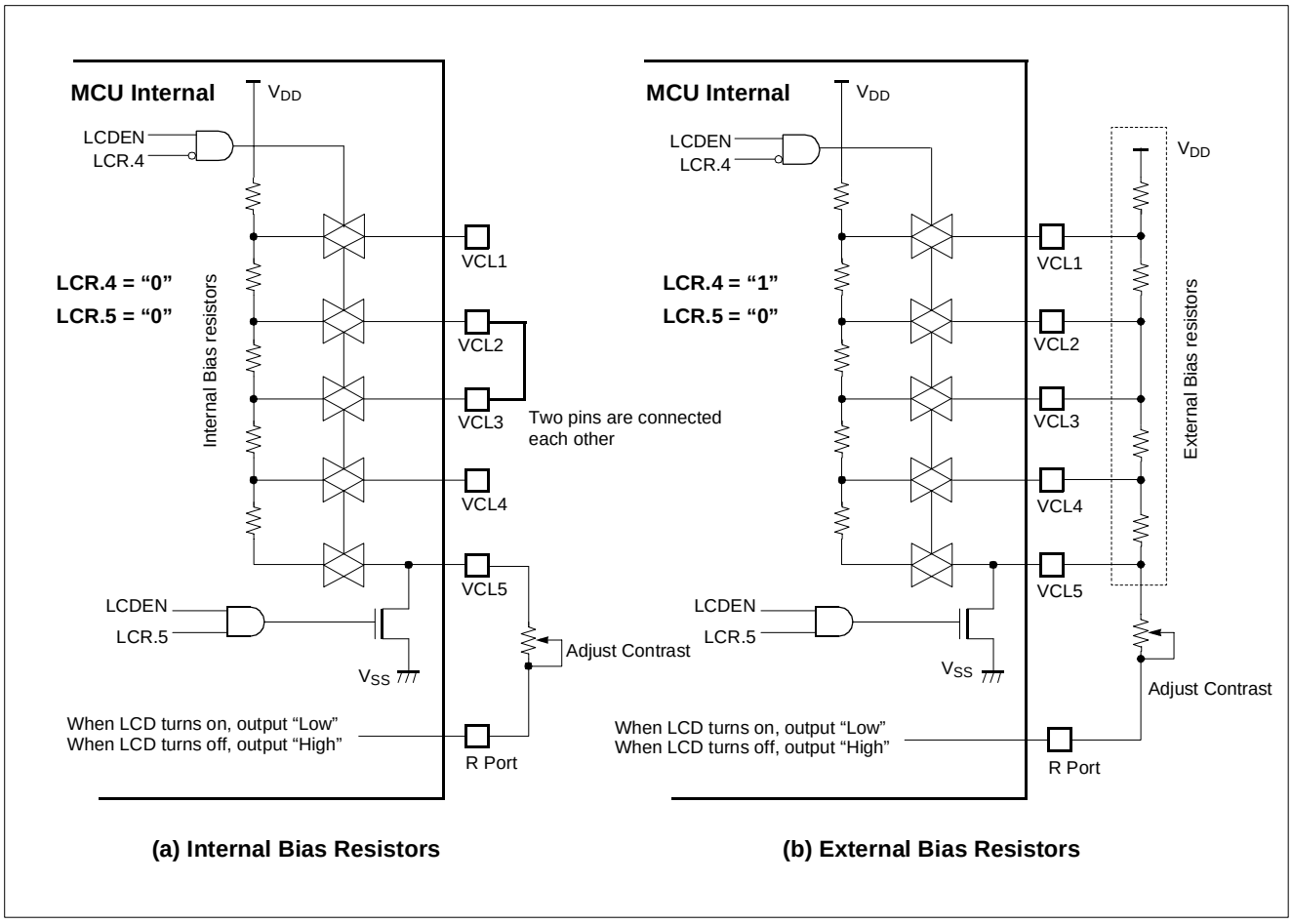


Figure 15-3 Application Example of Adjusting the Contrast

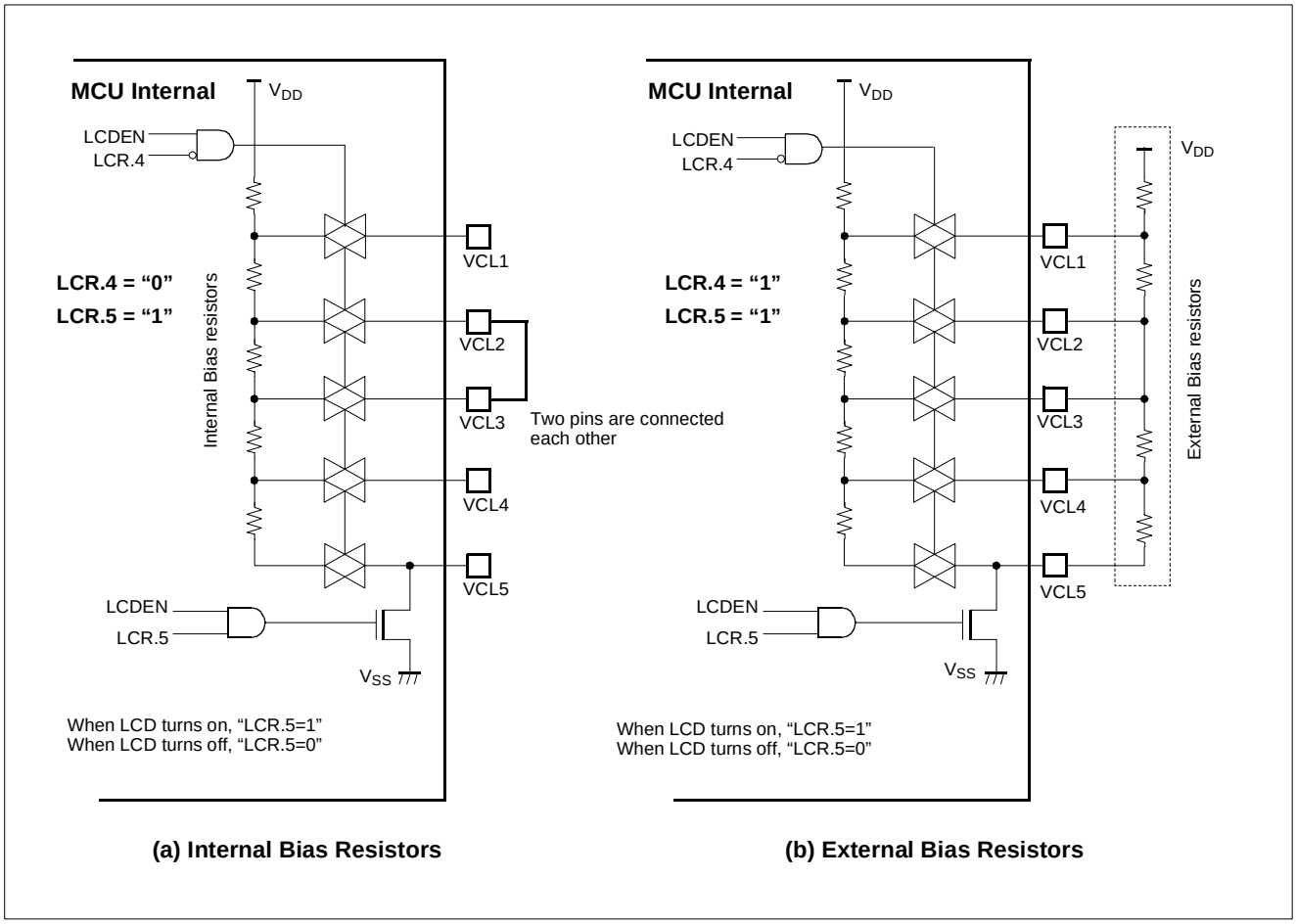


Figure 15-4 Application Example for No Adjusting of Contrast

15.4 LCD Display Memory

Display data are stored to the display data area (page 12) in the data memory.
The display data stored to the display data area (address $0C00_H$-$0C47_H$) are read automatically and sent to the LCD driver by the hardware. The LCD driver generates the segment signals and common signals in accordance with the display data and drive method.

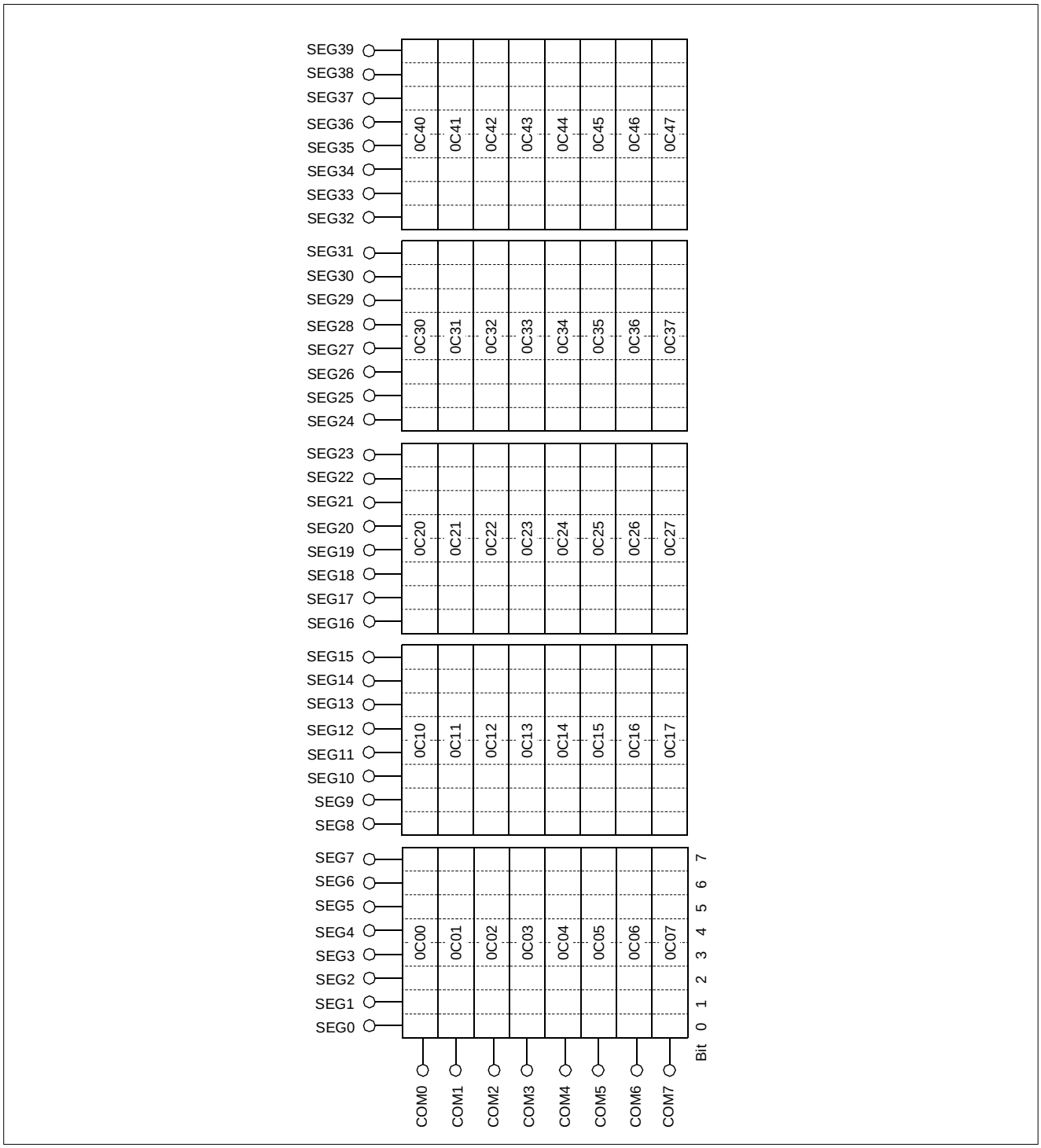


Figure 15-5 LCD Display Memory

Therefore, display patterns can be changed by only overwriting the contents of the display data area with a program. The table look up instruction is mainly used for this overwriting.
Figure 15-5 shows the correspondence between the display data area and the SEG/COM pins. The LCD lights when the display data is "1" and turn off when "0".

LCD display memory in this location that are not used for LCD display can be allocated for general purpose use.

15.5 LCD Port Selection

Segment pins are also used for normal I/O pins. The LCD port selection register LPMR is used to set R*n* pin for ordinary digital input.

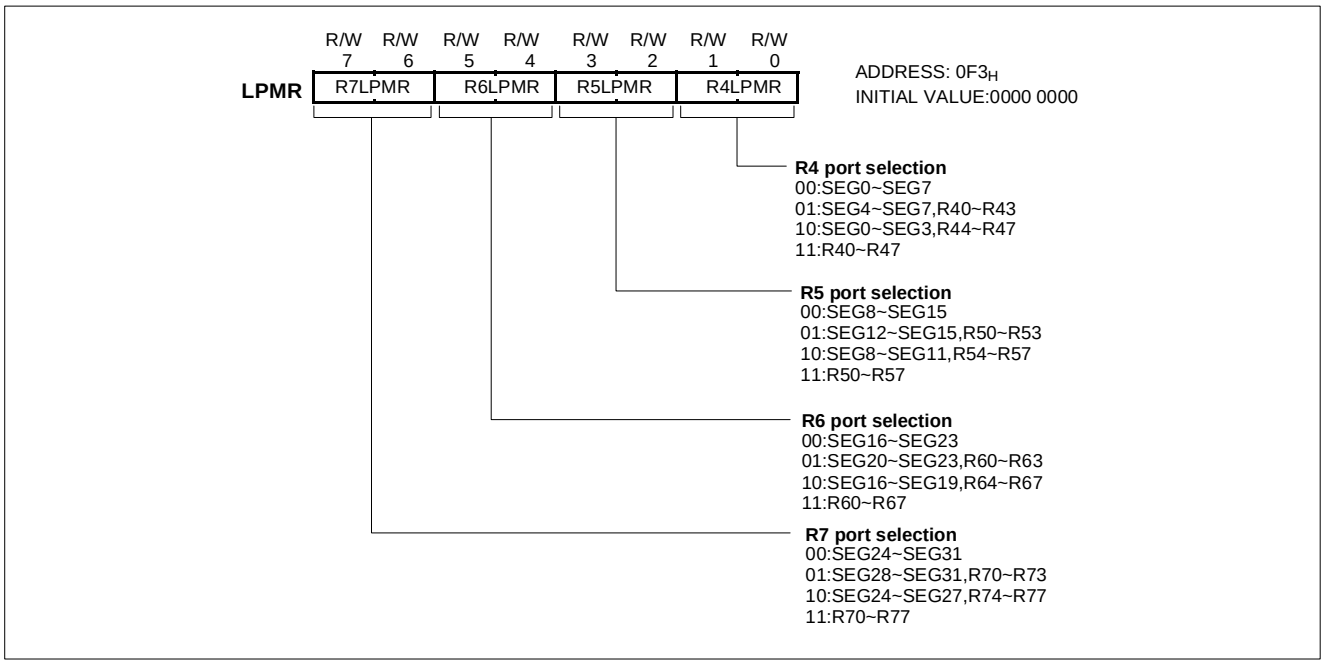


Figure 15-6 LCD Port Selection Register

15.6 Control Method of LCD Driver

Initial Setting

Flow chart of initial setting is shown in Figure 15-7 .

Example: Driving of LCD

```
Select Frame Frequency   LDM     LCR,#23H        ;fF=512Hz (fSUB= 32.768kHz)
                         :
                         SETG
                         LDM     RPR,#12         ;Select LCD Memory
                                                 ;area (Bank C)
Clear                    LDX     #0
LCD Display      C_LCD1: LDA     #0              ;RAM Clear
Memory                                           ;(0C00H->0C47H)
                         STA     {X}+
                         CMPX    #048H
                         BNE     C_LCD1
                         CLRG
                         LDM     RPR,#00         ;Bank=0
                         :
Turn on LCD              SET1B   LCR.7           ;Enable display
                         :
                         :
```

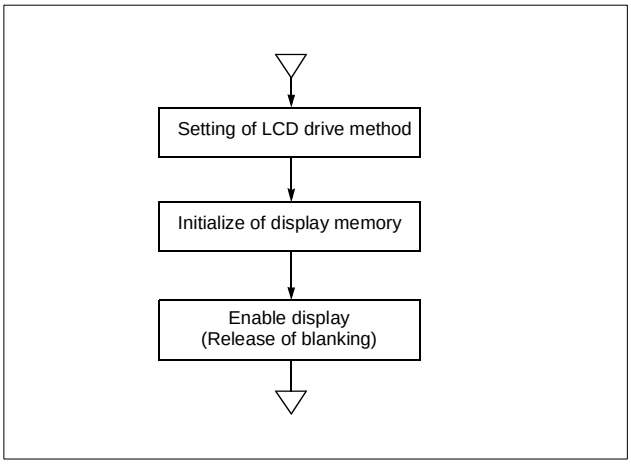


Figure 15-7 Initial Setting of LCD Driver

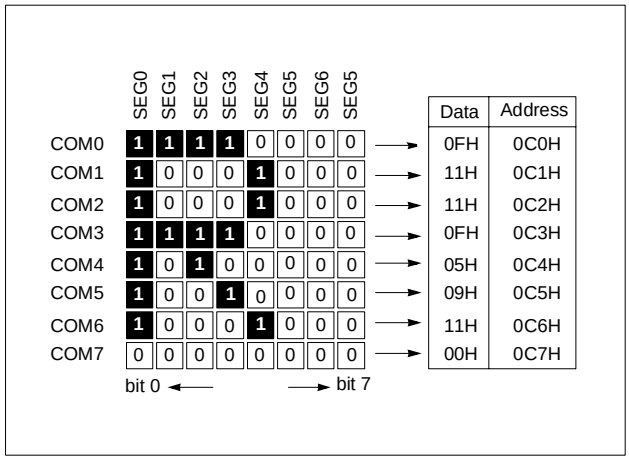


Figure 15-8 Example of Connection COM & SEG

Display Data

Normally, display data are kept permanently in the program memory and then stored at the display data area by the table look-up instruction. This can be explained using 5x7 dot matrix character display with 1/8 duty LCD as an example as well as any LCD panel. The COM and SEG connections to the LCD and display data are the same as those shown is Figure 15-8 . Programming example for displaying character is shown below.

Write into the LCD Memory

```
        :
        :
        SETG                        ;Set G-flag
        LDM     RPR,#12             ;Select Bank C to access LCD
        LDX     #0
        LDY     #0
LP:     LDA     !DTBL+Y             ;Load font data
        STA     {X}+
        INC     Y
        CMPY    #8
        BNE     LP
        CLRG                        ;Clear G-flag
        :
        :
```

Font data

```
DTBL:   DB      0FH,11H,11H,0FH
        DB      05H,09H,11H,00H
```

15.7 LCD Waveform

1/8 Duty, 1/4 Bias Drive

In this case, VCL2 and VCL3 pins are shorted each other (VCL2=VCL3) to use as 1/4 bias.

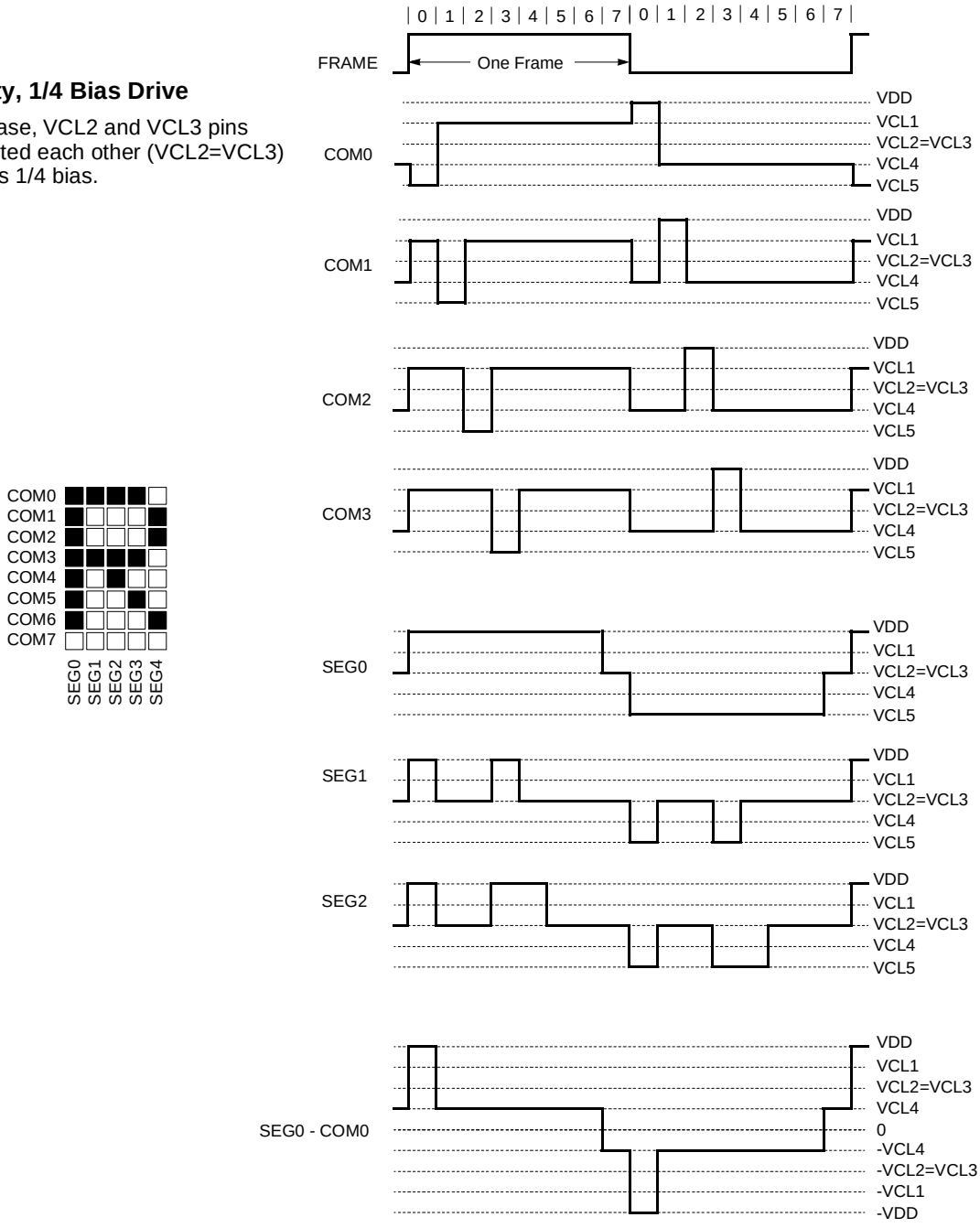


Figure 15-9 Example of LCD drive output

1/8 Duty, 1/4 Bias Drive

In this case, VCL2 and VCL3 pins are shorted each other (VCL2=VCL3) to use as 1/4 bias.

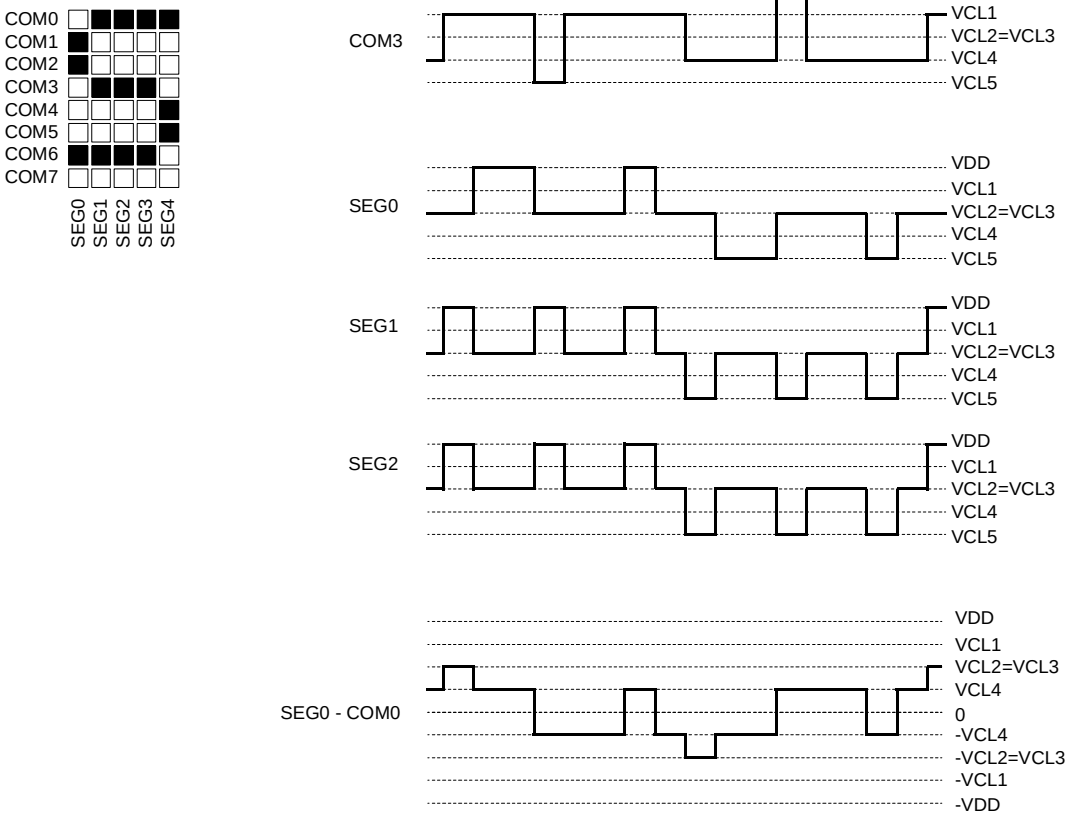


Figure 15-10 Example of LCD drive output

16. WATCHDOG TIMER

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or the like, and resumes the CPU to the normal state.
The watchdog timer signal for detecting malfunction can be selected either a reset CPU or a interrupt request.

When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.

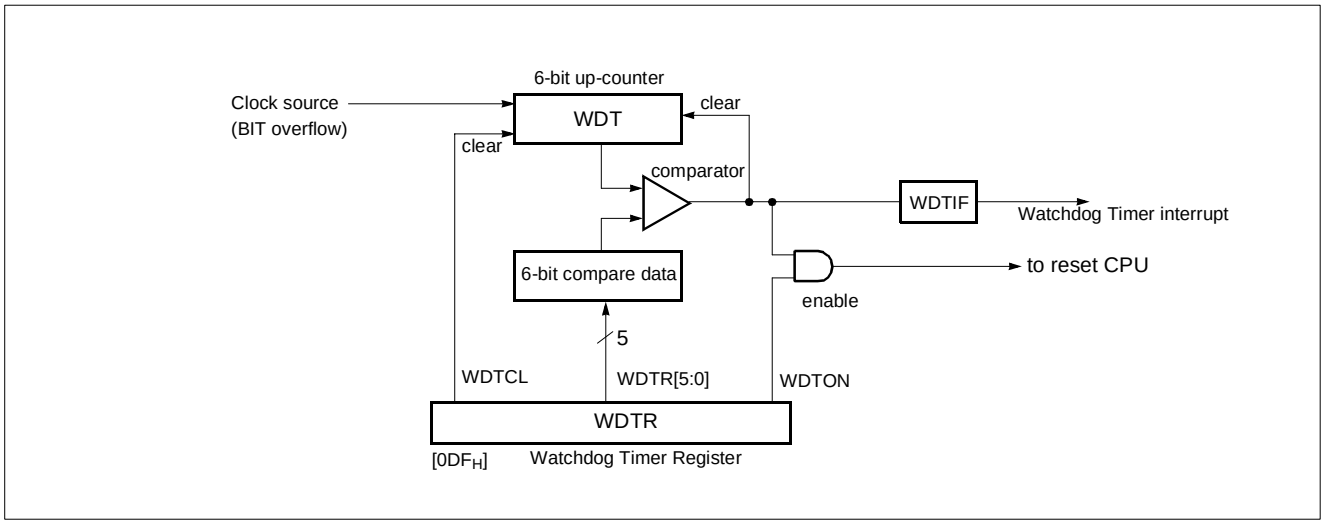


Figure 16-1 Block Diagram of Watchdog Timer

Watchdog Timer Control

Figure 16-2 shows the watchdog timer control register. The watchdog timer is automatically disabled after reset.

The CPU malfunction is detected during setting of the detection time, selecting of output, and clearing of the binary counter. Clearing the binary counter is repeated within the detection time.

If the malfunction occurs for any cause, the watchdog timer output will become active at the rising overflow from the binary counters unless the binary counter is cleared. At this time, when WDTON=1, a reset is generated, which drives the $\overline{\text{RESET}}$ pin to low to reset the internal hardware. When WDTON=0, a watchdog timer interrupt (WDTIF) is generated.

The watchdog timer temporarily stops counting in the STOP mode, and when the STOP mode is released, it automatically restarts (continues counting).

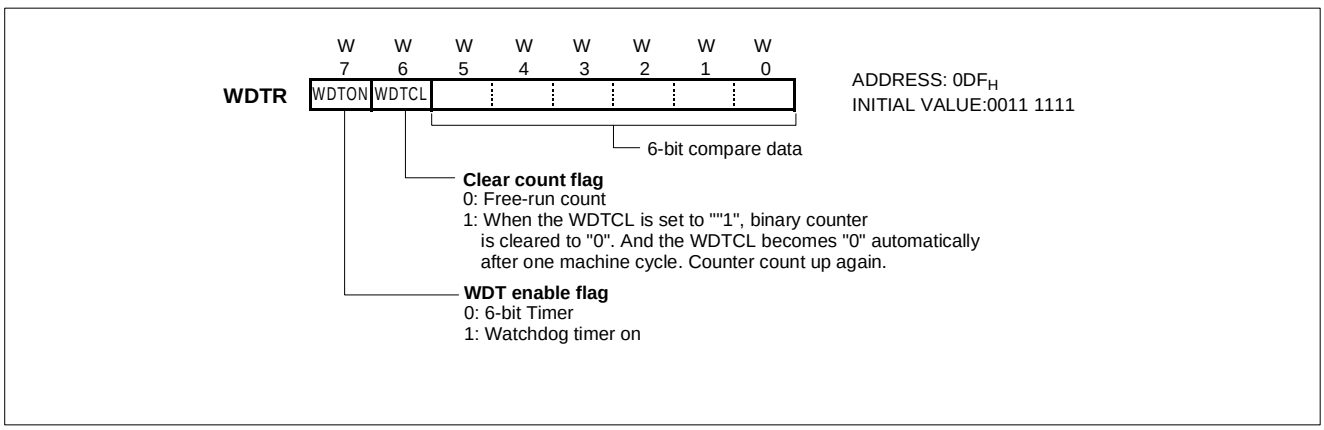


Figure 16-2 WDTR: Watchdog Timer Data Register

Example: Sets the watchdog timer detection time to 0.5 sec at 4.19MHz

```
        LDM   CKCTLR,#0EH          ;Select 1/512 clock source
        LDM   WDTR,#0CFH           ;WDTON ← 1, Clear Counter

        LDM   WDTR,#0CFH           ;Clear counter
        :
Within WDT
detection time
        :
        LDM   WDTR,#0CFH           ;Clear counter
        :
Within WDT
detection time
        :
        LDM   WDTR,#0CFH           ;Clear counter
```

Enable and Disable Watchdog

Watchdog timer is enabled by setting WDTON (bit 7 in WDTR) to "1". WDTON is initialized to "0" during reset and it should be set to "1" to operate after reset is released.

Example: Enables watchdog timer reset

```
:
LDM    WDTR,#0FFH  ;WDTON ← 1
:
:
```

The watchdog timer is disabled by clearing bit 7 (WDTON) of WDTR. The watchdog timer is halted in STOP mode and restarts automatically after STOP mode is released.

Watchdog Timer Interrupt

The watchdog timer can be also used as a simple 6-bit timer by clearing bit 7 of WDTR to "0". The interval of watchdog timer interrupt is decided by Basic Interval Timer.

Interval equation is shown as below.

$$T = WDTR \times Interval\ of\ BIT$$

The stack pointer (SP) should be initialized before using the watchdog timer output as an interrupt source.

Example: 6-bit timer interrupt set up.

```
LDX    #03FH
TXSP                ;SP ← 3F
LDM    WDTR,#3FH    ;WDTON ← 0
:                   ;WDTCL ← 0

:
```

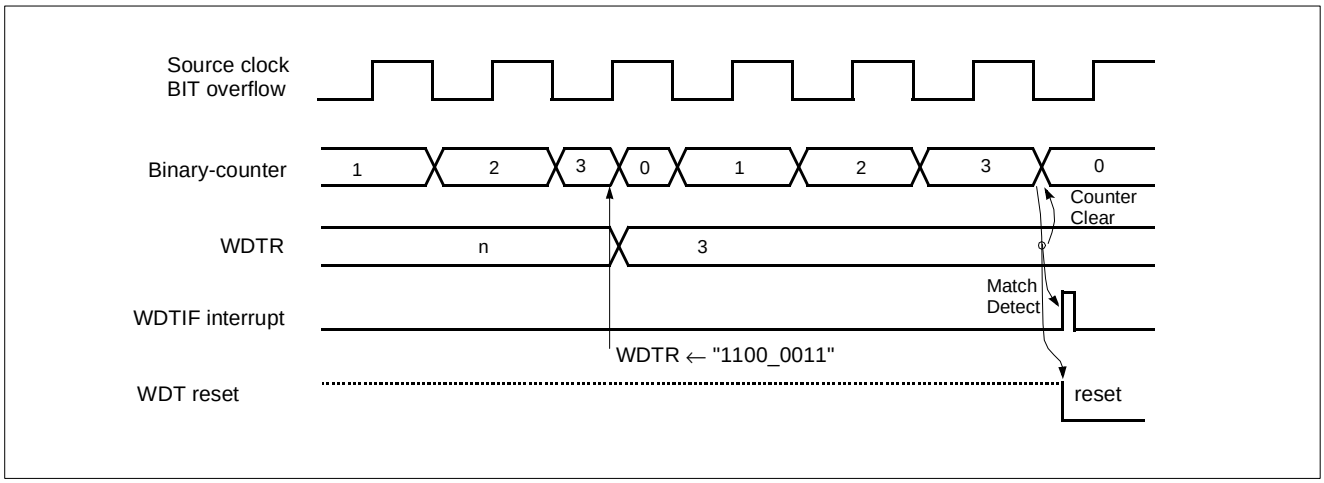


Figure 16-3 Watchdog timer Timing

If the watchdog timer output becomes active, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the internal hardware.

The main clock oscillator also turns on when a watchdog timer reset is generated in sub clock mode.

17. BUZZER DRIVER

The buzzer driver block consists of 6-bit binary counter, buzzer register, and clock source selector. It generates square-wave which has very wide range frequency (500Hz ~ 125kHz at f_{MAIN} = 4MHz) by user software.

A 50% duty pulse can be output to R13/BUZ pin to use for piezo-electric buzzer drive.

The port mode register PMR1 (address $0F6_H$) should be set to "1", the R13 will be configured as BUZ pin regardless of port direction register R1DD.

The frequency of output signal is controlled by the buzzer control register BUR.

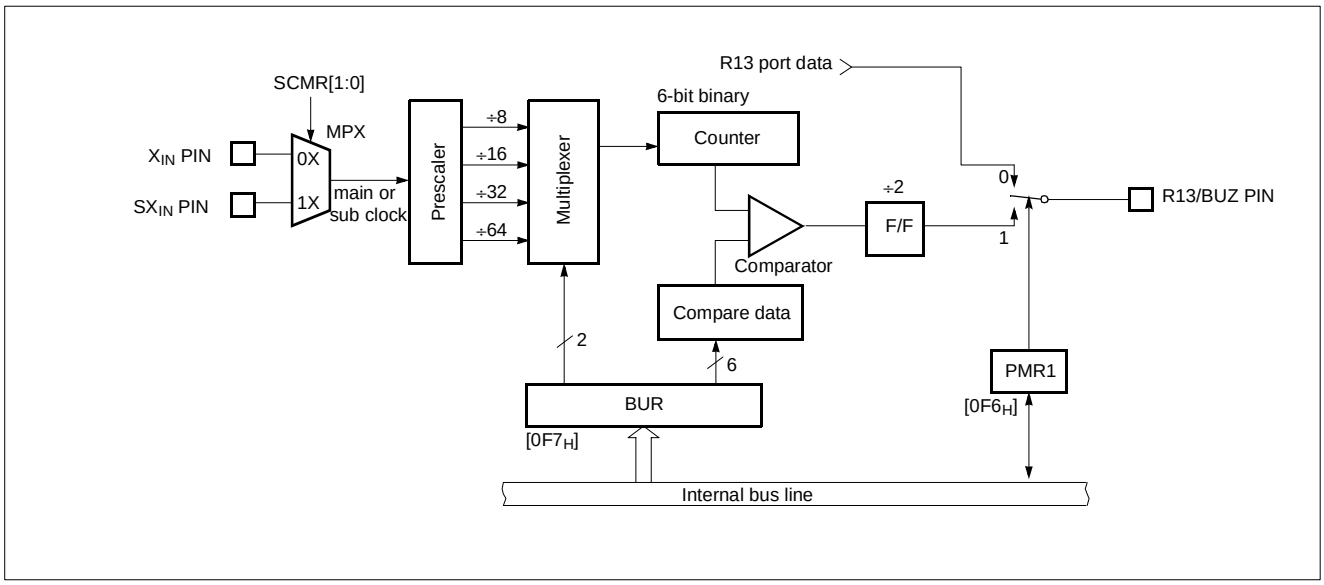


Figure 17-1 Block Diagram of Buzzer Driver

The bit 0 to bit 5 of BUR determine output frequency for buzzer driving.

The 6-bit counter is cleared and starts the counting by writing signal at BUR register. It is incremental from 00H until it matches 6-bit BUR value.

BUR is undefined after reset, so it must be initialized to between 1_H and $3F_H$ by software.

f_{BUZ}: BUZ pin frequency
Prescaler ratio: Prescaler divide ratio by BUR[7:6]
BUR value: 6-bit compare data, BUR[5:0]

Note that BUR is a write-only register.

Equation of frequency calculation is shown below.

$$f_{BUZ}(Hz) = \frac{Oscillator\ \ frequency}{2 \times Prescaler\ ratio \times (BUR\ value + 1)}$$

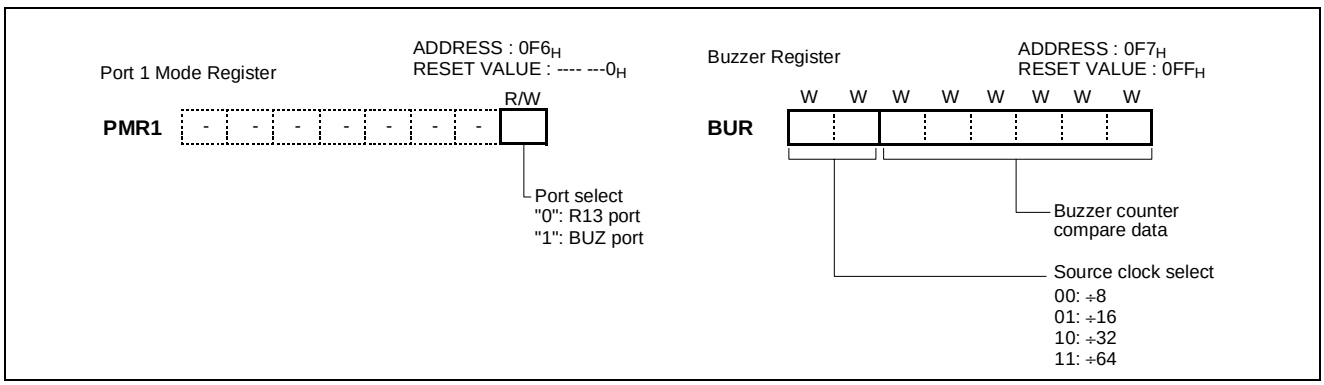


Figure 17-2 PMR1 and Buzzer Register

When main-frequency is 4.194304MHz, buzzer frequency is shown as below and if sub-frequency is selected as clock source, buzzer frequency is used after dividing by 128.

BUR [5:0]	Output frequency (kHz)			
	÷8	÷16	÷32	÷64
01	131.072	65.536	32.768	16.384
02	87.381	43.691	21.846	10.923
03	65.536	32.768	16.384	8.192
04	52.429	26.214	13.107	6.554
05	43.691	21.845	10.923	5.462
06	37.338	18.725	9.362	4.682
07	32.768	16.384	8.192	4.096
08	29.127	14.564	7.282	3.641
09	26.214	13.107	6.554	3.277
0A	23.831	11.916	5.958	2.979
0B	21.845	10.923	5.462	2.731
0C	20.165	10.082	5.041	2.521
0D	18.725	9.362	4.681	2.341
0E	17.476	8.738	4.369	2.185
0F	16.384	8.192	4.096	2.048
10	15.420	7.710	3.855	1.928
11	14.564	7.282	3.641	1.821
12	13.797	6.899	3.450	1.725
13	13.107	6.554	3.277	1.639
14	12.483	6.242	3.121	1.561
15	11.916	5.958	2.979	1.490
16	11.398	5.699	2.850	1.425
17	10.923	5.461	2.731	1.366
18	10.486	5.243	2.622	1.311
19	10.082	5.041	2.251	1.261
1A	9.709	4.855	2.428	1.214
1B	9.362	4.681	2.341	1.171
1C	9.039	4.520	2.260	1.130
1D	8.738	4.369	2.185	1.093
1E	8.456	4.228	2.114	1.057
1F	8.192	4.096	2.048	1.024

BUR [5:0]	Output frequency (kHz)			
	÷8	÷16	÷32	÷64
20	7.944	3.972	1.986	0.993
21	7.710	3.855	1.928	0.964
22	7.490	3.745	1.873	0.936
23	7.282	3.641	1.821	0.910
24	7.085	3.542	1.771	0.885
25	6.899	3.449	1.725	0.862
26	6.722	3.361	1.681	0.840
27	6.554	3.277	1.639	0.819
28	6.394	3.197	1.599	0.799
29	6.242	3.121	1.561	0.780
2A	6.096	3.048	1.524	0.762
2B	5.958	2.979	1.490	0.745
2C	5.825	2.913	1.457	0.728
2D	5.699	2.849	1.425	0.712
2E	5.578	2.789	1.395	0.697
2F	5.461	2.731	1.366	0.683
30	5.350	2.675	1.338	0.669
31	5.243	2.621	1.311	0.655
32	5.140	2.570	1.285	0.642
33	5.041	2.521	1.261	0.630
34	4.946	2.473	1.237	0.618
35	4.855	2.427	1.214	0.607
36	4.766	2.383	1.192	0.596
37	4.681	2.341	1.171	0.585
38	4.599	2.300	1.150	0.575
39	4.520	2.260	1.130	0.565
3A	4.443	2.222	1.111	0.555
3B	4.369	2.185	1.093	0.546
3C	4.297	2.149	1.075	0.537
3D	4.228	2.114	1.057	0.528
3E	4.161	2.081	1.041	0.520
3F	4.069	2.048	1.024	0.512

18. POWER DOWN OPERATION

GMS81C3004 has 2 power-down mode. In power-down mode, power consumption is reduced considerably that in Battery operation Battery life can be extended a lot.

Sleep mode is entered by setting bit 0 of Sleep Mode Register, and STOP Mode is entered by STOP instruction.

18.1 SLEEP Mode

In this mode, the internal oscillation circuits remain active.

Oscillation continues and peripherals are operate normally but CPU stops. Movement of all Peripherals is shown in Table 18-1. Sleep mode is entered by setting bit 0 of SMR (address $0DE_H$).

It is released by RESET or interrupt. To be release by interrupt, interrupt should be enabled before Sleep mode.

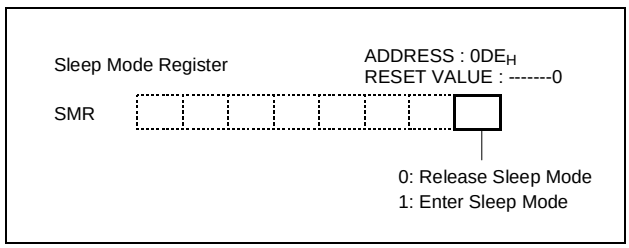


Figure 18-1 SLEEP Mode Register

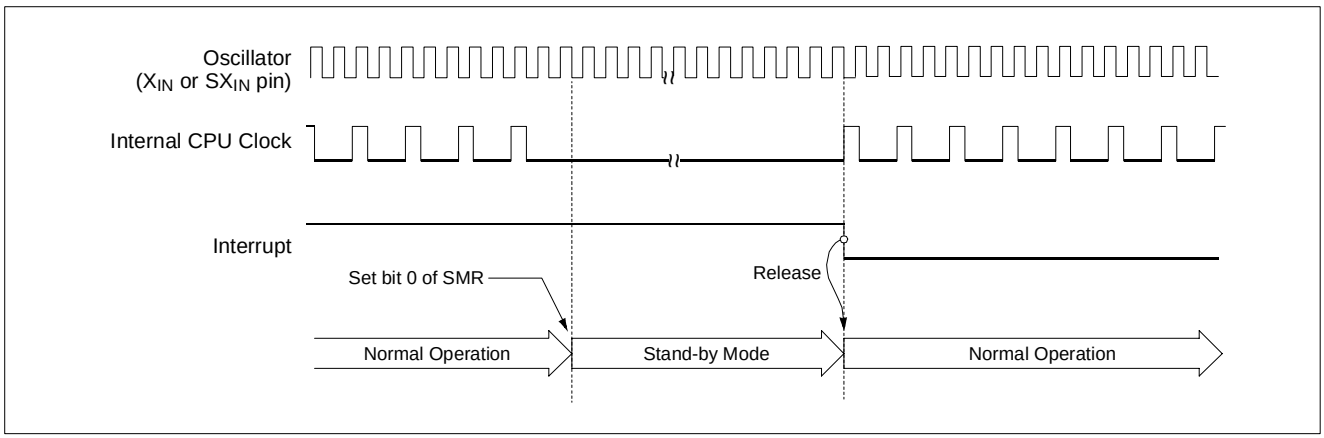


Figure 18-2 Sleep Mode Release Timing by External Interrupt

.

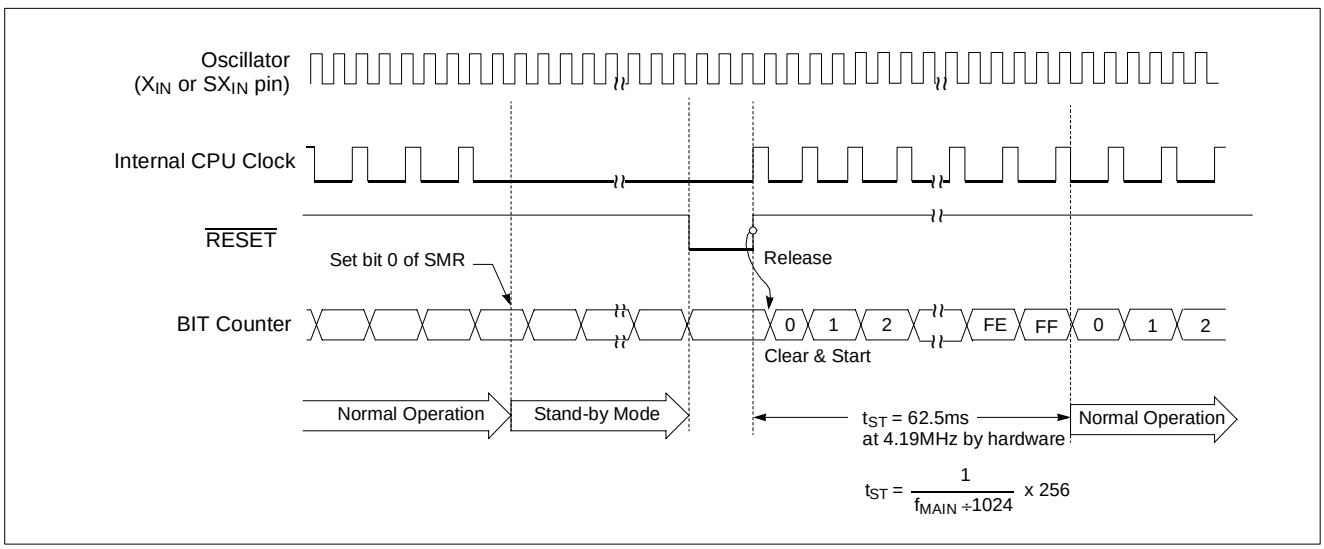


Figure 18-3 SLEEP Mode Release Timing by RESET pin

18.2 STOP Mode

For applications where power consumption is a critical factor, device provides reduced power of STOP.

Start The Stop Operation

An instruction that STOP causes to be the last instruction is executed before going into the STOP mode. In the Stop mode, the on-chip main-frequency oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Control registers are held. The port pins output the values held by their respective port data register, the port direction registers. The status of peripherals during Stop mode is shown below.

Peripheral	STOP Mode	Sleep Mode
CPU	All CPU operations are disabled	All CPU operations are disabled
RAM	Retain	Retain
LCD driver	LCDdriver operates continuously	LCD driver operates continuously
Basic Interval Timer	Halted	BIT operates continuously
Timer/Event counter 1	Halted (Only when the Event counter mode is enabled, Timer 1 operates normally)	Timer/Event counter 1 operates continuously
Watch Timer	Watch Timer operates continuously	Watch Timer operates continuously
Key Scan	Active	Active
X_{IN} PIN	LOW	Oscillation
X_{OUT} PIN	LOW	Oscillation
Main-oscillation	Stop	Oscillation
Sub-oscillation	Oscillation	Oscillation
I/O ports	Retain	Retain
Control Registers	Retain	Retain
Release method	by RESET, by Key Scan interrupt, Watch Timer interrupt, Timer interrupt ($\overline{EC1}$), by External interrupt	by RESET, All interrupts

Table 18-1 Peripheral Operation during Power Down Mode

Note: *Since the X_{IN} pin is connected internally to GND to avoid current leakage due to the crystal oscillator in STOP mode, do not use STOP instruction when an external clock is used as the main system clock.*

In the Stop mode of operation, V_{DD} can be reduced to minimize power consumption. Be careful, however, that V_{DD} is not reduced before the Stop mode is invoked, and that V_{DD} is restored to its normal operating level before the Stop mode is terminated.

The reset should not be activated before V_{DD} is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize.
And after STOP instruction, at least two or more NOP instruction should be written as shown in example below.

Example)

```
:
LDM    CKCTLR,#0000_1110B
STOP
NOP
NOP
:
```

The Interval Timer Register CKCTLR should be initialized ($0F_H$ or $0E_H$) by software in order that oscillation stabilization time should be longer than 20ms before STOP mode.

Release the STOP mode

The exit from STOP mode is using hardware reset or external interrupt, watch timer, key scan or timer/counter.

To release STOP mode, corresponding interrupt should be enabled before STOP mode.
Specially as a clock source of Timer/Event counter, $\overline{EC1}$ pin can release it by Timer/Event counter Interrupt re-

quest.

Reset redefines all the control registers but does not change the on-chip RAM. External interrupts allow both on-chip RAM and Control registers to retain their values.

Start-up is performed to acquire the time for stabilizing oscillation. During the start-up, the internal operations are all stopped.

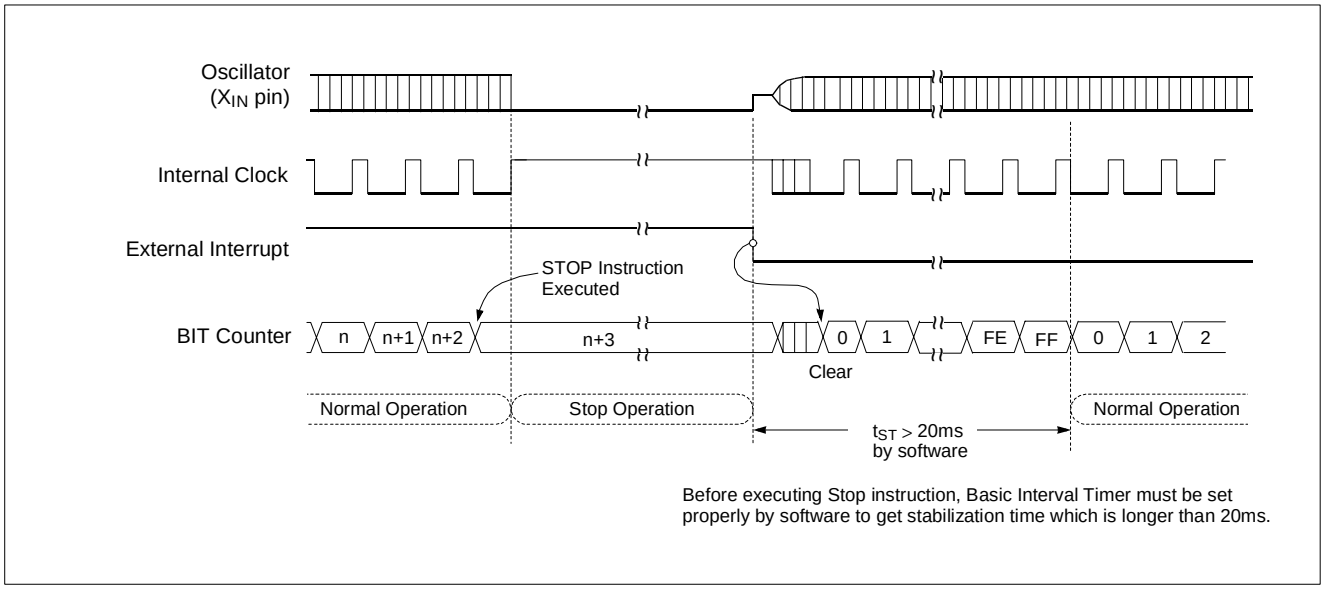


Figure 18-4 STOP Mode Release Timing by External Interrupt

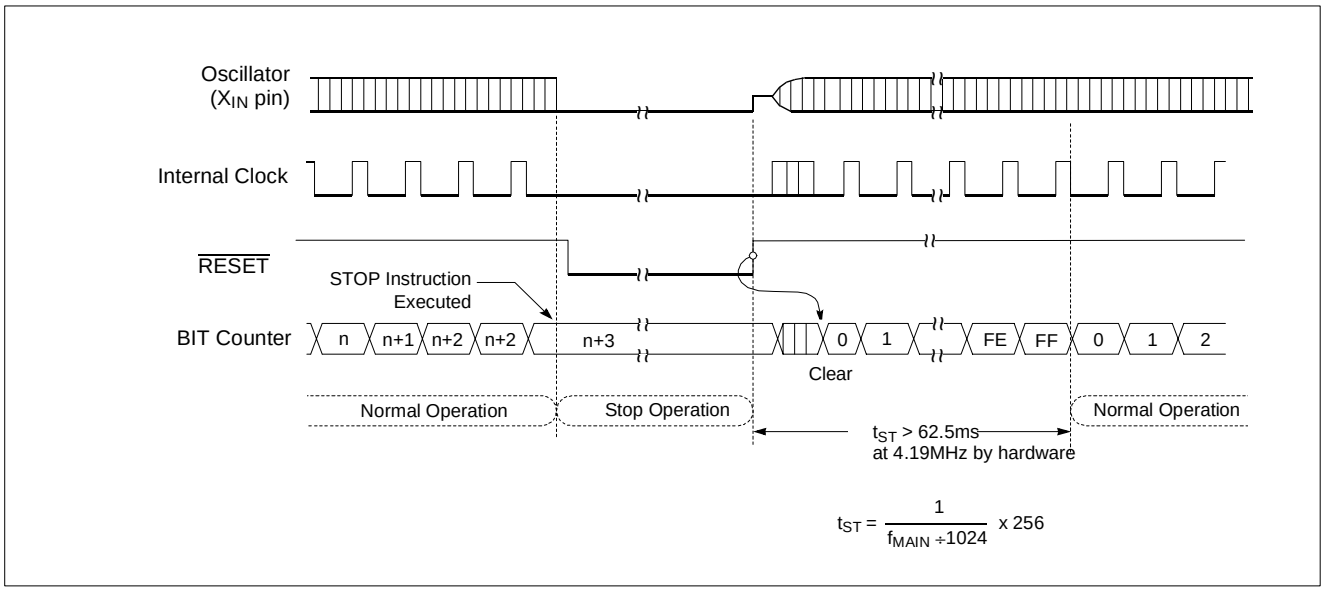


Figure 18-5 STOP Mode Release Timing by RESET

Minimizing Current Consumption

The Stop mode is designed to reduce power consumption. To minimize current drawn during Stop mode, the user should turn-off output drivers that are sourcing or sinking current, if it is practical.

Note: *In the STOP operation, the power dissipation associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the STOP feature. This point should be little current flows when the input level is stable at the power voltage level (V_{DD}/V_{SS}); however, when the input level becomes higher than the power voltage level (by approximately 0.3V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port puts the pin signal into the high-impedance state, a current flow across the ports input transistor, requiring it to fix the level by pull-up or other means.*

It should be set properly that current flow through port doesn't exist.

First consider the setting to 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 voltage level (not V_{SS}or V_{DD}) is applied to input pin, there can be little current (max. 1mA at around 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.

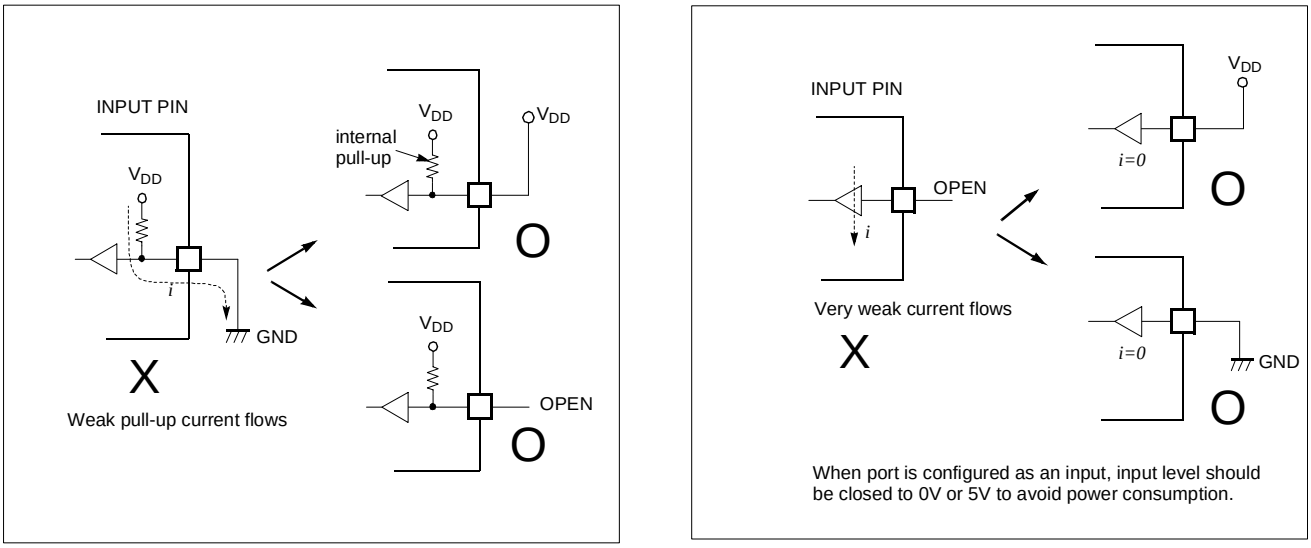


Figure 18-6 Application Example of Unused Input Port

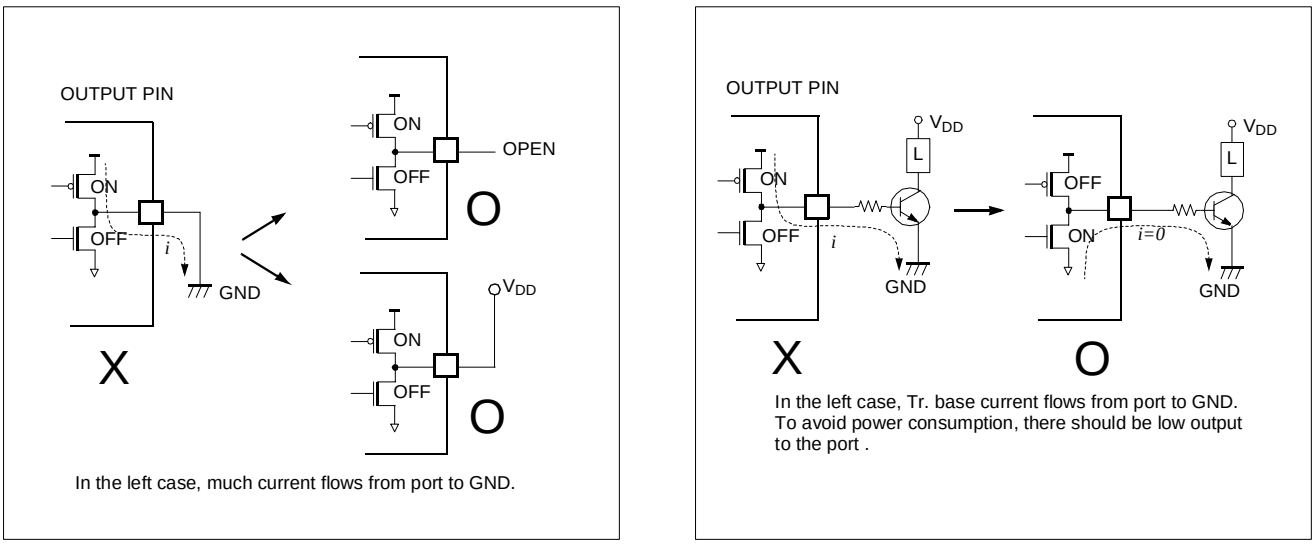


Figure 18-7 Application Example of Unused Output Port

19. OSCILLATOR CIRCUIT

The GMS81C3004 has two oscillation circuits internally. X_{IN} and X_{OUT} are input and output for main frequency and SX_{IN} and SX_{OUT} are input and output for sub frequency, respectively, inverting amplifier which can be configured for being used as an on-chip oscillator, as shown in Figure 19-1 .

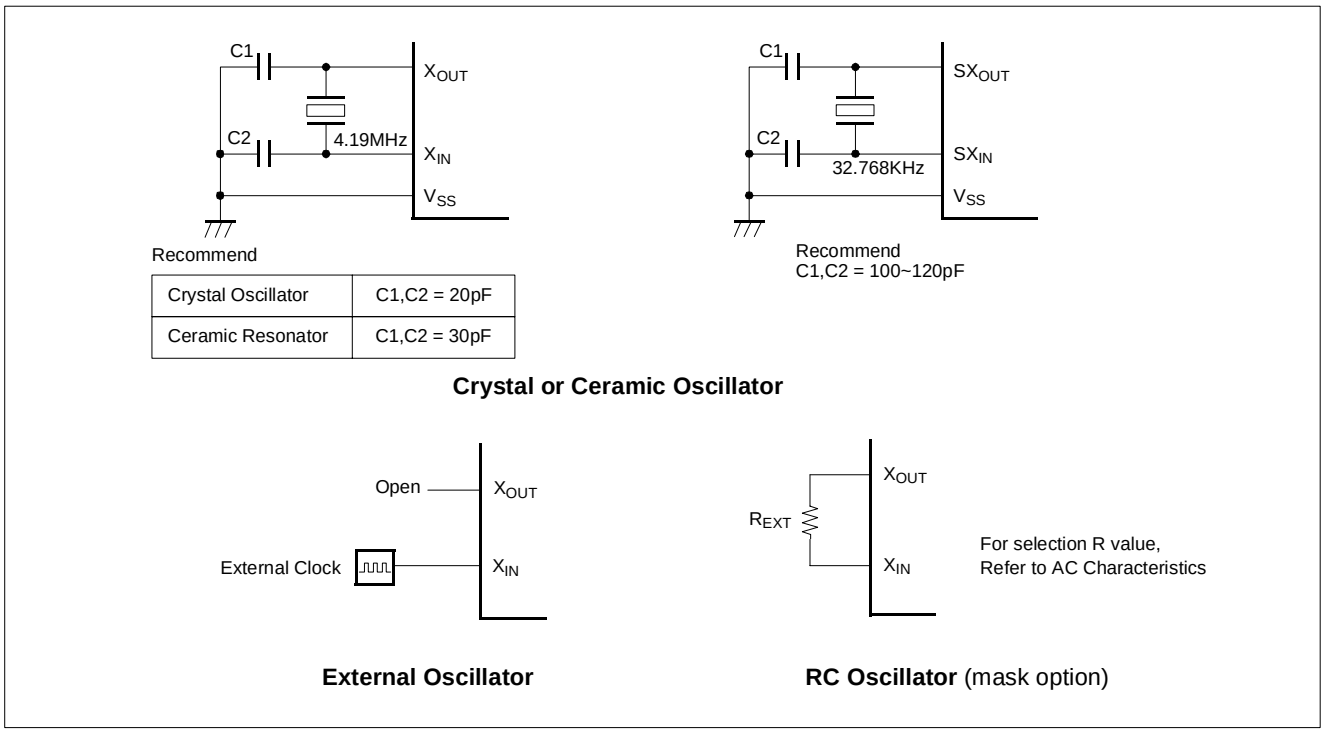


Figure 19-1 Oscillation Circuit

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Since each crystal and ceramic resonator have their own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Since each crystal and ceramic resonator have their own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

In addition, see Figure 19-2 for the layout of the crystal.

Note: *Minimize the wiring length. Do not allow the wiring to intersect with other signal conductors. Do not allow the wiring to come near changing high current. Set the potential of the grounding position of the oscillator capacitor to that of Vss. Do not ground it to any ground pattern where high current is present. Do not fetch signals from the oscillator.*

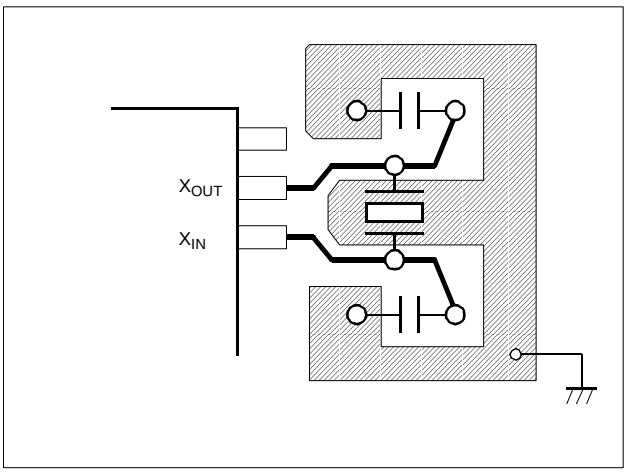


Figure 19-2 Layout of Oscillator PCB circuit

20. RESET

The GMS81C3004 have two types of reset generation procedures; one is an external reset input, the other is a watchdog timer reset. Table 20-1 shows on-chip hardware initialization by reset action.

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	($FFFF_H$) - ($FFFE_H$)	Peripheral clock	Off
RAM page register (RPR)	0	Watchdog timer	Disable
G-flag (G)	0	Control registers	Refer to Table 8-1 on page 22
Operation mode	Main-frequency clock	Power fail detector	Disable

Table 20-1 Initializing Internal Status by Reset Action

20.1 External Reset Input

The reset input is the RESET pin, which is the input to a Schmitt Trigger. A reset in accomplished by holding the RESET pin low for at least 8 oscillator periods, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset, 64ms (at 4 MHz) add with 7 oscillator periods are required to start execution as shown in Figure 20-2 .

Internal RAM is not affected by reset. When V_{DD} is turned on, the RAM content is indeterminate. Therefore, this RAM should be initialized before read or tested it.

When the RESET pin input goes to high, the reset operation is released and the program execution starts at the vector address stored at addresses $FFFE_H$ - $FFFF_H$.

A connection for simple power-on-reset is shown in Figure 20-1 .

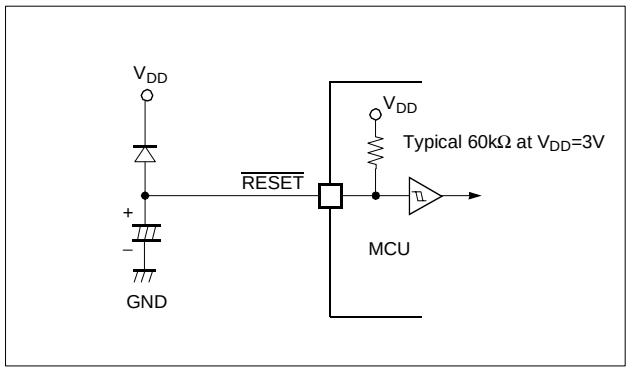


Figure 20-1 Simple Power-on-Reset Circuit

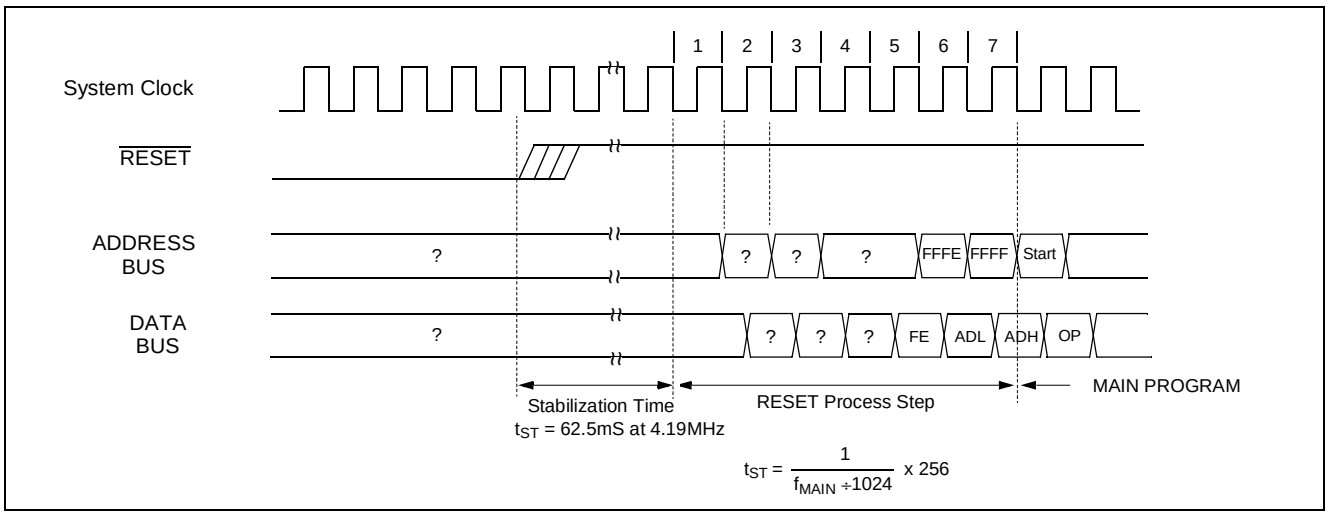


Figure 20-2 Timing Diagram after RESET

20.2 Watchdog Timer Reset

Refer to “16. WATCHDOG TIMER” on page 64.

21. POWER FAIL PROCESSOR

The GMS81C3004 has an on-chip low voltage detection circuitry to detect the V_{DD} voltage. A configuration register, LVDR, can enable or disable the low voltage detect circuitry. Whenever V_{DD} falls close to or below 3.4V, the LVD flag1, LVDF0 is just set to "1", and if it recovering 3.4V, LVDF0 is holded to "1". If V_{DD} falls below around 2.2V range, the low voltage situation may reset MCU according to setting of LVDR. Refer to “7.3 DC Electrical Characteristics” on page 10.

In the in-circuit emulator, power fail function is not implemented and user can not experiment with it. Therefore, after final development of user program, this function may be experimented or evaluated.

Note: *Power fail processor function is not available on 3V operation, because this function will detect power fail at all the time.*

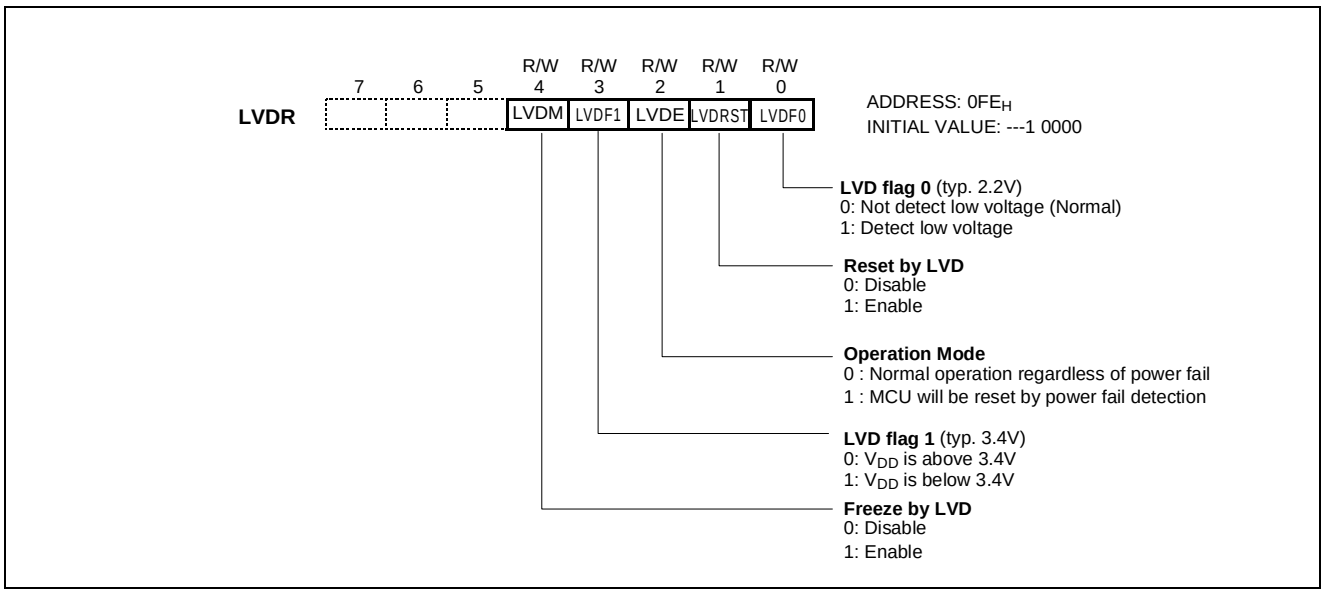


Figure 21-1 Low Voltage Detector Register

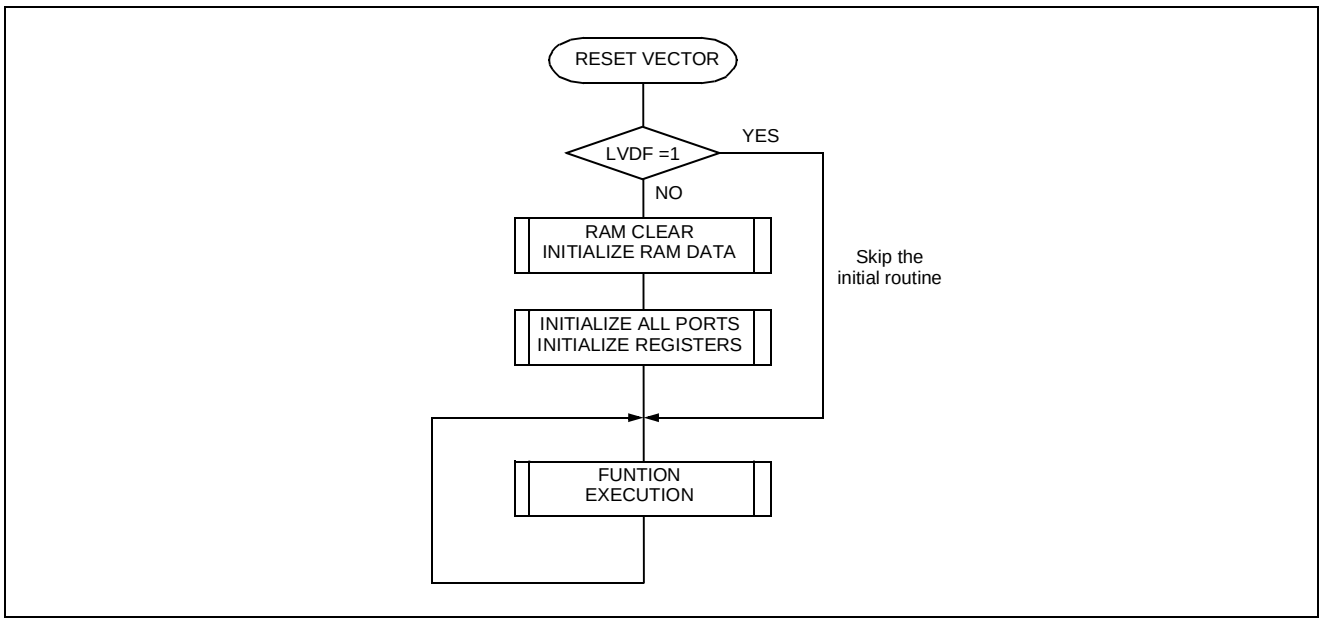


Figure 21-2 Example S/W of RESET by Power fail

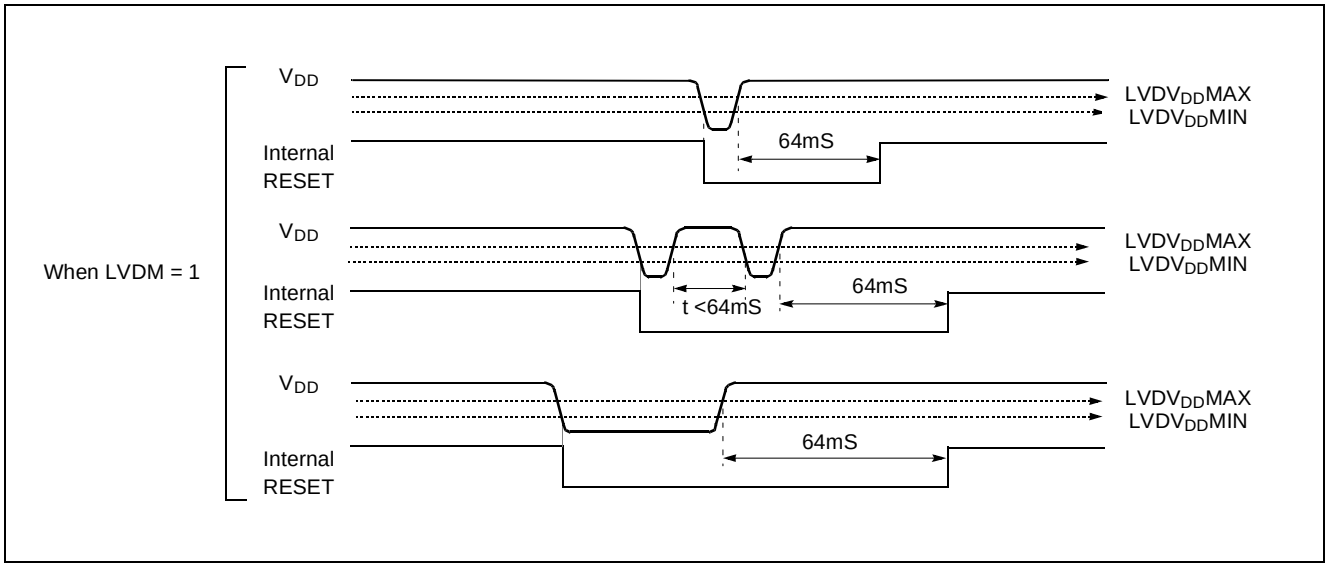


Figure 21-3 Power Fail Processor Situations

APPENDIX

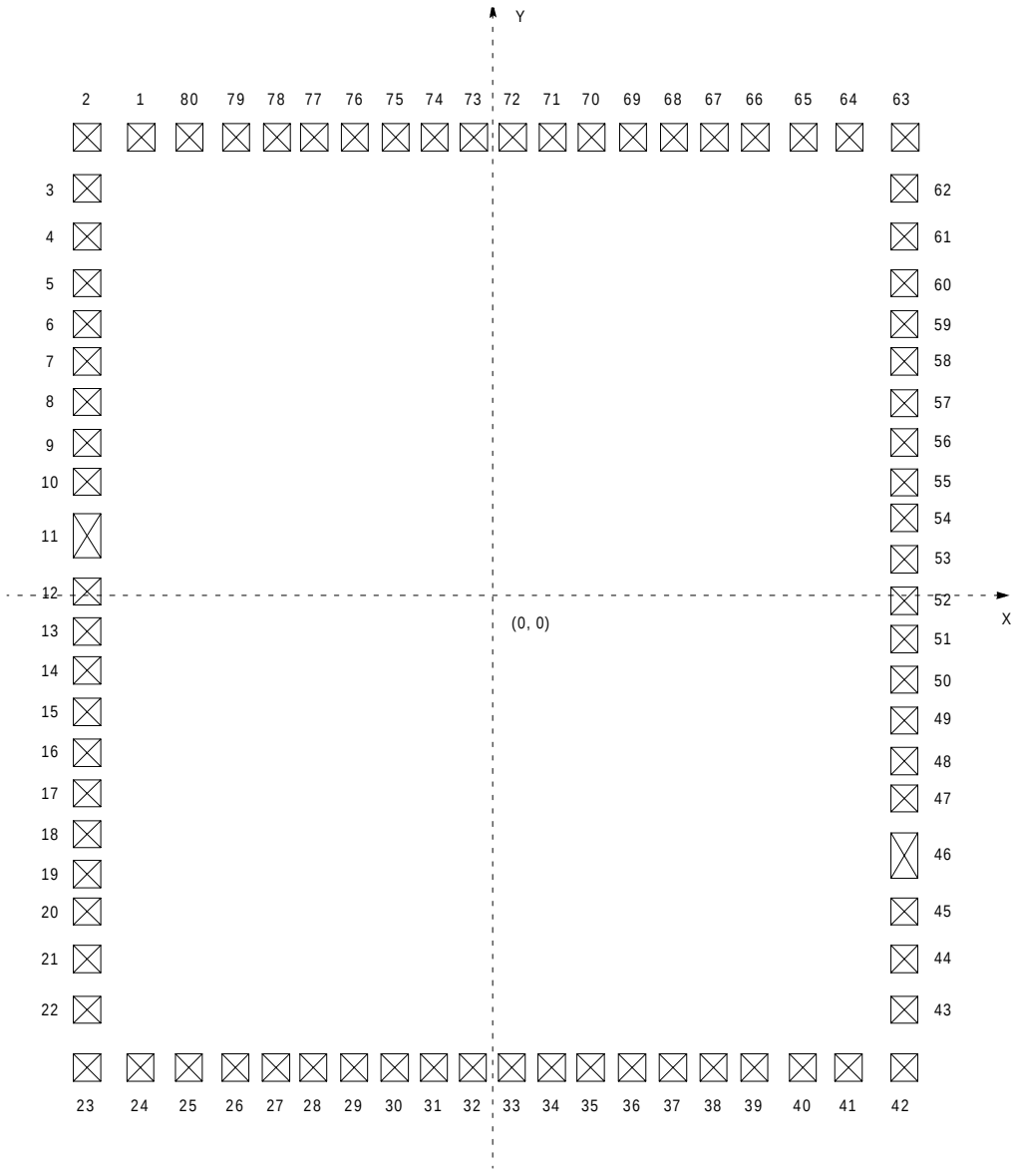
A. CONTROL REGISTER LIST

Address	Register Name	Symbol	R/W	Initial Value								Page
				7	6	5	4	3	2	1	0	
00C0	R0 port data register	R0	R/W	Undefined								30
00C1	R1 port data register	R1	R/W	Undefined								30
00C2	R2 port data register	R2	R/W	Undefined								31
00C4	R4 port data register	R4	R/W	Undefined								31
00C5	R5 port data register	R5	R/W	Undefined								31
00C6	R6 port data register	R6	R/W	Undefined								32
00C7	R7 port data register	R7	R/W	Undefined								32
00C8	R0 port I/O direction register	R0DD	W	0	0	0	0	0	0	0	0	30
00C9	R1 port I/O direction register	R1DD	W	0	0	0	0	0	0	0	0	30
00CA	R2 port I/O direction register	R2DD	W	-	-	-	-	-	0	0	0	31
00CC	R4 port I/O direction register	R4DD	W	0	0	0	0	0	0	0	0	31
00CD	R5 port I/O direction register	R5DD	W	0	0	0	0	0	0	0	0	31
00CE	R6 port I/O direction register	R6DD	W	0	0	0	0	0	0	0	0	32
00CF	R7 port I/O direction register	R7DD	W	0	0	0	0	0	0	0	0	32
00D4	R0 port pull-up resistor selection register	PUR0	W	0	0	0	0	0	0	0	0	30
00D5	R1 port pull-up resistor selection register	PUR1	W	0	0	0	0	0	0	0	0	30
00D6	R2 port pull-up resistor selection register	PUR2	W	-	-	-	-	-	0	0	0	31
00D8	External Interrupt Edge selection register	IESR	W	-	-	0	0	0	0	0	0	51
00D9	R0 port mode register	PMR0	W	-	-	-	-	0	0	0	0	51
00DA	Interrupt enable low register	IENL	R/W	-	-	-	0	-	-	-	0	47
00DB	Interrupt enable high register	IENH	R/W	-	-	0	0	-	0	0	0	47
00DC	Interrupt request flag low register	IRQL	R/W	-	-	-	0	-	-	-	0	47
00DD	Interrupt request flag high register	IRQH	R/W	-	-	0	0	-	0	0	0	47
00DE	Sleep mode register	SMR	W	-	-	-	-	-	-	-	0	68
00DF	Watchdog Timer Register	WDTR	W	0	0	1	1	1	1	1	1	64
00E4	Timer 1 mode register	TM1	R/W	-	-	-	0	0	0	0	0	40
00E5	Timer 1 count register	T1	R	0	0	0	0	0	0	0	0	40
	Timer 1 data register	TDR1	W	Undefined								40
00EC	Comparator mode register	CMR	W	0	0	-	0	0	0	0	0	45
00ED	Comparator channel selection register	CSR	W	0	-	-	-	-	-	0	0	45
00F0	Watch timer mode register	WTMR	W	-	-	-	-	0	0	0	0	43
00F1	LCD control register	LCR	R/W	0	-	0	0	-	0	0	0	56
00F3	LCD port selection register	LPMR	R/W	0	0	0	0	0	0	0	0	60
00F4	Key Scan control register	KSCR	R/W	0	0	0	0	0	0	0	0	54
00F6	R1 port mode register	PMR1	R/W	-	-	-	-	-	-	-	0	66
00F7	Buzzer register	BUR	W	0	0	0	0	0	0	0	0	66
00F8	RAM paging register	RPR	R/W	0	0	0	0	0	0	0	0	25
00F9	Basic interval timer mode register	BITR	R	Undefined								39
	Clock control register	CKCTLR	W	-	-	-	-	0	1	1	1	39
00FA	System clock mode register	SCMR	R/W	-	-	-	-	0	0	0	0	34
00FB	Peripheral clock control register	PCOR	W	-	-	-	-	-	-	-	0	34
00FE	LVD mode register	LVDR	R/W	-	-	-	1	0	0	0	0	75

B. PAD COORDINATION

Device	GMS81C3004
Chip size	2670μm × 2980μm
Pad Size	95μm × 95μm

B.1 Pad Layout



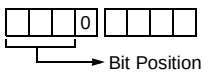
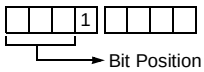
B.2 Bonding Pad Coordination

Pad No.	Pin Name	Pad Coordination (X,Y)
1	R76 / SEG30	(-1077.5, 1412.5)
2	R77 / SEG31	(-1257.5, 1412.5)
3	SEG32	(-1257.5, 1232.5)
4	SEG33	(-1257.5, 1082.5)
5	SEG34	(-1257.5, 935.0)
6	SEG35	(-1257.5, 815.0)
7	SEG36	(-1257.5, 695.0)
8	SEG37	(-1257.5, 575.0)
9	SEG38	(-1257.5, 455.0)
10	SEG39	(-1257.5, 335.0)
11	V_{SS}	(-1257.5, 180.0)
12	COM0	(-1257.5, 25.0)
13	COM1	(-1257.5, -95.0)
14	COM2	(-1257.5, -215.0)
15	COM3	(-1257.5, -335.0)
16	COM4	(-1257.5, -455.0)
17	COM5	(-1257.5, -575.0)
18	COM6	(-1257.5, -695.0)
19	COM7	(-1257.5, -815.0)
20	VCL1	(-1257.5, -935.0)
21	VCL2	(-1257.5, -1082.5)
22	VCL3	(-1257.5, -1232.5)
23	VCL4	(-1257.5, -1412.5)
24	VCL5	(-1077.5, -1412.5)
25	R10 / KS0	(-922.5, -1412.5)
26	R11 / KS1	(-780.0, -1412.5)
27	R12 / KS2	(-660.0, -1412.5)
28	R13 / KS3	(-540.0, -1412.5)
29	R14/ CMP0 / KS4	(-420.0, -1412.5)
30	R15 / CMP1 / KS5	(-300.0, -1412.5)
31	R16 / CMP2 / KS6	(-180.0, -1412.5)
32	R17 / CMP3 / KS7	(-60.0, -1412.5)
33	R00 / INT0	(60.0, -1412.5)
34	R01 / INT1	(180.0, -1412.5)
35	R02 /INT2	(300.0, -1412.5)
36	R03 / $\overline{EC1}$	(420.0, -1412.5)
37	R04	(540.0, -1412.5)
38	R05	(660.0, -1412.5)
39	R06 / LCDCK	(780.0, -1412.5)
40	R07	(922.5, -1412.5)

Pad No.	Pin Name	Pad Coordination (X,Y)
41	R20	(1077.5, -1412.5)
42	R21	(1257.5, -1412.5)
43	R22	(1257.5, -1232.5)
44	$\overline{RESET}$	(1257.5, -1082.5)
45	TEST	(1257.5, -935.0)
46	V_{DD}	(1257.5, -780.0)
47	X_{OUT}	(1257.5, -625.0)
48	X_{IN}	(1257.5, -505.0)
49	SX_{OUT}	(1257.5, -385.0)
50	SX_{IN}	(1257.5, -265.0)
51	SEG0 / R40	(1257.5, -145.0)
52	SEG1 / R41	(1257.5, -25.0)
53	SEG2 / R42	(1257.5, 95.0)
54	SEG3 / R43	(1257.5, 215.0)
55	SEG4 / R44	(1257.5, 335.0)
56	SEG5 / R45	(1257.5, 455.0)
57	SEG6 / R46	(1257.5, 575.0)
58	SEG7 / R47	(1257.5, 695.0)
59	SEG8 / R50	(1257.5, 815.0)
60	SEG9 / R51	(1257.5, 935.0)
61	SEG10 / R52	(1257.5, 1082.0)
62	SEG11 / R53	(1257.5, 1232.5)
63	SEG12 / R54	(1257.5, 1412.5)
64	SEG13 / R55	(1077.5, 1412.5)
65	SEG14 / R56	(922.5, 1412.5)
66	SEG15 / R57	(780.0, 1412.5)
67	SEG16 / R60	(660.0, 1412.5)
68	SEG17 / R61	(540.0, 1412.5)
69	SEG18 / R62	(420.0, 1412.5)
70	SEG19 / R63	(300.0, 1412.5)
71	SEG20 / R64	(180.0, 1412.5)
72	SEG21 / R65	(60.0, 1412.5)
73	SEG22 / R66	(-60.0, 1412.5)
74	SEG23 / R67	(-180.0, 1412.5)
75	SEG24 / R70	(-300.0, 1412.5)
76	SEG25 / R71	(-420.0, 1412.5)
77	SEG26 / R72	(-540.0, 1412.5)
78	SEG27 / R73	(-660.0, 1412.5)
79	SEG28 / R74	(-780.0, 1412.5)
80	SEG29 / R75	(-922.5, 1412.5)

C. INSTRUCTION

C.1 Terminology List

Terminology	Description
A	Accumulator
X	X - register
Y	Y - register
PSW	Program Status Word
#imm	8-bit Immediate data
dp	Direct Page Offset Address
!abs	Absolute Address
[ ]	Indirect expression
{ }	Register Indirect expression
{ }+	Register Indirect expression, after that, Register auto-increment
.bit	Bit Position
A.bit	Bit Position of Accumulator
dp.bit	Bit Position of Direct Page Memory
M.bit	Bit Position of Memory Data (000_H~$0FFF_H$)
rel	Relative Addressing Data
upage	U-page ($0FF00_H$~$0FFFF_H$) Offset Address
n	Table CALL Number (0~15)
+	Addition
x	 Upper Nibble Expression in Opcode
y	 Upper Nibble Expression in Opcode
–	Subtraction
×	Multiplication
/	Division
( )	Contents Expression
∧	AND
∨	OR
⊕	Exclusive OR
~	NOT
←	Assignment / Transfer / Shift Left
→	Shift Right
↔	Exchange
=	Equal
≠	Not Equal

C.2 Instruction Map

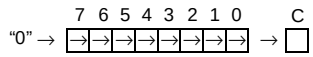
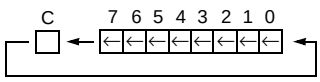
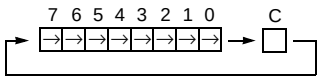
LOW / HIGH	00000 00	00001 01	00010 02	00011 03	00100 04	00101 05	00110 06	00111 07	01000 08	01001 09	01010 0A	01011 0B	01100 0C	01101 0D	01110 0E	01111 0F
000	-	SET1 dp.bit	BBS A.bit,rel	BBS dp.bit,rel	ADC #imm	ADC dp	ADC dp+X	ADC !abs	ASL A	ASL dp	TCALL 0	SETA1 .bit	BIT dp	POP A	PUSH A	BRK
001	CLRC				SBC #imm	SBC dp	SBC dp+X	SBC !abs	ROL A	ROL dp	TCALL 2	CLRA1 .bit	COM dp	POP X	PUSH X	BRA rel
010	CLRG				CMP #imm	CMP dp	CMP dp+X	CMP !abs	LSR A	LSR dp	TCALL 4	NOT1 M.bit	TST dp	POP Y	PUSH Y	PCALL Upage
011	DI				OR #imm	OR dp	OR dp+X	OR !abs	ROR A	ROR dp	TCALL 6	OR1 OR1B	CMPX dp	POP PSW	PUSH PSW	RET
100	CLRV				AND #imm	AND dp	AND dp+X	AND !abs	INC A	INC dp	TCALL 8	AND1 AND1B	CMPY dp	CBNE dp+X	TXSP	INC X
101	SETC				EOR #imm	EOR dp	EOR dp+X	EOR !abs	DEC A	DEC dp	TCALL 10	EOR1 EOR1B	DBNE dp	XMA dp+X	TSPX	DEC X
110	SETG				LDA #imm	LDA dp	LDA dp+X	LDA !abs	TXA	LDY dp	TCALL 12	LDC LDCB	LDX dp	LDX dp+Y	XCN	DAS
111	EI				LDM dp,#imm	STA dp	STA dp+X	STA !abs	TAX	STY dp	TCALL 14	STC M.bit	STX dp	STX dp+Y	XAX	STOP

LOW / HIGH	10000 10	10001 11	10010 12	10011 13	10100 14	10101 15	10110 16	10111 17	11000 18	11001 19	11010 1A	11011 1B	11100 1C	11101 1D	11110 1E	11111 1F
000	BPL rel	CLR1 dp.bit	BBC A.bit,rel	BBC dp.bit,rel	ADC {X}	ADC !abs+Y	ADC [dp+X]	ADC [dp]+Y	ASL !abs	ASL dp+X	TCALL 1	JMP !abs	BIT !abs	ADDW dp	LDX #imm	JMP [!abs]
001	BVC rel				SBC {X}	SBC !abs+Y	SBC [dp+X]	SBC [dp]+Y	ROL !abs	ROL dp+X	TCALL 3	CALL !abs	TEST !abs	SUBW dp	LDY #imm	JMP [dp]
010	BCC rel				CMP {X}	CMP !abs+Y	CMP [dp+X]	CMP [dp]+Y	LSR !abs	LSR dp+X	TCALL 5	MUL	TCLR1 !abs	CMPW dp	CMPX #imm	CALL [dp]
011	BNE rel				OR {X}	OR !abs+Y	OR [dp+X]	OR [dp]+Y	ROR !abs	ROR dp+X	TCALL 7	DBNE Y	CMPX !abs	LDYA dp	CMPY #imm	RETI
100	BMI rel				AND {X}	AND !abs+Y	AND [dp+X]	AND [dp]+Y	INC !abs	INC dp+X	TCALL 9	DIV	CMPY !abs	INCW dp	INC Y	TAY
101	BVS rel				EOR {X}	EOR !abs+Y	EOR [dp+X]	EOR [dp]+Y	DEC !abs	DEC dp+X	TCALL 11	XMA {X}	XMA dp	DECW dp	DEC Y	TYA
110	BCS rel				LDA {X}	LDA !abs+Y	LDA [dp+X]	LDA [dp]+Y	LDY !abs	LDY dp+X	TCALL 13	LDA {X}+	LDX !abs	STYA dp	XAY	DAA
111	BEQ rel				STA {X}	STA !abs+Y	STA [dp+X]	STA [dp]+Y	STY !abs	STY dp+X	TCALL 15	STA {X}+	STX !abs	CBNE dp	XYX	NOP

C.3 Instruction Set

Arithmetic / Logic Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	ADC #imm	04	2	2	Add with carry.	
2	ADC dp	05	2	3	A ← (A) + (M) + C	
3	ADC dp + X	06	2	4		
4	ADC !abs	07	3	4		NV--H-ZC
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	
10	AND dp	85	2	3	A ← (A) ∧ (M)	
11	AND dp + X	86	2	4		
12	AND !abs	87	3	4		N-----Z-
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	
18	ASL dp	09	2	4	C 7 6 5 4 3 2 1 0	N-----ZC
19	ASL dp + X	19	2	5	☐ ← ← ← ← ← ← ← ← ← “0”	
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	Compare accumulator contents with memory contents	N-----ZC
25	CMP !abs + Y	55	3	5	(A) - (M)	
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	
30	CMPX dp	6C	2	3	(X) - (M)	N-----ZC
31	CMPX !abs	7C	3	4		
32	CMPY #imm	7E	2	2	Compare Y contents with memory contents	
33	CMPY dp	8C	2	3	(Y) - (M)	N-----ZC
34	CMPY !abs	9C	3	4		
35	COM dp	2C	2	4	1’S Complement : (dp) ← ~(dp)	N-----Z-
36	DAA	DF	1	3	Decimal adjust for addition	N-----ZC
37	DAS	CF	1	3	Decimal adjust for subtraction	N-----ZC
38	DEC A	A8	1	2	Decrement	N-----Z-
39	DEC dp	A9	2	4	M ← (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-

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
44	DIV	9B	1	12	Divide : YA / X Q: A, R: Y	NV--H-Z-
45	EOR #imm	A4	2	2	Exclusive OR	
46	EOR dp	A5	2	3	A ← (A) ⊕ (M)	
47	EOR dp + X	A6	2	4		
48	EOR !abs	A7	3	4		N-----Z-
49	EOR !abs + Y	B5	3	5		
50	EOR [dp + X]	B6	2	6		
51	EOR [dp] + Y	B7	2	6		
52	EOR { X }	B4	1	3		
53	INC A	88	1	2	Increment	N-----ZC
54	INC dp	89	2	4	M ← (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		
62	LSR !abs	58	3	5		
63	MUL	5B	1	9	Multiply : YA ← Y × A	N-----Z-
64	OR #imm	64	2	2	Logical OR	
65	OR dp	65	2	3	A ← (A) ∨ (M)	
66	OR dp + X	66	2	4		
67	OR !abs	67	3	4		N-----Z-
68	OR !abs + Y	75	3	5		
69	OR [dp + X]	76	2	6		
70	OR [dp] + Y	77	2	6		
71	OR { X }	74	1	3		
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		
75	ROL !abs	38	3	5		
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		
79	ROR !abs	78	3	5		
80	SBC #imm	24	2	2	Subtract with Carry	
81	SBC dp	25	2	3	A ← (A) - (M) - ~(C)	
82	SBC dp + X	26	2	4		
83	SBC !abs	27	3	4		NV--HZC
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 ↔ A_3~A_0	N-----Z-

Register / Memory Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	LDA #imm	C4	2	2	Load accumulator	
2	LDA dp	C5	2	3	A ← (M)	
3	LDA dp + X	C6	2	4		
4	LDA !abs	C7	3	4		
5	LDA !abs + Y	D5	3	5		N-----Z-
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 ← (M) , X ← X + 1	
10	LDM dp,#imm	E4	3	5	Load memory with immediate data : (M) ← imm	--------
11	LDX #imm	1E	2	2	Load X-register	
12	LDX dp	CC	2	3	X ← (M)	N-----Z-
13	LDX dp + Y	CD	2	4		
14	LDX !abs	DC	3	4		
15	LDY #imm	3E	2	2	Load Y-register	
16	LDY dp	C9	2	3	Y ← (M)	N-----Z-
17	LDY dp + X	D9	2	4		
18	LDY !abs	D8	3	4		
19	STA dp	E5	2	4	Store accumulator contents in memory	
20	STA dp + X	E6	2	5	(M) ← A	
21	STA !abs	E7	3	5		
22	STA !abs + Y	F5	3	6		--------
23	STA [dp + X]	F6	2	7		
24	STA [dp] + Y	F7	2	7		
25	STA { X }	F4	1	4		
26	STA { X }+	FB	1	4	X- register auto-increment : (M) ← A, X ← X + 1	
27	STX dp	EC	2	4	Store X-register contents in memory	
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 memory	
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	--------

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 subtract without carry YA ← (YA) - (dp +1) (dp)	NV--H-ZC

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 ← (C) ∧ (M .bit)	-------C
2	AND1B M.bit	8B	3	4	Bit AND C-flag and NOT : C ← (C) ∧ ~(M .bit)	-------C
3	BIT dp	0C	2	4	Bit test A with memory :	MM----Z-
4	BIT !abs	1C	3	5	Z ← (A) ∧ (M) , N ← (M_7) , V ← (M_6)	
5	CLR1 dp.bit	y1	2	4	Clear bit : (M.bit) ← “0”	--------
6	CLRA1 A.bit	2B	2	2	Clear A bit : (A.bit) ← “0”	--------
7	CLRC	20	1	2	Clear C-flag : C ← “0”	-------0
8	CLRG	40	1	2	Clear G-flag : G ← “0”	--0-----
9	CLRV	80	1	2	Clear V-flag : V ← “0”	-0--0---
10	EOR1 M.bit	AB	3	5	Bit exclusive-OR C-flag : C ← (C) ⊕ (M .bit)	-------C
11	EOR1B M.bit	AB	3	5	Bit exclusive-OR C-flag and NOT : C ← (C) ⊕ ~(M .bit)	-------C
12	LDC M.bit	CB	3	4	Load C-flag : C ← (M .bit)	-------C
13	LDCB M.bit	CB	3	4	Load C-flag with NOT : C ← ~(M .bit)	-------C
14	NOT1 M.bit	4B	3	5	Bit complement : (M .bit) ← ~(M .bit)	--------
15	OR1 M.bit	6B	3	5	Bit OR C-flag : C ← (C) ∨ (M .bit)	-------C
16	OR1B M.bit	6B	3	5	Bit OR C-flag and NOT : C ← (C) ∨ ~(M .bit)	-------C
17	SET1 dp.bit	x1	2	4	Set bit : (M.bit) ← “1”	--------
18	SETA1 A.bit	0B	2	2	Set A bit : (A.bit) ← “1”	--------
19	SETC	A0	1	2	Set C-flag : C ← “1”	-------1
20	SETG	C0	1	2	Set G-flag : G ← “1”	--1-----
21	STC M.bit	EB	3	6	Store C-flag : (M .bit) ← C	--------
22	TCLR1 !abs	5C	3	6	Test and clear bits with A : A - (M) , (M) ← (M) ∧ ~(A)	N-----Z-
23	TSET1 !abs	3C	3	6	Test and set bits with A : A - (M) , (M) ← (M) ∨ (A)	N-----Z-

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 ← (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 ← (pc) + rel	
5	BCC rel	50	2	2/4	Branch if carry bit clear if (C) = 0 , then pc ← (pc) + rel	--------
6	BCS rel	D0	2	2/4	Branch if carry bit set if (C) = 1 , then pc ← (pc) + rel	--------
7	BEQ rel	D0	2	2/4	Branch if equal if (Z) = 1 , then pc ← (pc) + rel	--------
8	BMI rel	90	2	2/4	Branch if minus if (N) = 1 , then pc ← (pc) + rel	--------
9	BNE rel	70	2	2/4	Branch if not equal if (Z) = 0 , then pc ← (pc) + rel	--------
10	BPL rel	10	2	2/4	Branch if minus if (N) = 0 , then pc ← (pc) + rel	--------
11	BRA rel	2F	2	4	Branch always pc ← (pc) + rel	--------
12	BVC rel	30	2	2/4	Branch if overflow bit clear if (V) = 0 , then pc ← (pc) + rel	--------
13	BVS rel	B0	2	2/4	Branch if overflow bit set if (V) = 1 , then pc ← (pc) + rel	--------
14	CALL !abs	3B	3	8	Subroutine call	
15	CALL [dp]	5F	2	8	M(sp)←(pc_H), sp←sp - 1, M(sp)← (pc_L), sp ←sp - 1, if !abs, pc← abs ; if [dp], pc_L← (dp), pc_H← (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) ≠ (M) , then pc ← (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) ≠ 0 , then pc ← (pc) + rel.	
20	JMP !abs	1B	3	3	Unconditional jump	
21	JMP [!abs]	1F	3	5	pc ← jump address	--------
22	JMP [dp]	3F	2	4		
23	PCALL upage	4F	2	6	U-page call M(sp) ←(pc_H), sp ←sp - 1, M(sp) ← (pc_L), sp ← sp - 1, pc_L ← (upage), pc_H ← "0FF_H" .	--------
24	TCALL n	nA	1	8	Table call : (sp) ←(pc_H), sp ← sp - 1, M(sp) ← (pc_L),sp ← sp - 1, pc_L ← (Table vector L), pc_H ← (Table vector H)	--------

Control Operation & Etc.

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	BRK	0F	1	8	Software interrupt : B ← "1", M(sp) ← (pc_H), sp ←sp-1, M(s) ← (pc_L), sp ← sp - 1, M(sp) ← (PSW), sp ← sp -1, pc_L ← ($0FFDE_H$) , pc_H ← ($0FFDF_H$) .	---1-0--
2	DI	60	1	3	Disable all interrupts : I ← "0"	-----0--
3	EI	E0	1	3	Enable all interrupt : I ← "1"	-----1--
4	NOP	FF	1	2	No operation	--------
5	POP A	0D	1	4	sp ← sp + 1, A ← M(sp)	--------
6	POP X	2D	1	4	sp ← sp + 1, X ← M(sp)	
7	POP Y	4D	1	4	sp ← sp + 1, Y ← M(sp)	
8	POP PSW	6D	1	4	sp ← sp + 1, PSW ← M(sp)	restored
9	PUSH A	0E	1	4	M(sp) ← A , sp ← sp - 1	--------
10	PUSH X	2E	1	4	M(sp) ← X , sp ← sp - 1	
11	PUSH Y	4E	1	4	M(sp) ← Y , sp ← sp - 1	
12	PUSH PSW	6E	1	4	M(sp) ← PSW , sp ← sp - 1	
13	RET	6F	1	5	Return from subroutine sp ← sp +1, pc_L ← M(sp), sp ← sp +1, pc_H ← M(sp)	--------
14	RETI	7F	1	6	Return from interrupt sp ← sp +1, PSW ← M(sp), sp ← sp + 1, pc_L ← M(sp), sp ← sp + 1, pc_H ← M(sp)	restored
15	STOP	EF	1	3	Stop mode (halt CPU, stop oscillator)	--------

D. MASK ORDER SHEET

MASK ORDER & VERIFICATION SHEET

GMS81C3004-LA ☐☐☐

Customer should write inside thick line box.

1. Customer Information

Company Name	
Application	
Order Date	YYYY MM DD . .
Tel: Fax:	
Name & Signature:	

2. Device Information

Package	☐ 80QFP ☐ DIE
OSC Opt.	☐ Crystal ☐ RC
Mask Data	File Name: (.OTP)
	Check Sum: ( )

☐ Hitel

☐ Chollian

☐ Internet

0000H	Set "FF" in this area
6FFFH	
7000H	.OTP file data
7FFFH	

(Please check mark into ☐)

3. Marking Specification (if 80QFP sale)

LGS
GMS81C3004-LAxxx
YYWW KOREA
○

#1 index mark

4. Delivery Schedule

	Date	Quantity	LG Confirmation
Customer Sample	YYYY MM DD . .	pcs	
Risk Order	YYYY MM DD . .	pcs	

5. ROM Code Verification

Verification Date: YYYY MM DD . .
Please confirm our verification data.
Check Sum:
Tel: Fax:
Name & Signature:

This box is written after "5.Verification".

Approval Date: YYYY MM DD . .
I agree with your verification data and confirm you to make mask set.
Tel: Fax:
Name & Signature:

LG Semicon