

HYNIX SEMICONDUCTOR INC.
8-BIT SINGLE-CHIP MICROCONTROLLERS

GMS81C7008

GMS81C7016

User's Manual (Ver. 2.02)



REVISION HISTORY

VERSION 2.02 (OCT., 2002) This book

Add product of 64LQFP package.

VERSION 2.01 (APR., 2001) This book

Delete product of 52SDIP package also, no longer produce 52pin MCU.

The company name Hyundai Electronics Industries Co., Ltd. changed to Hynix Semiconductor Inc.

VERSION 2.00 (FEB., 2001)

Delete product of 52LQFP package.

Fixed some errata that pin number 25 and 26 on 52SDIP package are reversed.

VERSION 1.02 (NOV., 2000)

Fixed the name of LCR register on page 39 and 75, the BUR register on page 66.

Version 2.02

**Published by
MCU Application Team**

©2001 Hynix semiconductor Inc. All right reserved.

Additional information of this manual may be served by Hynix semiconductor offices in Korea or Distributors and Representatives listed at address directory.

Hynix semiconductor reserves the right to make changes to any information here in at any time without notice.

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

Table of Contents

1. OVERVIEW	1	14. ANALOG DIGITAL CONVERTER	59
Description	1	15. SERIAL COMMUNICATION	61
Features	1	Transmission/Receiving Timing	62
Development Tools	2	The method of Serial I/O	63
Ordering Information	2	The Method to Test Correct Transmission	63
2. BLOCK DIAGRAM	3	16. BUZZER FUNCTION	64
3. PIN ASSIGNMENT	4	17. INTERRUPTS	66
4. PACKAGE DIMENSION	6	Interrupt Sequence	68
5. PIN FUNCTION	8	BRK Interrupt	69
6. PORT STRUCTURES	11	Multi Interrupt	69
7. ELECTRICAL CHARACTERISTICS ...	13	External Interrupt	70
Absolute Maximum Ratings	13	Key Scan Interrupt	70
Recommended Operating Conditions	13	18. LCD DRIVER	72
DC Electrical Characteristics	13	LCD Control Registers	72
A/D Converter Characteristics	15	Duty and Bias Selection of LCD driver	74
AC Characteristics	15	Selecting Frame Frequency	74
Serial Interface Timing Characteristics	17	LCD Display Memory	77
Typical Characteristics	18	Control Method of LCD Driver	78
8. MEMORY ORGANIZATION	20	19. WATCH / WATCHDOG TIMER	80
Registers	20	Watch Timer	80
Program Memory	23	Watchdog Timer	80
Data Memory	26	20. POWER DOWN OPERATION	83
List of Control Registers	27	SLEEP Mode	83
Addressing Mode	30	STOP Mode	84
9. I/O PORTS	34	21. OSCILLATOR CIRCUIT	87
Registers for Port	34	22. RESET	88
I/O Ports Configuration	35	External Reset Input	88
10. CLOCK GENERATOR	39	Watchdog Timer Reset	88
11. OPERATION MODE	41	23. POWER FAIL PROCESSOR	89
Operation Mode Switching	42	24. DEVELOPMENT TOOLS	91
12. BASIC INTERVAL TIMER	44	OTP Programming	91
13. TIMER/EVENT COUNTER	46	Emulator EVA. Board Setting	92
8-bit Timer / Counter Mode	49	A. MASK ORDER SHEET	i
16-bit Timer / Counter Mode	53	B. INSTRUCTION	ii
8-bit Capture Mode	54	Terminology List	ii
16-bit Capture Mode	55	Instruction Map	iii
Timer output port mode	55	Instruction Set	iv
PWM Mode	56	C. SOFTWARE EXAMPLE	x

GMS81C7008/16

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH LCD DRIVER & A/D CONVERTER

1. OVERVIEW

1.1 Description

The GMS81C7008/7016 is advanced CMOS 8-bit microcontrollers with 8K/16K bytes of ROM. There are a powerful microcontroller which provides a highly flexible and cost effective solution to many LCD applications. These provide the following standard features: 16K/8K bytes of mask type ROM or 16K bytes OTP ROM, 448 bytes of RAM, 8-bit timer/counter, 8-bit A/D converter, 10 bit high speed PWM Output, programmable buzzer driving port, 8-bit basic interval timer, watch dog timer, serial peripheral interface, on chip oscillator and clock circuitry. They also come with 4com/24seg LCD driver. In addition, it support power saving mode to reduce power consumption.

Device name	ROM Size	RAM Size	I/O	OTP	Package
GMS81C7008	8K bytes	448 bytes	49	GMS87C7016	64SDIP, 64MQFP, 64LQFP
GMS81C7016	16K bytes	448 bytes	49	GMS87C7016	

1.2 Features

- **8K/16K Bytes On-chip Programmable ROM**
 - Internal Built-in Resistor Circuit for Bias
- **448 Bytes of On-chip Data RAM**
(Included stack area and 27 nibbles LCD Display RAM)
- **Instruction Execution Time**
1μs at 4MHz (2cycle NOP Instruction)
- **One 8-bit Basic Interval Timer**
- **One Watch Timer**
- **One Watchdog Timer**
- **Four 8-bit Timer/Event Counter**
(or Two 16-bit Timer/Event Counter)
- **Two channel 10-bit High Speed PWM Output**
- **Three External Interrupt input ports**
- **One Programmable 6-bit Buzzer Driving port**
- 500Hz ~ 250kHz@4MHz
- **49 I/O Ports**
- **Eight channel 8-bit A/D converter**
- **One 8-bit Serial Communication Interface**
- **LCD Display/ Controller**
 - Static Mode (27SEG x 1COM, Static)
 - 1/2 Duty Mode (26SEG x 2COM, 1/2 or 1/3 Bias)
 - 1/3 Duty Mode (25SEG x 3COM, 1/3 Bias)
 - 1/4 Duty Mode (24SEG x 4COM, 1/3 Bias)
- **Thirteen Interrupt sources**
 - Basic Interval Timer: 1
 - External input: 3
 - Timer/Event counter: 4
 - ADC: 1
 - Serial Interface: 1
 - WT:1
 - WDT: 1
 - Key Scan: 1
- **Main Clock Oscillation (1.0~4.5MHz)**
 - Crystal
 - Ceramic Resonator
 - External R Oscillator (Built-in Capacitor)
- **Sub Clock Oscillation**
- 32.768kHz Crystal Oscillator
- **Power Saving Operation Mode**
 - Main / Sub Active mode changeable
 - 2/8/16/64 divided system clock selectable
- **Power Down Mode**
 - STOP mode
 - SLEEP mode
 - Sub active Mode
- **2.7V to 5.5V Wide Operating Voltage Range**
- **Noise Immunity Circuit for EMS**

- Power fail processor
- Built in Noise filter
- 64SDIP, 64MQFP, 64LQFP package types

- Available 16K bytes OTP version

1.3 Development Tools

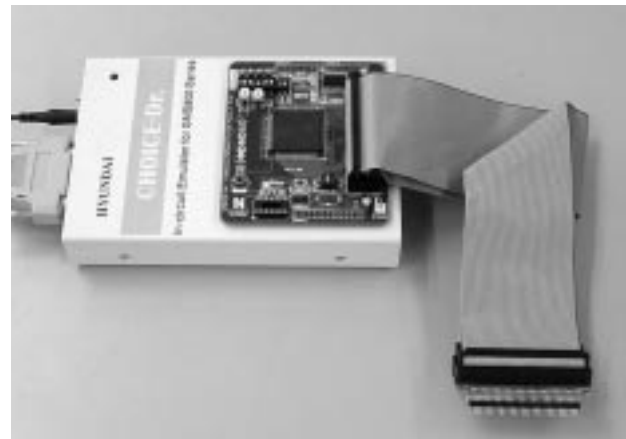
Note: There are several setting switches in the Emulator. User should read carefully and do setting properly before developing the program refer to "24.2 Emulator EVA. Board Setting" on page 92. Otherwise, the Emulator may not work properly.

Software	- MS- Window base assembler - Linker / Editor / Debugger
Hardware (Emulator)	- CHOICE-Dr. - CHOICE-Dr. EVA 81C51/81C7X B/D
OTP programmer	- CHOICE-SIGMA (Single type) - CHOICE-GANG4 (4-gang type)

The GMS81C7008/16 is supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Dr.TM and OTP programmers. There are two different type programmers, one is single type, another is gang type. For more detail, refer to OTP Programming chapter. Macro assembler operates under the MS-

Windows 95/98TM.

Please contact sales part of Hynix semiconductor.

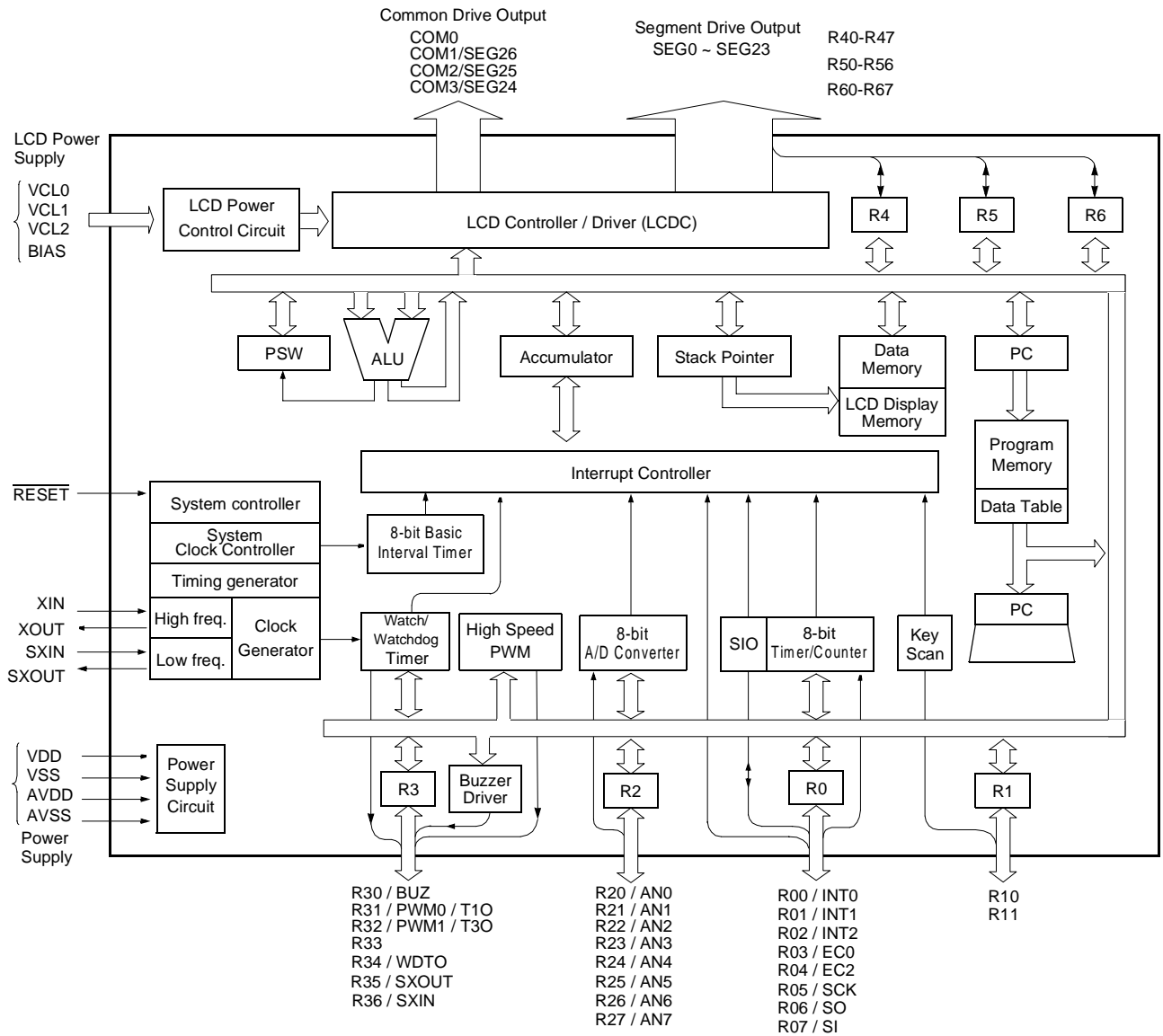


1.4 Ordering Information

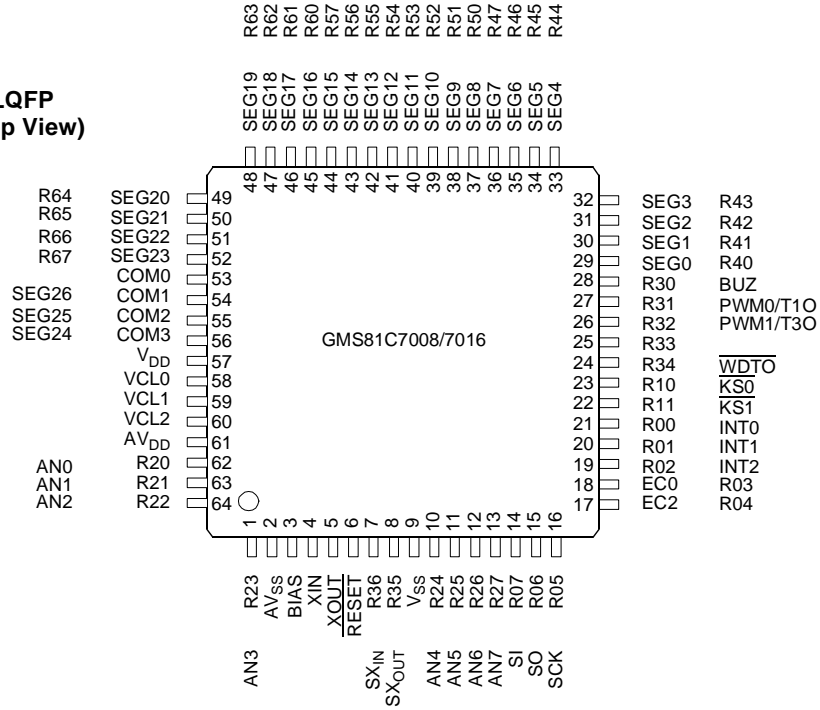
	Device name	ROM Size (bytes)	RAM size	Package
Mask ROM version	GMS81C7008 K	8K bytes	448 bytes	64SDIP
	GMS81C7016 K	16K bytes	448 bytes	64SDIP
	GMS81C7008 Q	8K bytes	448 bytes	64MQFP
	GMS81C7016 Q	16K bytes	448 bytes	64MQFP
	GMS81C7008 LQ	8K bytes	448 bytes	64LQFP
	GMS81C7016 LQ	16K bytes	448 bytes	64LQFP
OTP ROM version	GMS87C7016 K	16K bytes OTP	448 bytes	64SDIP
	GMS87C7016 Q	16K bytes OTP	448 bytes	64MQFP
	GMS87C7016 LQ	16K bytes OTP	448 bytes	64LQFP

2. BLOCK DIAGRAM

GMS81C7008/7016

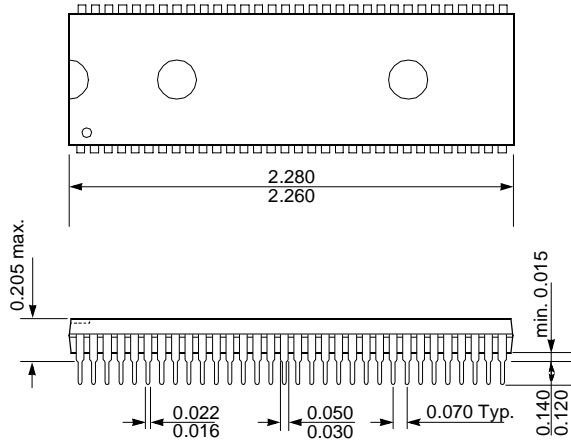


**64LQFP
(Top View)**

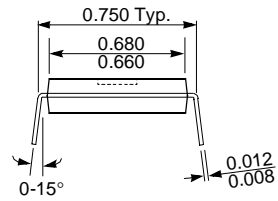


4. PACKAGE DIMENSION

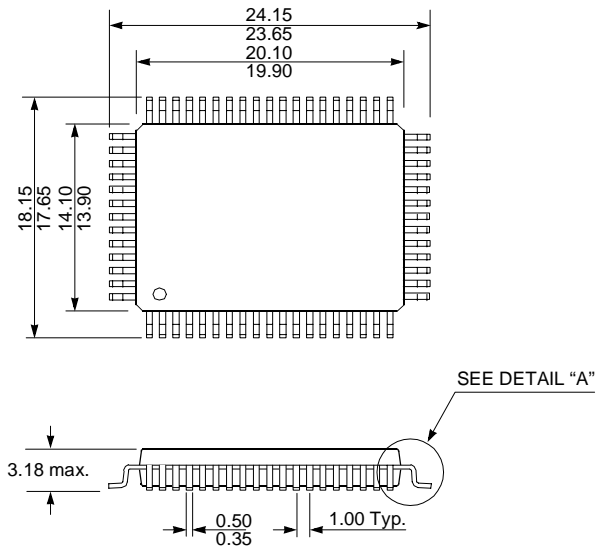
64SDIP



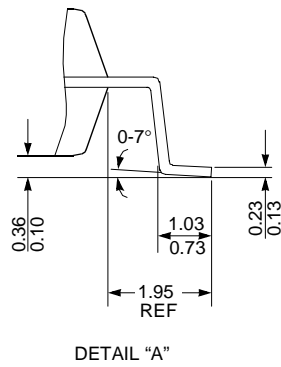
UNIT: INCH

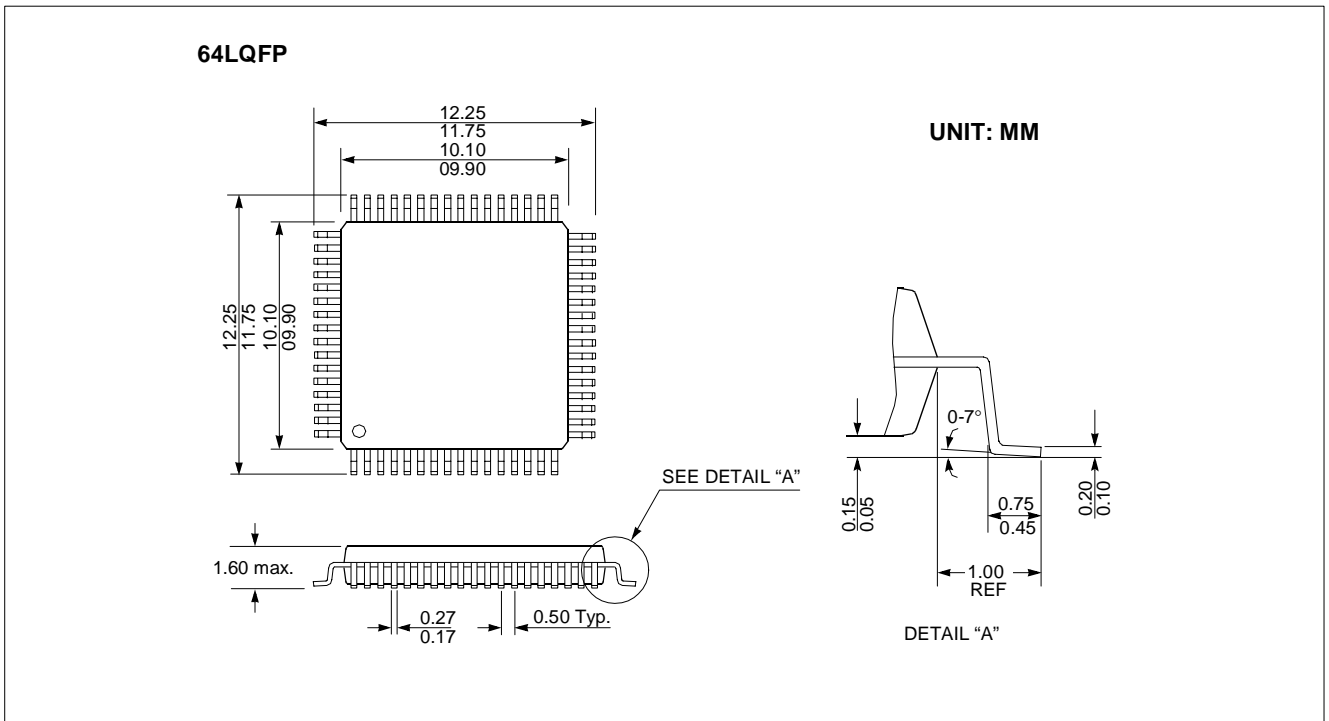


64MQFP



UNIT: MM





5. PIN FUNCTION

V_{DD}: Supply voltage.

V_{SS}: Circuit ground.

RESET: Reset the MCU.

AV_{DD}: Supply voltage to the ladder resistor of ADC circuit. To enhance the resolution of analog to digital converter, use independent power source as well as possible, other than digital power source.

AV_{SS}: ADC circuit ground.

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.

BIAS: LCD bias voltage input pin.

VCL0~VCL2: LCD driver power supply pins. The voltage on each pin is VCL2 > VCL1 > VCL0. For details, Refer to "18. LCD DRIVER" on page 72.

COM0~COM3: LCD common signal output pins. Also, the pins of COM1, COM2 and COM3 are shared with LCD segment signal outputs of SEG26, SEG25, SEG24 as application requirement.

SX_{IN}: Input to the internal subsystem clock operating circuit. In addition, SX_{IN} is shared with the R36 which is selected by the software option.

SX_{OUT}: Output from the inverting subsystem oscillator amplifier. In addition, SX_{OUT} is shared with the R35 which is selected by the software option.

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 schmitt trigger inputs. Also, pull-up resistors and open-drain outputs are software assignable.

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	EC0 (Event counter input 0)
R04	EC2 (Event counter input 2)
R05	SCK (Serial clock)
R06	SO (Serial data output)
R07	SI (Serial data input)

R10~R11: R1 is a 2-bit CMOS bidirectional I/O port. R1 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs are soft-

ware assignable.

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

Port pin	Alternate function
R00	$\overline{KS0}$ (Key scan 0)
R01	$\overline{KS1}$ (Key scan 1)

R20~R27: R2 is an 8-bit CMOS bidirectional I/O port. R2 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs are software assignable.

In addition, R2 is shared with the ADC input.

Port pin	Alternate function
R20	AN0 (Analog Input 0)
R21	AN1 (Analog Input 1)
R22	AN2 (Analog Input 2)
R23	AN3 (Analog Input 3)
R24	AN4 (Analog Input 4)
R25	AN5 (Analog Input 5)
R26	AN6 (Analog Input 6)
R27	AN7 (Analog Input 7)

R30~R36: R3 is a 7-bit CMOS bidirectional I/O port. R3 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs are software assignable.

In addition, R3 serves the functions of the various following special features.

Port pin	Alternate function
R30	BUZ (Buzzer driving output)
R31	PWM0 / T1O (PWM 0 output / Timer 1 output)
R32	PWM1 / T3O (PWM 1 output / Timer 3 output)
R33	-
R34	$\overline{WDT0}$ (Watchdog timer output)
R35	SX _{OUT} (Sub clock output)
R36	SX _{IN} (Sub clock input)

SEG0~SEG7: These pins generate LCD segment signal output. Every LCD segment pins are shared with normal R4 input/output port. R4 is an 8-bit CMOS bidirectional I/O port. R4 pins 1 or 0 written to the Port Direction Register can be used as outputs or in-

puts.

LCD pin function	Port pin
SEG0 (LCD segment 0 signal output)	R40
SEG1 (LCD segment 1 signal output)	R41
SEG2 (LCD segment 2 signal output)	R42
SEG3 (LCD segment 3 signal output)	R43
SEG4 (LCD segment 4 signal output)	R44
SEG5 (LCD segment 5 signal output)	R45
SEG6 (LCD segment 6 signal output)	R46
SEG7 (LCD segment 7 signal output)	R47

SEG8~SEG15: These pins generate LCD segment signal output. Every LCD segment pins are shared with normal R5 input/output port. R5 is an 8-bit CMOS bidirectional I/O port. R5 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

LCD pin function	Port pin
SEG8 (LCD segment 8 signal output)	R50
SEG9 (LCD segment 9 signal output)	R51
SEG10 (LCD segment 10 signal output)	R52
SEG11 (LCD segment 11 signal output)	R53
SEG12 (LCD segment 12 signal output)	R54
SEG13 (LCD segment 13 signal output)	R55
SEG14 (LCD segment 14 signal output)	R56
SEG15 (LCD segment 15 signal output)	R57

SEG16~SEG23: These pins generate LCD segment signal output.

Every LCD segment pins are shared with normal R6 input/output port. R6 is an 8-bit CMOS bidirectional I/O port. R6 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

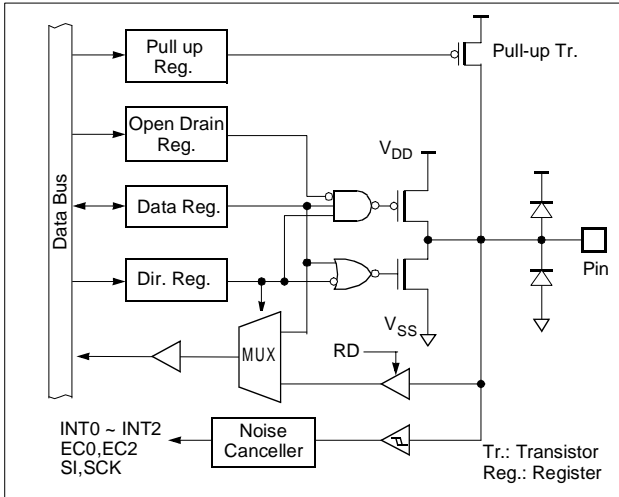
LCD pin function	Port pin
SEG16 (LCD segment 16 signal output)	R60
SEG17 (LCD segment 17 signal output)	R61
SEG18 (LCD segment 18 signal output)	R62
SEG19 (LCD segment 19 signal output)	R63
SEG20 (LCD segment 20 signal output)	R64
SEG21 (LCD segment 21 signal output)	R65
SEG22 (LCD segment 22 signal output)	R66
SEG23 (LCD segment 23 signal output)	R67

PIN NAME (Alternate)	In/Out (Alternate)	Function	
		Basic	Alternate
V _{DD}	-	Supply voltage	
V _{SS}	-	Circuit ground	
RESET	I	Reset signal input	
AV _{DD}	-	Supply voltage input pin for ADC	
AV _{SS}	-	Ground level input pin for ADC	
X _{IN}	I	Oscillation input	
X _{OUT}	O	Oscillation output	
BIAS	I	LCD bias voltage input	
VCL0~VCL2	I	LCD driver power supply	
COM0	O	LCD common signal output	
COM1(SEG26)	O(O)	LCD common signal output	LCD segment signal output
COM2(SEG25)	O(O)		
COM3(SEG24)	O(O)		
R00 (INT0)	I/O (I)	8-bit general I/O ports	External interrupt 0 input
R01 (INT1)	I/O (I)		External interrupt 1 input
R02 (INT2)	I/O (I)		External interrupt 2 input
R03 (EC0)	I/O (I)		Timer/Counter 0 external input
R04 (EC2)	I/O (I)		Timer/Counter 1 external input
R05 (SCK)	I/O (I/O)		Serial clock I/O
R06 (SO)	I/O (O)		Serial data output
R07 (SI)	I/O (I)		Serial data input
R10, R11($\overline{KS0}$, $\overline{KS1}$)	I/O (I)	2-bit general I/O ports	Key scan input
R20~R27(AN0~AN7)	I/O(I)	8-bit general I/O ports	Analog voltage input
R30(BUZ)	I/O(O)	7-bit general I/O ports	Buzzer driving output
R31(PWM0 / T1O)	I/O(O)		PWM 0 output / Timer 1 output
R32(PWM1 / T3O)	I/O(O)		PWM 1 output / Timer 2 output
R33	I/O		-
R34($\overline{WDT0}$)	I/O(O)		Watchdog timer output
R35(SX _{OUT})	I/O(O)		Sub clock output
R36(SX _{IN})	I/O(I)		Sub clock input
SEG0 ~ SEG7 (R40~R47)	O (I/O)		LCD segment signal output
SEG8 ~ SEG15 (R50~R57)	O (I/O)	LCD segment signal output	8-bit general I/O ports
SEG16 ~ SEG23 (R60~R67)	O (I/O)	LCD segment signal output	8-bit general I/O ports

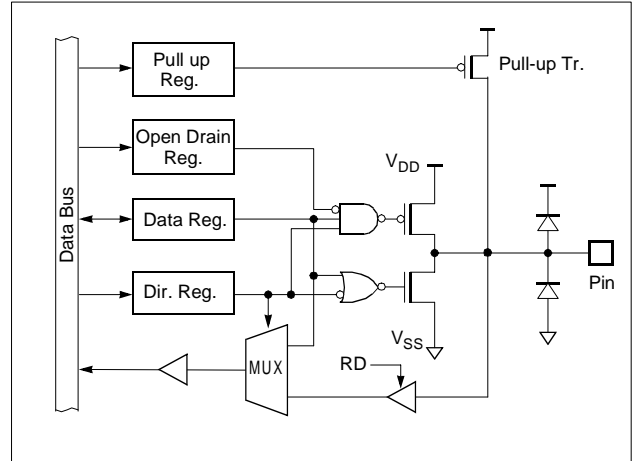
Table 5-1 Port Function Description

6. PORT STRUCTURES

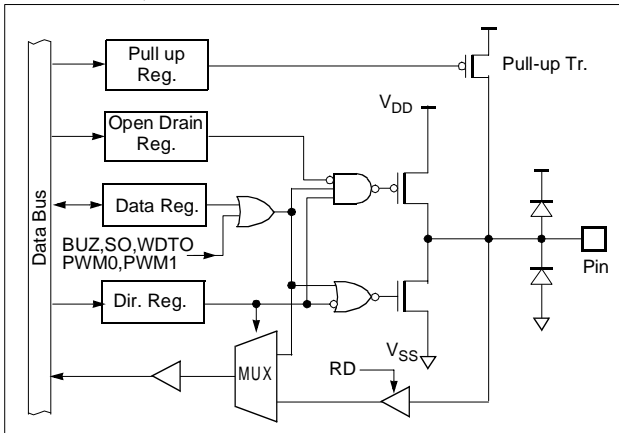
R00/INT0, R01/INT1, R02/INT2, R03/EC0, R04/EC2, R05/SCK, R07/S



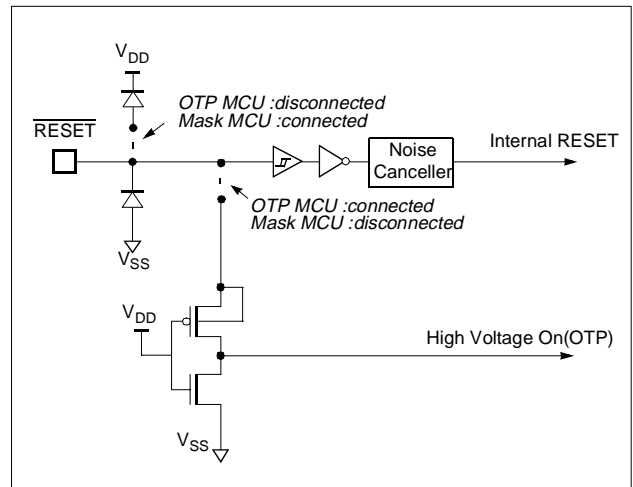
R10~R11, R33, R35, R36



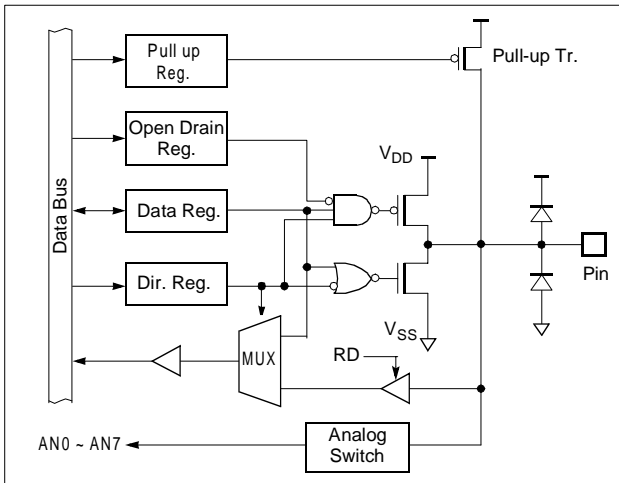
R30/BUZ, R31/PWM0/T10, R32/PWM1/T30, R34/WDTO, R06



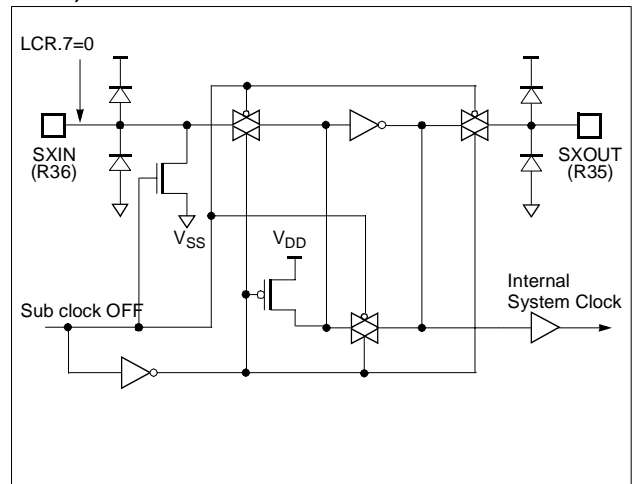
RESET



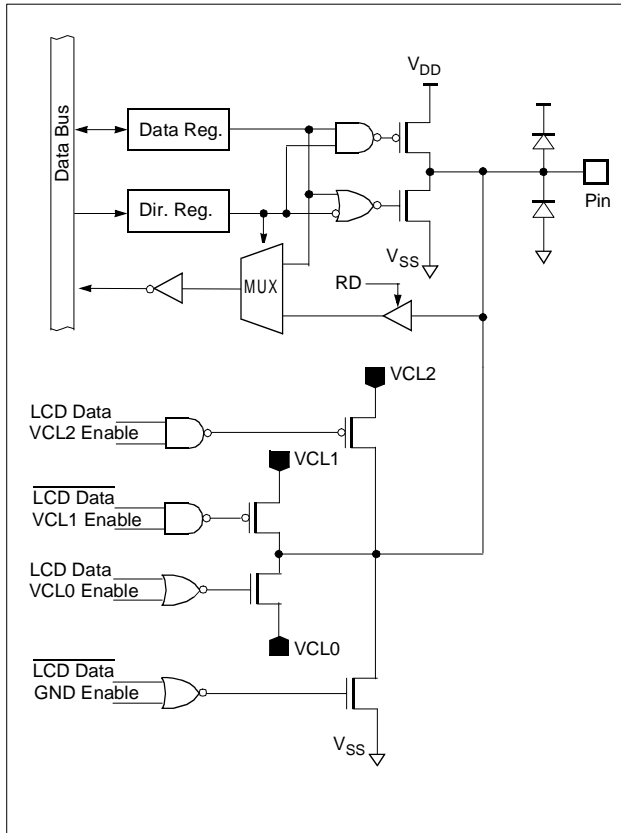
R20/AN0~R27/AN7



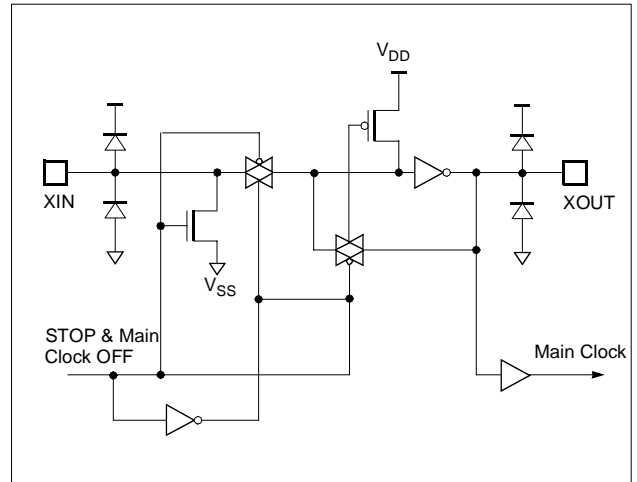
SXIN, SXOUT



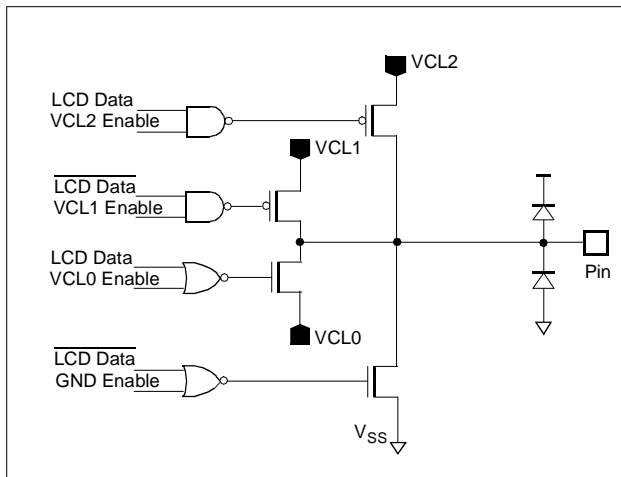
R40~R47, R50~R57, R60~R67 / SEG0~SEG23



XIN, XOUT



COM0~COM3 / SEG24~SEG26



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)
..... 15 mA

Maximum current (ΣI_{OL}) 100 mA

Maximum current (ΣI_{OH}) 60 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}	$f_{XIN}=4.19MHz$ $f_{SXIN}=32.768kHz$	2.7	5.5	V
Operating Frequency	f_{XIN}	$V_{DD}=2.7\sim 5.5V$	1	4.5	MHz
Sub Operating Frequency	f_{SXIN}	$V_{DD}=2.7\sim 5.5V$	30	35	kHz
Operating Temperature	T_{OPR}		-20	+85	°C

7.3 DC Electrical Characteristics

($T_A=-20\sim 85^\circ C$, $V_{DD}=2.7\sim 5.5V$),

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH1}	\overline{RESET} , R0 (except R06)	$0.8 V_{DD}$	-	V_{DD}	V
	V_{IH2}	Other pins	$0.7 V_{DD}$	-	V_{DD}	V
Input Low Voltage	V_{IL1}	\overline{RESET} , R0 (except R06)	0	-	$0.2 V_{DD}$	V
	V_{IL2}	Other pins	0	-	$0.3 V_{DD}$	V
Output High Voltage	V_{OH1}	R0,R1,R2,R3 $I_{OH1}=-0.5mA$	$V_{DD}-0.1$	-	-	V
	V_{OH2}	SEG, COM $I_{OH2}=-30\mu A$	-	-	0.4	V
Output Low Voltage	V_{OL1}	R0,R1,R2,R3 $I_{OL1}=0.4mA$	-	-	0.2	V
	V_{OL2}	SEG, COM $I_{OL2}=30\mu A$	$V_{DD}-0.2$	-	-	V
Input High Leakage Current	I_{IH1}	$V_{IN}=V_{DD}$, All input pins except X_{IN} , SX_{IN}	-	-	1	μA
	I_{IH2}	$V_{IN}=V_{DD}$, X_{IN} , SX_{IN}	-	-	20	μA

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Input Low Leakage Current	I_{IL1}	$V_{IN}=0$, All input pins except X_{IN} , SX_{IN}	-	-	-1	μA
	I_{IL2}	$V_{IN}=0$, X_{IN} , SX_{IN}	-	-	-20	μA
Pull-up Resistor ¹	R_{PORT}	$V_{IN}=0V$, $V_{DD}=5.5V$, R0, R1, R2	60	160	350	k Ω
LCD Voltage Dividing Resistor	R_{LCD}	$V_{DD}=5.5V$	45	65	85	k Ω
Voltage Drop $ V_{DD-COMn} $, $n=0\sim 3$	V_{DC}	$V_{DD}=2.7 \sim 5.5V$ -15 μA per common pin	-	-	120	mV
Voltage Drop $ V_{DD-SEGn} $, $n=0\sim 26$	V_{DS}	$V_{DD}=2.7 \sim 5.5V$ -15 μA per segment pin	-	-	120	mV
V_{CL2} Output Voltage	V_{CL2}	$V_{DD}=2.7 \sim 5.5V$, 1/3 bias BIAS pin and V_{CL2} pin are shorted	$V_{DD}-0.3$	V_{DD}	$V_{DD}+0.3$	V
V_{CL1} Output Voltage	V_{CL1}		0.66 V_{DD} -0.2	0.66 V_{DD}	0.66 V_{DD} +0.3	
V_{CL0} Output Voltage	V_{CL0}		0.33 V_{DD} -0.3	0.33 V_{DD}	0.33 V_{DD} +0.3	
RC Oscillation Frequency	f_{RC}	R=60k Ω , $V_{DD}=5V$	1	2	3	MHz
Supply Current ¹ () means at 3V operation	I_{DD1}	Main clock operation mode ² $V_{DD}=5.5V\pm 10\%$, $X_{IN}=4MHz$, $SX_{IN}=32kHz$	-	2.9 (1.3)	7.0 (3.0)	mA
	I_{DD2}	Sleep mode (Main active) ³ $V_{DD}=5.5V\pm 10\%$, $X_{IN}=4MHz$, $SX_{IN}=32kHz$	-	0.4 (0.1)	1.7 (1.0)	mA
	I_{DD3}	Stop mode ² $V_{DD}=5V\pm 10\%$, $X_{IN}=0Hz$, $SX_{IN}=32kHz$	-	2.0 (1.0)	12 (5)	μA
	I_{DD4}	Sub clock operation mode ⁴ $V_{DD}=5.5V\pm 10\%$, $X_{IN}=0Hz$, $SX_{IN}=32kHz$	-	350 (70)	500 (200)	μA
	I_{DD5}	Sleep mode (Sub active) ⁵ $V_{DD}=3V\pm 10\%$, $X_{IN}=0Hz$, $SX_{IN}=32kHz$	-	10 (3)	50 (20)	μA
	I_{DD6}	Stop mode ⁴ $V_{DD}=5V\pm 10\%$, $X_{IN}=0Hz$, $SX_{IN}=0Hz$ SX_{IN} , $SXOUT$ are used as R35, R36.	-	1.0 (0.5)	12 (5)	μA

1. 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.
2. This mode set System Clock Mode Register(SCMR) to xxx0000_B that is $f_{XIN}/2$
3. This mode set SCMR to xxx0000_B ($f_{XIN}/2$) and set SMR to "1".
4. Main-frequency clock stops and sub-frequency clock in not used and set SCMR to xxx0011_B.
5. Main-frequency clock stops and sub-frequency clock in not used, set SCMR to xxx0011_B and set SMR to "1".

7.4 A/D Converter Characteristics

($T_A=25^\circ\text{C}$, $V_{SS}=0\text{V}$, $V_{DD}=5.0\text{V}$, $AV_{DD}=5.0\text{V}$ @ $f_{XIN}=4\text{MHz}$)

Parameter	Symbol	Test Condition	Specifications			Unit
			Min.	Typ. ¹	Max.	
Analog Input Voltage Range	V_{AIN}	$V_{DD}=AV_{DD}=5.0\text{V}$	$V_{SS}-0.3$	-	$AV_{DD}+0.3$	V
Non-linearity Error	N_{NLE}		-	± 1.0	± 1.5	LSB
Differential Non-linearity Error	N_{DNLE}		-	± 1.0	± 1.5	LSB
Zero Offset Error	N_{ZOE}		-	± 0.5	± 1.5	LSB
Full Scale Error	N_{FSE}		-	± 0.25	± 0.5	LSB
Gain Error	N_{GE}		-	± 1.0	± 1.5	LSB
Overall Accuracy	N_{ACC}		-	± 1.0	± 1.5	LSB
AV_{DD} Input Current	I_{REF}		-	-	200	μA
Conversion Time	T_{CONV}		-	-	20	μs
Analog Power Supply Input Range	AV_{DD}	$V_{DD}=5.0\text{V}$ $V_{DD}=3.0\text{V}$	3.0 2.7	-	V_{DD}	V

1. Data in "Typ" column is at 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

7.5 AC Characteristics

($T_A=-20\sim+85^\circ\text{C}$, $V_{DD}=5\text{V}\pm 10\%$, $V_{SS}=0\text{V}$)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Operating Frequency	f_{MAIN}	X_{IN}	0.455	-	4.2	MHz
	f_{SUB}	SX_{IN}	30	32.768	35	kHz
External Clock Pulse Width	t_{MCPW}	X_{IN}	80	-	-	nS
	t_{SCPW}	SX_{IN}	14.7	-	-	μS
External Clock Transition Time	t_{MRCP}, t_{MFPCP}	X_{IN}	-	-	20	nS
	t_{SRCP}, t_{SFPCP}	SX_{IN}	-	-	3	μS
Main oscillation Stabilizing Time	t_{MST}	X_{IN}, X_{OUT} at 4MHz	-	-	20	mS
Sub oscillation Stabilizing Time	t_{SST}	SX_{IN}, SX_{OUT}	-	0.5	1	S
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}	EC0, EC2	2	-	-	t_{SYS}^1

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

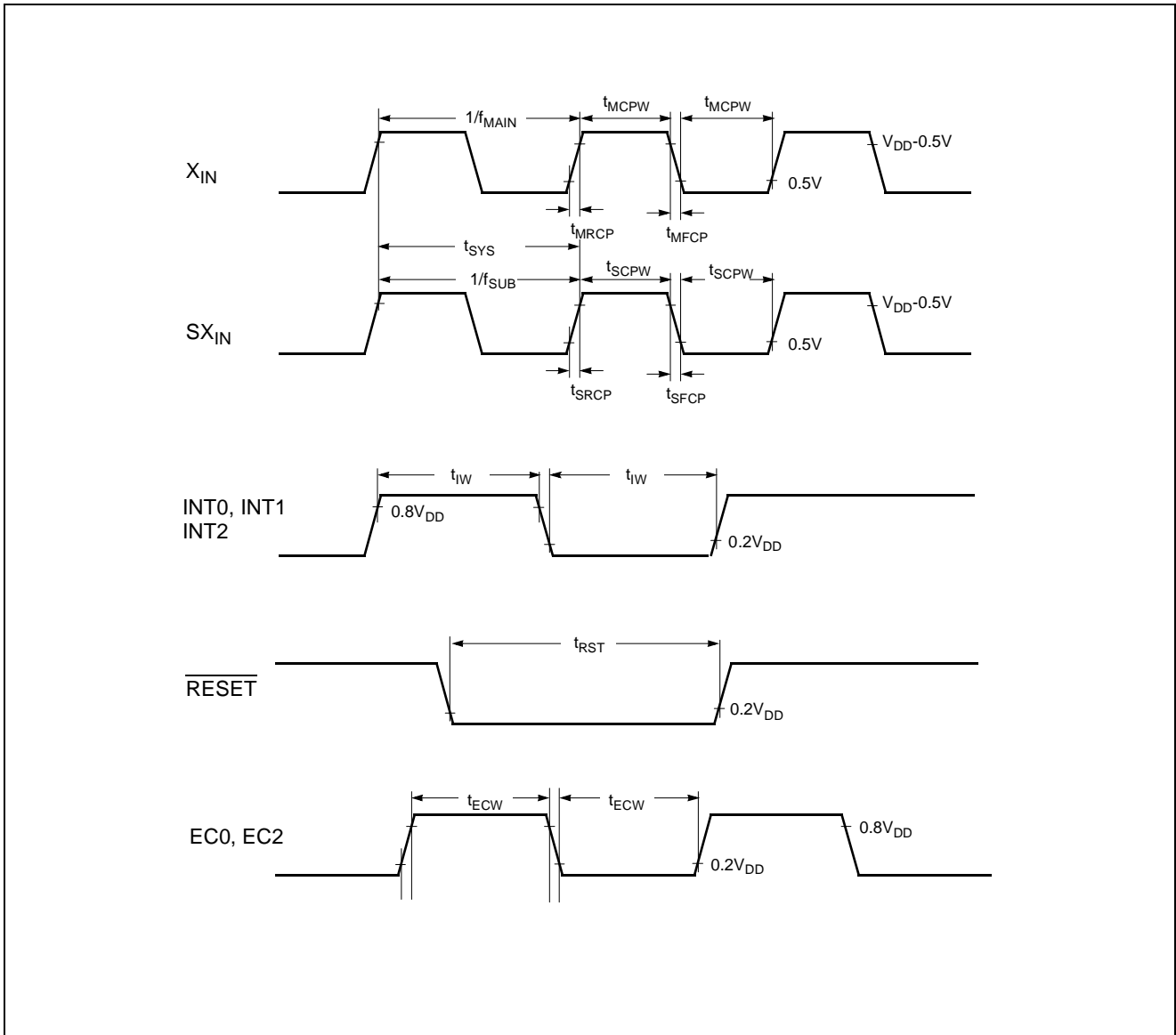


Figure 7-1 Timing Chart

7.6 Serial Interface Timing Characteristics

($T_A = -20 \sim +85^\circ\text{C}$, $V_{DD} = 2.7 \sim 5.5\text{V}$, $V_{SS} = 0\text{V}$, $f_{XIN} = 4\text{MHz}$)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Serial Input Clock Pulse	t_{SCYC}	SCK	$2t_{SYS} + 200$	-	8	ns
Serial Input Clock Pulse Width	t_{SCKW}	SCK	$t_{SYS} + 70$	-	8	ns
SIN Input Setup Time (External SCK)	t_{SUS}	SIN	100	-	-	ns
SIN Input Setup Time (Internal SCK)	t_{SUS}	SIN	200	-	-	ns
SIN Input Hold Time	t_{HS}	SIN	$t_{SYS} + 70$	-	-	ns
Serial Output Clock Cycle Time	t_{SCYC}	SCK	$4t_{SYS}$	-	$16t_{SYS}$	ns
Serial Output Clock Pulse Width	t_{SCKW}	SCK	$t_{SYS} - 30$	-	-	ns
Serial Output Clock Pulse Transition Time	t_{FSCK} t_{RSCK}	SCK	-	-	30	ns
Serial Output Delay Time	t_{SOUT}	SO	-	-	100	ns

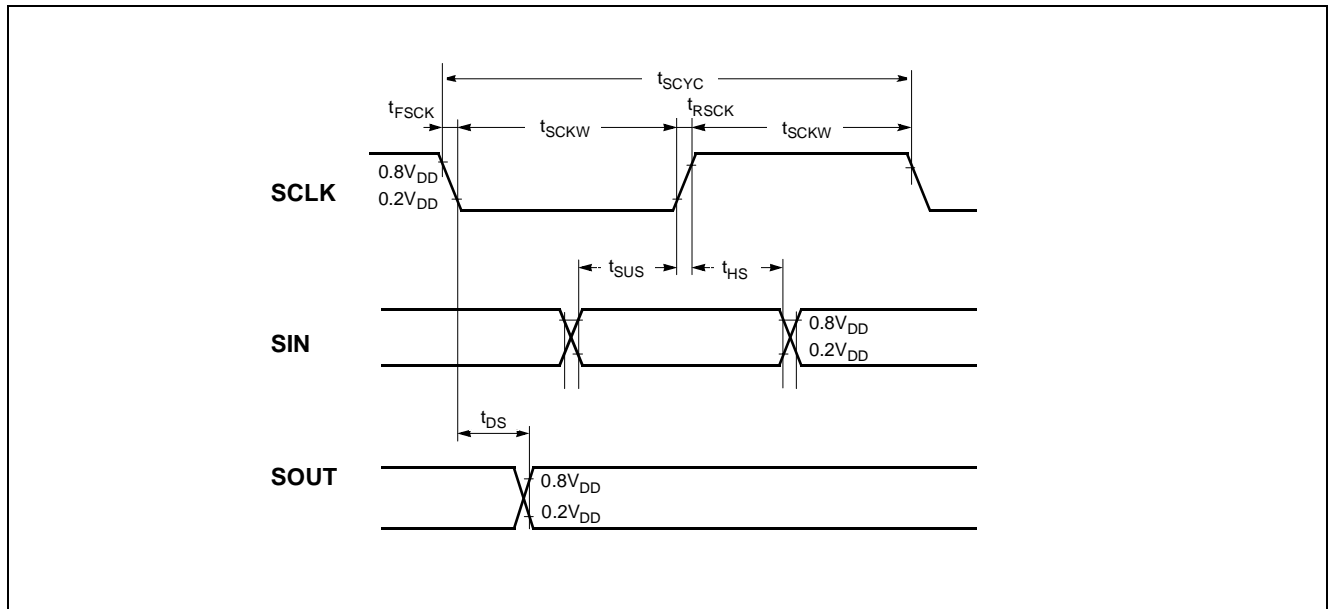


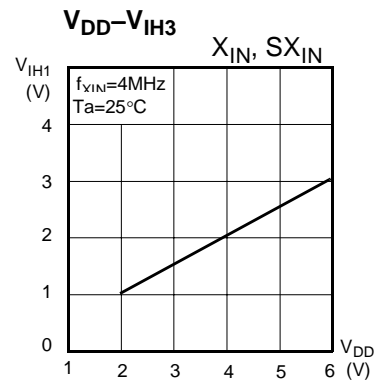
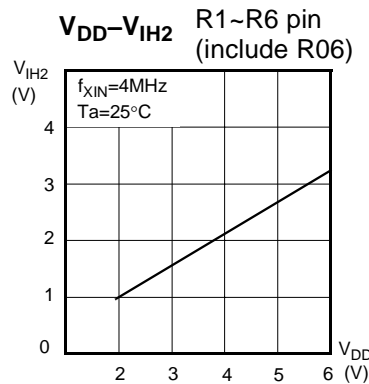
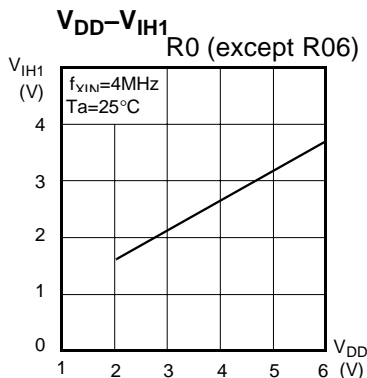
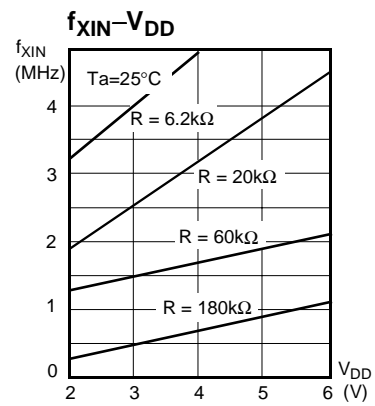
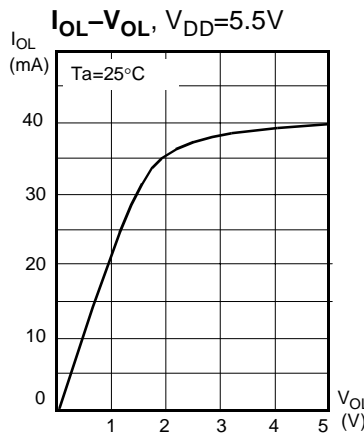
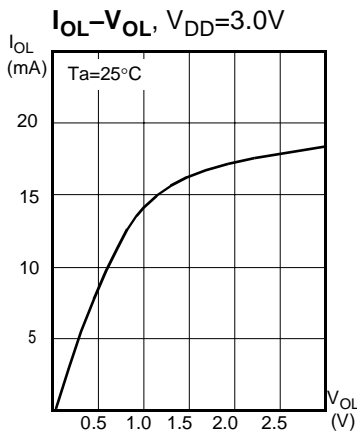
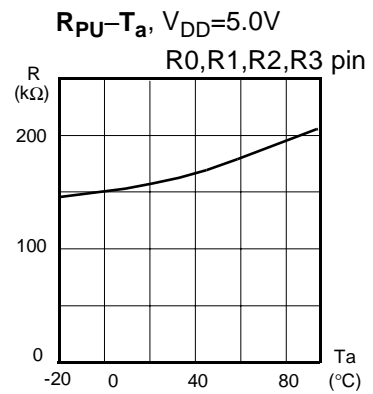
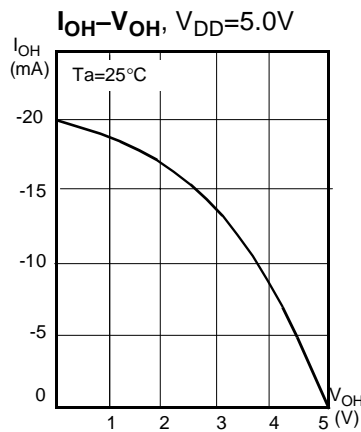
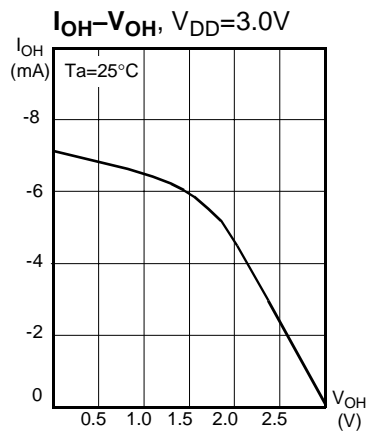
Figure 7-2 Serial I/O Timing Chart

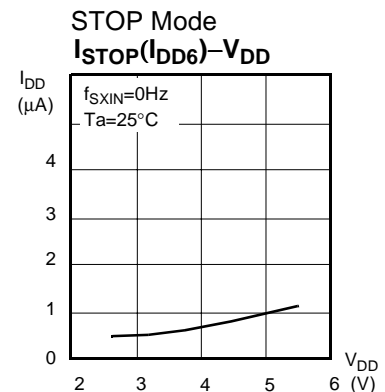
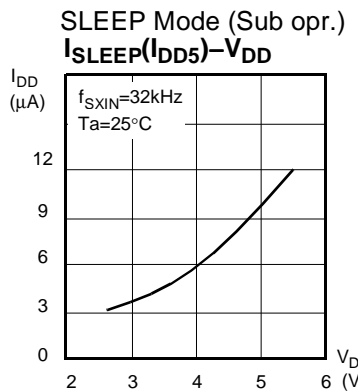
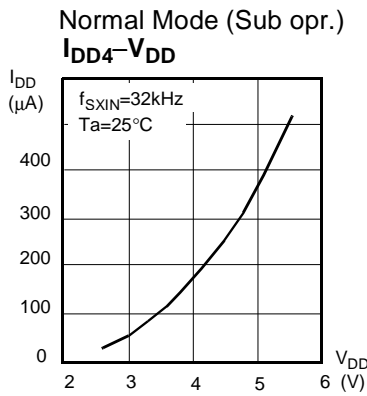
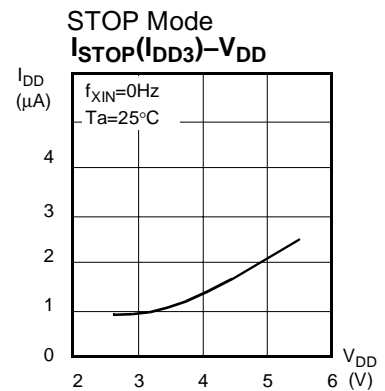
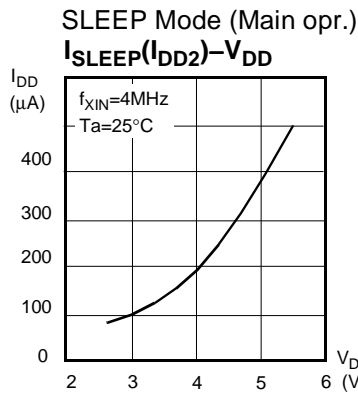
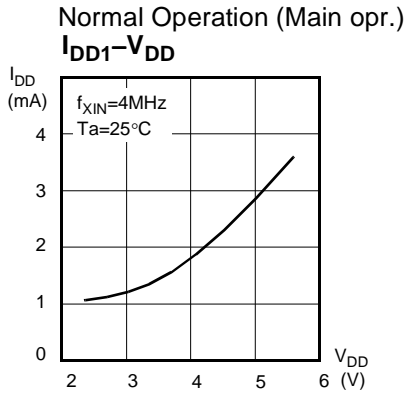
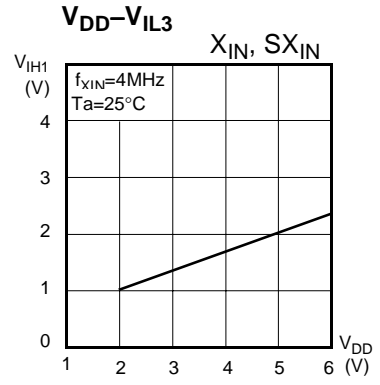
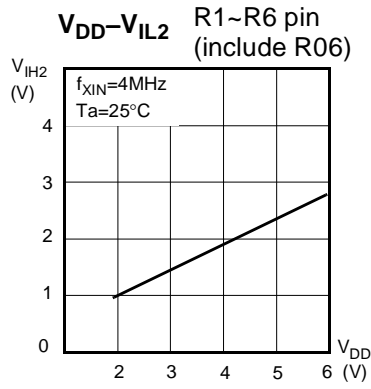
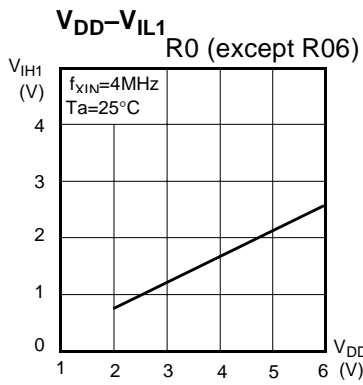
7.7 Typical Characteristics

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

In some graphs or tables the data presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for information only and devices are guaranteed 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





8. MEMORY ORGANIZATION

The GMS81C7008/16 has separate address spaces for Program memory and Data Memory. Program memory can only be read, not written to. It can be up to 8K/16K bytes of Program memory.

8.1 Registers

This device has six registers that are the Program Counter (PC), an 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.

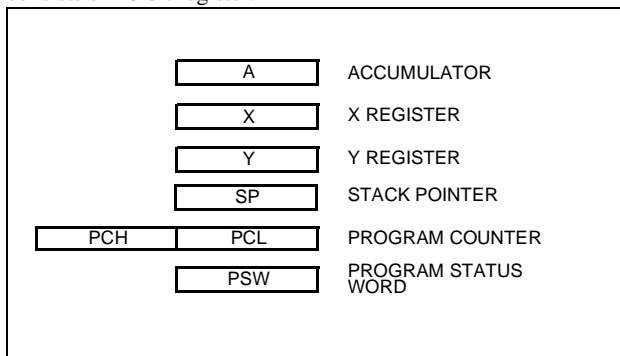


Figure 8-1 Configuration of Registers

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.

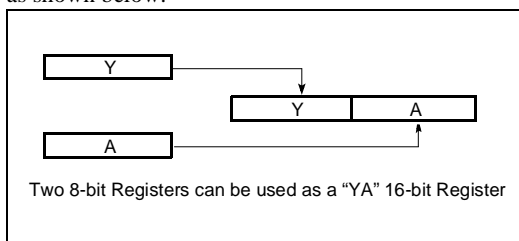


Figure 8-2 Configuration of YA 16-bit Register

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.

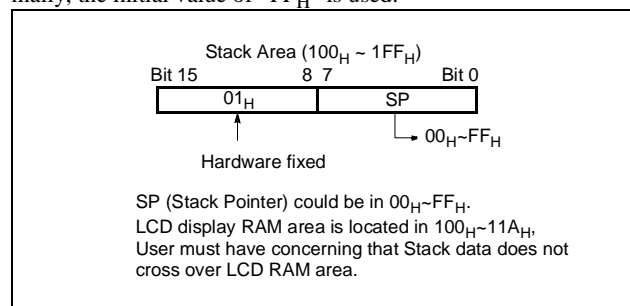
Stack Pointer: The Stack Pointer is an 8-bit register used for occurrence interrupts and calling out subroutines. Stack Pointer identifies the location in the stack to be access (save or restore).

Generally, SP is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in ex-

cess 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 $011B_H$ to $01FF_H$ 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.

The stack area is located in 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.



Note: 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 not 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.

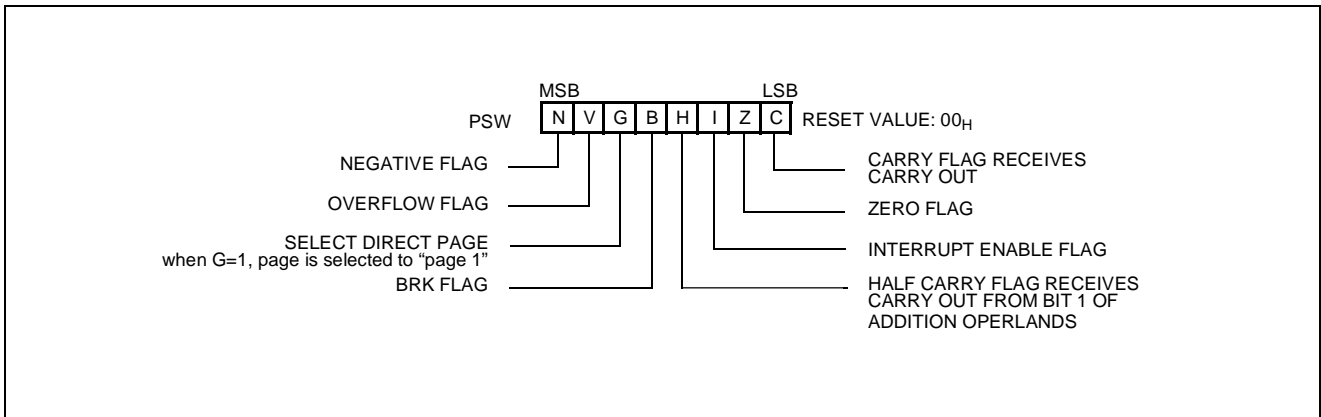


Figure 8-3 PSW (Program Status Word) Register

[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 00H to 0FFH when this flag is "0". If it is set to "1", addressing area is assigned by RPR register (address 0F3H). It is set by SETG in-

struction and cleared by CLRG.

RAM Page	Instruction	Bit1 of RPR	Bit0 of RPR
0 page	CLRG	X	X
0 page	SETG	0	0
1 page	SETG	0	1
Reserved	SETG	1	0
Reserved	SETG	1	1

When content of RPR is above 2, malfunction will be occurred.

[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(7FH) or -128(80H). 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.

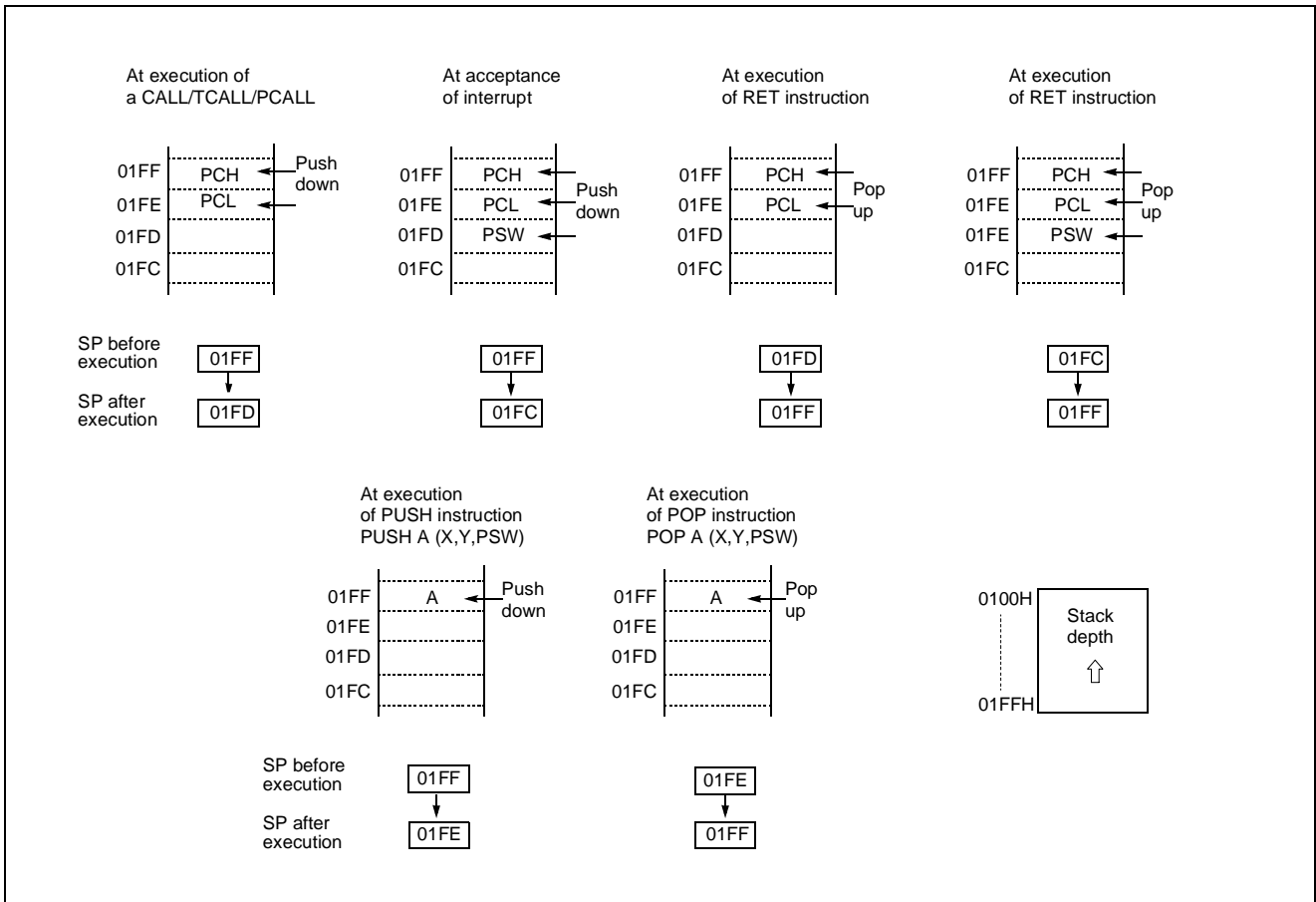


Figure 8-4 Stack Operation

8.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 8K/16K 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.

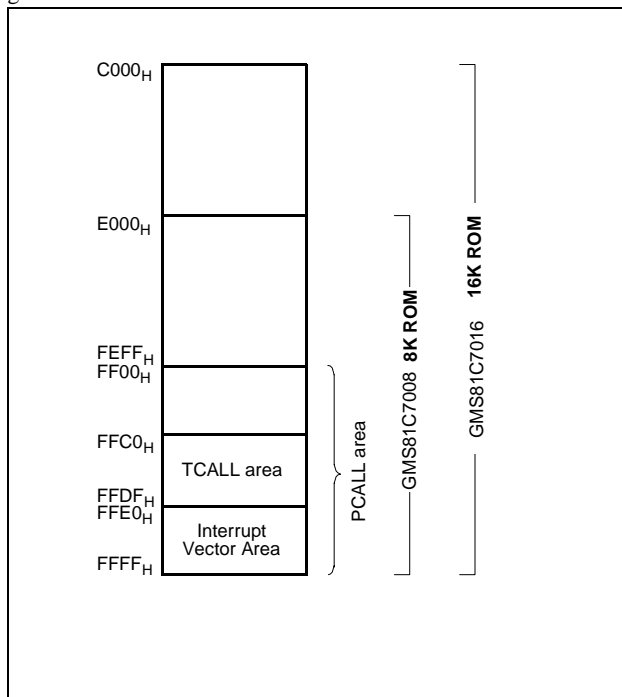


Figure 8-5 Program Memory Map

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

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	Timer/Counter 3
E2	Timer/Counter 2
E4	Watch Timer
E6	A/D Converter
E8	Serial Peripheral Interface
EA	-
EC	-
EE	External Interrupt 2
F0	Timer/Counter 1
F2	Timer/Counter 0
F4	External Interrupt 1
F6	External Interrupt 0
F8	Watchdog Timer
FA	Basic Interval Timer
FC	Key Scan
FE	RESET

NOTE:
"-" means reserved area.

Figure 8-6 Interrupt Vector Area

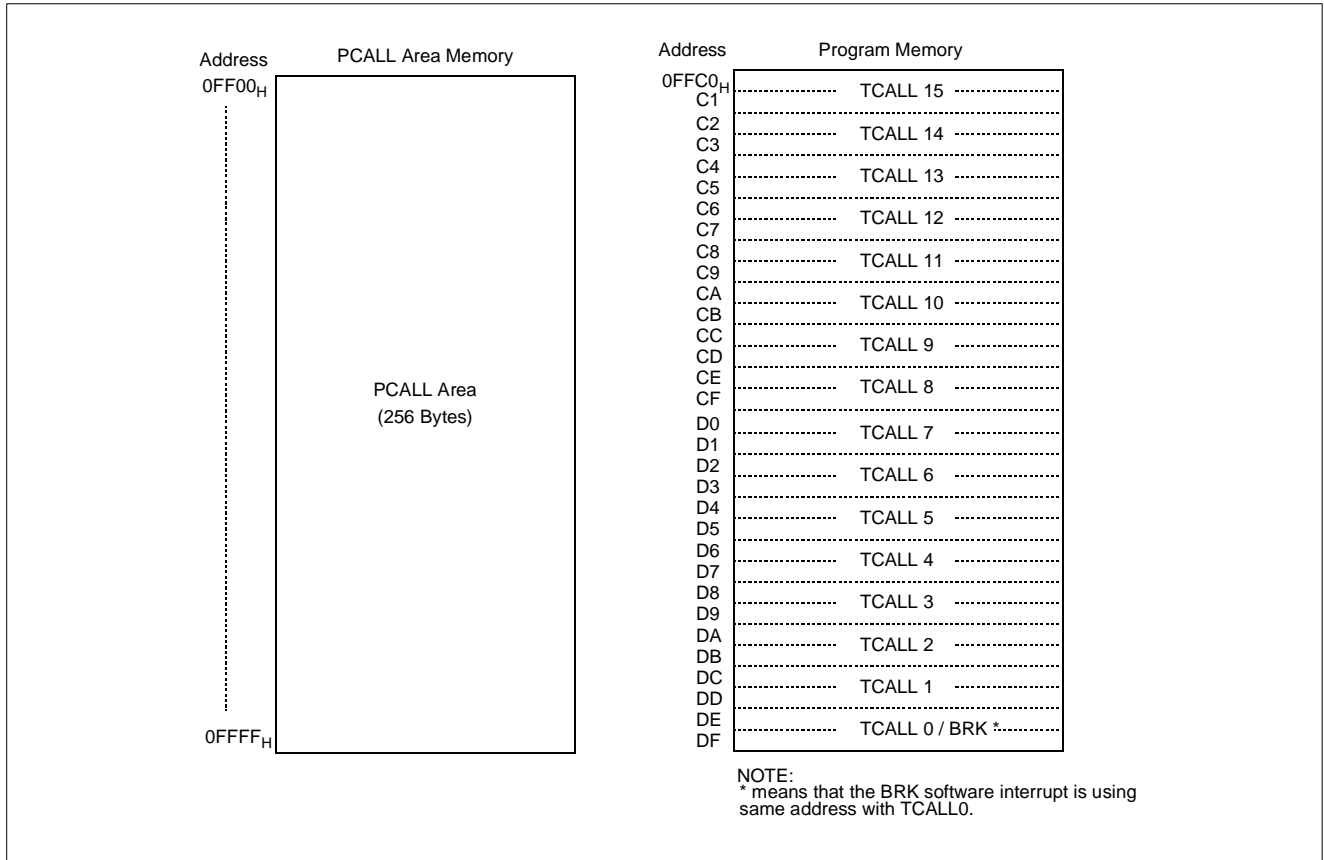
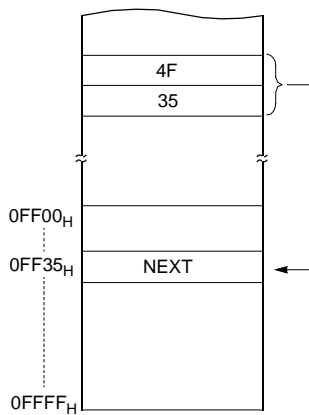


Figure 8-7 PCALL and TCALL Memory Area

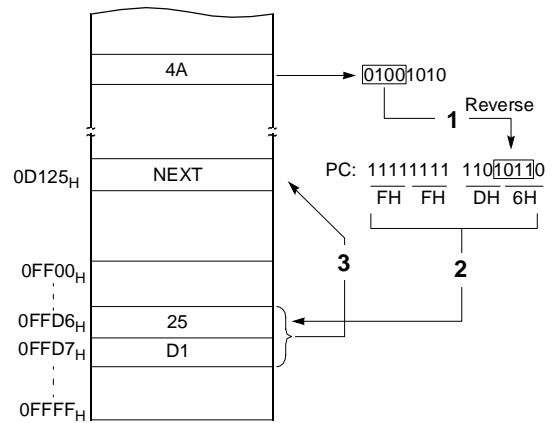
PCALL → rel

4F35 PCALL 35H



TCALL → n

4A TCALL 4



Example: The usage software example of Vector address for GMS81C7016.

```

ORG    0FFE0H

DW     TIMER3           ; Timer-3
DW     TIMER2           ; Timer-2
DW     WATCH_TIMER     ; Watch Timer
DW     ADC              ; ADC
DW     SIO              ; Serial Interface
DW     NOT_USED        ; -
DW     NOT_USED        ; -
DW     INT2            ; Int.2
DW     TIMER1         ; Timer-1
DW     TIMER0         ; Timer-0
DW     INT1           ; Int.1
DW     INT0           ; Int.0
DW     WD_TIMER      ; Watchdog Timer
DW     BIT_TIMER     ; Basic Interval Timer
DW     KEYSKAN       ; Key Scan Timer
DW     RESET         ; Reset

;
ORG    0C000H           ; in case of 16K ROM Start address
;
ORG    0E000H           ; in case of 8K ROM Start address

;*****
;
;      MAIN      PROGRAM      *
;*****
;
RESET:  LDM      SCMR,#0           ;When main clock mode
        DI              ;Disable All Interrupts
        LDM      WDTR,#0         ;Disable Watch Dog Timer
        LDM      RPR,#1
        CLRG
RAM_CLR: LDX      #0
        LDA      #0              ;RAM Clear(!0000H ~ !00BFH)
        STA      {X}+
        CMPX     #0C0H
        BNE     RAM_CLR
        SETG
RAM_CLR1: LDX      #0
        LDA      #0
        STA      {X}+
        CMPX     #1BH           ;DISPLAY RAM Clear(!0100H ~ !011AH)
        BNE     RAM_CLR1
        CLRG
;
        LDX      #0FFH         ;Stack Pointer Initialize
        TXSP
;
        LDM      R0, #0         ;Normal Port 0
        LDM      R0DD,#82H     ;Normal Port Direction
        LDM      ROPU,#0       ;Normal Pull Up
        :
        :
        LDM      TDR0,#250     ;8us x 250 = 2000us
        LDM      TM0,#0000_1111B ;Start Timer0, 8us at 4MHz
        LDM      IRQH,#0
        LDM      IRQL,#0
        LDM      IENH,#0000_1110B ;Enable INT0, INT1, Timer0
        LDM      IENL,#0
        LDM      IEDS,#15H     ;Select falling edge detect on INT pin
        LDM      PMR,#3H       ;Set external interrupt pin(INT0, INT1)
        EI                    ;Enable master interrupt
    
```

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.

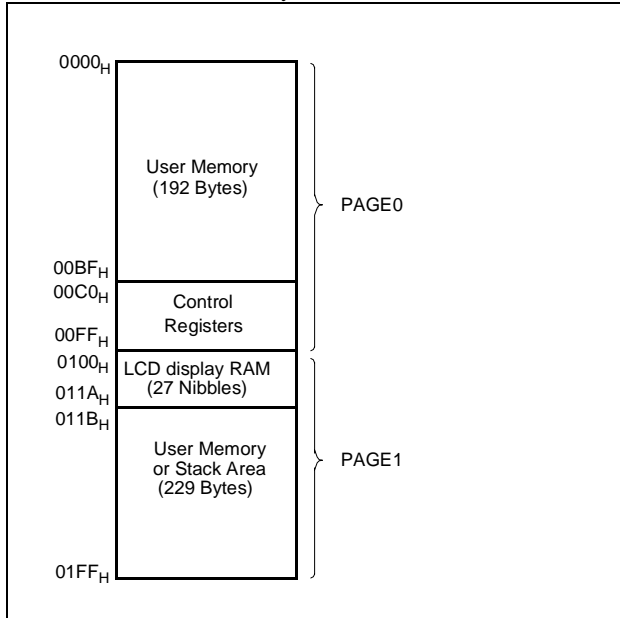


Figure 8-8 Data Memory Map

User Memory

The both GMS81C7008/16 has 448×8 bits for the user memory (RAM).

There are two page internal RAM. Page is selected by G-flag and RAM page selection register RPR. When G-flag is cleared to "0", always page 0 is selected regardless of RPR value. If G-flag is set to "1", page will be selected according to RPR value.

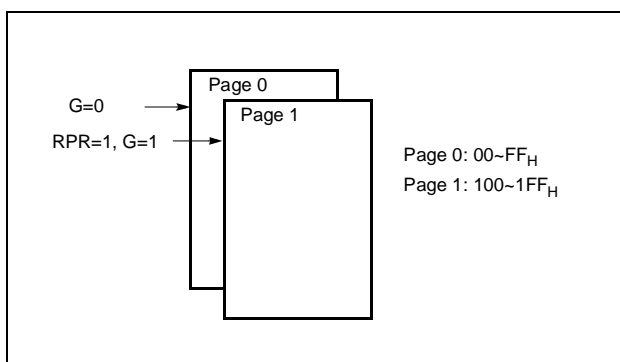


Figure 8-9 RAM page configuration

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 (SET1, CLR1). Do not use read-modify-write instruction. Use byte manipulation instruction, for example "LDM".

Example: To write at CKCTLR

```
LDM CKCTLR,#09H ;Divide ratio(+16)
```

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 22.

8.4 List of Control Registers

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	0	0	0	0	0	0	0	0	0	page 35
00C1	R1 port data register	R1	R/W	-	-	-	-	-	-	0	0	0	page 35
00C2	R2 port data register	R2	R/W	0	0	0	0	0	0	0	0	0	page 35
00C3	R3 port data register	R3	R/W	-	0	0	0	0	0	0	0	0	page 35
00C4	R4 port data register	R4	R/W	0	0	0	0	0	0	0	0	0	page 36
00C5	R5 port data register	R5	R/W	0	0	0	0	0	0	0	0	0	page 36
00C6	R6 port data register	R6	R/W	0	0	0	0	0	0	0	0	0	page 37
00C8	R0 port I/O direction register	R0DD	W	0	0	0	0	0	0	0	0	0	page 37
00C9	R1 port I/O direction register	R1DD	W	-	-	-	-	-	-	0	0	0	page 38
00CA	R2 port I/O direction register	R2DD	W	0	0	0	0	0	0	0	0	0	page 38
00CB	R3 port I/O direction register	R3DD	W	-	0	0	0	0	0	0	0	0	page 37
00CC	R4 port I/O direction register	R4DD	W	0	0	0	0	0	0	0	0	0	page 38
00CD	R5 port I/O direction register	R5DD	W	0	0	0	0	0	0	0	0	0	page 38
00CE	R6 port I/O direction register	R6DD	W	0	0	0	0	0	0	0	0	0	page 38
00D0	R0 port pull-up register	R0PU	W	0	0	0	0	0	0	0	0	0	page 35
00D1	R1 port pull-up register	R1PU	W	-	-	-	-	-	-	0	0	0	page 35
00D2	R2 port pull-up register	R2PU	W	0	0	0	0	0	0	0	0	0	page 35
00D3	R3 port pull-up register	R3PU	W	-	0	0	0	0	0	0	0	0	page 35
00D4	R0 port open drain control register	R0CR	W	0	0	0	0	0	0	0	0	0	page 35
00D5	R1 port open drain control register	R1CR	W	-	-	-	-	-	-	0	0	0	page 35
00D6	R2 port open drain control register	R2CR	W	0	0	0	0	0	0	0	0	0	page 35
00D7	R3 port open drain control register	R3CR	W	-	0	0	0	0	0	0	0	0	page 35
00D8	Ext. interrupt edge selection register	IEDS	R/W	-	-	0	0	0	0	0	0	0	page 71
00D9	Port mode register	PMR	R/W	0	0	0	0	0	0	0	0	0	page 64, page 71
00DA	Interrupt enable lower byte register	IENL	R/W	0	-	-	0	0	0	0	0	0	page 67
00DB	Interrupt enable upper byte register	IENH	R/W	-	0	0	0	0	0	0	0	0	page 67
00DC	Interrupt request flag lower byte register	IRQL	R/W	0	-	-	0	0	0	0	0	0	page 66
00DD	Interrupt request flag upper byte register	IRQH	R/W	-	0	0	0	0	0	0	0	0	page 66
00DE	Sleep mode register	SMR	W	-	-	-	-	-	-	-	0	0	page 83
00DF	Watch dog timer register	WDTR	R/W	-	-	0	1	0	0	1	0	0	page 81
00E0	Timer0 mode register	TM0	R/W	-	-	0	0	0	0	0	0	0	page 47
00E1	Timer0 counter register	T0	R	0	0	0	0	0	0	0	0	0	page 47
	Timer0 data register	TDR0	W	1	1	1	1	1	1	1	1	1	page 47
	Timer0 input capture register	CDR0	R	0	0	0	0	0	0	0	0	0	page 47
00E2	Timer1 mode register	TM1	R/W	0	0	0	0	0	0	0	0	0	page 47

Table 8-1 Control Registers

Address	Register Name	Symbol	R/W	Initial Value								Page	
				7	6	5	4	3	2	1	0		
00E3	Timer1 data register	TDR1	W	1	1	1	1	1	1	1	1	1	page 47
	PWM0 pulse period register	T1PPR	W	1	1	1	1	1	1	1	1	1	page 56
00E4	Timer1 counter register	T1	R	0	0	0	0	0	0	0	0	0	page 47
	Timer1 input capture register	CDR1	R	0	0	0	0	0	0	0	0	0	page 47
	Timer1 pulse duty register	T1PDR	R/W	0	0	0	0	0	0	0	0	0	page 56
00E5	PWM0 high register	PWM0HR	W	-	-	-	-	0	0	0	0	0	page 56
00E6	Timer2 mode register	TM2	R/W	-	-	0	0	0	0	0	0	0	page 48
00E7	Timer2 counter register	T2	R	0	0	0	0	0	0	0	0	0	page 48
	Timer2 data register	TDR2	W	1	1	1	1	1	1	1	1	1	page 48
	Timer2 input capture register	CDR2	R	0	0	0	0	0	0	0	0	0	page 48
00E8	Timer3 mode register	TM3	R/W	0	0	0	0	0	0	0	0	0	page 48
00E9	Timer3 data register	TDR3	W	1	1	1	1	1	1	1	1	1	page 48
	PWM1 pulse period register	T3PPR	W	1	1	1	1	1	1	1	1	1	page 56
00EA	Timer3 counter register	T3	R	0	0	0	0	0	0	0	0	0	page 48
	Timer3 input capture register	CDR3	R	0	0	0	0	0	0	0	0	0	page 48
	Timer3 pulse duty register	T3PDR	R/W	0	0	0	0	0	0	0	0	0	page 48
00EB	PWM1 high register	PWM1HR	W	-	-	-	-	0	0	0	0	0	page 56
00EC	A/D converter mode register	ADCM	R/W	-	0	0	0	0	0	0	0	1	page 60
00ED	A/D converter data register	ADR	R	Undefined								page 60	
00EF	Watch timer mode register	WTMR	R/W	-	0	-	-	0	0	0	0	0	page 81
00F0	Key scan port mode register	KSMR	R/W	-	-	-	-	-	-	0	0	0	page 71
00F1	LCD control register	LCR	R/W	0	0	0	0	0	0	0	0	0	page 73
00F2	LCD port mode register high	LPMR	R/W	-	-	0	0	0	0	0	0	0	page 73
00F3	RAM paging register	RPR	R/W	-	-	-	-	-	-	0	0	0	page 26, page 73
00F4	Basic interval timer register	BITR	R	0	0	0	0	0	0	0	0	0	page 45
	Clock control register	CKCTLR	W	-	-	-	0	0	1	1	1	1	page 45
00F5	System clock mode register	SCMR	R/W	0	0	0	0	0	0	0	0	0	page 40
00FB	LVD register	LVDR	R/W	0	0	0	0	0	-	-	-	-	page 89
00FD	Buzzer data register	BUR	W	0	0	0	0	0	0	0	0	0	page 64
00FE	Serial I/O mode register	SIOM	R/W	0	0	0	0	0	0	0	0	1	page 61
00FF	Serial I/O Data register	SIOR	R/W	Undefined								page 61	

Table 8-1 Control Registers

W

Registers are controlled by byte manipulation instruction such as LDM etc., do not use bit manipulation instruction such as SET1, CLR1 etc. If bit manipulation instruction is used on these registers, content of other seven bits are may varied to unwanted value.

R/W

Registers are controlled by both bit and byte manipulation instruction.

- : this bit location is reserved.

Three registers are mapped on same address.

Address	Timer/Counter mode	Capture mode	PWM mode
E1 _H	T0 [R], TDR0 [W]	CDR0 [R], TDR0 [W]	-
E3 _H	TDR1 [W]	TDR1 [W]	T1PPR [W]
E4 _H	T1 [R]	CDR1 [R]	T1PDR [R/W]
E7 _H	T2 [R], TDR2 [W]	CDR2 [R], TDR2 [W]	-
E9 _H	TDR3 [W]	TDR3 [W]	T3PPR [W]
EA _H	T3 [R]	CDR3 [R]	T3PDR [R/W]

Two registers are mapped on same address.

Address	Basic Interval Timer
F4 _H	BITR [R], CKCTLR [W]

8.5 Addressing Mode

The GMS800 series MCU uses six addressing modes;

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

(1) Register Addressing

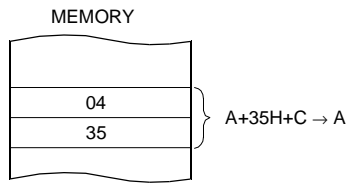
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

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

Example:

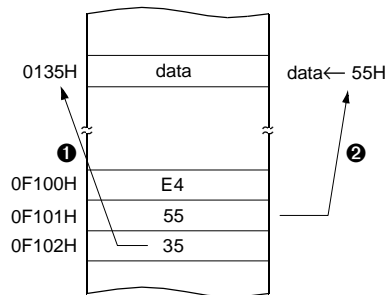
```
0435   ADC   #35H
```



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

Example: G=1, RPR=01

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

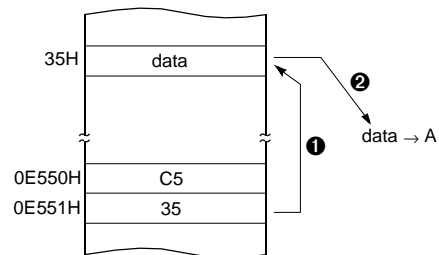


(3) Direct Page Addressing → dp

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

Example; G=0

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



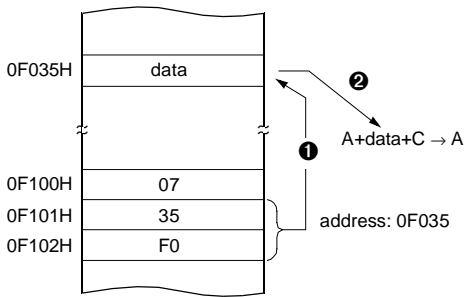
(4) Absolute Addressing → !abs

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;

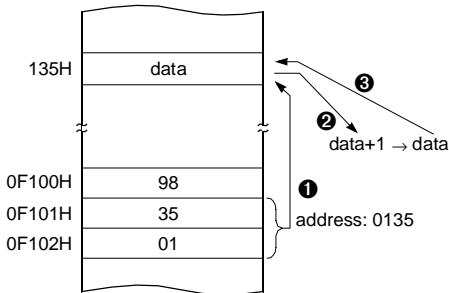
```
0735F0 ADC !0F035H ;A ←ROM[0F035H]
```



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

Example; Addressing accesses the address 0135H regardless of G-flag.

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



(5) Indexed Addressing

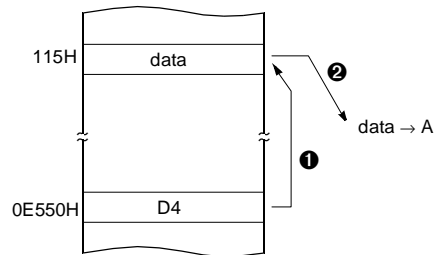
X indexed direct page (no offset) → {X}

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

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

Example; X=15H, G=1

```
D4 LDA {X} ;ACC←RAM[X]
```



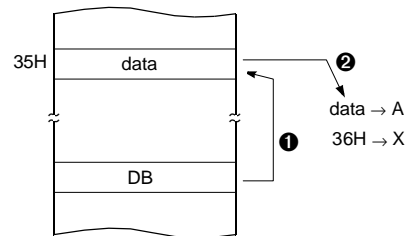
X indexed direct page, auto increment → {X}+

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=35H

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



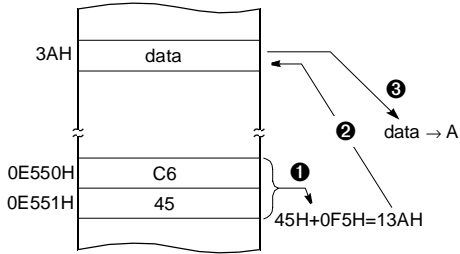
X indexed direct page (8 bit offset) → dp+X

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=0F5H

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.

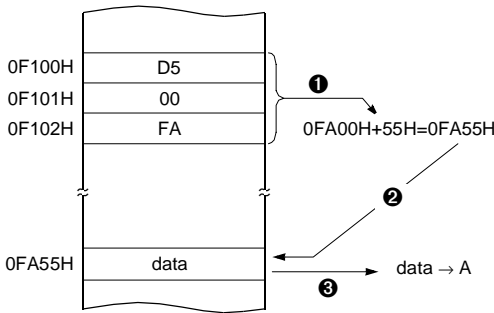
This is same with above (2). Use Y register instead of X.

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=55H

D500FA LDA !0FA00H+Y



(6) Indirect Addressing

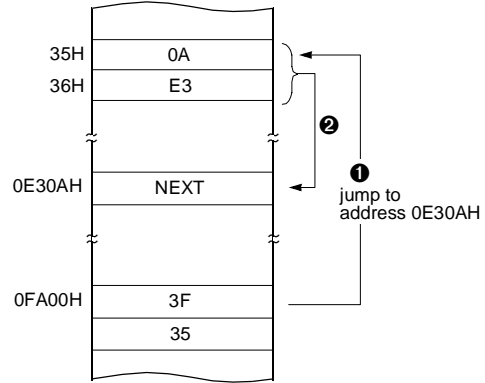
Direct page indirect → [dp]

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]



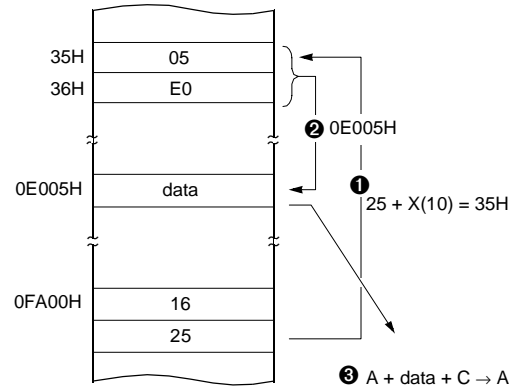
X indexed indirect → [dp+X]

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=10H

1625 ADC [25H+X]



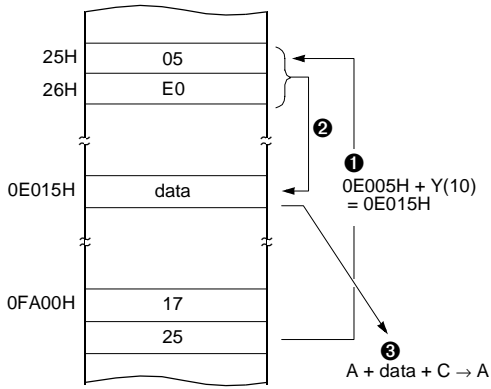
Y indexed indirect → [dp]+Y

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

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

Example; G=0, Y=10H

1725 ADC [25H]+Y



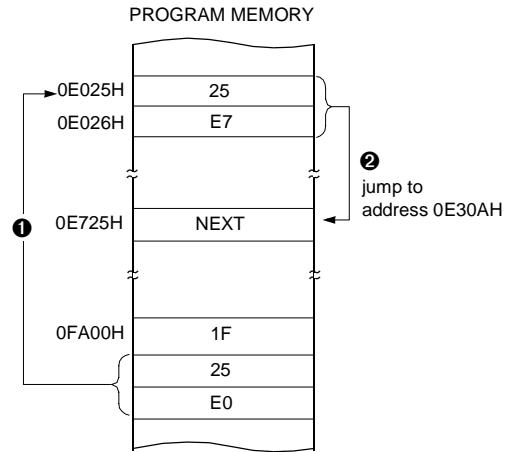
Absolute indirect → [!abs]

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

JMP

Example; G=0

1F25E0 JMP [!0E025H]



9. I/O PORTS

The GMS81C7008/16 has seven ports (R0, R1, R2, R3, R4, R5 and R6), and LCD segment port SEG0~SEG23, and LCD common port COM0~COM3, which are multiplexed with SEG24~SEG26.

9.1 Registers for Port

Port Data Registers

The Port Data Registers in I/O buffer in each seven ports (R0,R1,R2,R3,R4,R5,R6) 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.

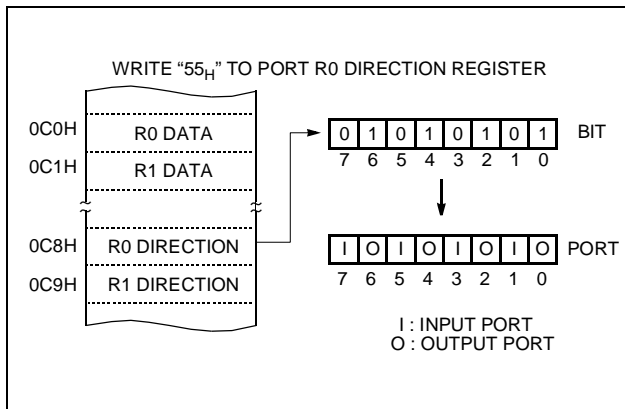


Figure 9-1 Example of port I/O assignment

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

Pull-up Control Registers

The R0, R1, R2 and R3 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 (PURn). The value of that resistor is typically 180kΩ.

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, R3 ports are used as a general purpose input port and R4, R5, R6 and R7 ports are used as LCD segment drive output port.

When a port is used as input, input logic is firmly either low or high, therefore external pull-down or pull-up resistors are required practically. The GMS81C7008/16 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 R0PU, R1PU, R2PU and R3PU.

When ports are configured as inputs and pull-up resistor is selected by software, they are pulled to high.

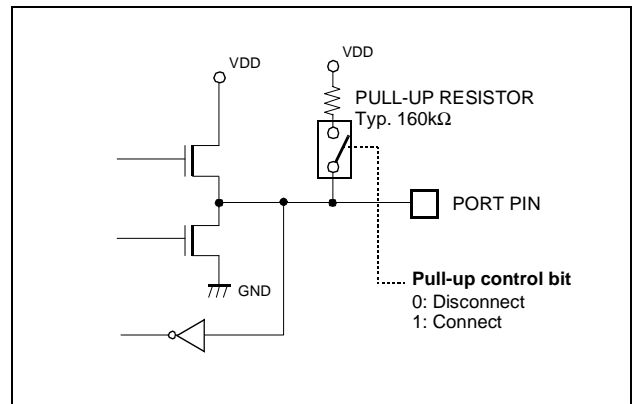


Figure 9-2 Pull-up Port Structure

Open drain port Registers

The R0, R1, R2 and R3 ports have open drain port resistors R0CR~R3CR.

Figure 9-3 shows a open drain port configuration by control register. It is selected as either push-pull port or open-drain port by R0CR, R1CR, R2CR and R3CR.

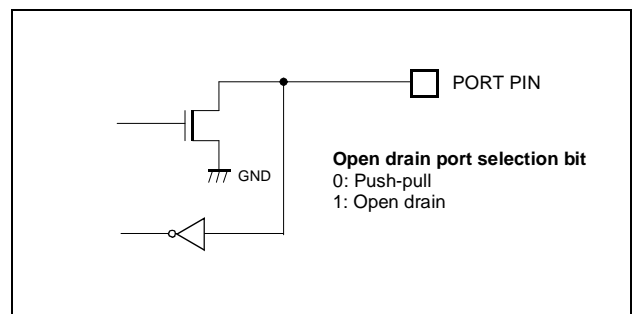


Figure 9-3 Open-drain Port Structure

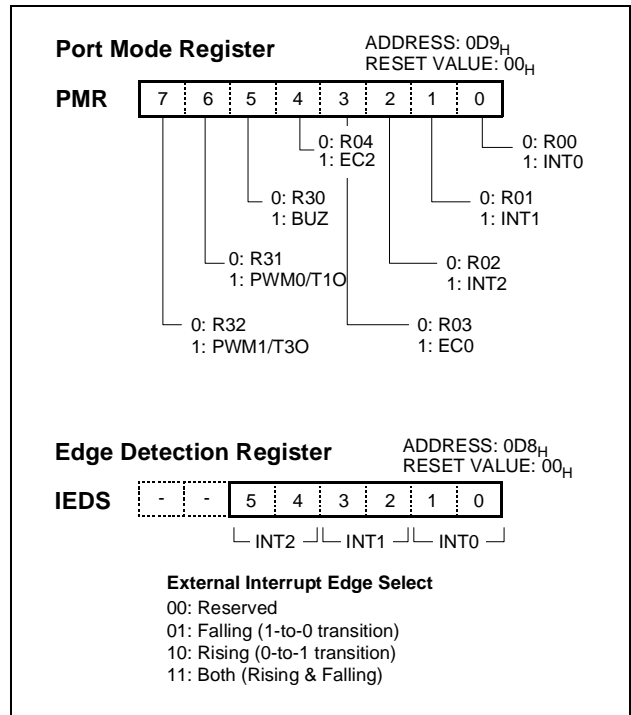
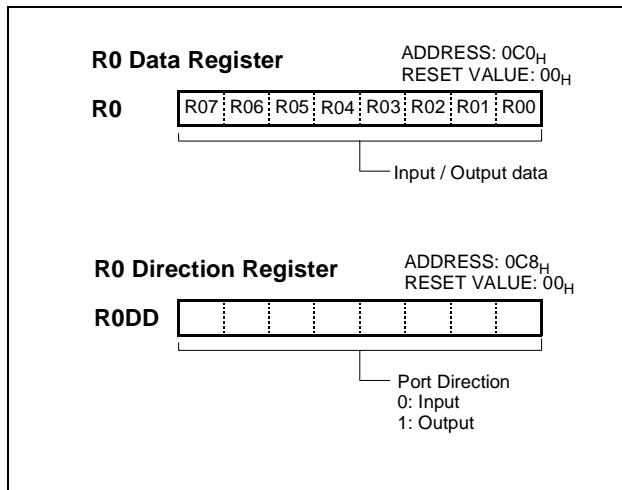
9.2 I/O Ports Configuration

R0 and R0DD register: 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). Each port also can be set individually as pull-up port through the ROPU (address 0D0_H), and as open drain register through the ROCR (address 0D4_H).

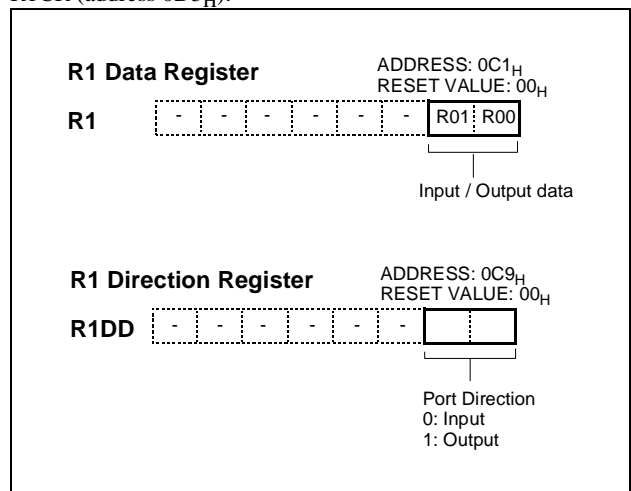
In addition, port R0 is multiplexed with various special features. The control register through the PMR (address 0D9_H) and the SIOM (address 0FE_H) control the selection of alternate function. After reset, this value is “0”, port may be used as normal I/O port. To use alternate function such as external interrupt, event counter input, serial interface data input, serial interface data output or serial interface clock, write “1” in the corresponding bit of PMR (address 0D9_H) and SIOM (address 0FE_H).

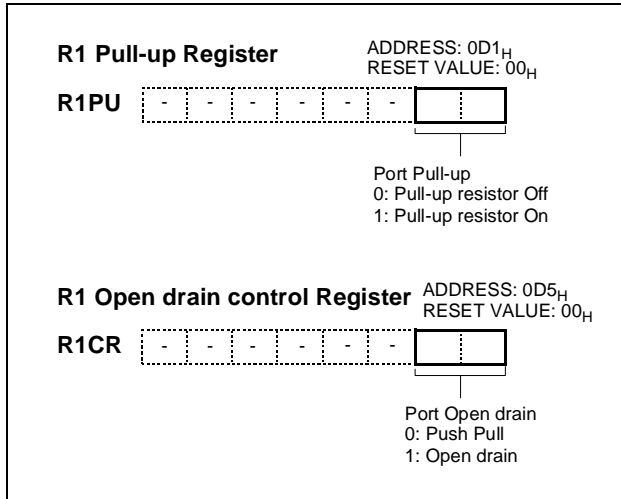
Port pin	Alternate function
R00	INT0 (External interrupt 0)
R01	INT1 (External interrupt 1)
R02	INT2 (External interrupt 2)
R03	EC0 (Event counter input 0)
R04	EC2 (Event counter input 2)
R05	SCK (Serial clock)
R06	SO (Serial data output)
R07	SI (Serial data input)

Regardless of the direction register R0DD, the control registers of PMR and SIOM are selected to use as alternate functions, port pin can be used as a corresponding alternate features



R1 and R1DD register: R1 is a 2-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). Each port also can be set individually as pull-up port through the R1PU (address 0D1_H), and as open drain register through the R1CR (address 0D5_H).

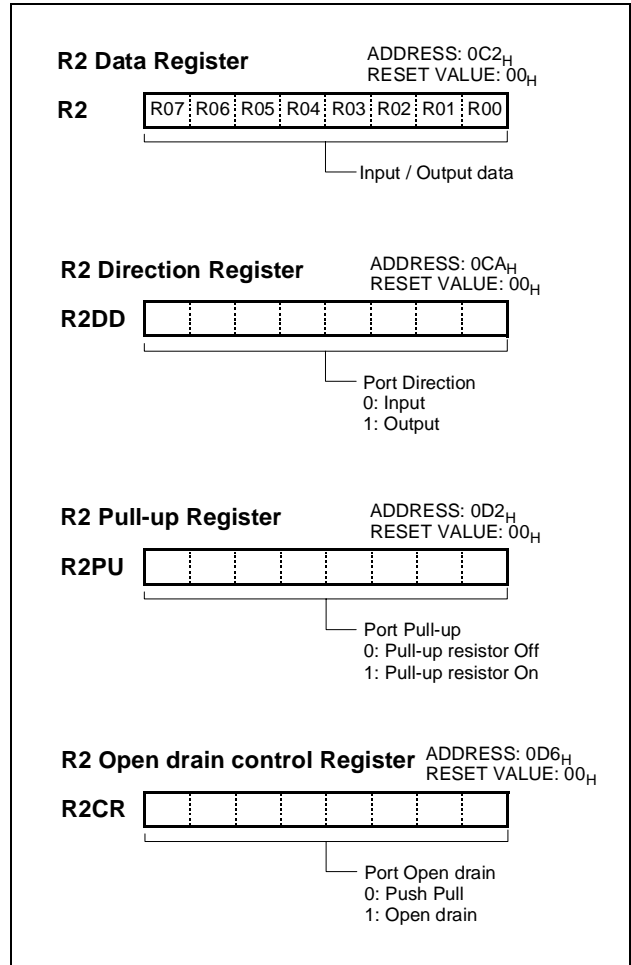




R2 and R2DD register: R2 is an 8-bit CMOS bidirectional I/O port (address 0C2_H). Each I/O pin can independently used as an input or an output through the R2DD register (address 0CA_H). Each port also can be set individually as pull-up port through the R2PU (address 0D2_H), and as open drain register through the R2CR (address 0D6_H).

In addition, port R2 is multiplexed with analog input port.

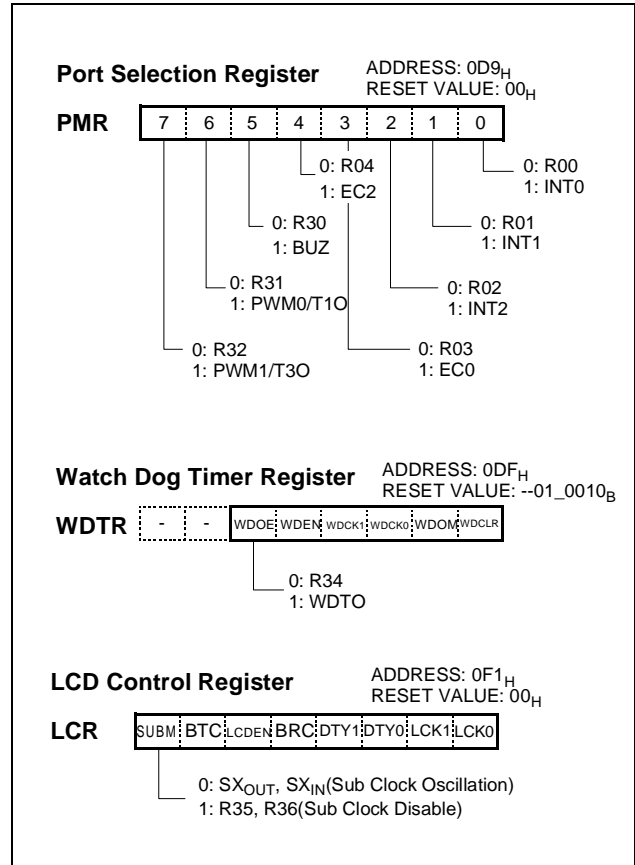
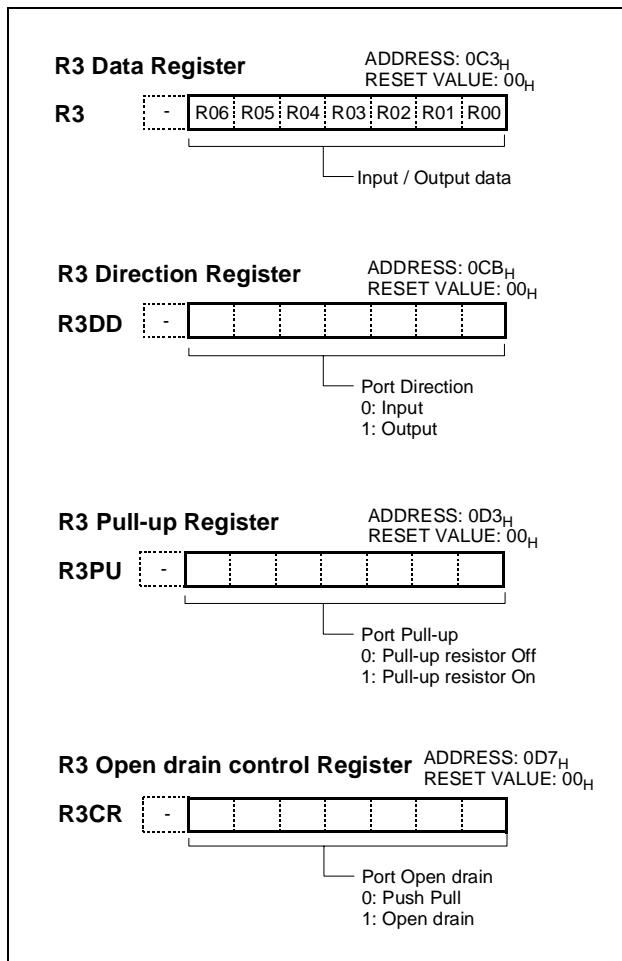
Port pin	Alternate function
R20	AN0 (Analog Input 0)
R21	AN1 (Analog Input 1)
R22	AN2 (Analog Input 2)
R23	AN3 (Analog Input 3)
R24	AN4 (Analog Input 4)
R25	AN5 (Analog Input 5)
R26	AN6 (Analog Input 6)
R27	AN7 (Analog Input 7)



R3 and R3DD register: R3 is an 8-bit CMOS bidirectional I/O port (address 0C3_H). Each I/O pin can independently used as an input or an output through the R3DD register (address 0CB_H). Each port also can be set individually as pull-up port through the R3PU (address 0D3_H), and as open drain register through the R3CR (address 0D7_H).

In addition, port R3 is multiplexed with various special features.

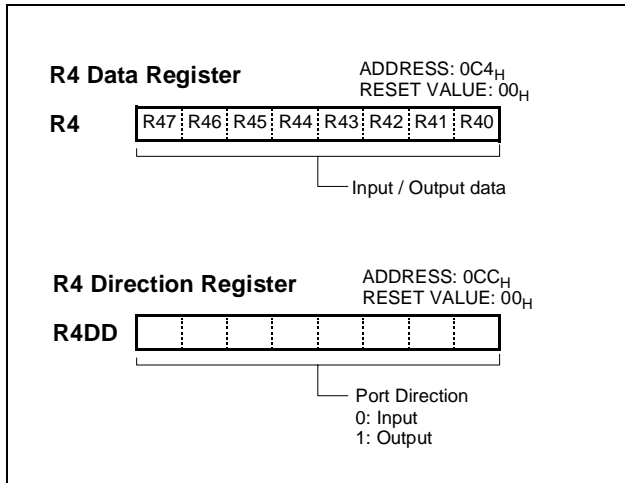
Port pin	Alternate function
R30	BUZ (Buzzer driving output)
R31	PWM0 / T1O (PWM 0 output / Timer 1 output)
R32	PWM1 / T3O (PWM 1 output / Timer 3 output)
R33	-
R34	WDTO (Watchdog timer output)
R35	SX _{OUT} (Sub clock output)
R36	SX _{IN} (Sub clock input)



R4 and R4DD register: 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).

After Reset, R4 port is used as LCD segment output SEG0~SEG7. To use general I/O ports user should be written appropriate value into the LPMR (0F3_H).

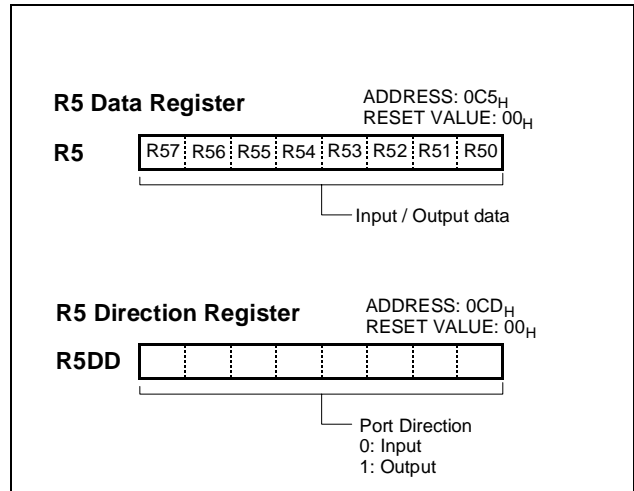
LCD pin function	Port pin
SEG0 (LCD segment 0 signal output)	R40
SEG1 (LCD segment 1 signal output)	R41
SEG2 (LCD segment 2 signal output)	R42
SEG3 (LCD segment 3 signal output)	R43
SEG4 (LCD segment 4 signal output)	R44
SEG5 (LCD segment 5 signal output)	R45
SEG6 (LCD segment 6 signal output)	R46
SEG7 (LCD segment 7 signal output)	R47



R5 and R5DD register: 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).

After Reset, R5 port is used as LCD segment output SEG8~SEG15. To use general I/O ports user should be written appropriate value into the LPMR (0F3_H).

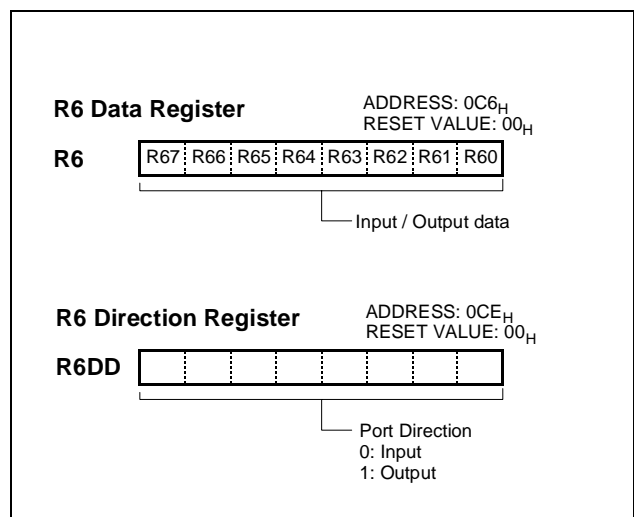
LCD pin function	Port pin
SEG8 (LCD segment 8 signal output)	R50
SEG9 (LCD segment 9 signal output)	R51
SEG10 (LCD segment 10 signal output)	R52
SEG11 (LCD segment 11 signal output)	R53
SEG12 (LCD segment 12 signal output)	R54
SEG13 (LCD segment 13 signal output)	R55
SEG14 (LCD segment 14 signal output)	R56
SEG15 (LCD segment 15 signal output)	R57



R6 and R6DD register: 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).

After Reset, R6 port is used as LCD segment output SEG16~SEG23. To use general I/O ports user should be written appropriate value into the LPMR (0F3_H).

LCD pin function	Port pin
SEG16 (LCD segment 16 signal output)	R60
SEG17 (LCD segment 17 signal output)	R61
SEG18 (LCD segment 18 signal output)	R62
SEG19 (LCD segment 19 signal output)	R63
SEG20 (LCD segment 20 signal output)	R64
SEG21 (LCD segment 21 signal output)	R66
SEG22 (LCD segment 22 signal output)	R66
SEG23 (LCD segment 23 signal output)	R67



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	
	X _{IN} = 4MHz	SX _{IN} = 32.768kHz
÷ 2	0.5 us	61 us
÷ 8	2.0 us	244 us
÷ 16	4.0 us	488 us
÷ 64	16.0 us	1953 us

The register is 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 STOP instruction.

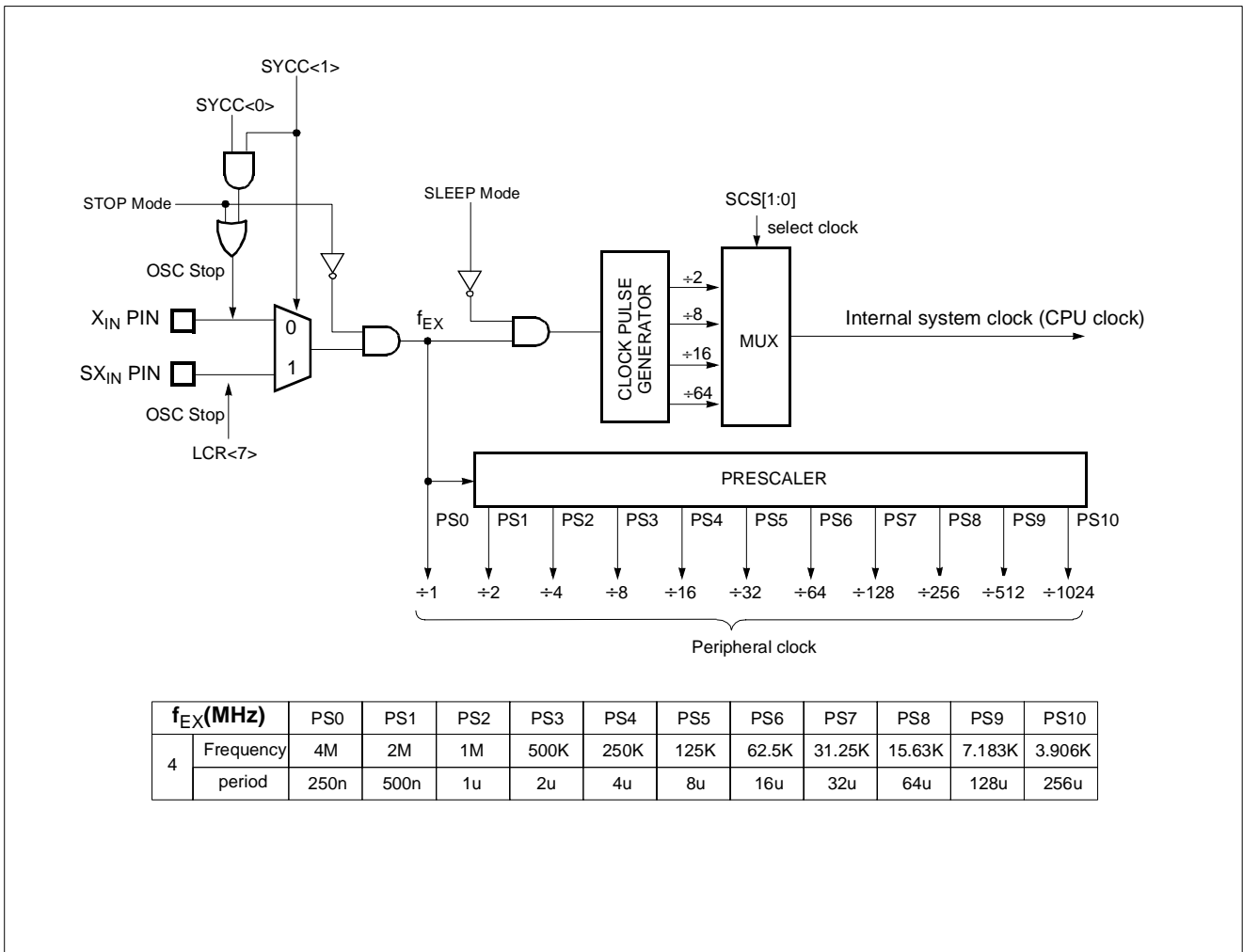


Figure 10-1 Block Diagram of Clock Generator

The system clock is decided by bit1 (SYCC1) of the system clock mode register(SCMR). In selection Sub clock, to oscillate or stop the Main clock is decided by bit0 (SYCC0) of SCMR. On the ini-

tial reset, internal system clock is PS1 which is the fastest and other clock can be provided by bit2 and bit3 of SCMR.

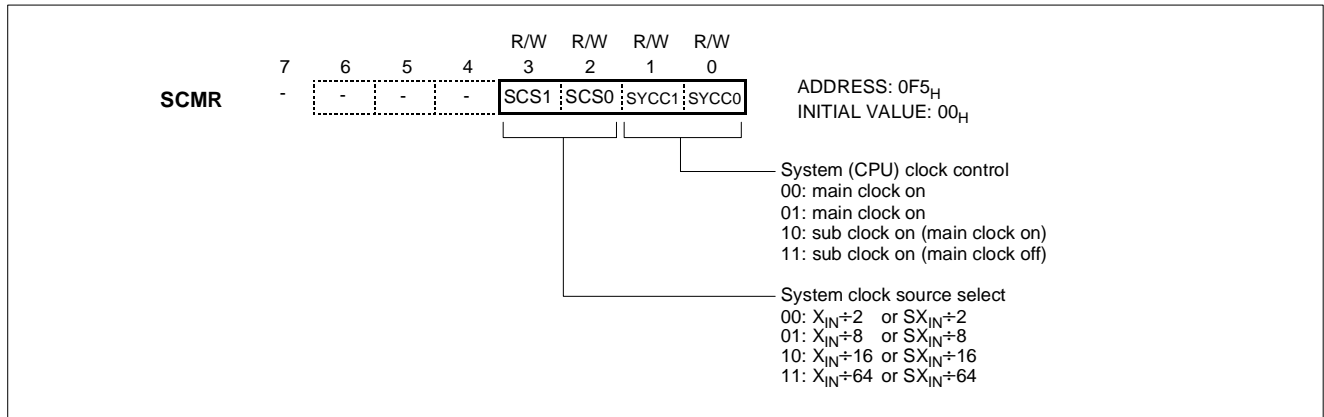


Figure 10-2 SCMR: System Clock Control Registers

11. 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 11-1 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 and low-frequency clock oscillation is active 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.

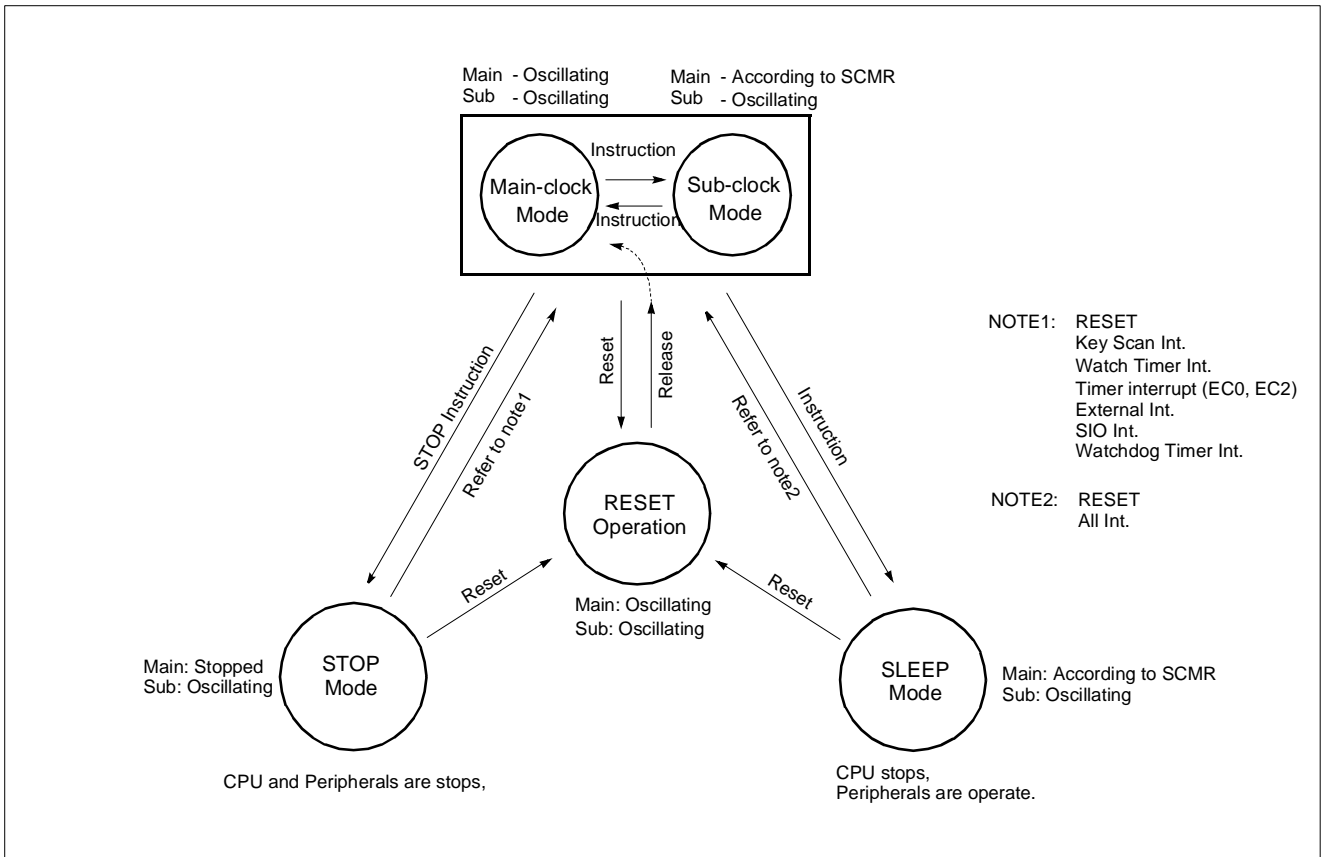


Figure 11-1 Operating Mode

11.1 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 “10_B” into lower 2 bits of SCMR to switch the main system clock to the sub-frequency clock.

Next, write “11_B” to turn off main frequency oscillation.

Example:

```

:
:
MOV SCMR, #0000_XX10B ; Switch to sub mode
MOV SCMR, #0000_XX11B ; Turn off main clock
:
:

```

Returning from sub clock operation to main clock operation

First, write “10_B” 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 GMS81C7008/16 is placed in main frequency operation mode.

Example:

```

:
:
:
MOV SCMR, #0000_XX10B ; Turn on main-clock
CALL DELAY ; Wait until stable
MOV SCMR, #0000_XX00B ; Move to main mode
:
:
:

```

```
;20ms software delay at fXIN=4MHz
```

```

DELAY: 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 operate normally.

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

For more detail, See "20.1 SLEEP Mode" on page 83

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 (EC0, EC2 pin), and External Interrupt.

For more details, see "20.2 STOP Mode" on page 84.

Note: In the STOP and Sub clock 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.

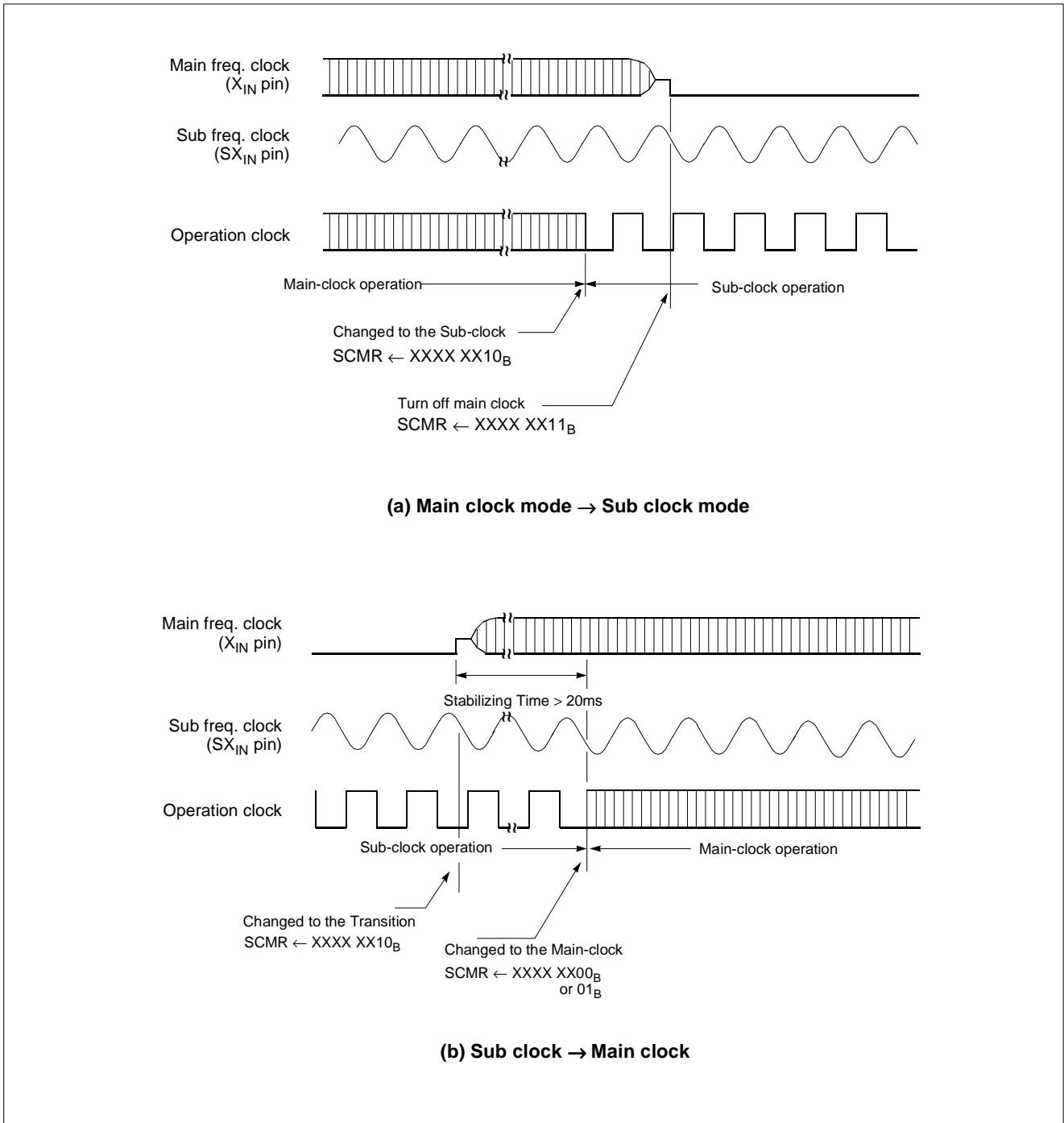


Figure 11-2 System Clock Switching Timing

12. BASIC INTERVAL TIMER

The GMS81C7008/16 has one 8-bit Basic Interval Timer that is free-run and can not stop. Block diagram is shown in Figure 12-1.

In addition, the Basic Interval Timer generates the time base for watchdog timer counting. 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 12-2.

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

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

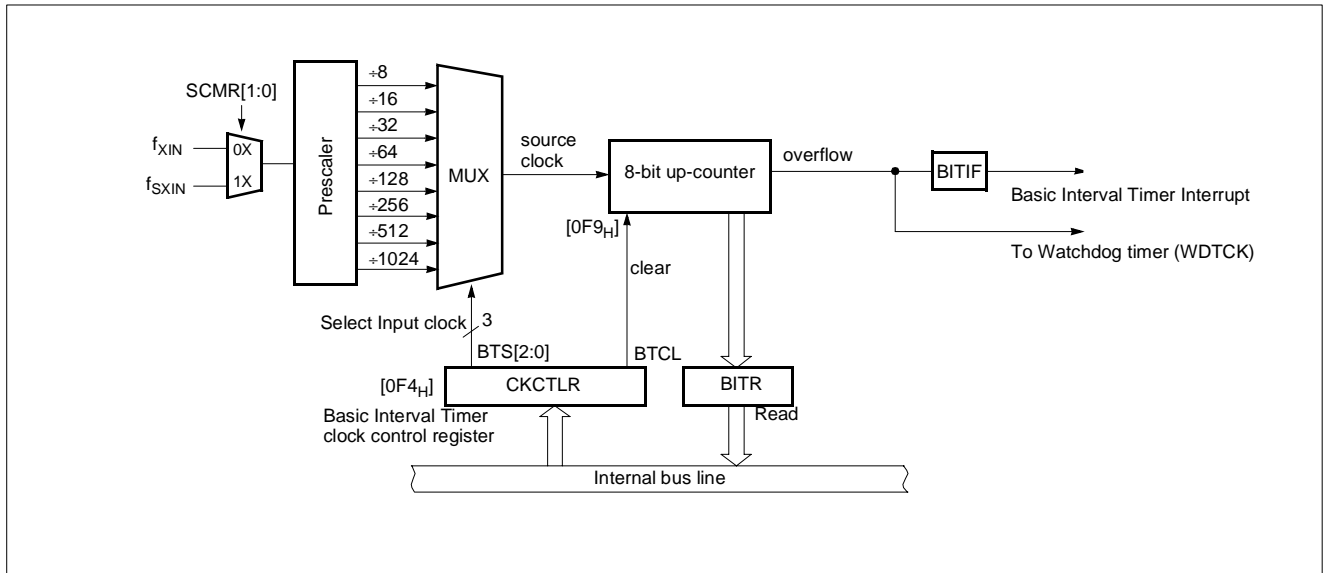


Figure 12-1 Block Diagram of Basic Interval Timer

BTS[2:0]	CPU Source clock	Interrupt (overflow) Period (ms)	
		@ f _{XIN} = 4MHz	@ f _{SXIN} = 32.768kHz
000	÷ 8	0.512	62.5ms
001	÷16	1.024	125ms
010	÷32	2.048	250ms
011	÷64	4.096	500ms
100	÷128	8.192	1000ms
101	÷256	16.384	2000ms
110	÷512	32.768	4000ms
111	÷1024	65.536	8000ms

Table 12-1 Basic Interval Timer Interrupt Time

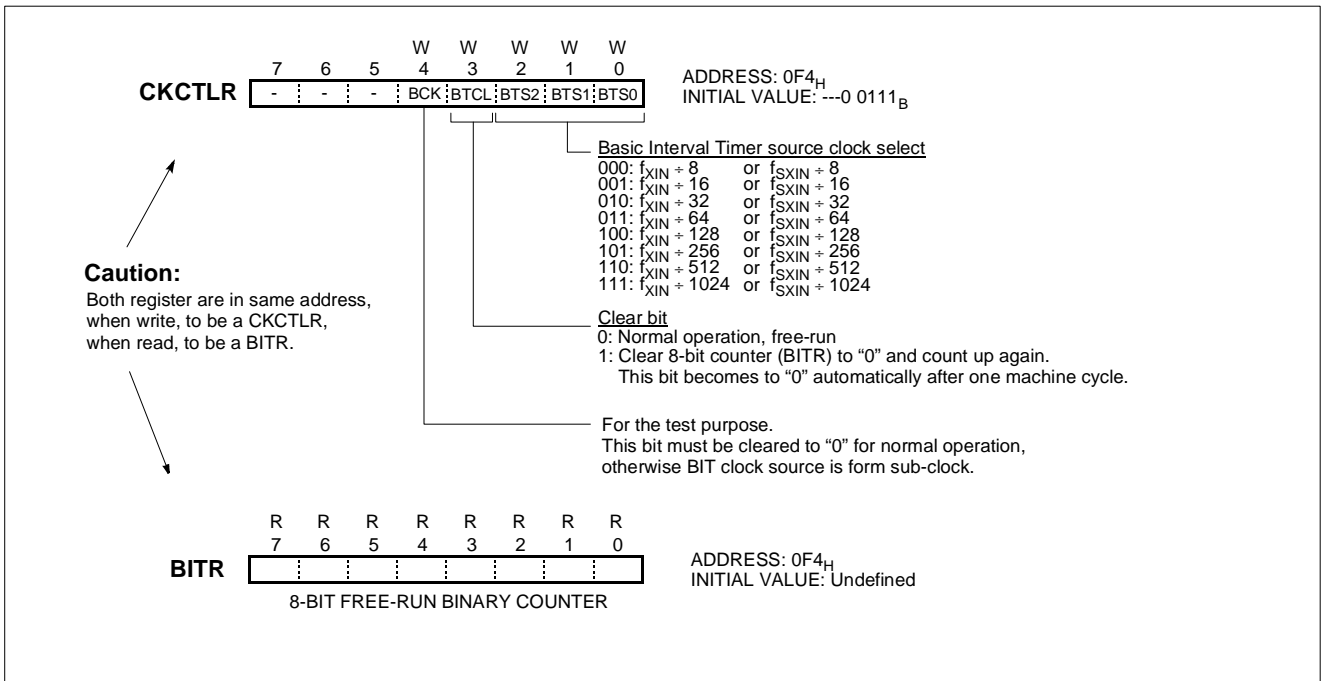


Figure 12-2 BITR: Basic Interval Timer Mode Register

Example 1:

Interrupt request flag is generated every 8.192ms at 4MHz.

```

:
LDM   CKCTLR, #0CH
SET1  BITE
EI
:
    
```

13. TIMER/EVENT COUNTER

The GMS81C7008/16 has four Timer/Event counters. Each module can generate an interrupt to indicate that an event has occurred (i.e. timer match).

Timer 0 and Timer 1 can be used either two 8-bit Timer/Counter or one 16-bit Timer/Counter with combine them. Also Timer 2 and Timer 3 can be joined as a 16-bit Timer/Counter.

In the “timer” function, the register is increased every internal clock input. Thus, one can think of it as counting internal clock input. The count rate is 1/2 to 1/2048 of the oscillator frequency.

In the “counter” function, the register is incremented in response to a 0-to-1 (rising edge) transition at its corresponding external input pin, EC0 or EC2 pin.

Example 1:

Timer 0 = 8-bit timer mode, 8ms interval at 4MHz

Timer 1 = 8-bit timer mode, 4ms interval at 4MHz

Timer 2 = 16-bit event counter mode

```
LDM   SCMR,#0      ;Main clock mode
LDM   TDR0,#249
LDM   TM0,#0001_0011B
LDM   TDR1,#124
LDM   TM1,#0000_1111B

LDM   TDR2,#1FH
LDM   TDR3,#4CH
LDM   TM2,#0001_1111B
LDM   TM3,#0100_1100B

SET1  T0E
SET1  T2E
EI
:
:
```

Example 2:

Timer0 = 16-bit timer mode, 0.5s at 4MHz

Timer2 = 2ms 8-bit timer mode at 4MHz

Timer3 = 250us 8-bit timer mode at 4MHz

```
LDM   SCMR,#0      ;Main clock mode
LDM   TDR0,#23H
LDM   TDR1,#0F4H
LDM   TM0,#0FH     ;FXIN/32, 8us
LDM   TM1,#4CH

LDM   TDR2,#249
LDM   TDR3,#124
LDM   TM2,#0FH     ;FXUN/32, 8us
LDM   TM3,#0DH     ;FXIN/8, 2us

SET1  T0E
SET1  T2E
SET1  T3E
EI
:
:
```

In addition the “capture” function, the register is incremented in response external or internal clock sources same with timer or counter function. When external clock edge input, the count register is captured into Capture data register correspondingly.

It has five operating modes: “8-bit timer/counter”, “16-bit timer/counter”, “8-bit capture”, “16-bit capture”, “PWM mode” which are selected by bit in Timer mode register TM n .

In operation of Timer 2, Timer 3, their operations are same with Timer 0, Timer 1, respectively.

When programming the software, you may refer to following example.

Example 3:

Timer0 = 8-bit timer mode, 2ms interval at 4MHz

Timer1 = 8-bit capture mode, 2us sampling count.

```
LDM   TDR0,#249    ;250x8=2000us
LDM   TM0,#0FH     ;FXIN/32, 8us

LDM   IEDS,#XXXX_01XXB ;FALLING
LDM   PMR,#XXXX_XX1XB  ;AS INT1
LDM   TDR1,#0FFH
LDM   TM1,#0001_1011B ;2us

SET1  T0E          ;ENABLE TIMER 0
SET1  T1E          ;ENABLE TIMER 1
SET1  INT1E        ;ENABLE EXT. INT1
EI
:
:
```

X: don't care.

Example 4:

Timer0 = 8-bit timer mode, 2ms interval at 4MHz

Timer2 = 16-bit capture mode, 8us sampling count.

```
LDM   TDR0,#249
LDM   TM0,#0FH

LDM   IEDS,#XX11_XXXXB
LDM   PMR4,#XXXX_X1XXB
LDM   TDR2,#0FFH           ;MAX
LDM   TDR3,#0FFH           ;MAX
LDM   TM2,#XX10_1111B     ;/32
LDM   TM3,#X10X_11XXB

SET1  T0E          ;ENABLE TIMER 0
SET1  T2E          ;ENABLE TIMER 2
SET1  INT2E        ;ENABLE EXT. INT2
EI
:
:
```

X: don't care.

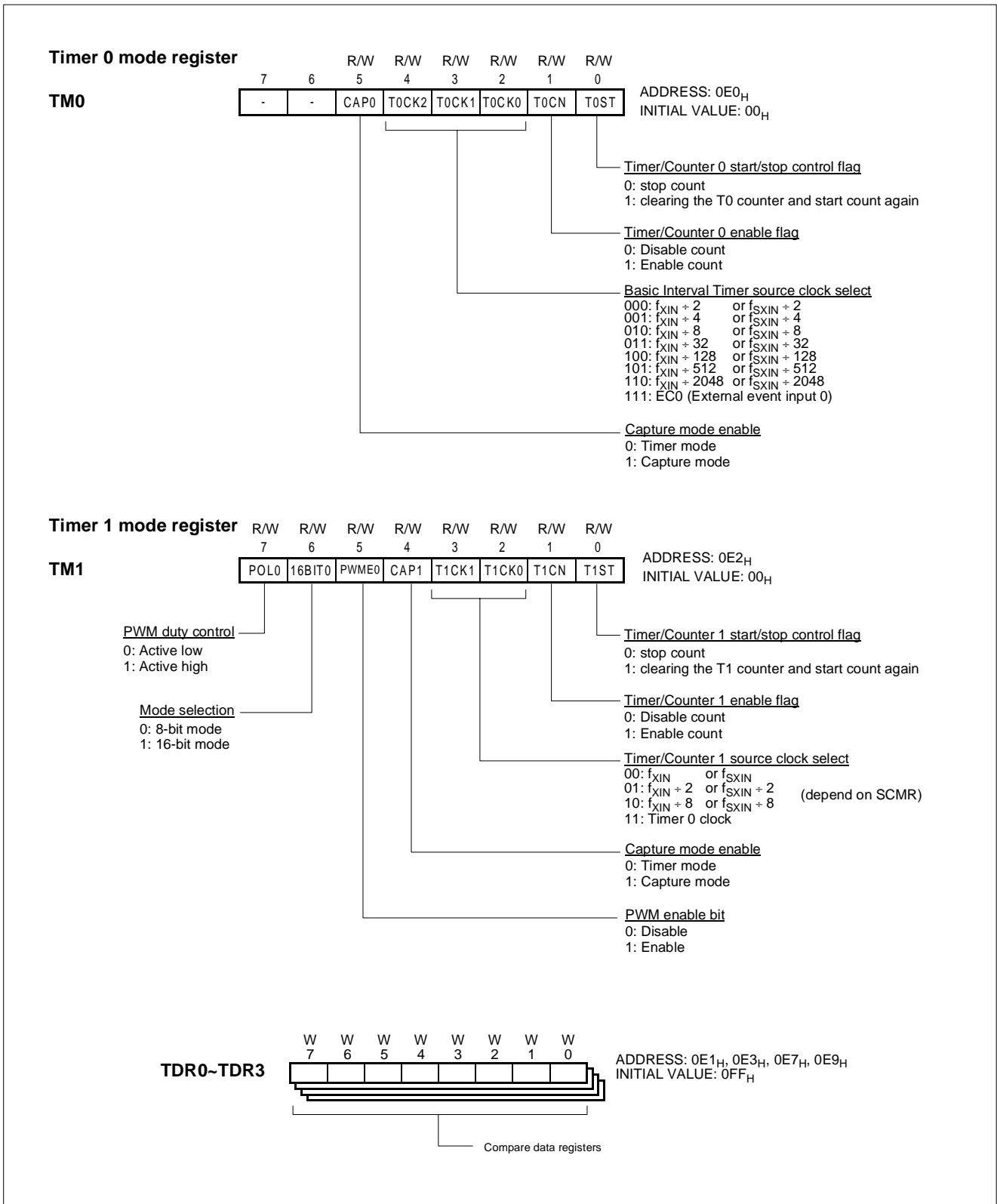


Figure 13-1 TM0, TM1, TDRn Registers

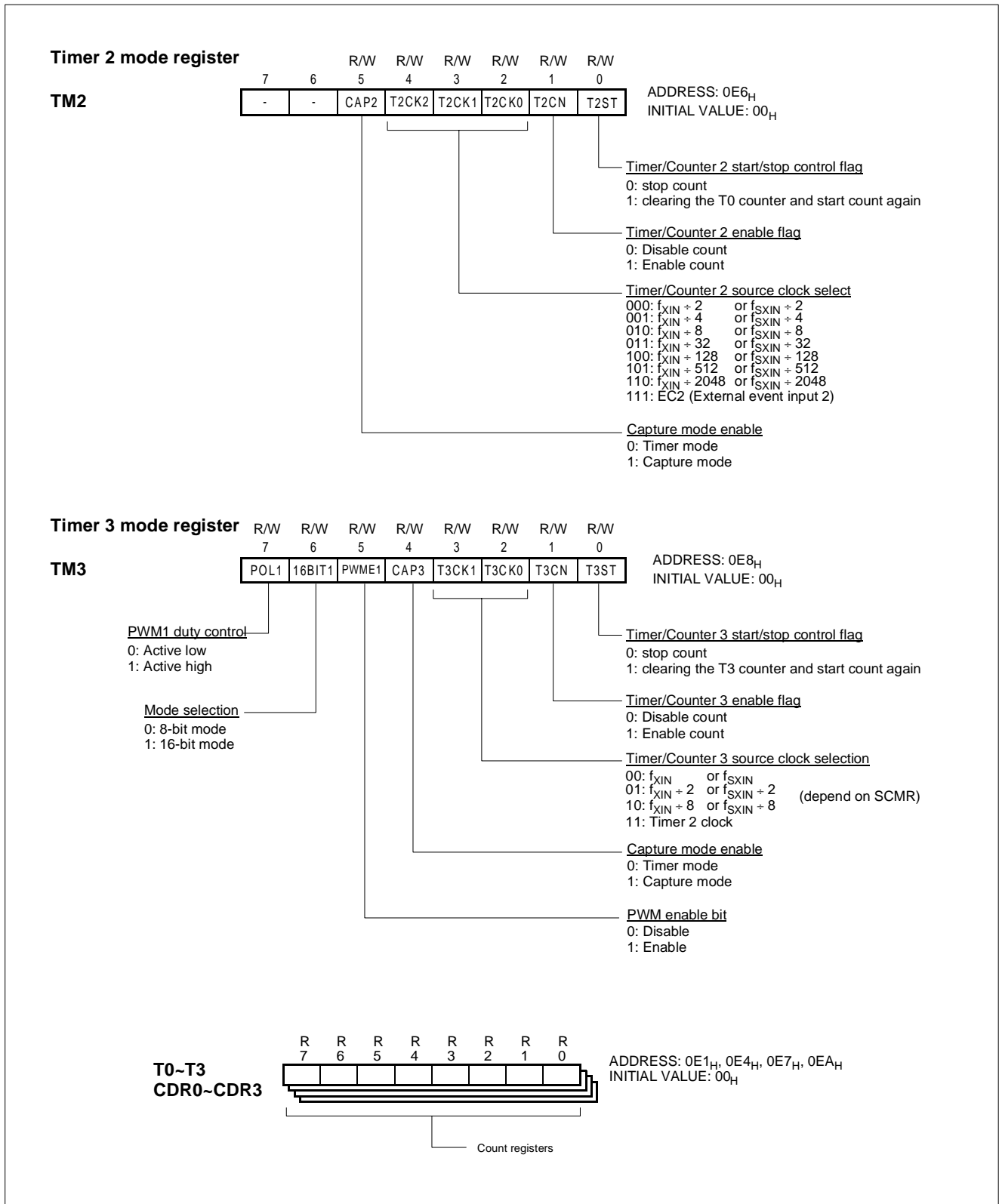


Figure 13-2 TM2, TM3 Registers

13.1 8-bit Timer / Counter Mode

The GMS81C7008/16 has four 8-bit Timer/Counters, Timer 0, Timer 1, Timer 2, Timer 3 which are shown in Figure 13-3, Figure 13-4.

The “timer” or “counter” function is selected by control registers TMn. To use as an 8-bit timer/counter mode, CAP0, CAP1,

16BIT0 and PWME0 bits should be cleared to “0”. These timers have each 8-bit count register and data register. The count register is increased by every internal or external clock input. The internal clock has a prescaler divide ratio option of 2~2048 selected by control bits of register TMn (n=0,1,2,3).

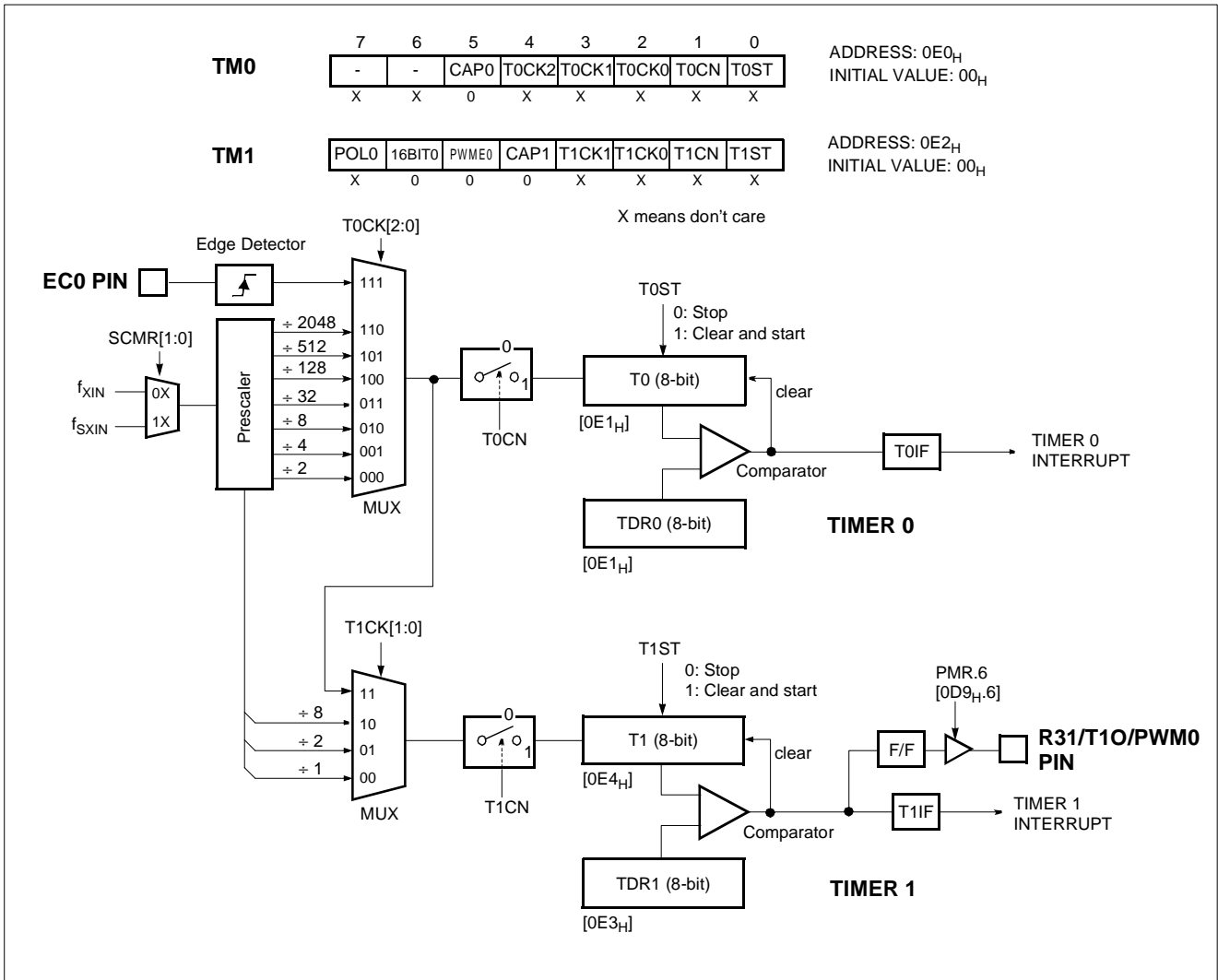


Figure 13-3 8-bit Timer/Counter 0, 1

Note: The contents of Timer data register TDRx should be initialized with 1H~FFH, not to 0H, because it is not to be defined before reset.

In the Timer 0, timer register T0 increments from 00H until it matches with TDR0 and then reset to 00H. The match output of Timer 0 generates Timer 0 interrupt (latched in T0IF bit)

As TDRx and Tx register are in same address, when reading it as a Tx, written to TDRx.

In counter function, the counter is increased every 0-to-1 (rising edge) transition of EC0 or EC2 pin. In order to use counter function, the bit 3 and bit 4 of the Port mode register PMR are set to "1" by software. The Timer 0 can be used as a counter by pin EC0 input. Similarly, Timer 2 can be used by pin EC2 input.

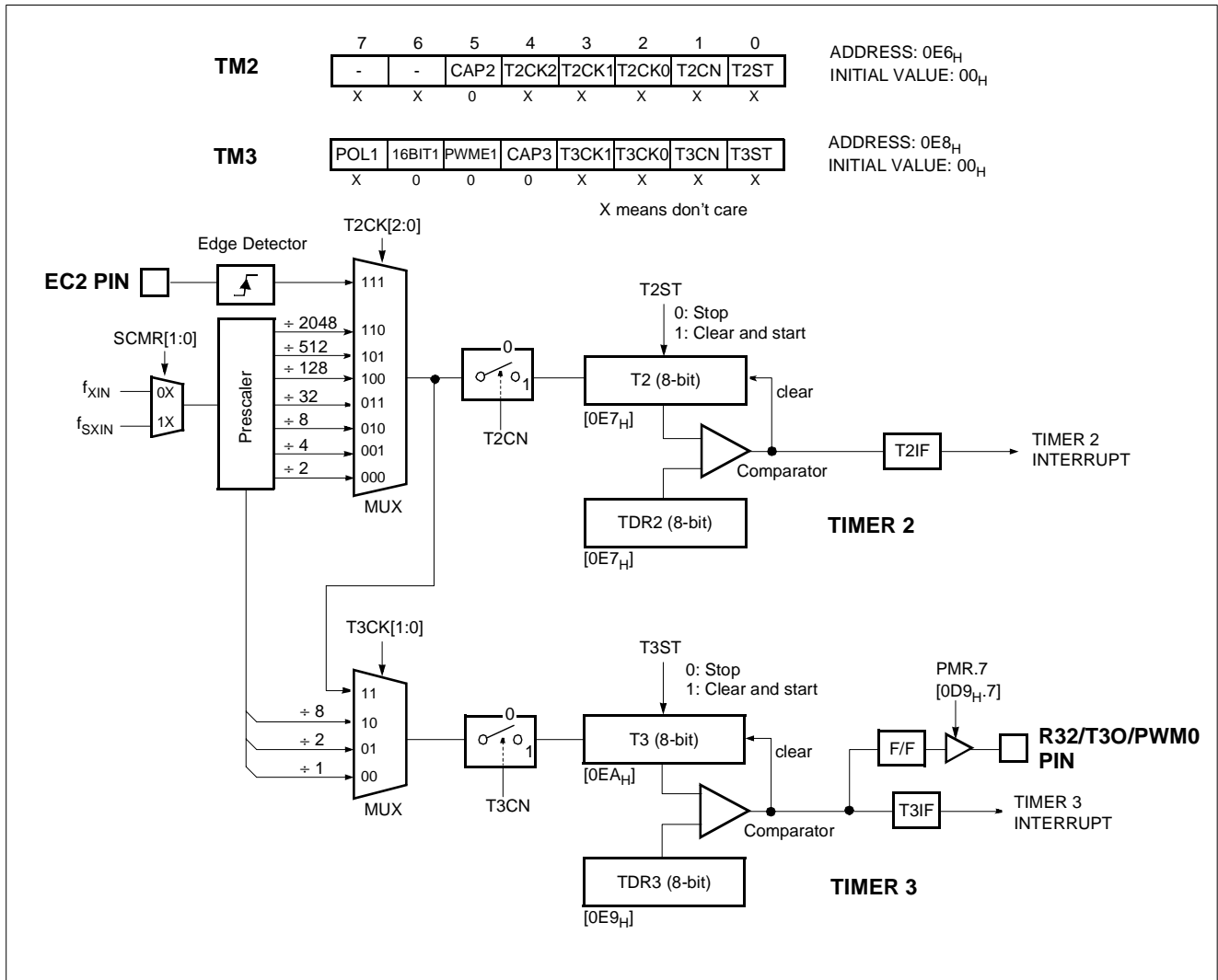


Figure 13-4 8-bit Timer/Counter 2, 3

8-bit 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 TDRn (n=0,1,2,3) are compared with the contents of up-counter, Tn (n=0,1,2,3). 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 TDRn can be re-written by software, time interval is set as you want.

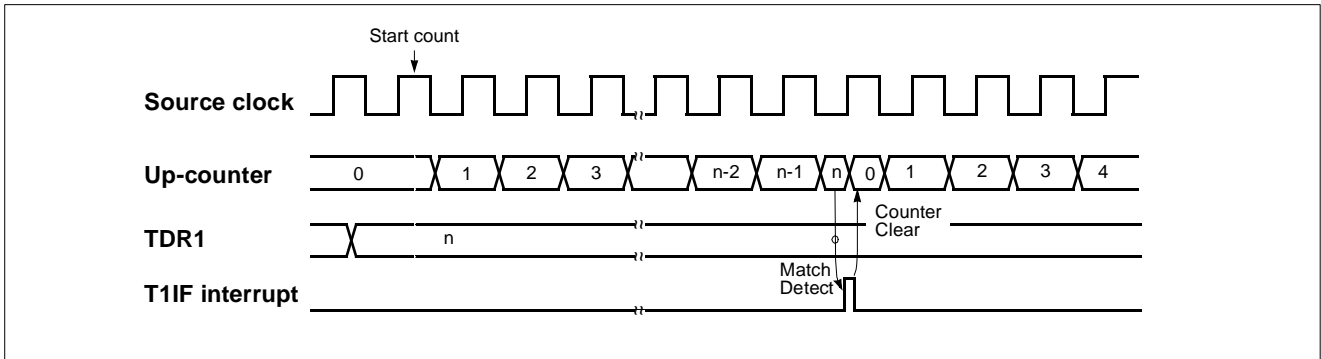


Figure 13-5 Timer Mode Timing Chart

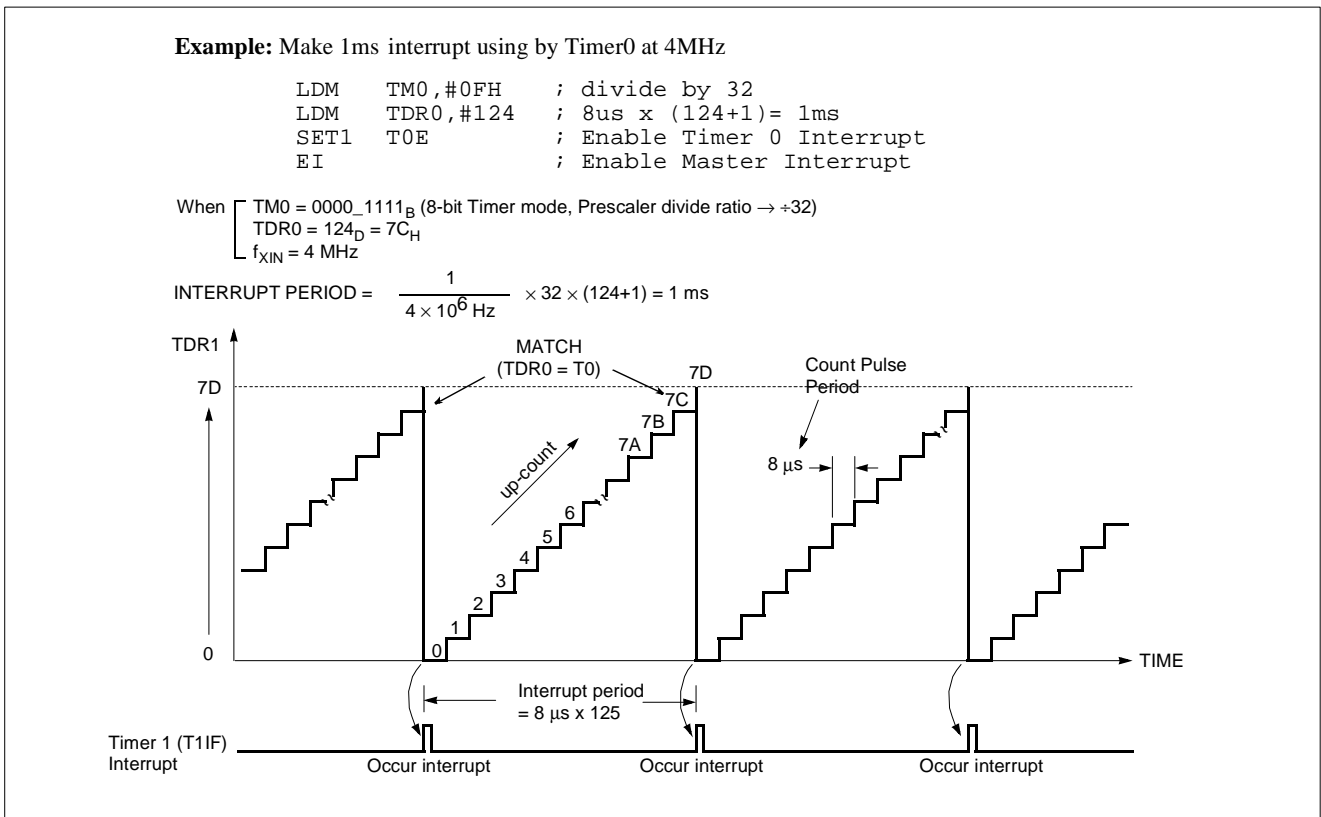


Figure 13-6 Timer Count Example

8-bit Event Counter Mode

In this mode, counting up is started by an external trigger. This trigger means rising edge of the EC0 or EC2 pin input. Source clock is used as an internal clock selected with timer mode register TM0, TM1, TM2 or TM3. The contents of timer data register TDRn (n = 0,1,2,3,.....,FF) are compared with the contents of the up-counter Tn. If a match is found, an timer interrupt request flag TnIF is generated, and the counter is cleared to "0". The counter is restart and count up continuously by every rising edge of the ECn pin input.

The maximum frequency applied to the ECn pin is $f_{XIN}/2$ [Hz].

In order to use event counter function, the bit 3, 4 of the Port Mode Register PMR (address 0D9H) is required to be set to "1".

After reset, the value of timer data register TDRn is undefined, it should be initialized to between 1H~FFH, not to "0". The interval period of Timer is calculated as below equation.

$$Period(sec) = \frac{1}{f_{XIN}} \times 2 \times Divide\ Ratio \times TDRn$$

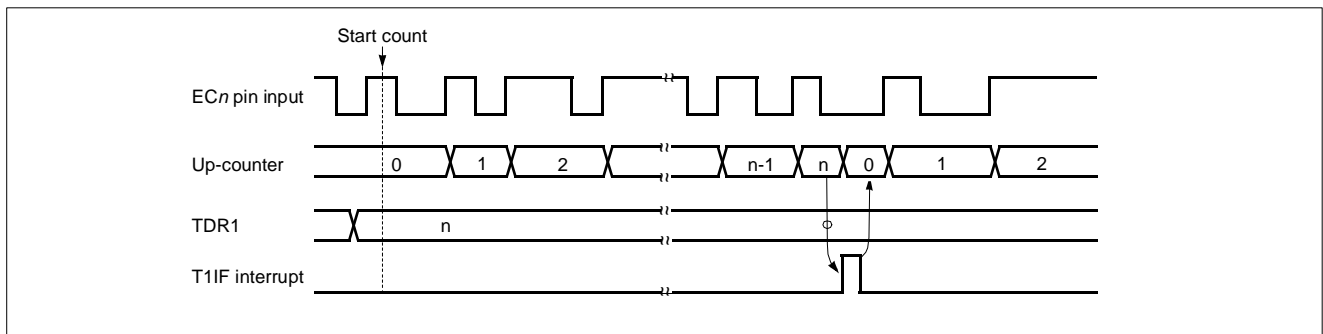


Figure 13-7 Event Counter Mode Timing Chart

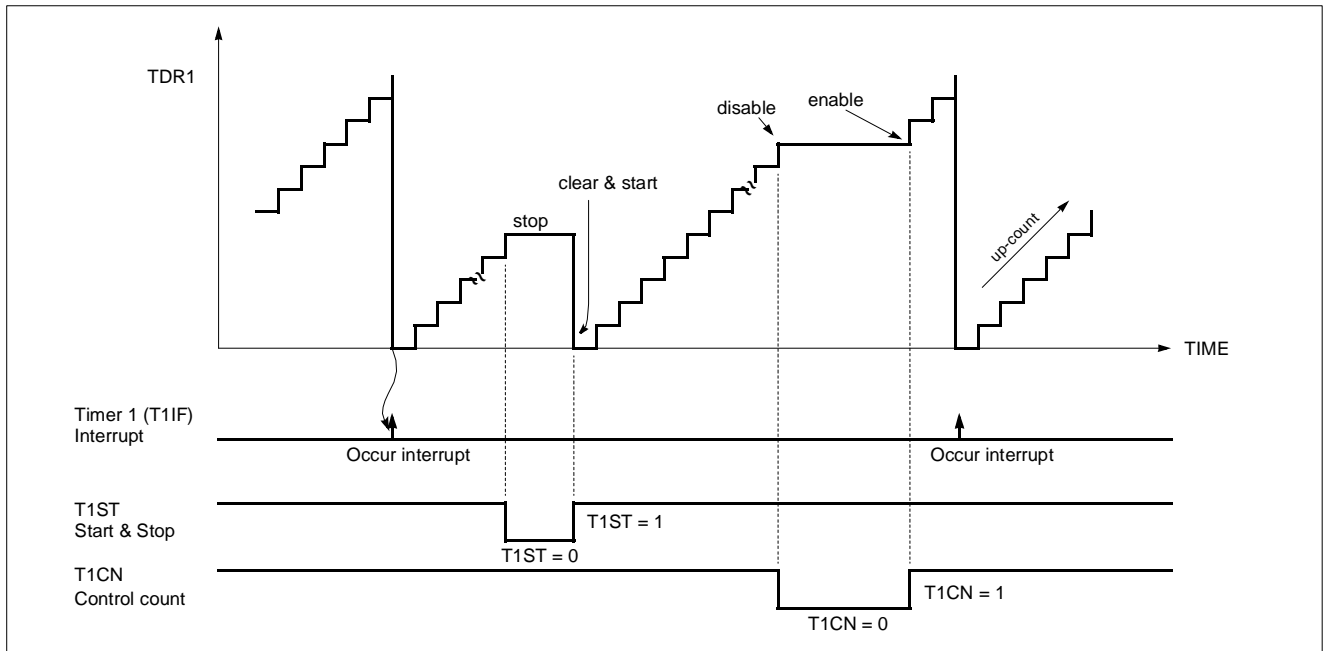


Figure 13-8 Count Operation of Timer / Event counter

13.2 16-bit Timer / Counter Mode

The Timer register is being run with all 16 bits. A 16-bit timer/counter register T0, T1 are incremented from 0000_H until it matches TDR0, TDR1 and then resets to 0000_H. The match output generates Timer 0 interrupt.

The clock source of the Timer 0 is selected either internal or external clock by bit T0SL1, T0SL0.

Even if the Timer 0 (including the Timer 1) is used as a 16-bit timer, the Timer 2 and Timer 3 can still be used as either two 8-bit timer or one 16-bit timer by setting the TM2. Reversely, even if the Timer 2 (including the Timer 3) is used as a 16-bit timer, the Timer 0 and Timer 1 can still be used as 8-bit timer independently.

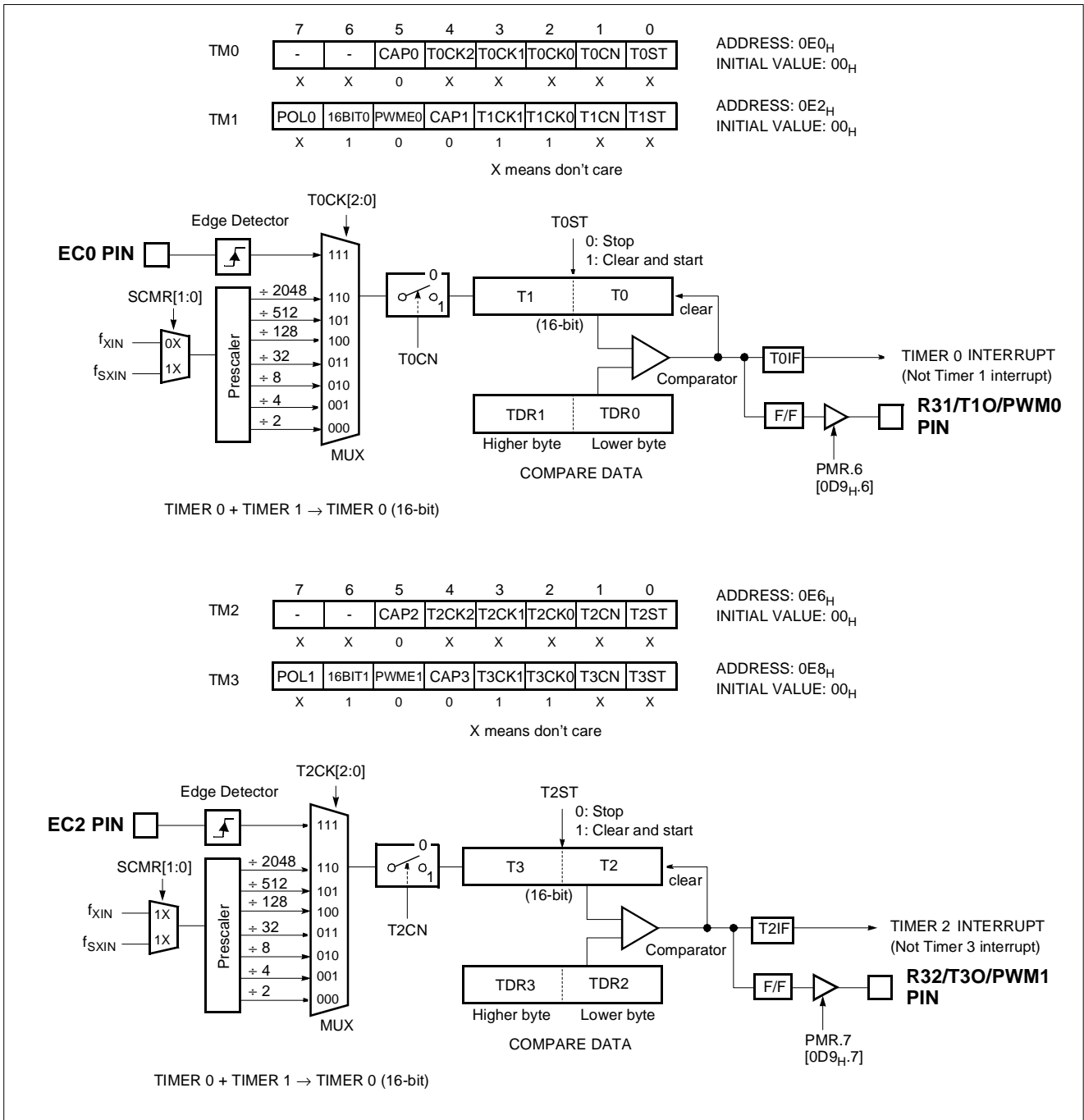


Figure 13-9 16-bit Timer/Counter

13.3 8-bit Capture Mode

The capture mode can be used to measure the pulse width between two edges. The Timer 0 capture mode is set by bit CAP0 of Timer Mode Register TM0, and the Timer 1 capture mode is set by CAP1 of Timer Mode Register TM1 as shown in Figure 13-10. Timer 2 and Timer 3 have same architecture with Timer 0 and Timer 1.

The Timer/Counter register is incremented in response internal or external input. This counting function is same with normal timer mode, and Timer interrupt is generate when timer register T0 (T1, T2, T3) increase and match TDR0 (TDR1, TDR2, TDR3).

Timer/Counter still does the above, but with the added feature that a edge transition at external input INTn pin causes the current

$$f_{timer} = \frac{f_{xin}}{2 \times \text{prescaler value} \times (TDR + 1)}$$

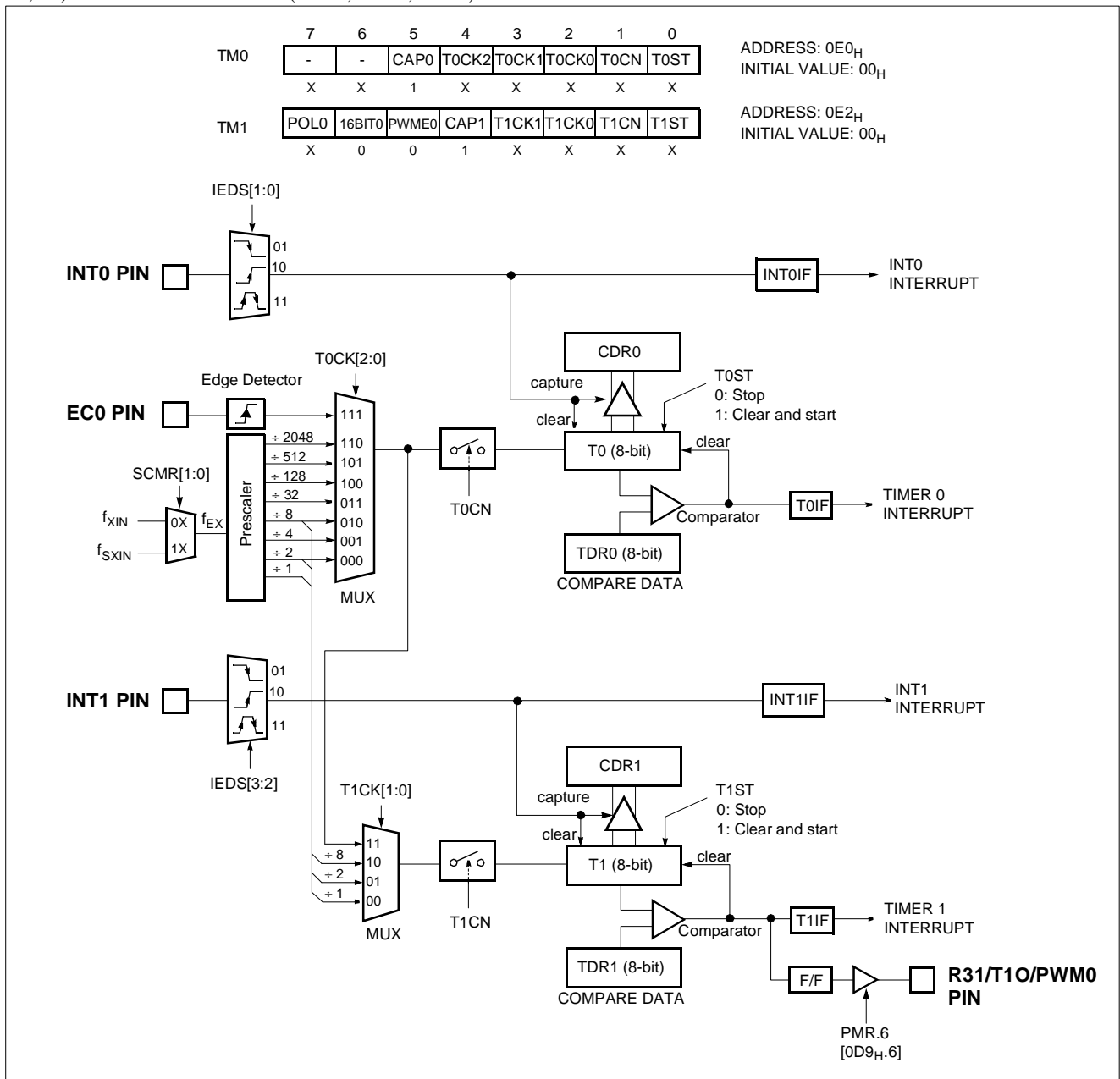


Figure 13-10 8-bit Capture Mode (Timer0/Timer1 case)

value in the Timer counter register (T0,T1), to be captured and stored into registers CDRn (CDR0, CDR1), respectively. After capture, the Timer counter register is cleared and restarts by hardware. At this time, reading the address E1H as a CDR0, not T0. T0, TDR0, CDR0 are located at same address. The other CDR1~CDR3 are same. Refer to Timer registers of page 29.

It has three transition modes: “falling edge”, “rising edge”, “both edge” which are selected by interrupt edge selection register IEDS. Refer to “17.4 External Interrupt” on page 70. In addition, the transition at INTn pin generate an interrupt.

Note: The CDRn and Tn are in same address. In the capture mode, reading operation is read as CDRn, not Tn because addressing path is opened to the CDRn.

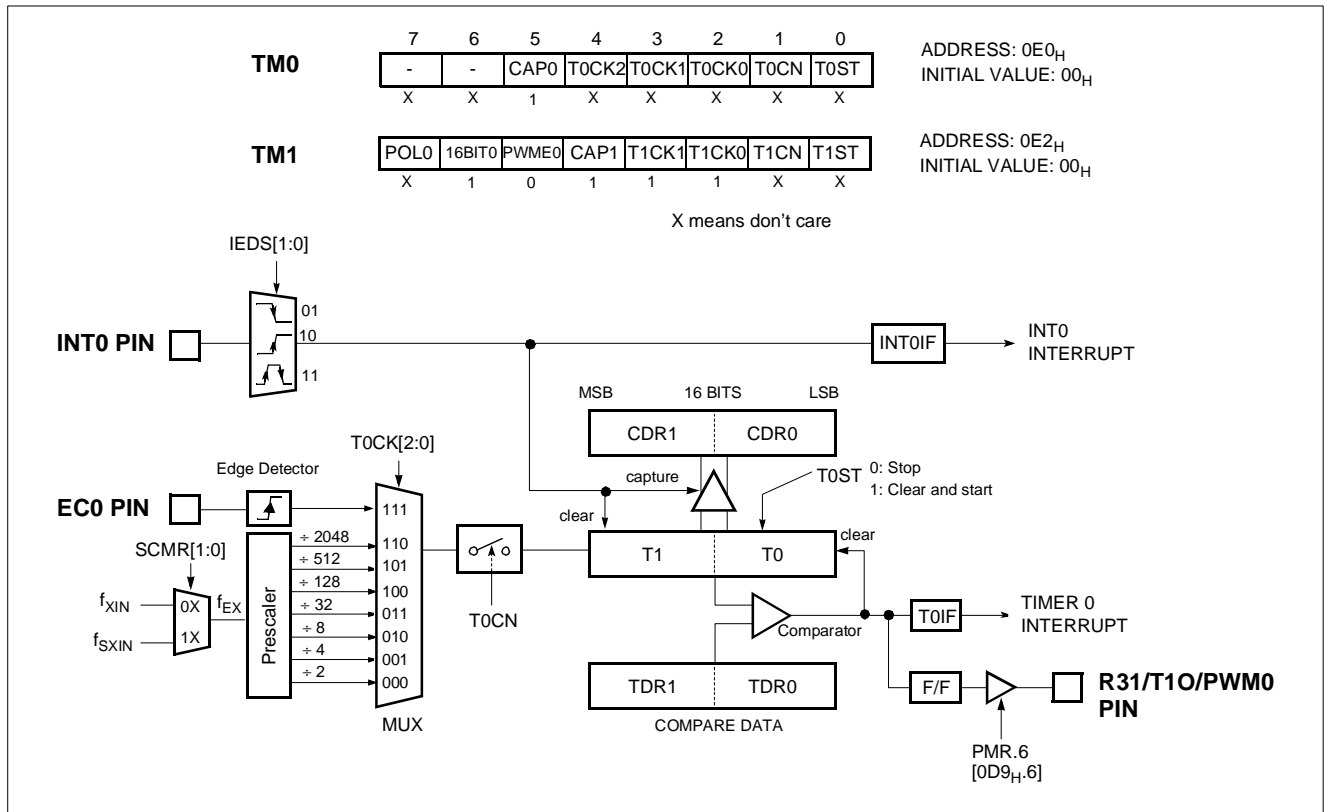


Figure 13-11 16-bit Capture Mode

13.4 16-bit Capture Mode

16-bit capture mode is the same as 8-bit capture, except that the Timer register is being run will 16 bits. Configuration is shown in

Figure 13-11.

13.5 Timer output port mode

The GMS81C7008/16 has a function of Timer compare output. To pulse out, the timer match can goes out to port pin (T10, T30) as shown in Figure 13-3, Figure 13-4 and Figure 13-9. Thus pulse out is generated by the timer match. These operation is implemented to pin T10, T30. This pin output the signal having 50% duty square wave and output frequency is same as below

equation.

To use this function, the bit 6 and bit 7 of Port Mode Register (PMR) are set or clear properly. In addition, 16-bit Timer output mode is available, also

13.6 PWM Mode

The GMS81C70xx has two high speed PWM (Pulse Width Modulation) functions which shared with Timer 1 and Timer 3.

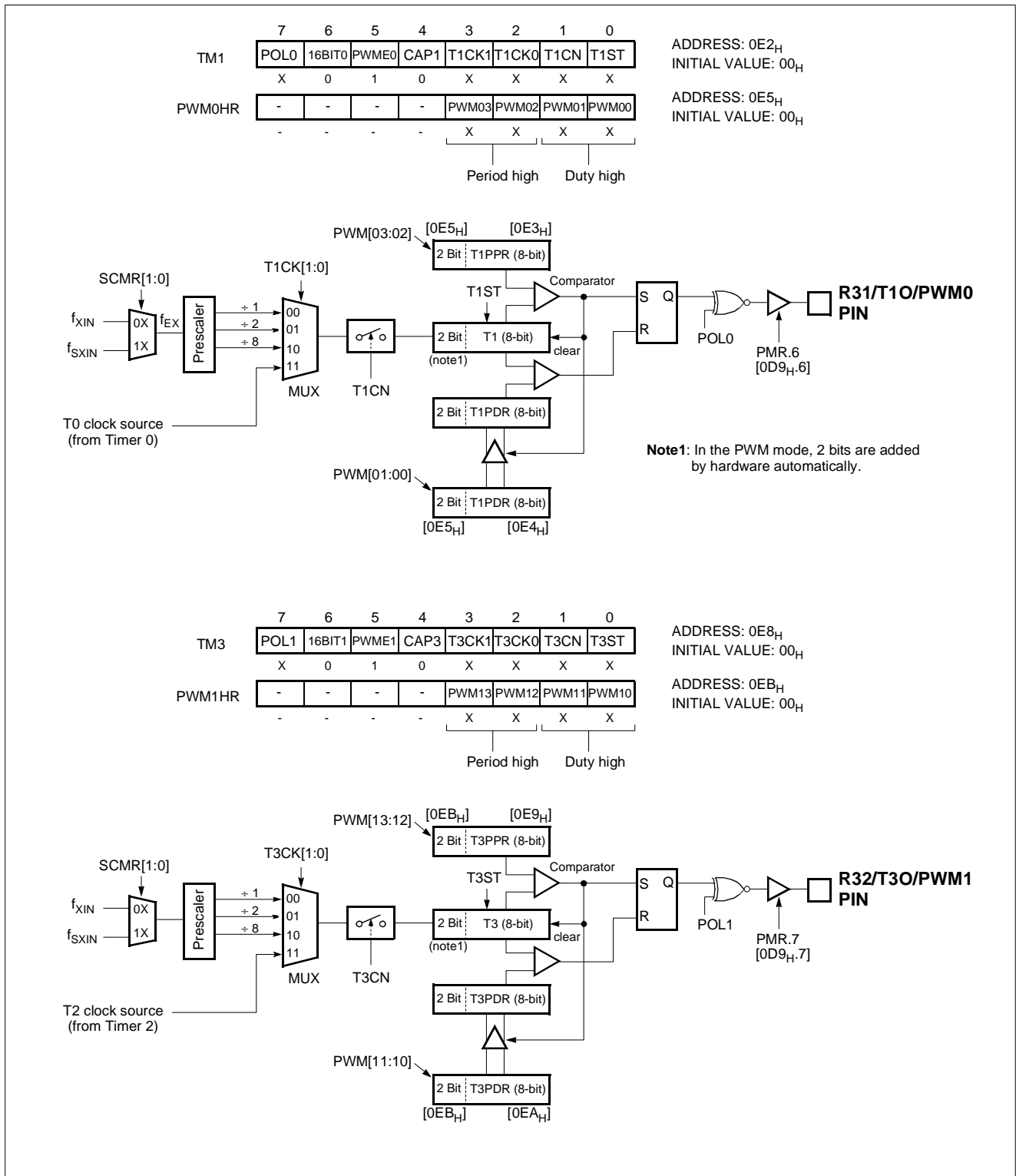


Figure 13-12 PWM Mode

Note: Whenever change the register content of Period or Duty of PWM output, the timer counter T_n must be stopped and restart again by software.

The PWM0 will be explained in this chapter. Other PWM1 has same architecture. Pin R32/T1O/PWM0 outputs up to a 10-bit resolution PWM output. This pin should be configure as a PWM output to set bit PRM0.6 to "1".

The period of the PWM output is determined by the T1PPR (PWM0 Period Register) and PWM0HR[3:2] and the duty is determined by the T1PDR (PWM0 Duty Register) and PWM0HR[1:0].

The user writes the lower 8-bit period value to the T1PPR and the higher 2-bit period value to the PWM0HR[3:2]. And writes duty value to the T1PDR and the PWM0HR[1:0] same way.

The T1PDR is configure as a double buffering for glitchless PWM output. In, the duty data is transferred from the master to the slave when the period data matched to the counted value. (i.e. at the beginning of next duty cycle)

The relation between frequency and resolution is in inverse proportion. Table 13-1 shows the PWM frequency in each clock source. If it needed higher frequency of PWM, it should be reduced resolution.

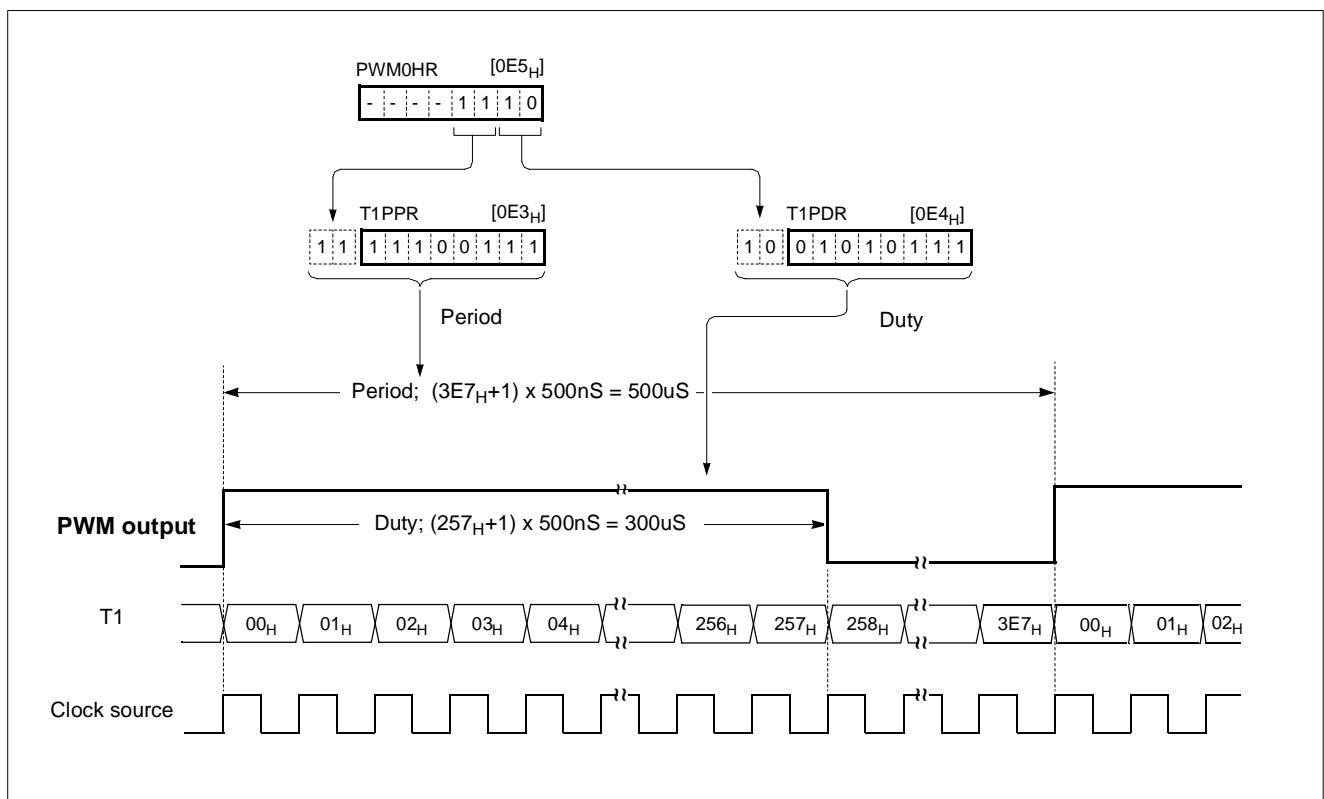


Figure 13-13 Example of Register setting

The bit POL0 of TM0 decides the polarity of duty cycle.

If the duty value is set same to the period value, the PWM output is determined by the bit POL0 (1: High, 0: Low). And if the duty value is set to "00_H", the PWM output is determined by the bit POL0 (1: Low, 0: High).

It can be changed duty value when the PWM output. However the changed duty value is output after the current period is over. And it can be maintained the duty value at present output when

changed only period value shown as Figure 13-14. As it were, the absolute duty time is not changed in varying frequency. But the changed period value must greater than the duty value.

At PWM output start command, one first pulse would be output abnormally. Because if user writes register values while timer is in operation, these register could be set with certain values at first. To prevent this operation, user must stop PWM timer clock and then set the duty and the period register values.

Resolution	PWM clock source		
	f_{XIN+1}	f_{XIN+2}	$f_{XIN+1024}$
10-bit	3.9kHz	1.95kHz	3.8Hz
9-bit	7.8kHz	3.9kHz	7.6Hz
8-bit	15.6kHz	7.8kHz	15.3Hz
7-bit	31.2kHz	15.6kHz	30.5Hz

Table 13-1 PWM Frequency vs. Resolution at 4MHz

Example:

Timer1 = 2kHz, 30% duty PWM mode

```

LDM    TM1, #00H
LDM    T1PPR, #0E8H
LDM    T1PDR, #58H
LDM    PWM0HR, 0000_1110B
LDM    TM1, #1010_1011B
    
```

Refer to Figure 13-13.

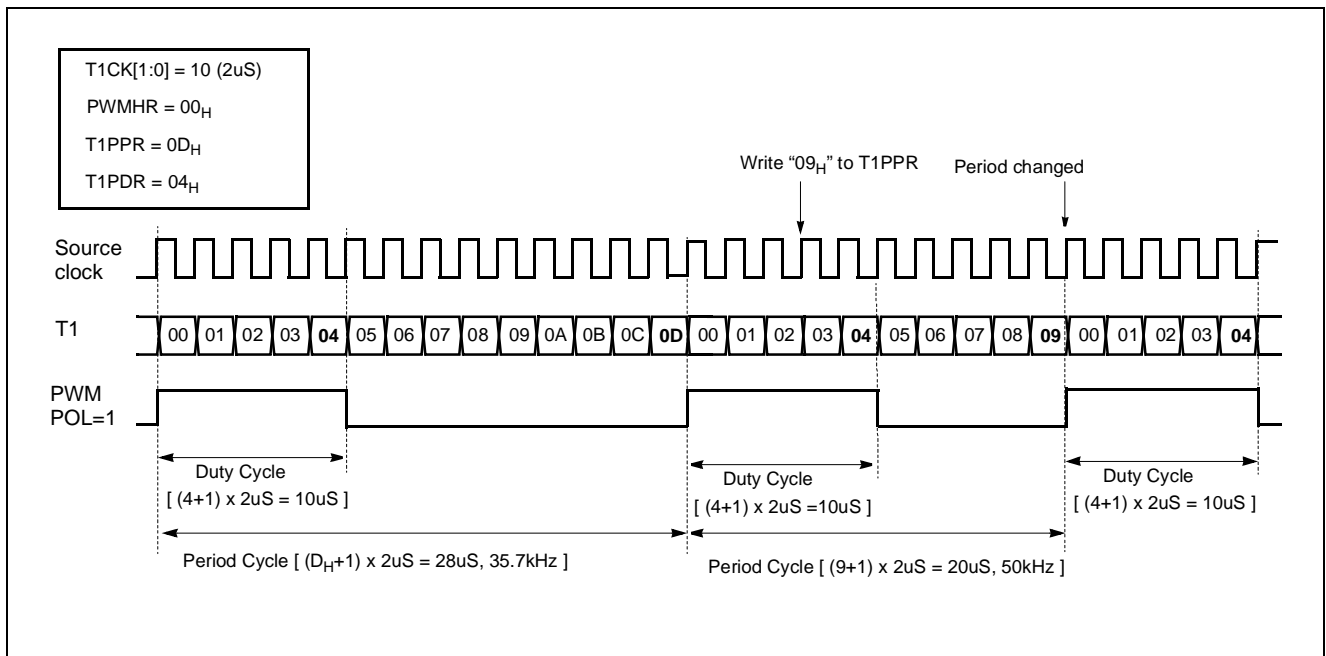


Figure 13-14 Example of changing the period in absolute duty cycle at 4MHz

14. ANALOG DIGITAL CONVERTER

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

The A/D module has two registers which are the control register ADCM and A/D result register ADR. The register ADCM, shown in Figure 14-4, controls the operation of the A/D converter module. The port pins can be configured as analog inputs or digital I/O. To use analog inputs, I/O is selected input mode by R2DD direction register.

How to Use A/D Converter

The processing of conversion is start when the start bit ADST is set to "1". After one cycle, it is cleared by hardware. The register ADR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADR, the A/D conversion status bit ADSF is set to "1", and the A/D interrupt flag AIF is set. The block diagram of the A/D module is shown in Figure 14-1. The A/D status bit ADSF is set automatically when A/D conversion is completed, cleared when A/D conversion is in process. The conversion time takes maximum 20 uS (at f_{XIN}=4 MHz).

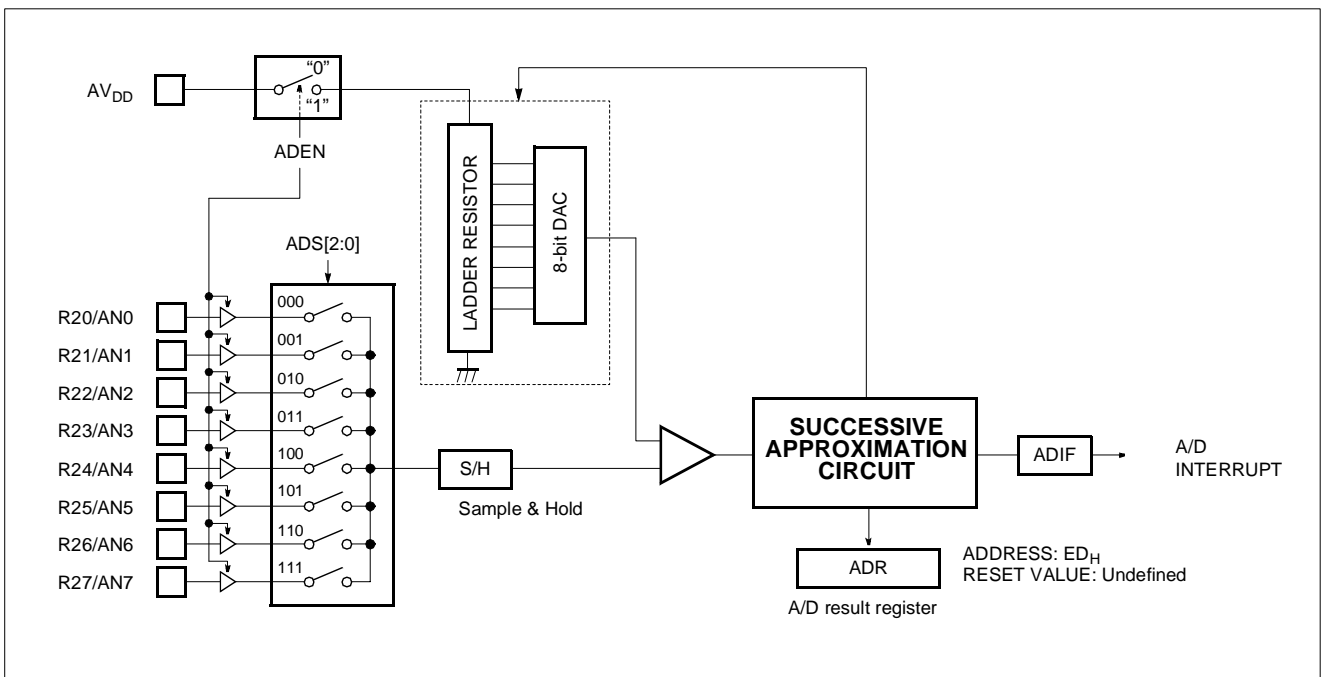


Figure 14-1 A/D Block Diagram

A/D Converter Cautions

(1) Input voltage range of AN0 to AN7

The input voltage of AN0 to AN7 should be within the specification range. In particular, if a voltage above AV_{DD} or below AV_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel can not be indeterminate. The conversion values of the other channels may also be affected.

(2) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AV_{DD} and AN0 to AN7. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 14-2 in order to reduce noise.

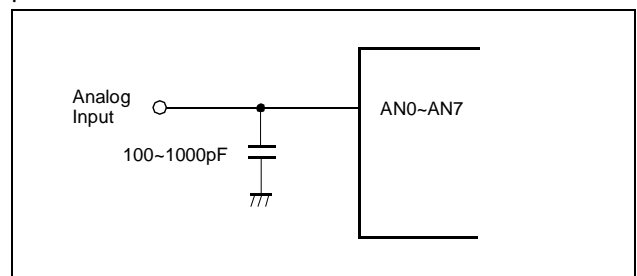


Figure 14-2 Analog Input Pin Connecting Capacitor

(3) AD pin sharing with normal I/O port

The analog input pins AN0 to AN7 also function as input/output port (PORT R20~R27) pins. When A/D conversion is performed with any of pins AN0 to AN7 selected, be sure not to execute a PORT input instruction while conversion is in progress, as this may reduce the conversion resolution.

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

(4) AV_{DD} pin input impedance

A series resistor string of approximately 10kΩ is connected between the AV_{DD} pin and the AV_{SS} pin.

Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{DD} pin and the AV_{SS} pin, and there will be a large reference voltage error.

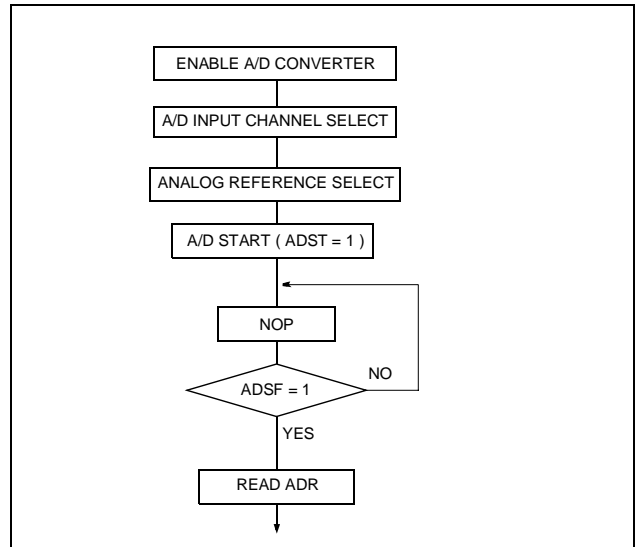


Figure 14-3 A/D converter Operation Flow

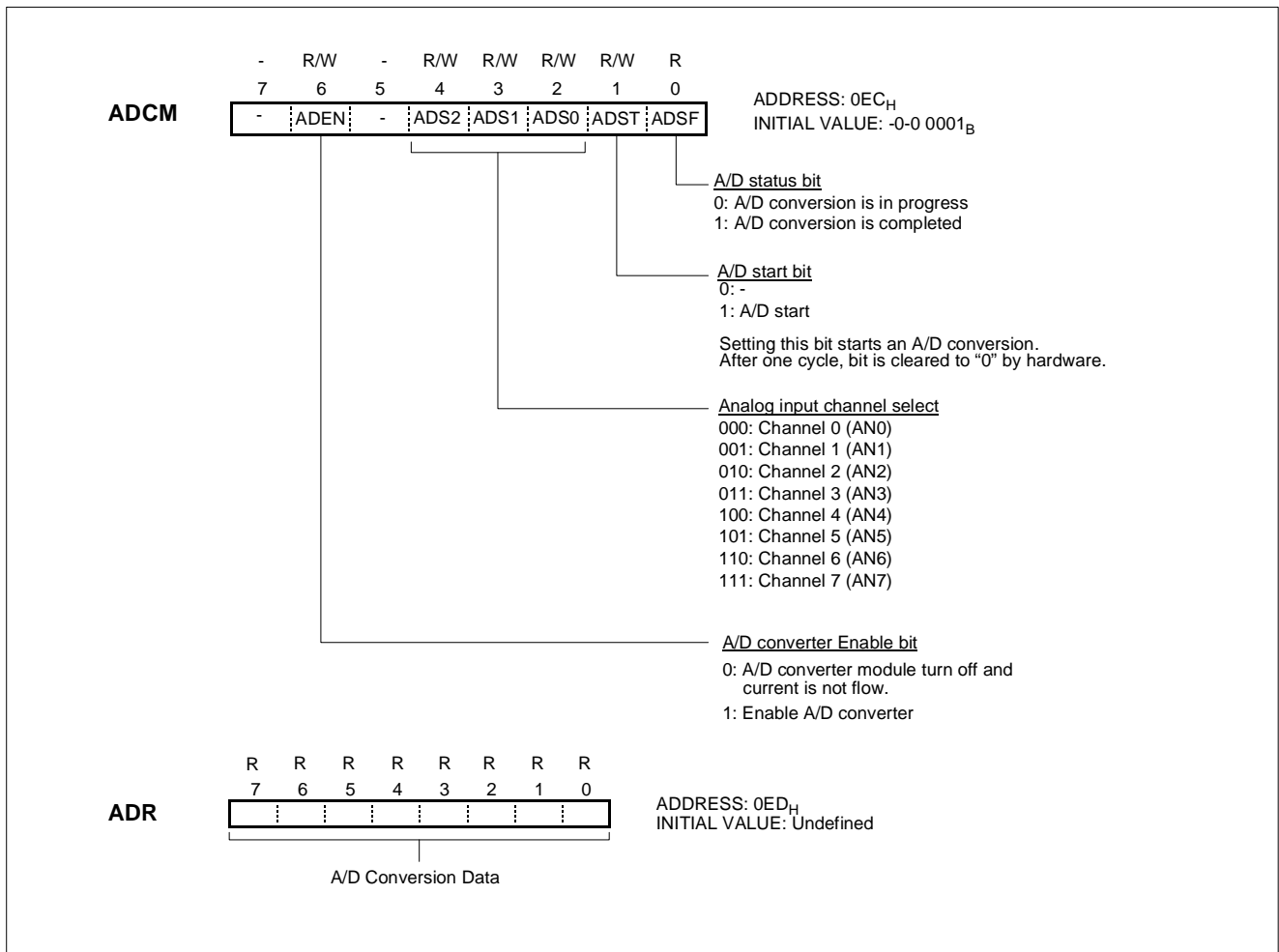


Figure 14-4 A/D Converter Control Register

15. SERIAL COMMUNICATION

The serial interface is used to transmit/receive 8-bit data serially. Serial communication block consists of serial I/O data register, serial I/O mode register, clock selection circuit, octal counter and control circuit as illustrated in Figure 15-1. Pin R07/SIN, R06/SOUT and R05/SCLK pins are controlled by the Serial Mode Register. The contents of the Serial I/O data register can be written into or read out by software.

The serial communication is activated by the instruction "SETI

SIOST". The octal counter is reset to "0" by this instruction, starts counting at the falling or rising edge (by POL selection) of the transmit clock (SCLK), and it increments at the every clock. A serial interrupt request flag is set when the eighth transmit clock signal is input (the serial interface is reset) or when serial communication is discontinued (the octal counter is reset).

The data in the Serial Data Register can be shifted synchronously with the transfer clock signal.

SCK1	SCK0	SCLK/R05 Port	Clock Source	Prescaler Divide Ratio
0	0	SCLK output	Internal clock	÷ 4
0	1	SCLK output	Internal clock	÷ 16
1	0	SCLK output	Internal clock	Use clock from Timer 0 overflow
1	1	SCLK input	External clock	-

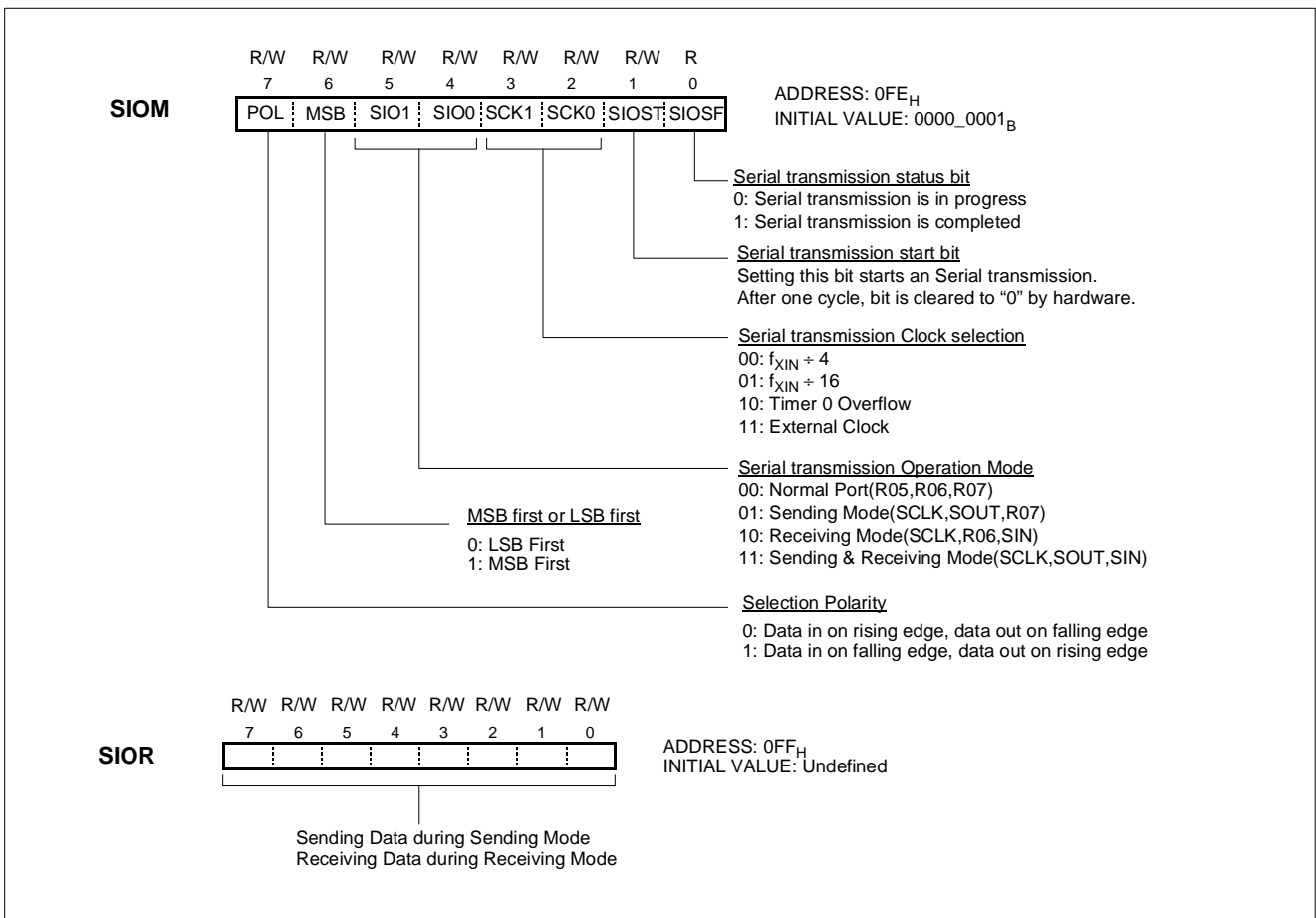


Figure 15-1 SCI Control Register

Serial I/O Mode Register(SIOM) controls serial I/O function.
 The POL bit control which edge

According to SCK1 and SCK0, the internal clock or external clock can be selected.

Serial I/O Data Register(SIOR) is an 8-bit shift register.

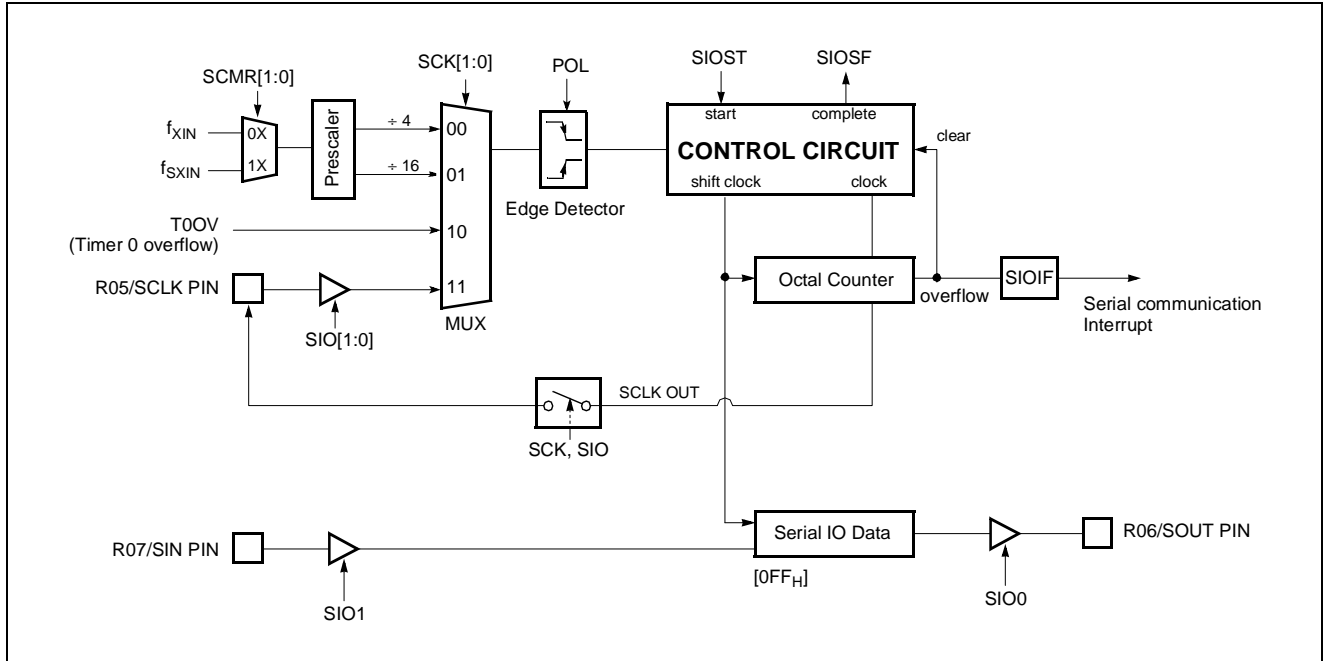


Figure 15-2 Block Diagram of SCI

15.1 Transmission/Receiving Timing

The serial transmission is started by setting SIOST(bit1 of SIOM) to “1”. After one cycle of SCK, SIOST is cleared automatically to “0”. The serial output data from 8-bit shift register is output at falling edge of SCLK. And input data

is latched at rising edge of SCLK pin. When transmission clock is counted 8 times, serial I/O counter is cleared as “0”. Transmission clock is halted in “H” state and serial I/O interrupt(SIOIF) occurred.

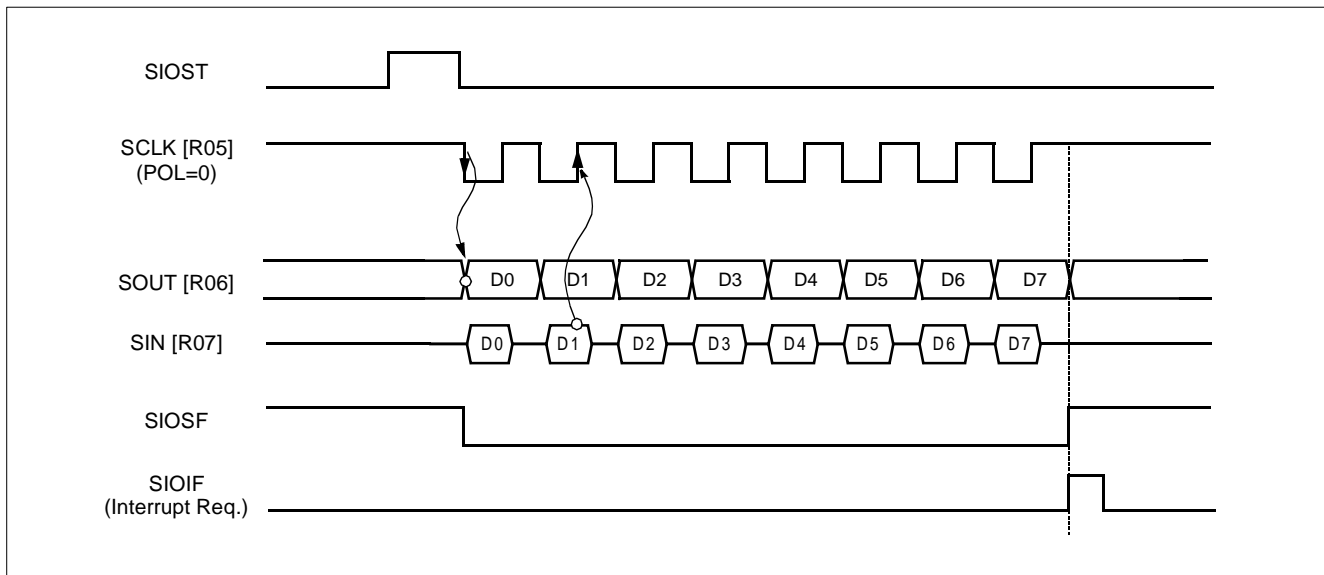


Figure 15-3 SPI Timing Diagram at POL=0

15.2 The method of Serial I/O

1. Select transmission/receiving mode

When external clock is used, the frequency should be less than 1MHz and recommended duty is 50%.

2. In case of sending mode, write data to be send to SIOR.

3. Set SIOST to "1" to start serial transmission.

If both transmission mode is selected and transmission is per-

formed simultaneously it would be made error.

4. The SIO interrupt is generated at the completion of SIO and SIOSF is set to "1". In SIO interrupt service routine, correct transmission should be tested.

5. In case of receiving mode, the received data is acquired by reading the SIOR.

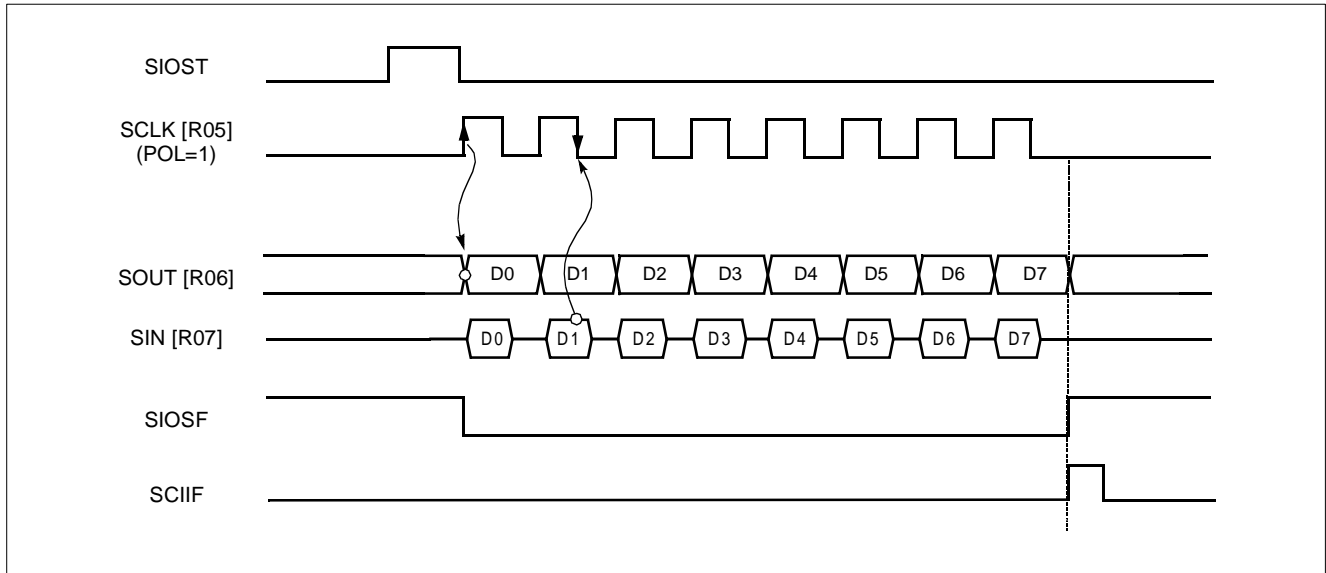


Figure 15-4 SPI Timing Diagram at POL=1

15.3 The Method to Test Correct Transmission

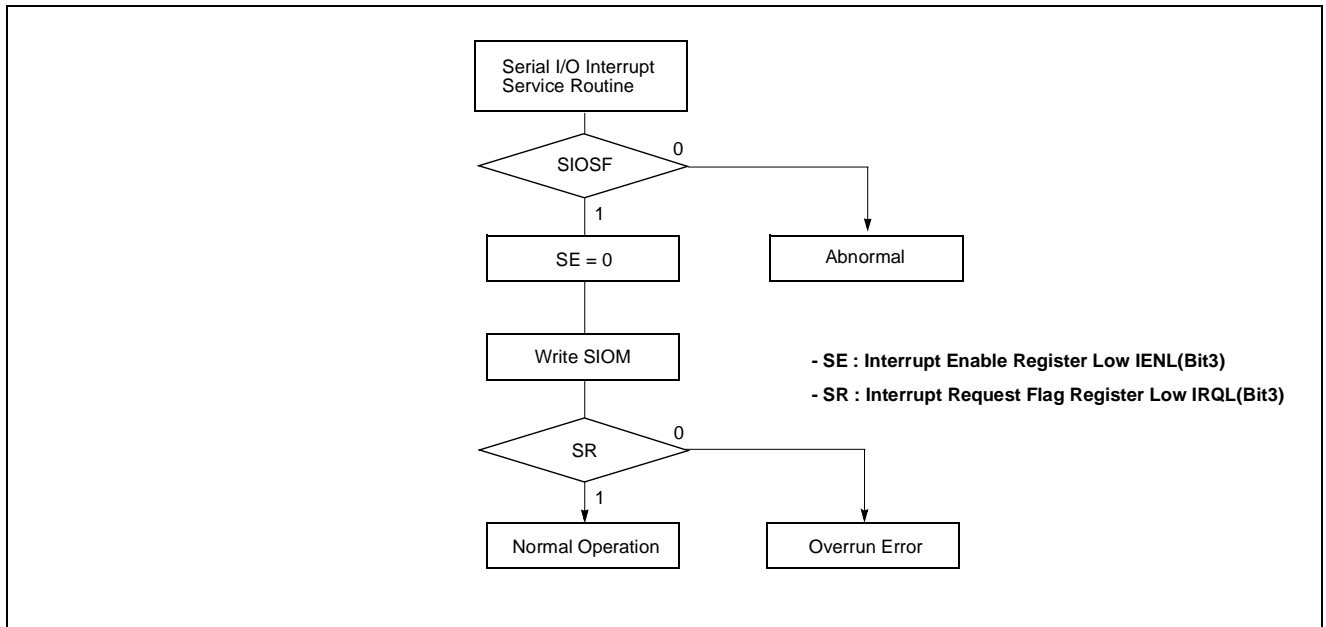


Figure 15-5 Serial Method to Test Transmission

16. BUZZER FUNCTION

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 ~ 250kHz at $f_{XIN}=4\text{MHz}$) by user software.

A 50% duty pulse can be output to R30/BUZ pin to use for piezo-electric buzzer drive. Pin R30 is assigned for output port of Buzzer driver by setting the bit 5 of PMR (address D9_H) to "1". At this time, the pin R30 must be defined as output mode (the bit 0 of R3DD=1).

Example: 2.4kHz output at 4MHz.

```

LDM R3DD, #XXXX_XXX1B
LDM BUR, #0111_0011B

SET1 PMR.5 ;BUZ ON
CLR1 PMR.5 ;BUZ OFF
    
```

X means don't care

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

Equation of frequency calculation is shown below.

$$f_{BUZ} = \frac{f_{XIN}}{2 \times DivideRatio \times (BUR[5:0] + 1)}$$

- f_{BUZ} : Buzzer frequency
- f_{XIN} : Oscillator frequency
- Divide Ratio: Prescaler divide ratio by BUCK[1:0]
- BUR: Lower 6-bit value of BUR. Buzzer period value.

The frequency of output signal is controlled by the buzzer control register BUR. The BUR[5:0] determine output frequency for buzzer driving.

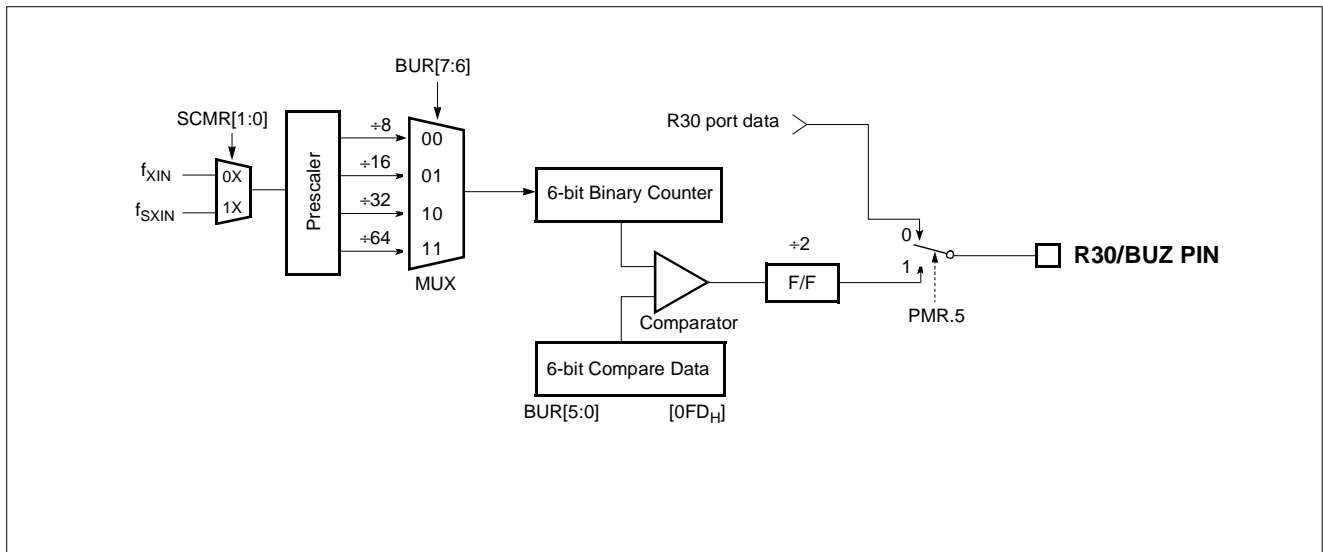


Figure 16-1 Block Diagram of Buzzer Driver

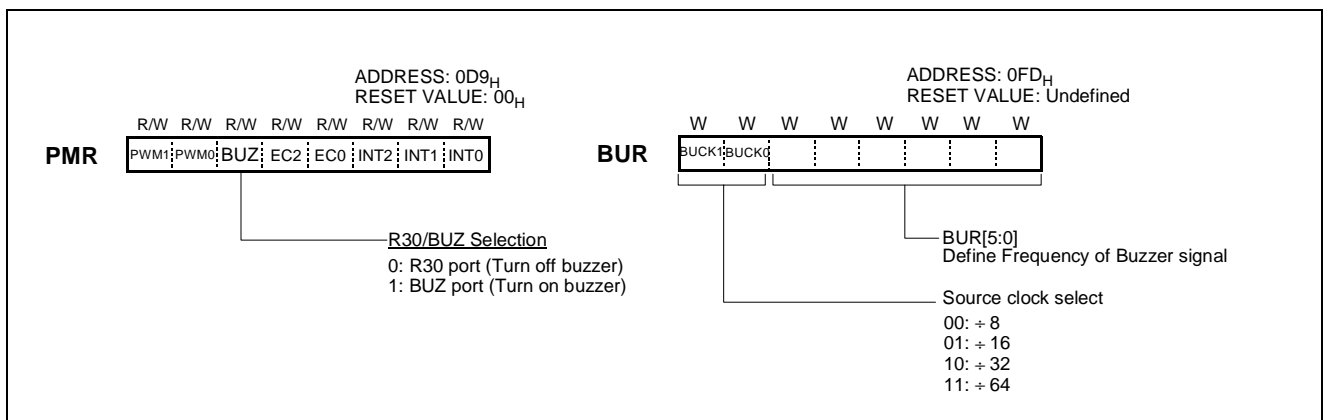


Figure 16-2 PMR and Buzzer Register

Note that BUR is a write-only register.

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

bit BUR value.

When main-frequency is 4MHz, buzzer frequency is shown as below table. The unit is kHz.

BUR [5:0]	BUCK[1:0]			
	00	01	10	11
00	250.000	125.000	62.500	31.250
01	125.000	62.500	31.250	15.625
02	83.333	41.667	20.833	10.417
03	62.500	31.250	15.625	7.813
04	50.000	25.000	12.500	6.250
05	41.667	20.833	10.417	5.208
06	35.714	17.857	8.929	4.464
07	31.250	15.625	7.813	3.906
08	27.778	13.889	6.944	3.472
09	25.000	12.500	6.250	3.125
0A	22.727	11.364	5.682	2.841
0B	20.833	10.417	5.208	2.604
0C	19.231	9.615	4.808	2.404
0D	17.857	8.929	4.464	2.232
0E	16.667	8.333	4.167	2.083
0F	15.625	7.813	3.906	1.953
10	14.706	7.353	3.676	1.838
11	13.889	6.944	3.472	1.736
12	13.158	6.579	3.289	1.645
13	12.500	6.250	3.125	1.563
14	11.905	5.952	2.976	1.488
15	11.364	5.682	2.841	1.420
16	10.870	5.435	2.717	1.359
17	10.417	5.208	2.604	1.302
18	10.000	5.000	2.500	1.250
19	9.615	4.808	2.404	1.202
1A	9.259	4.630	2.315	1.157
1B	8.929	4.464	2.232	1.116
1C	8.621	4.310	2.155	1.078
1D	8.333	4.167	2.083	1.042
1E	8.065	4.032	2.016	1.008
1F	7.813	3.906	1.953	0.977

BUR [5:0]	BUCK[1:0]			
	00	01	10	11
20	7.576	3.788	1.894	0.947
21	7.353	3.676	1.838	0.919
22	7.143	3.571	1.786	0.893
23	6.944	3.472	1.736	0.868
24	6.757	3.378	1.689	0.845
25	6.579	3.289	1.645	0.822
26	6.410	3.205	1.603	0.801
27	6.250	3.125	1.563	0.781
28	6.098	3.049	1.524	0.762
29	5.952	2.976	1.488	0.744
2A	5.814	2.907	1.453	0.727
2B	5.682	2.841	1.420	0.710
2C	5.556	2.778	1.389	0.694
2D	5.435	2.717	1.359	0.679
2E	5.319	2.660	1.330	0.665
2F	5.208	2.604	1.302	0.651
30	5.102	2.551	1.276	0.638
31	5.000	2.500	1.250	0.625
32	4.902	2.451	1.225	0.613
33	4.808	2.404	1.202	0.601
34	4.717	2.358	1.179	0.590
35	4.630	2.315	1.157	0.579
36	4.545	2.273	1.136	0.568
37	4.464	2.232	1.116	0.558
38	4.386	2.193	1.096	0.548
39	4.310	2.155	1.078	0.539
3A	4.237	2.119	1.059	0.530
3B	4.167	2.083	1.042	0.521
3C	4.098	2.049	1.025	0.512
3D	4.032	2.016	1.008	0.504
3E	3.968	1.984	0.992	0.496
3F	3.906	1.953	0.977	0.488

Table 16-1 Buzzer Frequency at 4MHz

17. INTERRUPTS

The GMS81C7008/16 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). Thirteen interrupt sources are provided. The configuration of interrupt circuit is shown in Figure 17-2.

The keyscan interrupt is generated when 1-to-0 transition is detected at KS0 or KS0 pin.

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

The Watchdog timer Interrupt is generated by WDTIF which is set by a match in Watchdog timer register.

The External Interrupts INT0 ~ INT2 each can be transition-activated (1-to-0 or 0-to-1 transition) by selection IEDS.

The flags that actually generate these interrupts are bit INT0IF, INT1IF and INT2IF in register IRQH and IRQL. 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 0 ~ Timer 3 Interrupts are generated by T0IF~T3IF which are set by a match in their respective timer/counter register.

The Serial Communication Interrupts are generated by SIOIF which is set by 8-bit serial data transmitting or receiving through SCK, SIN, SOUT pin.

The AD converter Interrupt is generated by ADIF which is set by finishing the analog to digital conversion.

The Watch Timer Interrupt is generated by WTIF which is set by an 14-bit binary counter overflow.

The interrupts are controlled by the interrupt master enable flag I-flag (bit 2 of PSW on page 21), the interrupt enable register (IENH, IENL), and the interrupt request flags (in IRQH and IRQL) except Power-on reset and software BRK interrupt. Below table shows the Interrupt priority.

Reset/Interrupt	Symbol	Priority
Hardware Reset	RESET	-
Key scan Interrupt	KS	1
Basic Interval Timer	BIT	2
Watchdog Timer	WDT	3
External Interrupt 0	INT0	4
External Interrupt 1	INT1	5
Timer/Counter 0	Timer 0	6
Timer/Counter 1	Timer 1	7
External Interrupt 2	INT2	8
Serial Communication	SCI	9
ADC Interrupt	ADC	10
Watch Timer Interrupt	WT	11
Timer/Counter 2	Timer 2	12
Timer/Counter 3	Timer 3	13

Vector addresses are shown in Figure 8-6 on page 23. Interrupt enable registers are shown in Figure 17-3. 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.

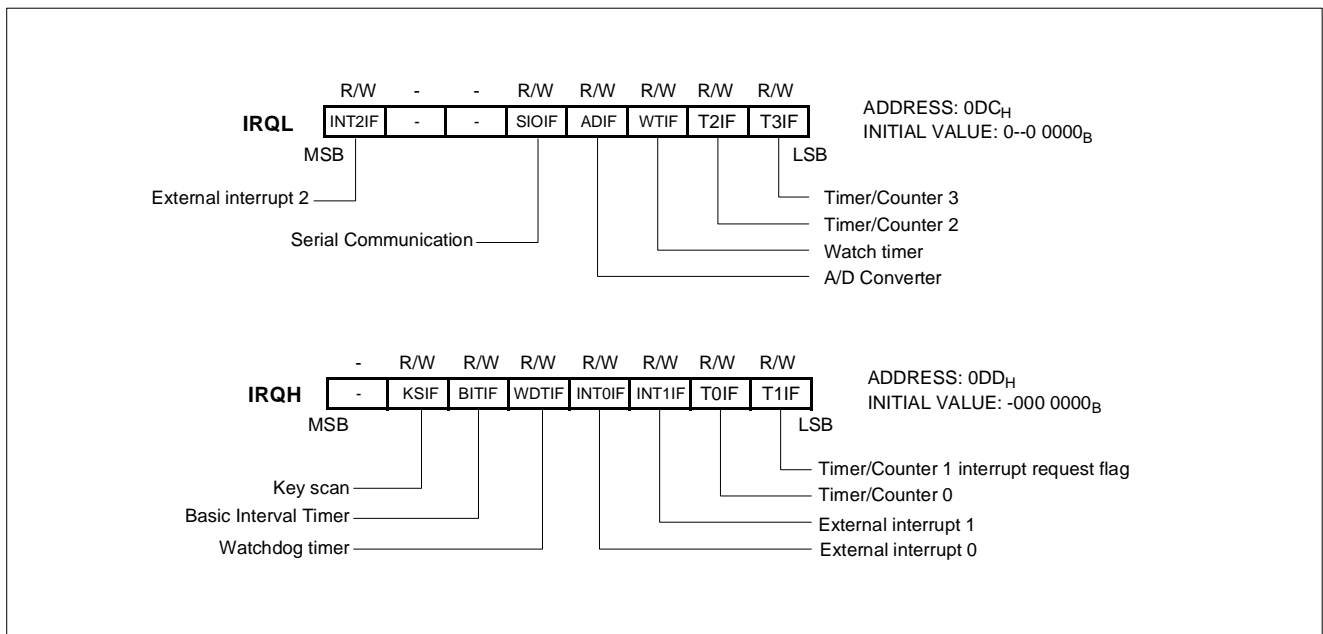


Figure 17-1 Interrupt Request Flag

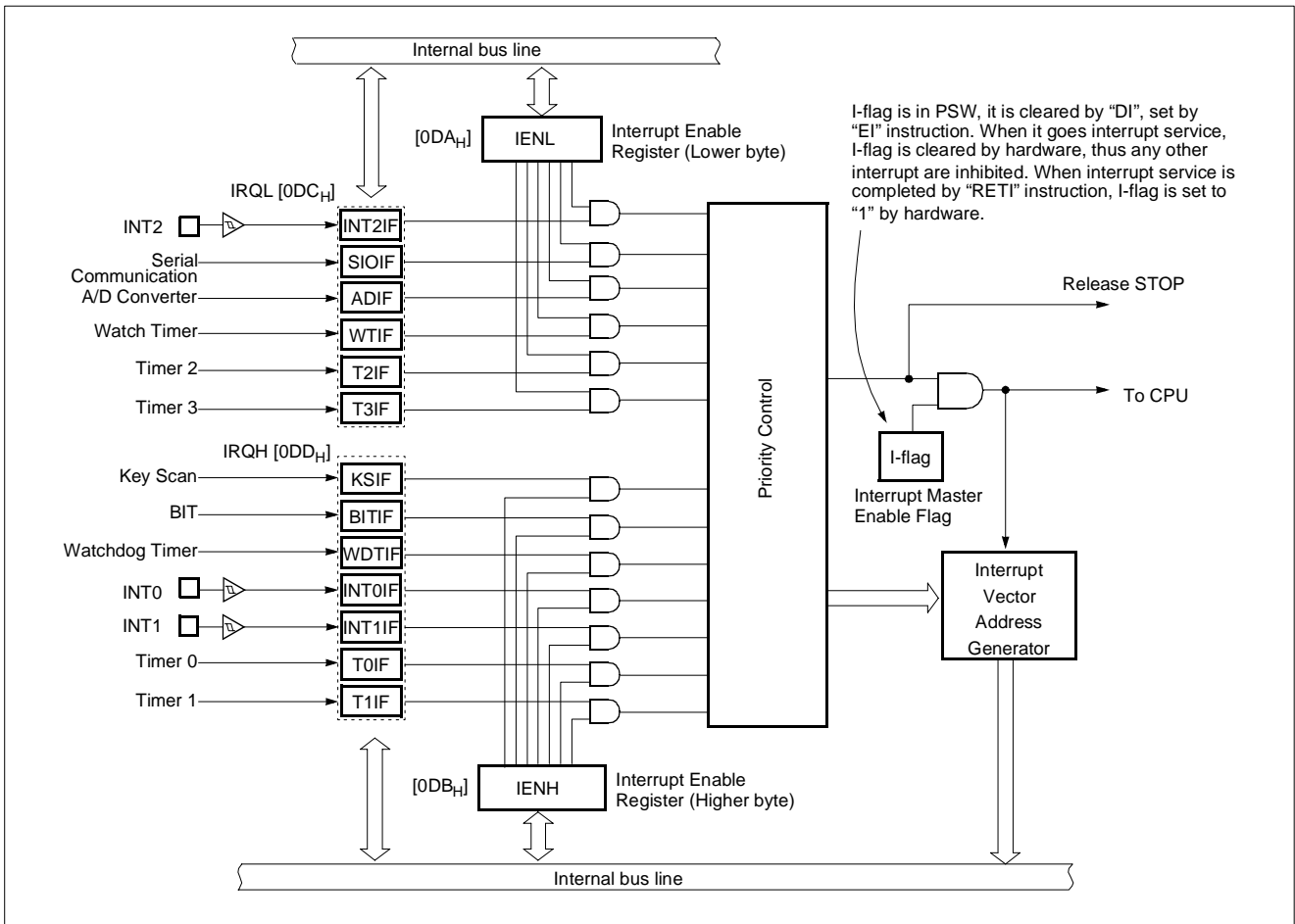


Figure 17-2 Block Diagram of Interrupt

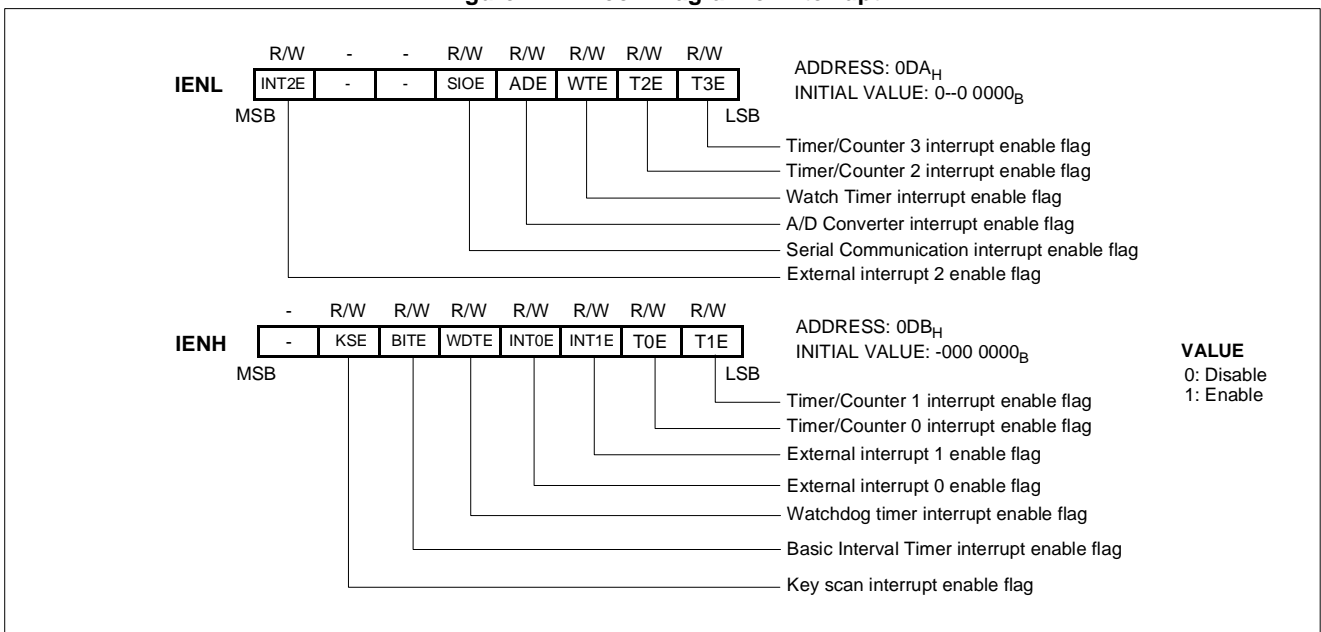


Figure 17-3 Interrupt Enable Flag

17.1 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 $8 f_{XIN}$ ($2 \mu s$ 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.

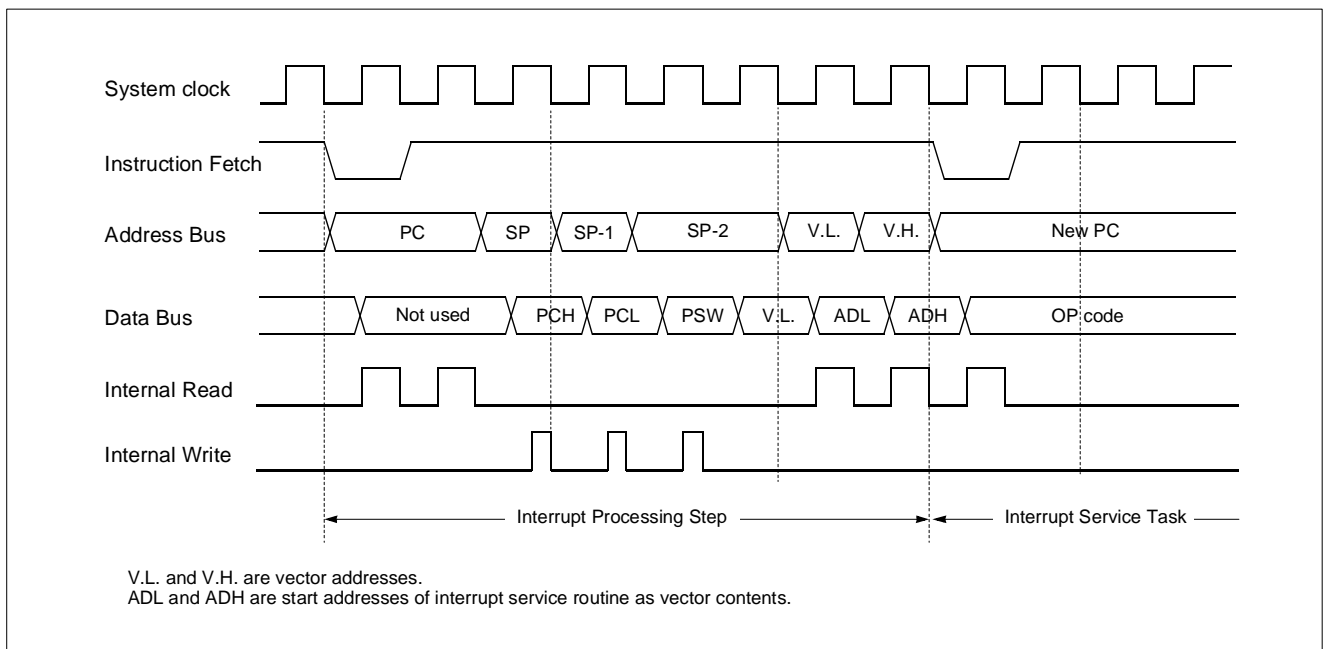
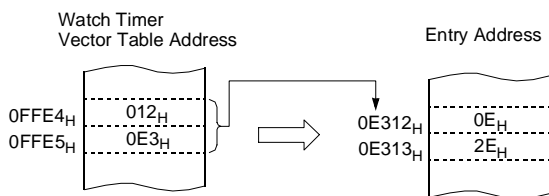


Figure 17-4 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



Correspondence between vector table address for Watch Timer 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

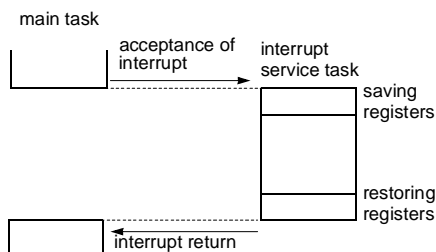
```

INTxx:  PUSH    A      ;SAVE ACC .
        PUSH    X      ;SAVE X REG.
        PUSH    Y      ;SAVE Y REG.

        [interrupt processing]

        POP     Y      ;RESTORE Y REG.
        POP     X      ;RESTORE X REG.
        POP     A      ;RESTORE ACC.
        RETI    ;RETURN
    
```

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



17.2 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 17-5.

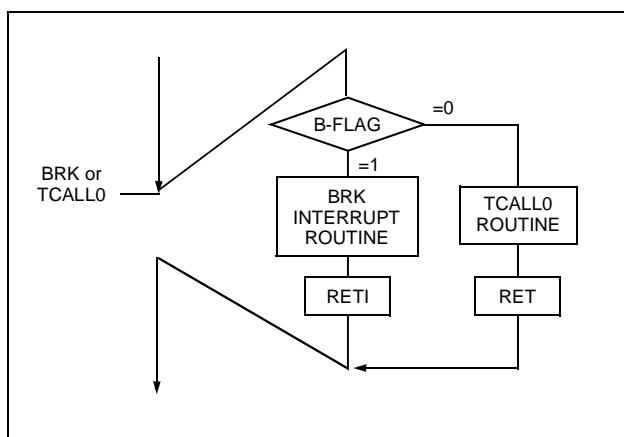


Figure 17-5 Execution of BRK/TCALL0

17.3 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.

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: During Timer1 interrupt is in progress, INTO interrupt serviced without any suspend.

```

TIMER1:  PUSH    A
        PUSH    X
        PUSH    Y
        LDM    IENH,#08H ; Enable INTO only
        LDM    IENL,#00H ; Disable other
        EI      ; Enable Interrupt
        :
        :
        :
        LDM    IENH,#0FFH ; Enable all interrupts
        LDM    IENL,#0FFH
        POP    Y
        POP    X
        POP    A
        RETI
    
```

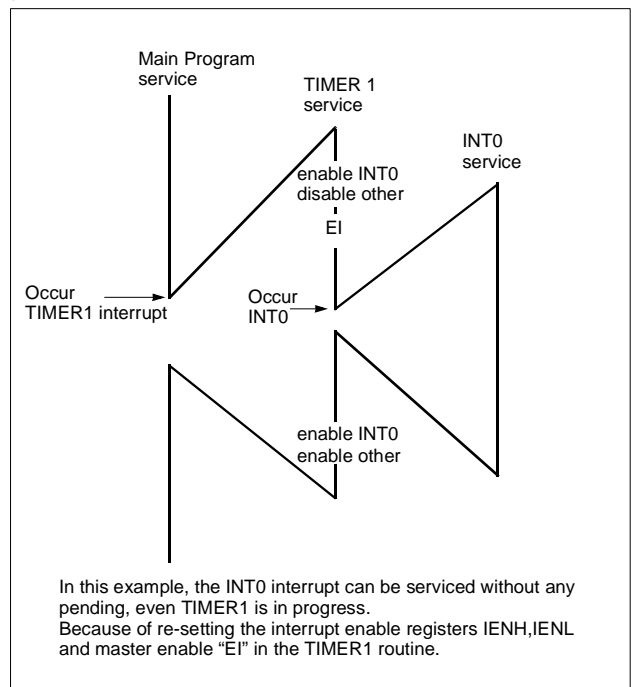


Figure 17-6 Execution of Multi Interrupt

17.4 External Interrupt

The external interrupt on INT0, INT1 and INT3 pins are edge triggered depending on the edge selection register IEDS (address 0D8_H) as shown in Figure 17-7.

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

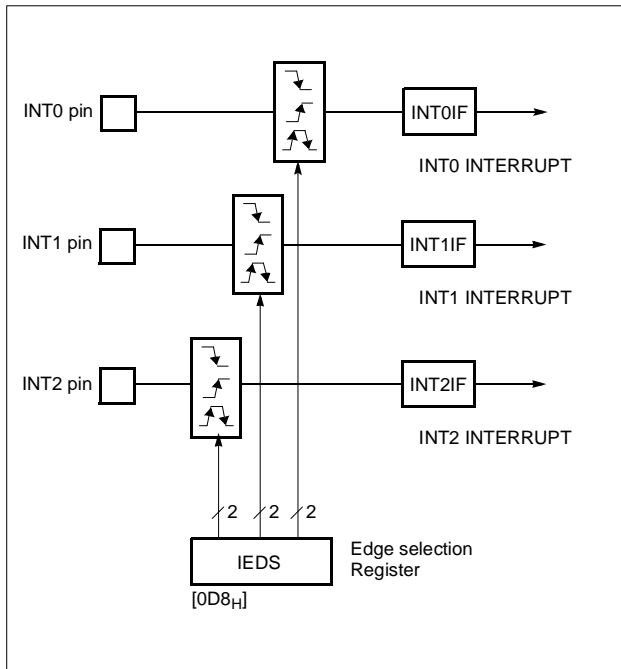


Figure 17-7 External Interrupt Block Diagram

INT0 ~ INT2 are multiplexed with general I/O ports (R00~R02). To use as an external interrupt pin, the bit of Port Mode Register PMR should be set to “1” correspondingly as shown in Figure 17-9.

17.5 Key Scan Interrupt

GMS81C7008/16 has the key-scan block which consists of Port selection Multiplexer, Interrupt controller, Key scan mode register and Falling edge detector shown as Figure 17-10.

When the key scan interrupt is used, key scan register KSMR (address 0F0_H) should be set to “1” as KS0 and

Example: To use as an INT0 and INT2

```

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

```

Response Time

The INT0 ~ INT2 edge are latched into INT1IF ~ 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.

Figure 17-8 shows interrupt response timings.

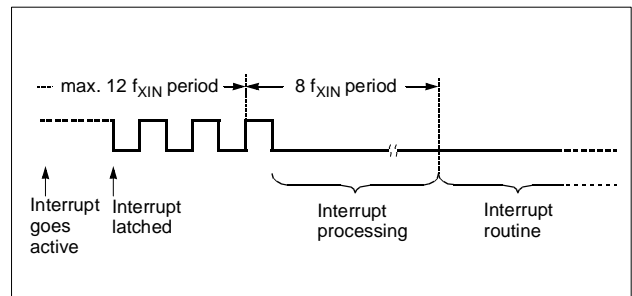


Figure 17-8 Interrupt Response Timing Diagram

KS1. After reset, initial setting is general R10 and R00 ports.

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.

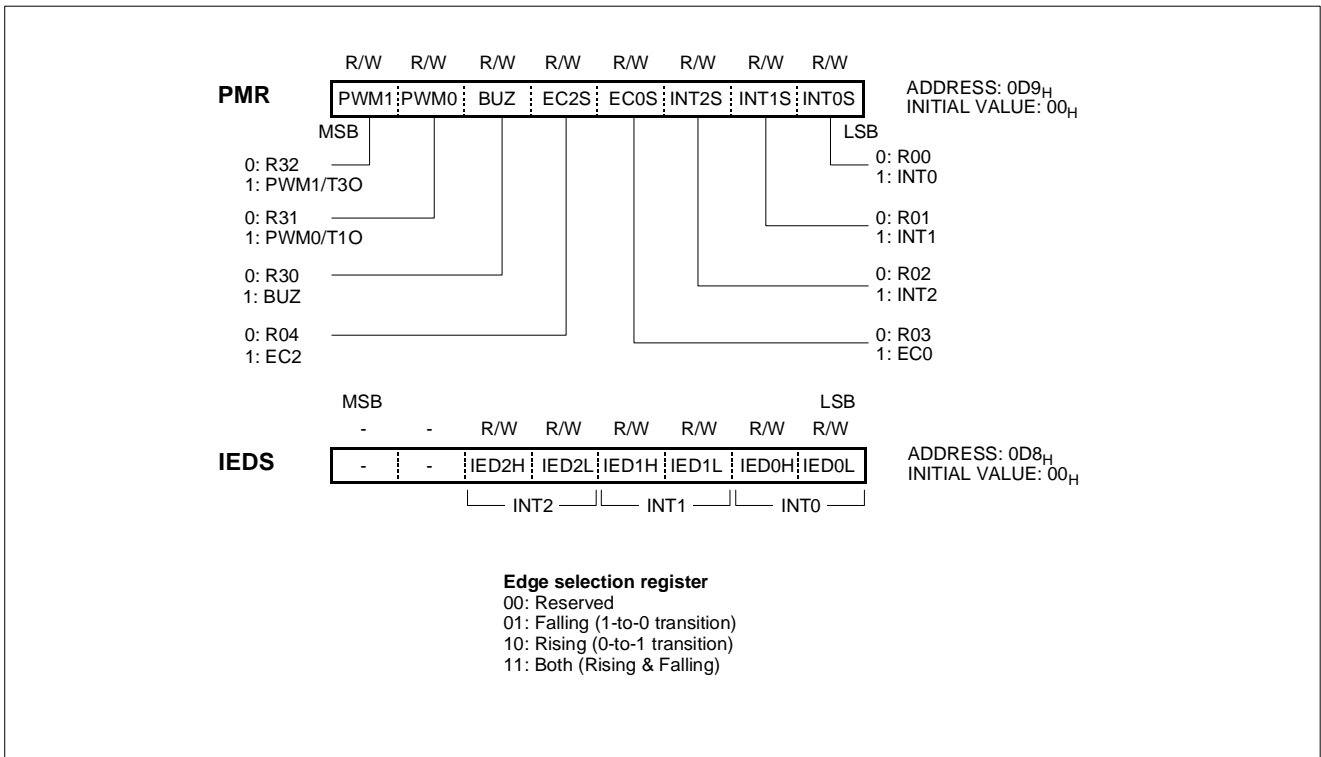


Figure 17-9 PMR and IEDS Registers

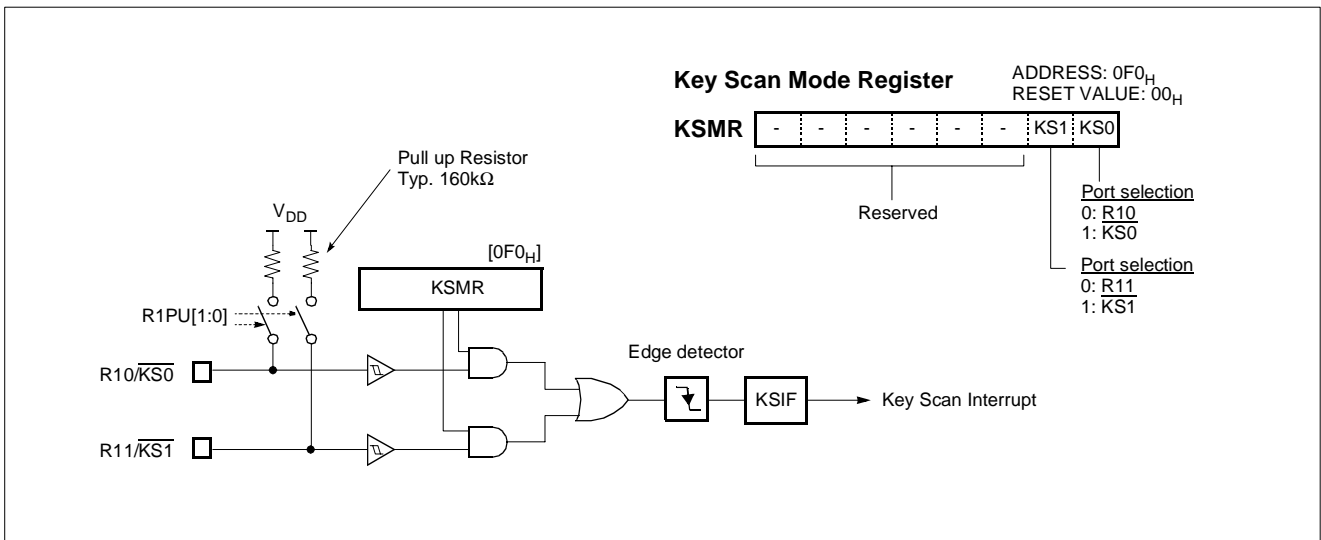


Figure 17-10 Key Scan Port Block Diagram

18. LCD DRIVER

The GMS81C7008/16 has the circuit that directly drives the liquid crystal display (LCD) and its control circuit. In addition, VCL n pin is provided as the drive power pin.

Basically, the GMS81C7008/16 has 24 seg.× 4 com. ports of LCD driver. Extend display modes are shown in left table.

Figure 18-1 shows the configuration of the LCD driver.

*******Caution*******

When you developing the software using by Emulator, you must select the External bias resistor mode because of no internal bias resistor inside the Emulator (EVA. chip).

GMS81C7008/16	
1/4 duty:	24 seg × 4com
1/3 duty:	25 seg × 3com
1/2 duty:	26 seg × 2com
Static:	27 seg × 1com

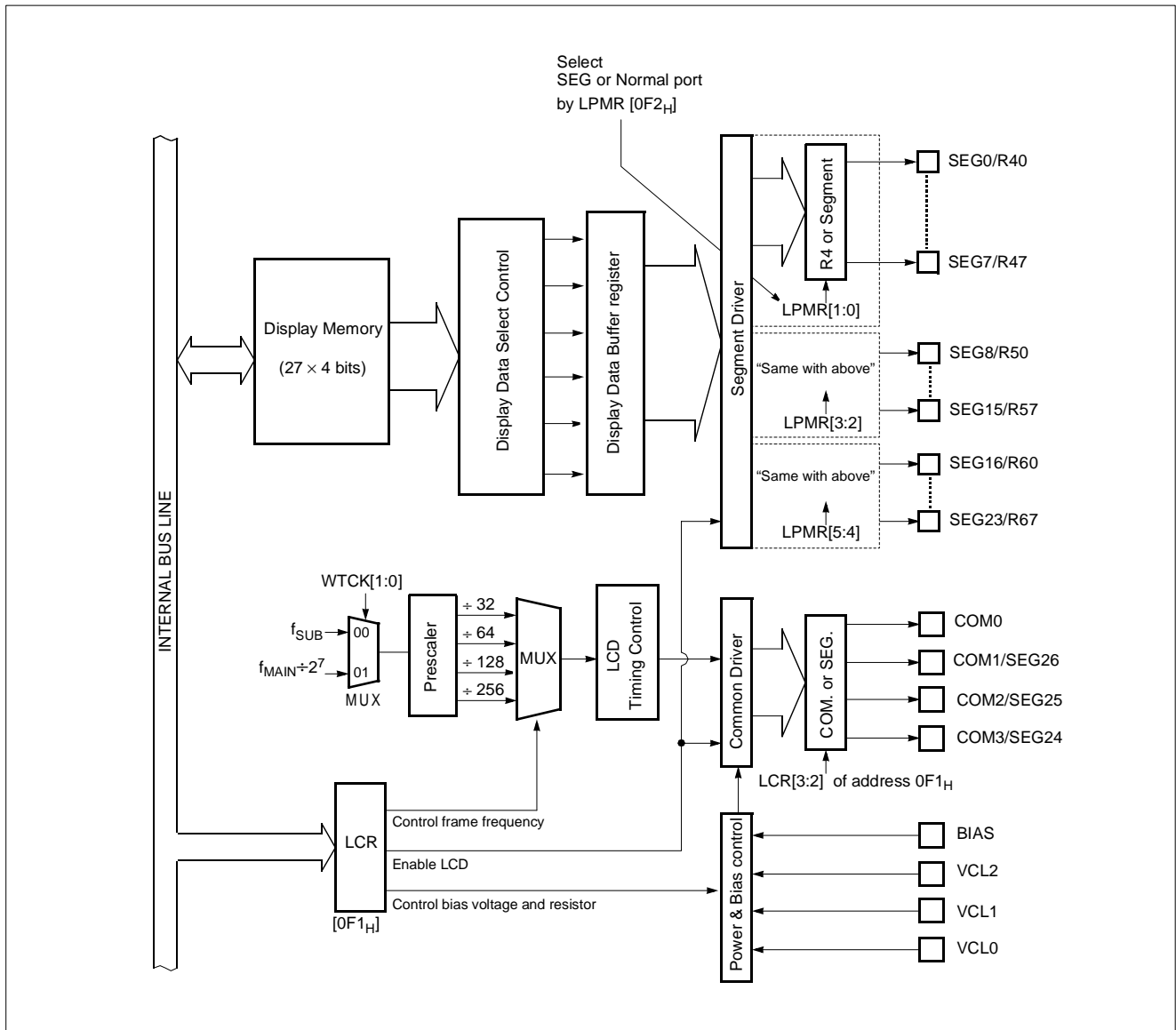


Figure 18-1 LCD Driver Block Diagram

18.1 LCD Control Registers

The LCD driver is controlled by the LCD control register LCR

which is shown in Figure 18-2. LCD block input the clock from

the Watch Timer. When LCD is operate, the Watch Timer much be enabled by WTEN (bit 6 of address 0EF_H).

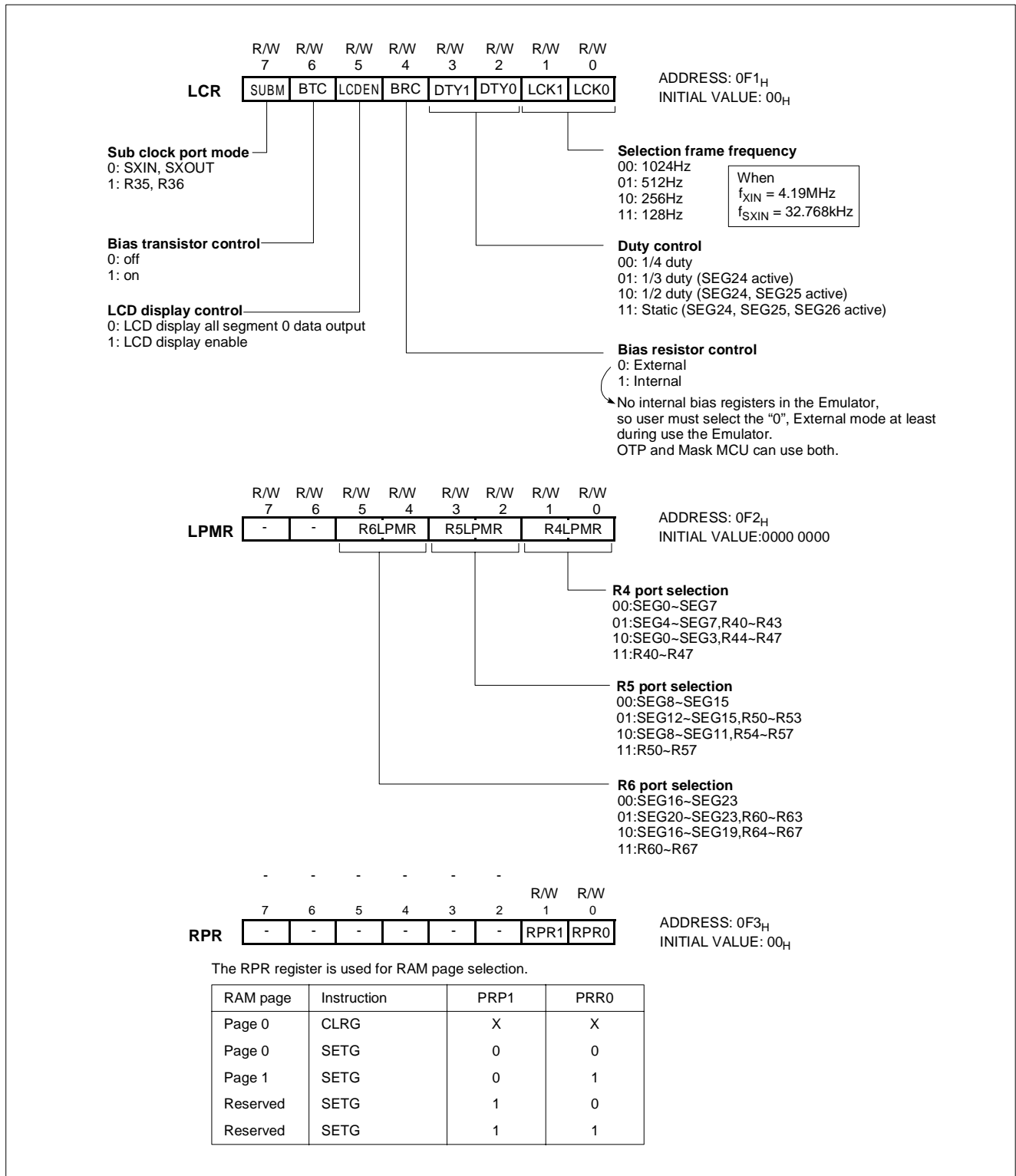


Figure 18-2 LCD Control Register

18.2 Duty and Bias Selection of LCD driver

5 kinds of driving methods can be selected by DTY (bits 3 and 2 of LCD Control Register and connection of VCL pin externally.

Figure 18-3 shows typical driving waveforms for LCD.)

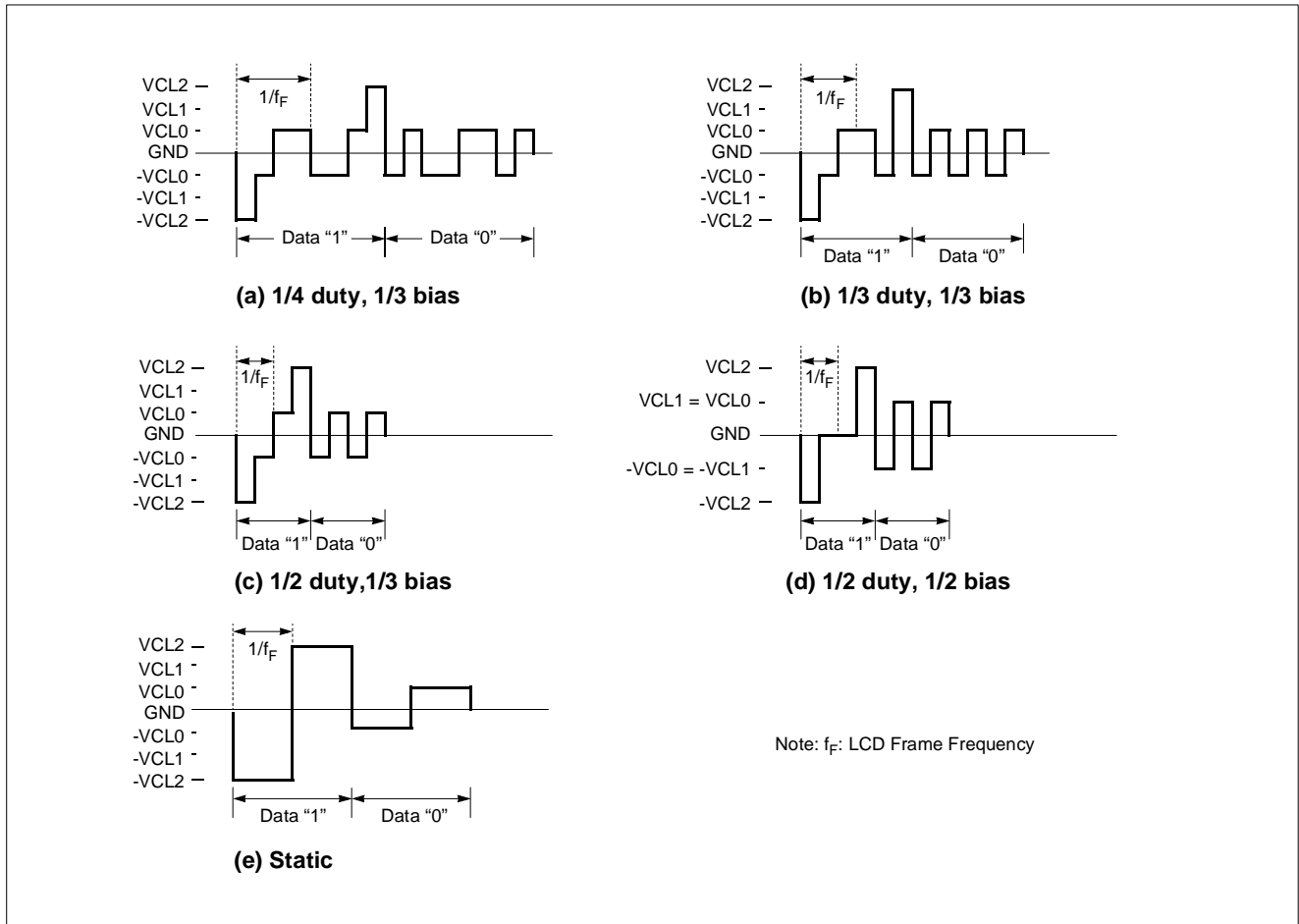


Figure 18-3 LCD drive waveform (Voltage COM-SEG Pins)

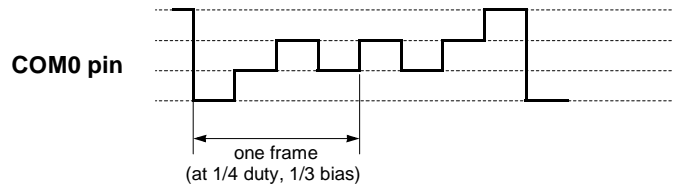
18.3 Selecting Frame Frequency

Frame frequency is set to the base frequency as shown in the following Table 18-1.

The LCK[1:0] of LCR determines the frequency of COM signal scanning of each segment output. The watch timer must be enabled when the LCD display is turned on. RESET clears the LCD control register LCR values to logic zero. The LCD display can continue to operate even during the SLEEP and STOP modes if a sub-frequency clock is oscillate and used as clock source of LCD driver.

LCK[1:0]	LCD clock	Frame Frequency (Hz) (When $f_{SUB} = 32.768$ kHz)
00	$f_{SUB} \div 32$	1024
01	$f_{SUB} \div 64$	512
10	$f_{SUB} \div 128$	256
11	$f_{SUB} \div 256$	128

Table 18-1 Setting of LCD Frame Frequency



LCD Port Selection

Segment pins are also used for normal I/O pins. The LCD port selection register LPMR is used to set Rn pin for ordinary digital input. Refer to LPMR register as shown in Figure 18-2.

Bias Resistor

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

The BTC (Bit 6 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. The LCD drive voltage (V_{CL2}) is given by the difference in potential ($V_{DD}-V_{CL2}$) between pins V_{DD} and V_{CL2} . Therefore, when the MCU operating voltage is 5V and LCD drive voltage are the same, the Bias pin is connected to the V_{CL2} pin as shown in (a) of Figure 18-5.

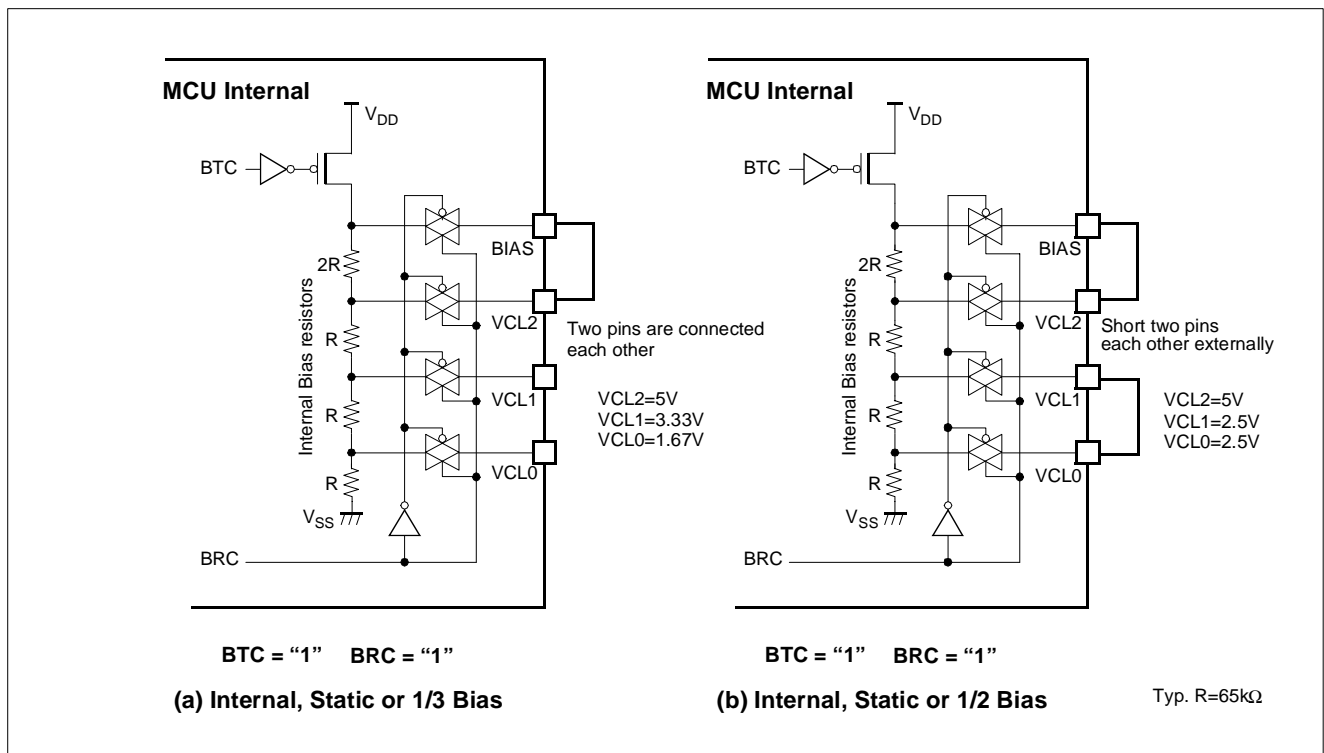


Figure 18-4 Application Example of 5V LCD Panel

When require supply 3V output to the LCD, the voltage of V_{CL2} becomes 3V as shown in Figure 18-5. Because V_{DD} is down to 3V through internal 2R resistor.

The LCD light only when the difference in potential between the segment and common output is $\pm VCL$, and turn off at all other times. During reset, the power switch of the LCD driver is turned off automatically, shutting off the VCL voltage.

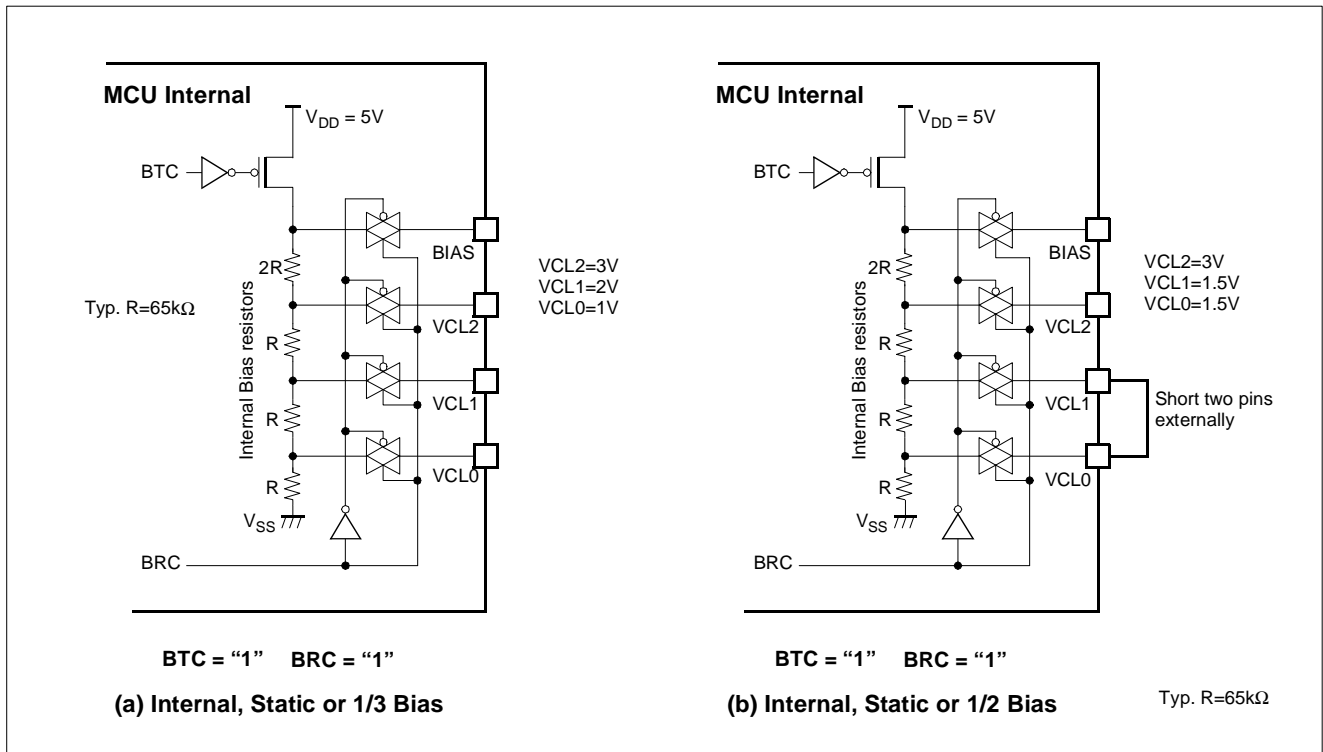


Figure 18-5 Application Example of 3V LCD Panel

Some user want to use external bias resistor instead of internal, you can connect external resistor as shown in Figure 18-6. And

the external capacitors are may required for stable display according to your system environment.

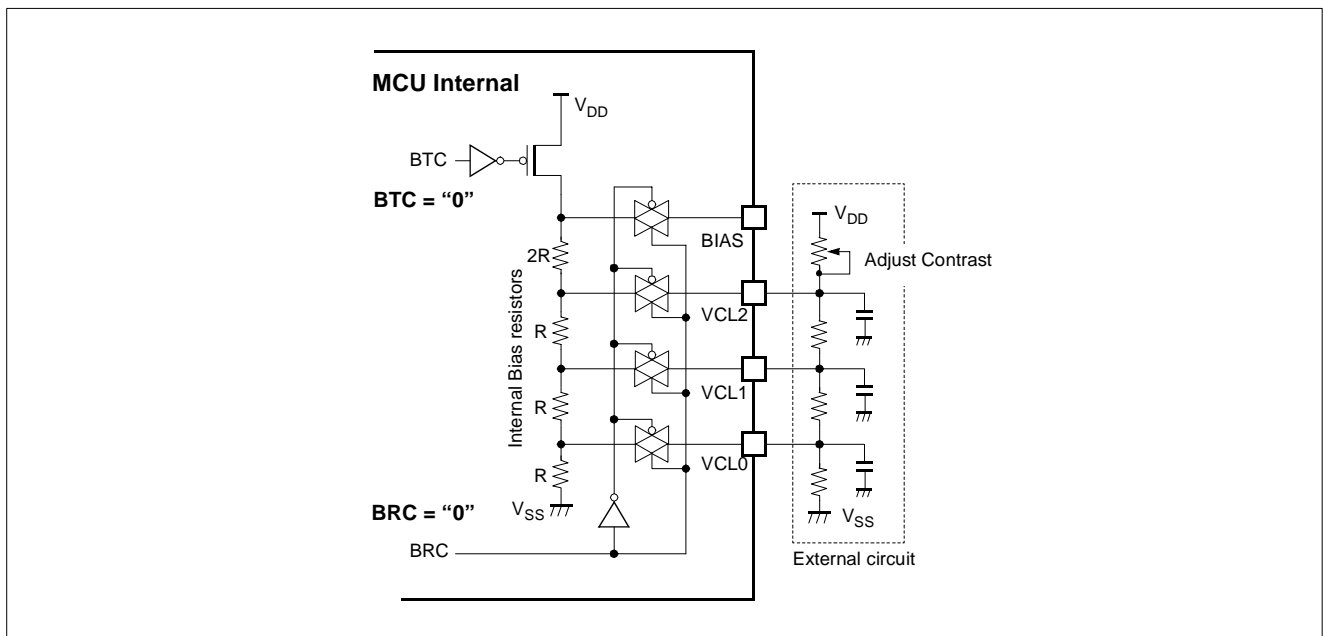


Figure 18-6 External Resistor

18.4 LCD Display Memory

Display data are stored to the display data area (address 100_H-11A_H) in the data memory. The display data stored to the display data area 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.

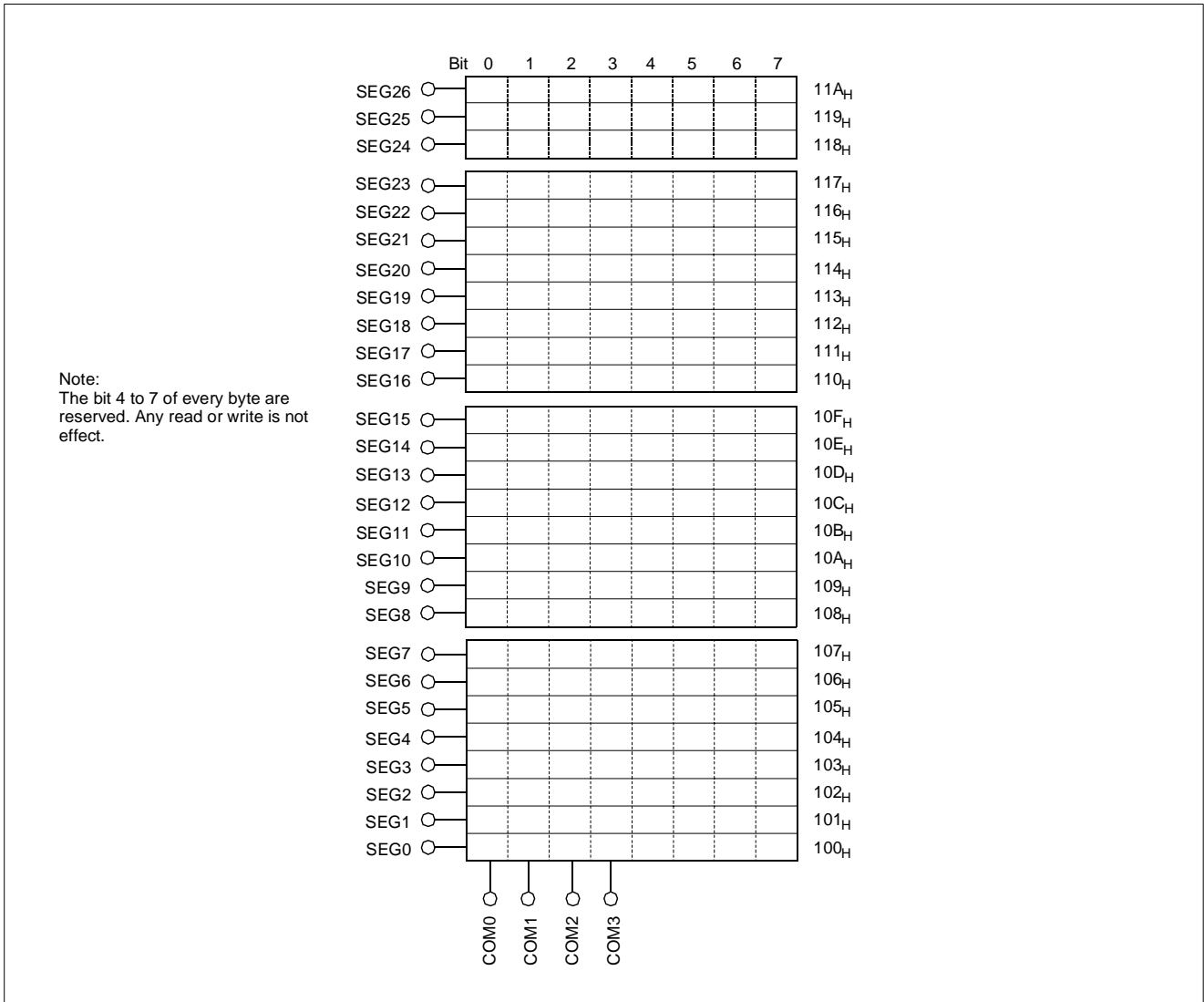


Figure 18-7 LCD Display Memory

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 18-7 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”.

The number of segment which can be driven differs depending on the LCD drive method, therefore, the number of display data area bits used to store the data also differs

(Refer to Figure 18-2). Consequently, data memory not

Drive methods	Bit 3	Bit 2	Bit 1	Bit 0
1/4 duty	COM3	COM2	COM1	COM0
1/3 duty	-	COM2	COM1	COM0
1/2 duty	-	-	COM1	COM0
Static	-	-	-	COM0

Table 18-2 The duty vs. COM port Configuration

example for displaying character is shown below.

```

Write into the LCD Memory
[
:
CLRG
LDX #DISPRAM
GOLCD: LDA {X}
TAY
LDA !FONT+Y ;LOAD FONT DATA
LDM RPR,#1 ;Set RPR = 1 to access LCD
SETG ;Set Page 1
LDX #0
STA {X}+ ;LOWER 4 BITS OF ACC. -> M(X)
XCN
STA {X} ;UPPER 4 BITS OF ACC. -> M(X+1)
CLRG ;Set Page = 0
:
:
]

Font data
[
FONT DB 1101_0111B ; "0"
DB 0000_0110B ; "1"
DB 1110_0011B ; "2"
DB 1010_0111B ; "3"
DB 0011_0110B ; "4"
DB 1011_0101B ; "5"
DB 1111_0101B ; "6"
DB 0000_0111B ; "7"
DB 1111_0111B ; "8"
DB 0011_0111B ; "9"
]

```

Note: When power on RESET, sub oscillation start up time is required. Enable LCD display after sub oscillation is stabilized, or LCD may occur flicker at power on time shortly.

19. WATCH / WATCHDOG TIMER

19.1 Watch Timer

The watch timer goes the clock continuously even during the power saving mode. When MCU is in the Stop or Sleep mode, MCU can wake up itself every 2Hz or 4Hz or 16Hz.

The watch timer consists of input clock selector, 14-bit binary counter, interval selector and Watch Timer Mode Register WTMR (address 0EF_H). The WTMR is 5-bit read/write register and shown in Figure 19-2. WTMR can select the clock input by 2 bits WTCK[1:0] and interval time selector by 2 bits WTIN[1:0] and enable/disable bit. The WTEN bit is set to "1" timer start counting. Input clocks can be selected among three different source which are sub clock or divided main clock ($f_{XIN} \div 128$) or main clock. For the switching between main and sub clock, rec-

ommend the oscillator 4.194304MHz as a main and 32.768kHz as a sub. Because above main frequency is equal to 128 times of sub frequency. Generally main clock (f_{XIN}) at WTCK=10_B is not used, it is just for test purpose in factory.

In the Stop Mode, the main clock is stopped but sub clock is oscillation continuously for watch clock operation. Output timer interval can be selected and Watch Timer Interrupt is generated.

```
LDM   IENL, #XXXX_X1XXB
EI
LDM   WTMR, #0100_1000B
```

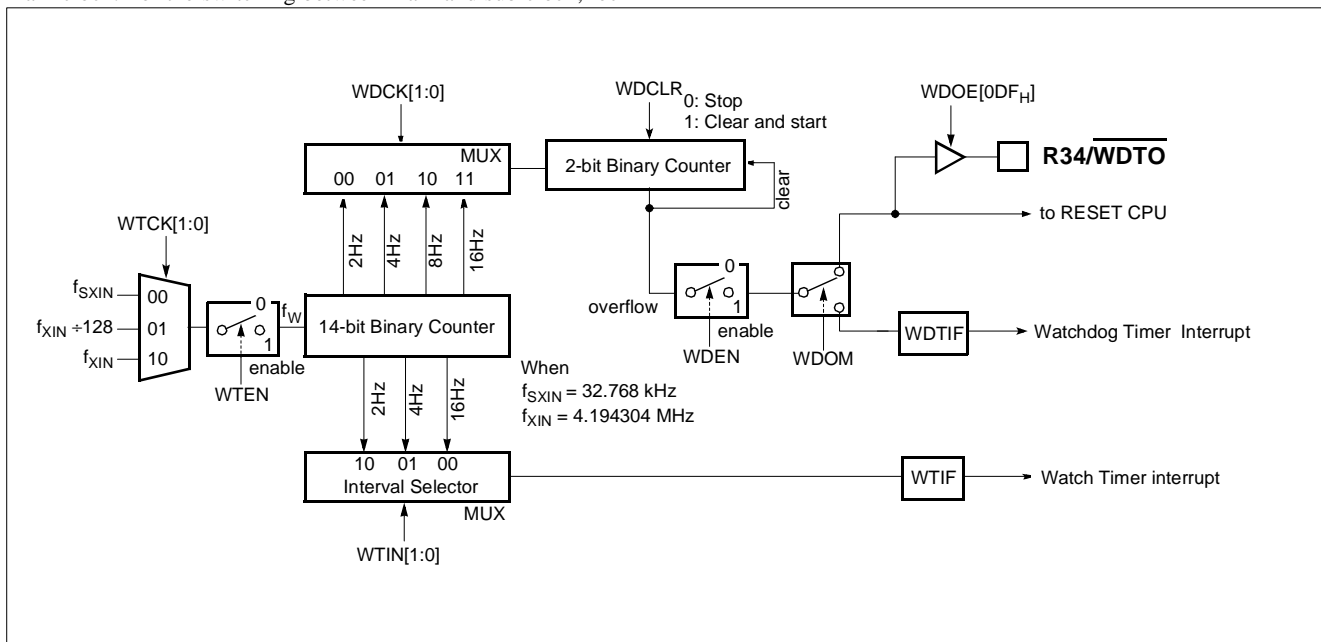


Figure 19-1 Block Diagram of Watchdog Timer

19.2 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 an interrupt request as you want.

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.

Watchdog Timer Control

Figure 19-2 shows the watchdog timer control register WDTR (address 0DF_H). The watchdog timer is automatically enabled initially and watchdog output to reset CPU but clock input source is disabled. To enable this function, you should write bit WTEN of WTMR (address 0EF_H) set to "1".

The CPU malfunction is detected during setting of the detection time, selecting of output, and clearing of the binary counter. Clearing the 2-bit binary counter by bit WDCLR of WDTR is repeated within the detection time.

If the malfunction occurs for any cause, the watchdog timer output will become active from the binary counters unless the binary counter is cleared. At this time, when WDOM=1, a reset is generated, which drives the RESET pin to low to reset the internal hardware. When WDOM=0, a watchdog timer interrupt (WDTIF) is generated instead of Reset function. This interrupt can be used general timer as user want.

When main clock is selected as clock input source on the STOP mode, clock input is stopped so the watchdog timer temporarily stops counting. The other side, when sub clock is selected as clock input source on the STOP mode, sub clock operates always

so the watchdog timer works continuously.

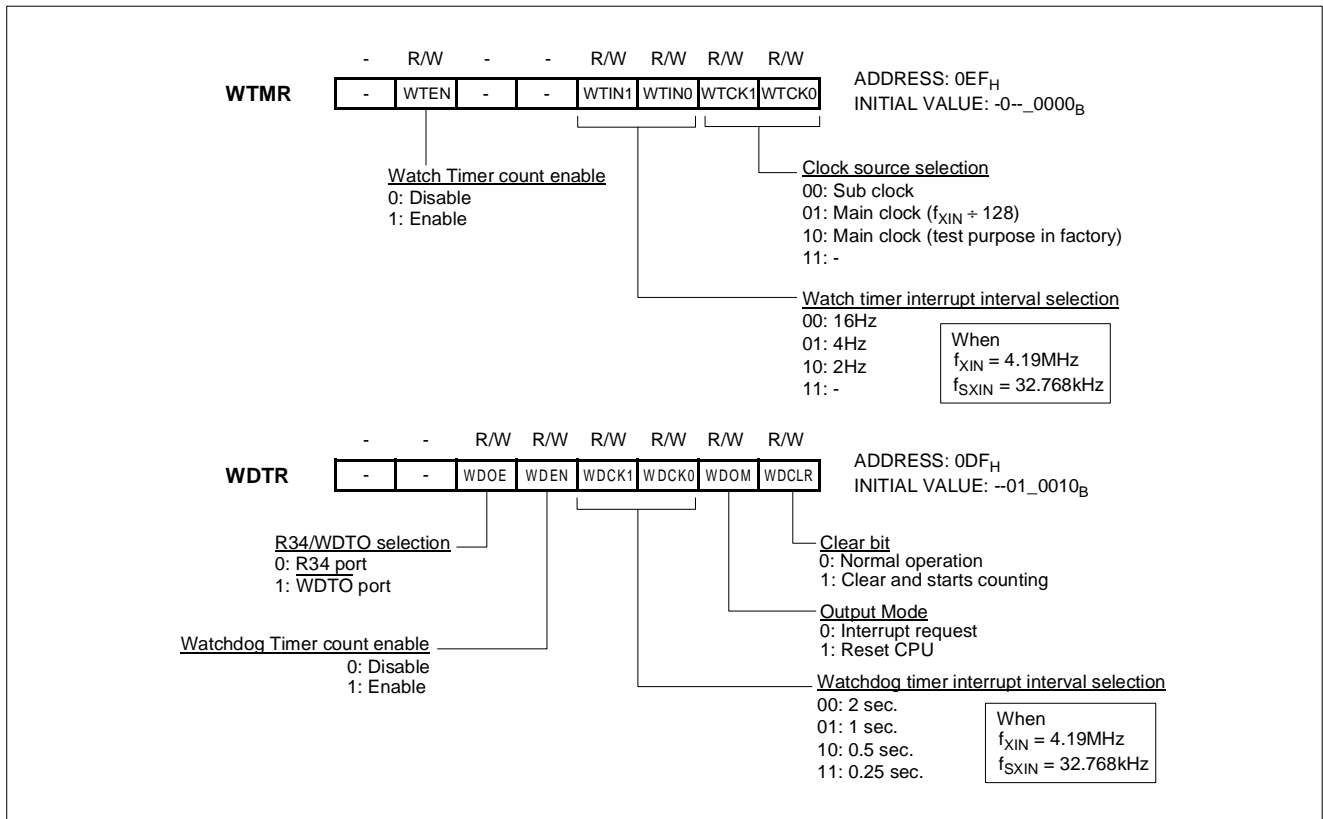


Figure 19-2 WTMR, WDTR: Watch Timer and Watchdog Timer Data Register

Example: Sets the watchdog timer detection time to 1 sec at 4.19MHz, 32.768kHz

```

LDM    WTMR, #0100_1000B ; Select sub clock as an input source
LDM    WDTR, #0001_0111B

Within 0.75 sec. SET1    WDCLR ; Clear counter
                :
                :
                :
                :
Within 0.75 sec. SET1    WDCLR ; Clear counter
                :
                :
                :
                :
                SET1    WDCLR ; Clear counter
    
```

Enable and Disable Watchdog

Watchdog timer is enabled by setting WDEN (bit 4 in CKCTLR) to “1”. WDEN is initialized to “1” during reset and it should be clear to “0” disable.

Example: Enables watchdog timer for Reset

```

:
LDM    WTMR, #0100_XXXXB; WTEN ← 1
LDM    WDTR, #00X1_XX11B; WDEN ← 1
:
    
```

The watchdog timer is disabled by clearing either bit 4 (WDEN) of WDTR or bit 6 (WTEN) of WTMR. The watchdog timer is halted in STOP mode and restarts automatically after STOP mode is released.

Clearing 2-bit binary counter of the Watchdog timer

The watchdog timer count the clock source as 14-bit binary

counter which is free run can not be cleared. The watchdog timer has 2-bit binary counter. It is incremented by 14-bit binary counter match as shown in Figure 19-1. Interrupt request flag or Reset signal are generated by overflow 2-bit binary counter.

During normal operation in the software, 2-bit binary counter

should be cleared by bit WDCLR of WDTR within watchdog timer overflow.

The time of clearing must be within 3 times of 14-bit binary counter interval as shown in Figure 19-3.

The worst case, watchdog time is just 3 times of 14-bit counter.

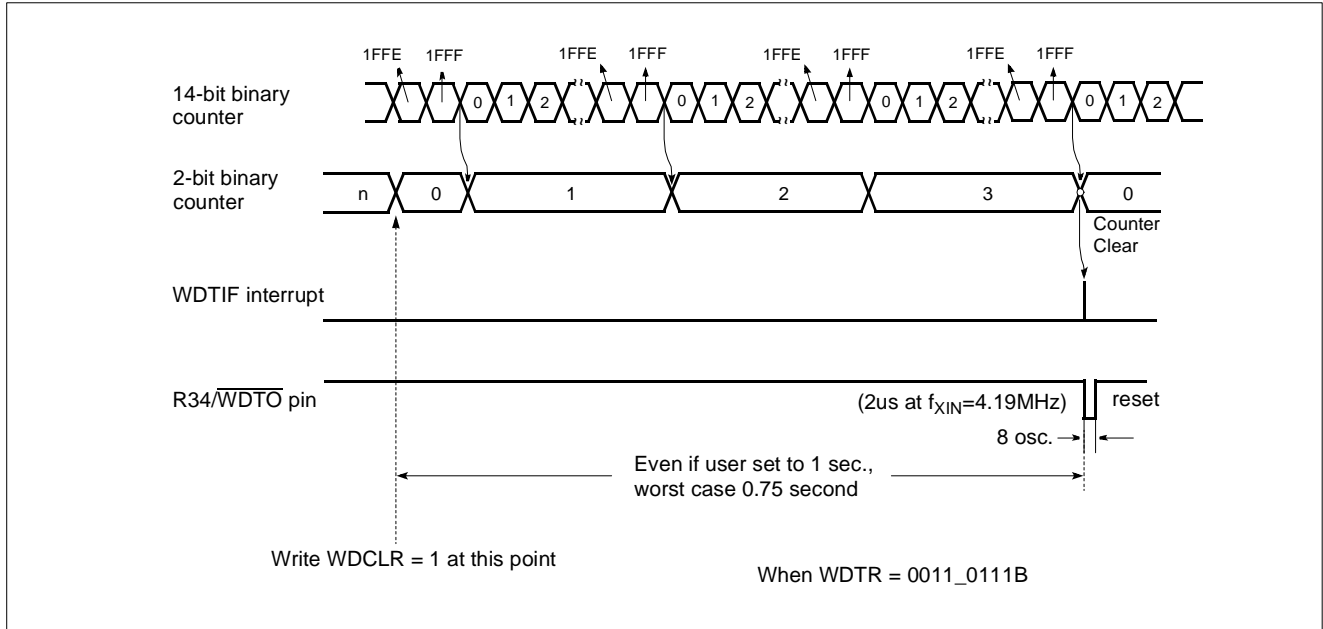


Figure 19-3 Watchdog timer Timing

If the watchdog timer output becomes active, a reset is generated, which drives the **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.

20. POWER DOWN OPERATION

The GMS81C7008/16 has two power-down modes. 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.

20.1 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 20-1. 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.

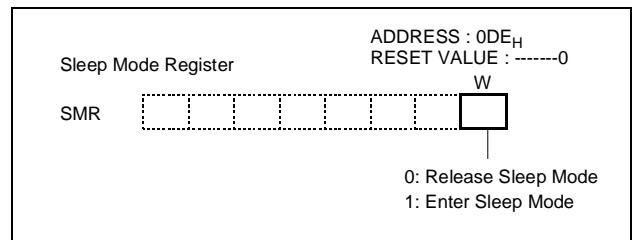


Figure 20-1 SLEEP Mode Register

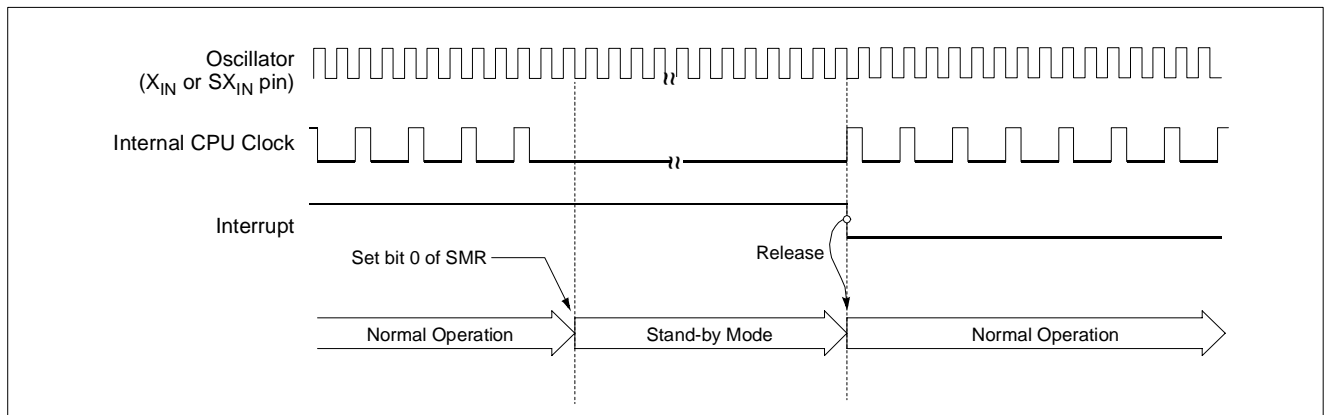


Figure 20-2 Sleep Mode Release Timing by External Interrupt

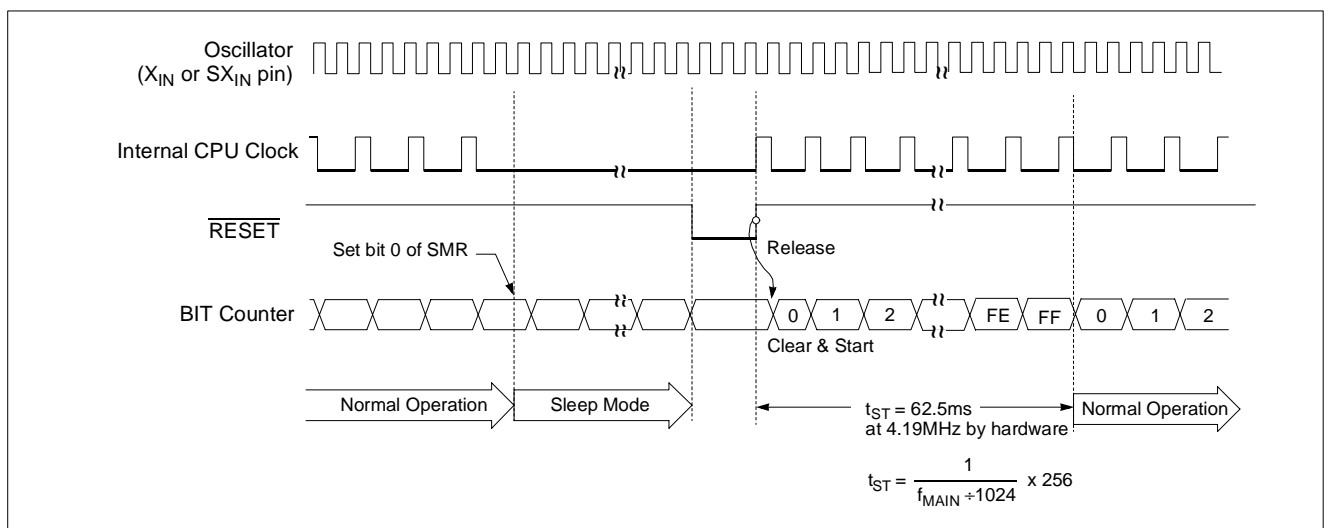


Figure 20-3 SLEEP Mode Release Timing by RESET pin

20.2 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	LCD driver operates continuously	LCD driver operates continuously
Basic Interval Timer	Halted	BIT operates continuously
Timer/Event counter	Halted (Only when the Event counter mode is enabled, Timer operates normally)	Timer/Event counter operates continuously
Watch Timer	Watch Timer operates continuously	Watch Timer operates continuously
Main-oscillation	Stop (X_{IN} pin = "L", X_{OUT} pin = "L")	Oscillation
Sub-oscillation	Oscillation	Oscillation
I/O ports	Retain	Retain
Control Registers	Retain	Retain
Release method	RESET, Key Scan interrupt, SIO interrupt, Watch Timer interrupt, Timer interrupt (EC0,2), External interrupt	RESET, All interrupts

Table 20-1 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, #0EB ; 32.8ms
;      LDM   CKCTLR, #0FB ; 65.5ms
;      STOP
;      NOP
;      NOP

```

The Interval Timer Register CKCTLR should be initialized (0FH or 0EH) 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, EC0 or EC2 pin can release it by Timer/Event counter Interrupt request.

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.

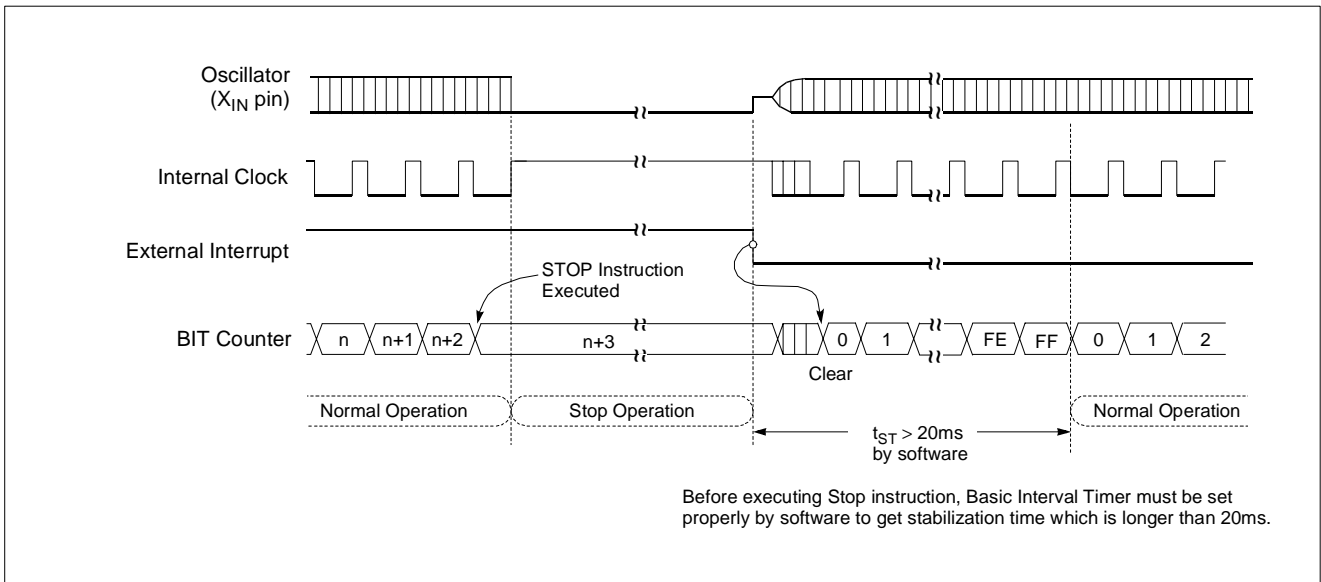


Figure 20-4 STOP Mode Release Timing by External Interrupt

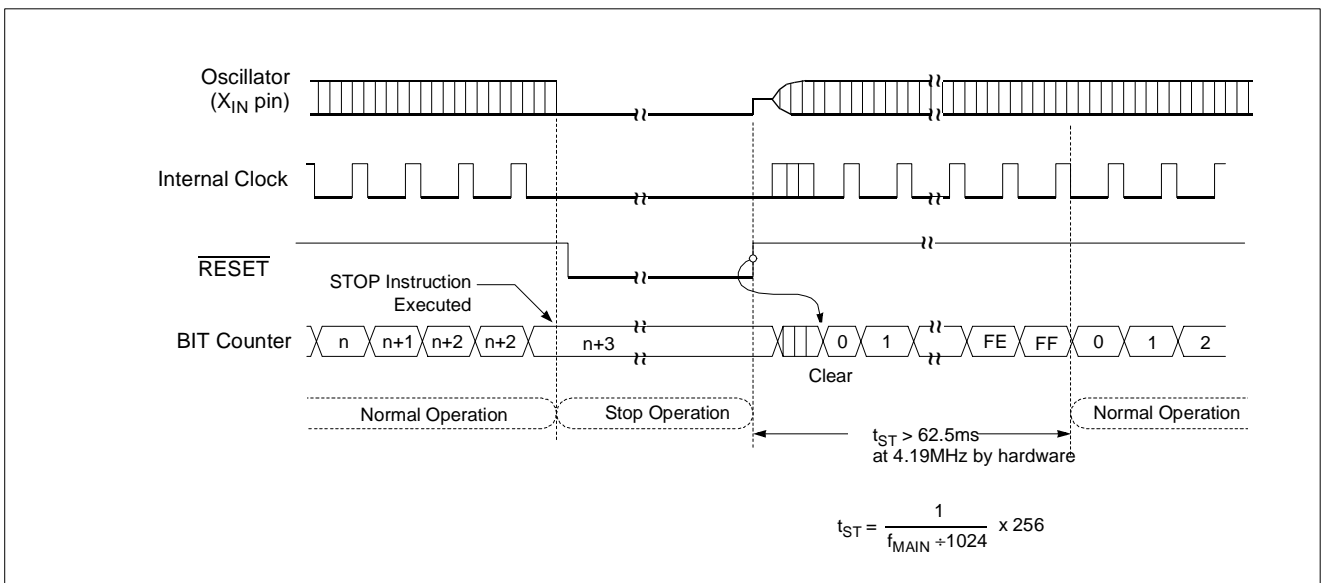


Figure 20-5 STOP Mode Release Timing by RESET

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 un-firmed 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.

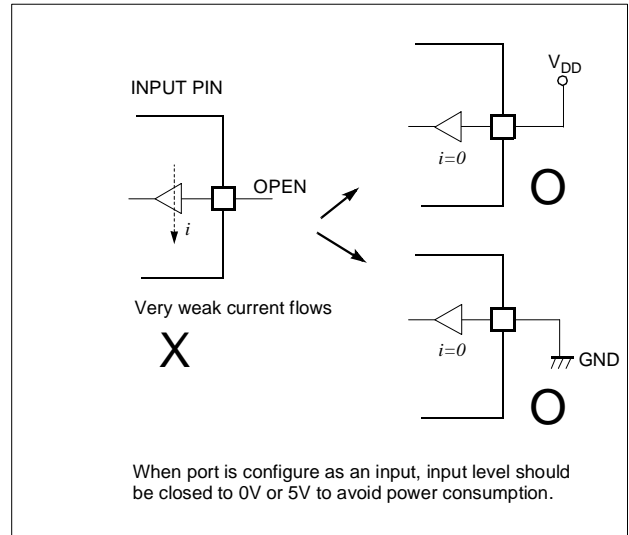
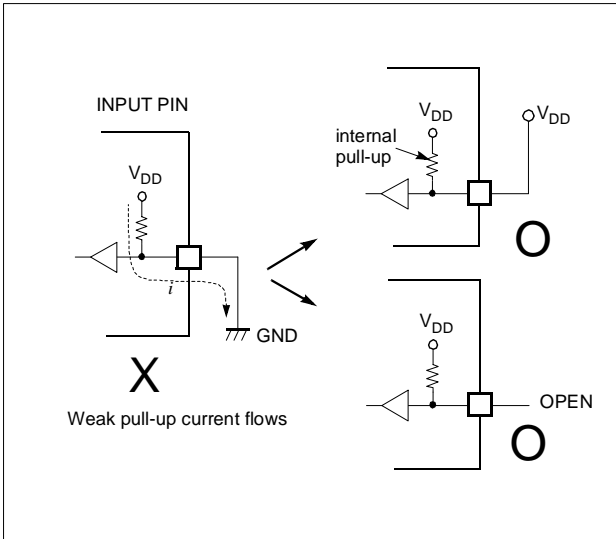


Figure 20-6 Application Example of Unused Input Port

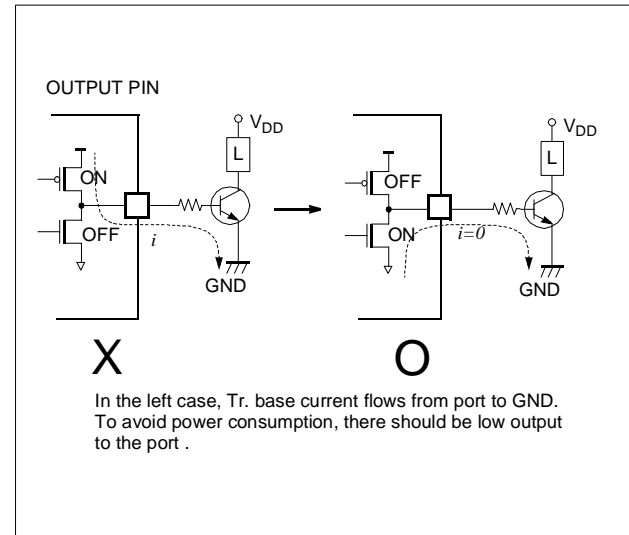
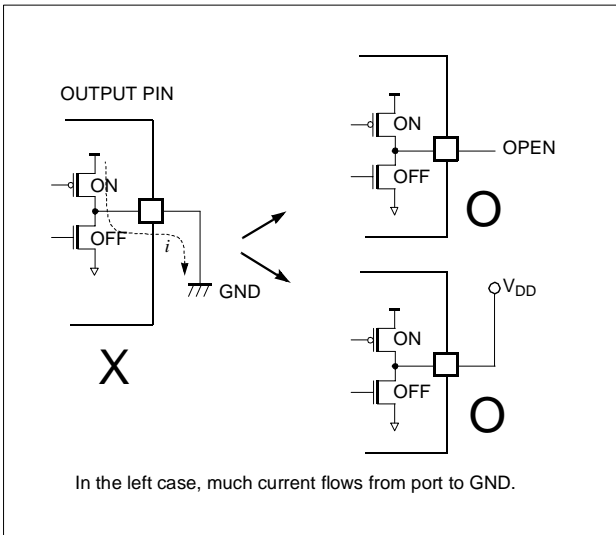


Figure 20-7 Application Example of Unused Output Port

21. OSCILLATOR CIRCUIT

The GMS81C7008/16 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 21-1. To use RC oscillation instead of crystal, user should check mark on the "A. MASK ORDER SHEET" on page i of the appendix of this manual. However in the OTP device, when the programming RC oscillation can be selected or not into the configuration bit. For more detail, refer to

"24.1 OTP Programming" on page 91.

Note: When using the sub clock oscillation, connect a resistor in series with R which is shown as below figure. In order to reduce the power consumption, the sub clock oscillator employs a low amplification factor circuit. Because of this, the sub clock oscillator is more sensitive to noise than the main system clock oscillator.

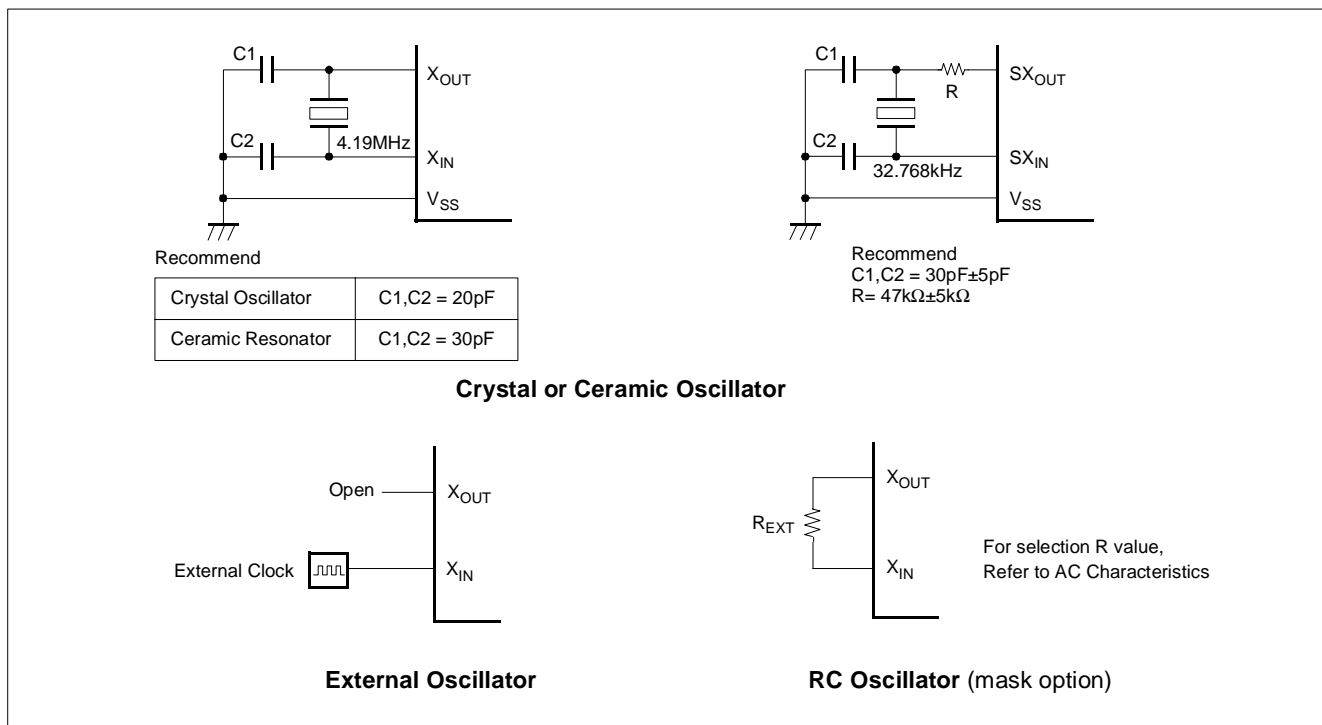


Figure 21-1 Oscillation Circuit

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 21-2 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 V_{SS} . Do not ground it to any ground pattern where high current is present. Do not fetch signals from the oscillator.

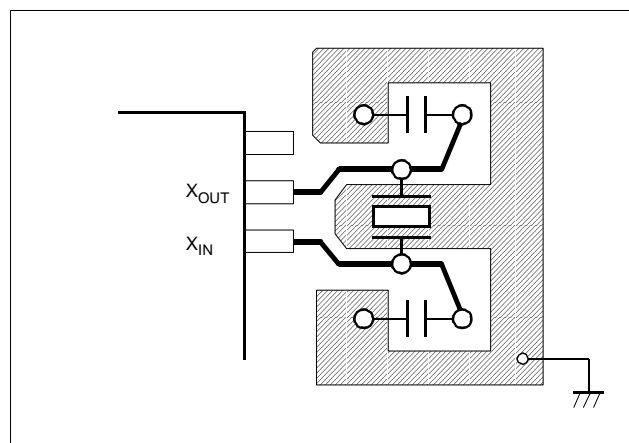


Figure 21-2 Recommend Layout of Oscillator PCB circuit

22. RESET

The GMS81C7008/16 has two types of reset generation procedures; one is an external reset input, the other is a watch-dog timer reset. Table 22-1 shows on-chip hardware initialization by reset action.

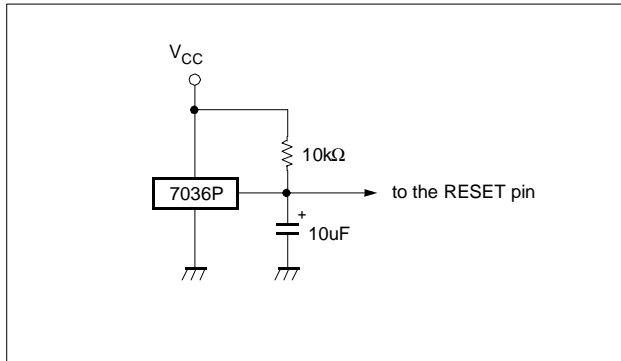


Figure 22-1 Simple Power-on-Reset Circuit.

On-chip Hardware	Initial Value
Program counter (PC)	(FFFF _H) - (FFFE _H)
G-flag (G)	0
Operation mode	Main operating mode
Peripheral clock	On
Watchdog timer	Disable (Because the Watch timer is disabled)
Control registers	Refer to Table 8-1 on page 27
Low voltage detector	Enable

Table 22-1 Initializing Internal Status by Reset Action

22.1 External Reset Input

The reset input is the RESET pin, which is the input to a Schmitt Trigger. A reset is 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 22-2.

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 .

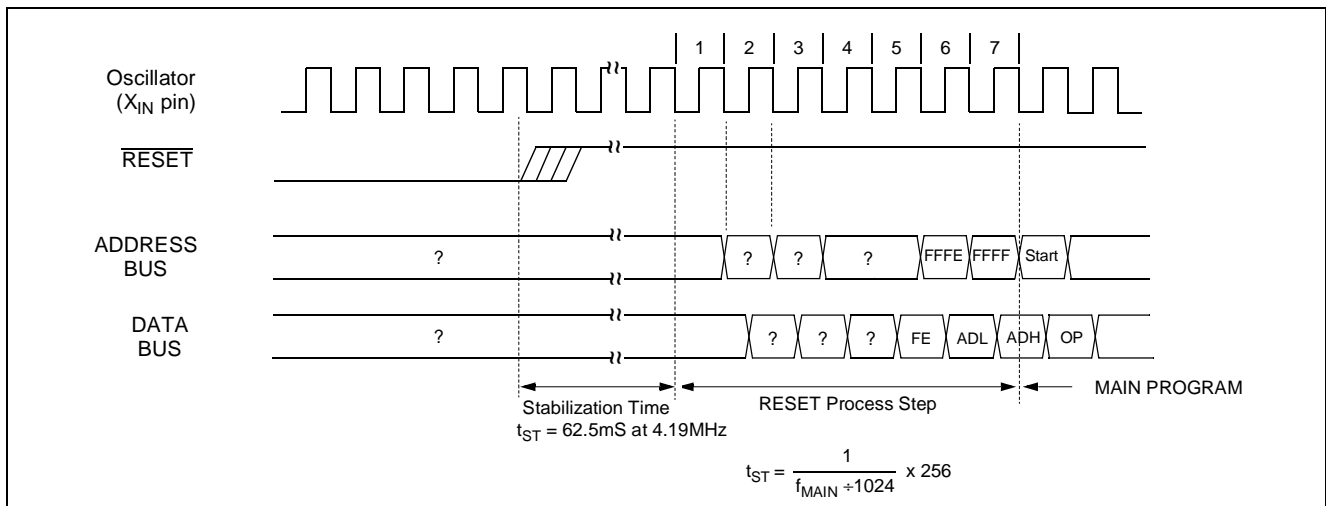


Figure 22-2 Timing Diagram after RESET

22.2 Watchdog Timer Reset

Refer to “18. LCD DRIVER” on page 72.

23. POWER FAIL PROCESSOR

The GMS81C7008/16 has an on-chip low voltage detection circuitry to detect the V_{DD} voltage. A configuration register, LVDR (address 0FB_H), can enable or disable the low voltage detect circuitry. Whenever V_{DD} falls close to or below 2.2V, the LVD0 is just set to "1", and if it recovering 3.4V, LVD0 is held to "1". If V_{DD} falls below around 3.4V range, the low voltage situation may reset the MCU or freeze the clock according to setting of bit 5 (LVDM) of LVDR . The bit 4 LVD1 function is same with LVD0 except different voltage level 2.1V. The detection voltage is varied very little. See "7.3 DC Electrical Characteristics" on page 13 for more detail voltage level.

In the in-circuit emulator, power fail function is not implemented and user may not use it. Therefore, after completed development of user program, this function may be experimented or evaluated using by OTP.

When power fail certainly occur the MCU was reset, program notify this Reset circumstance cause by LVD function. So, does not erase the all RAM contents and operates subsequently as shown in Figure .

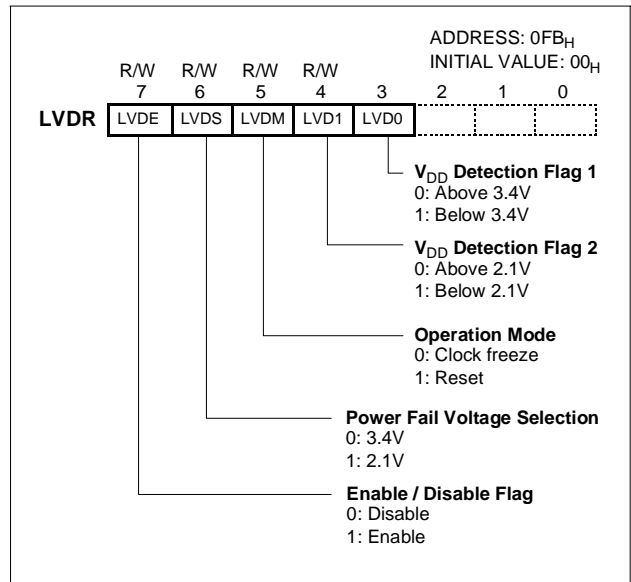


Figure 23-1 Low Voltage Detector Register

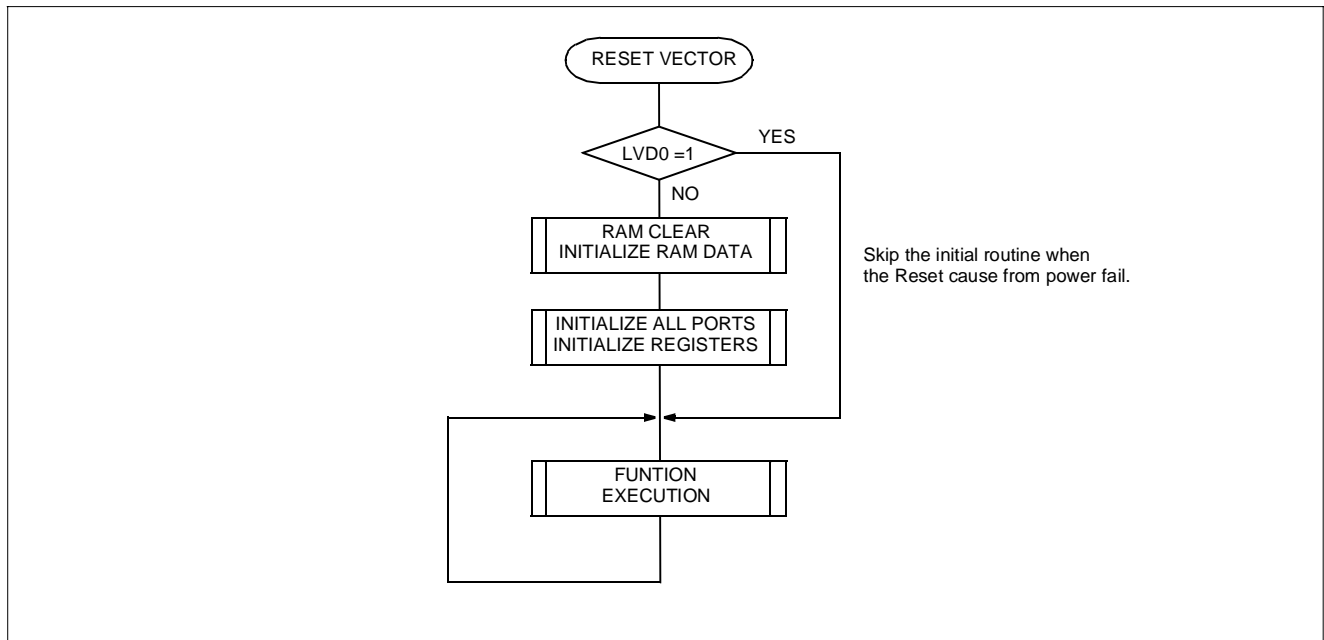


Figure 23-2 Example S/W of RESET by Power fail

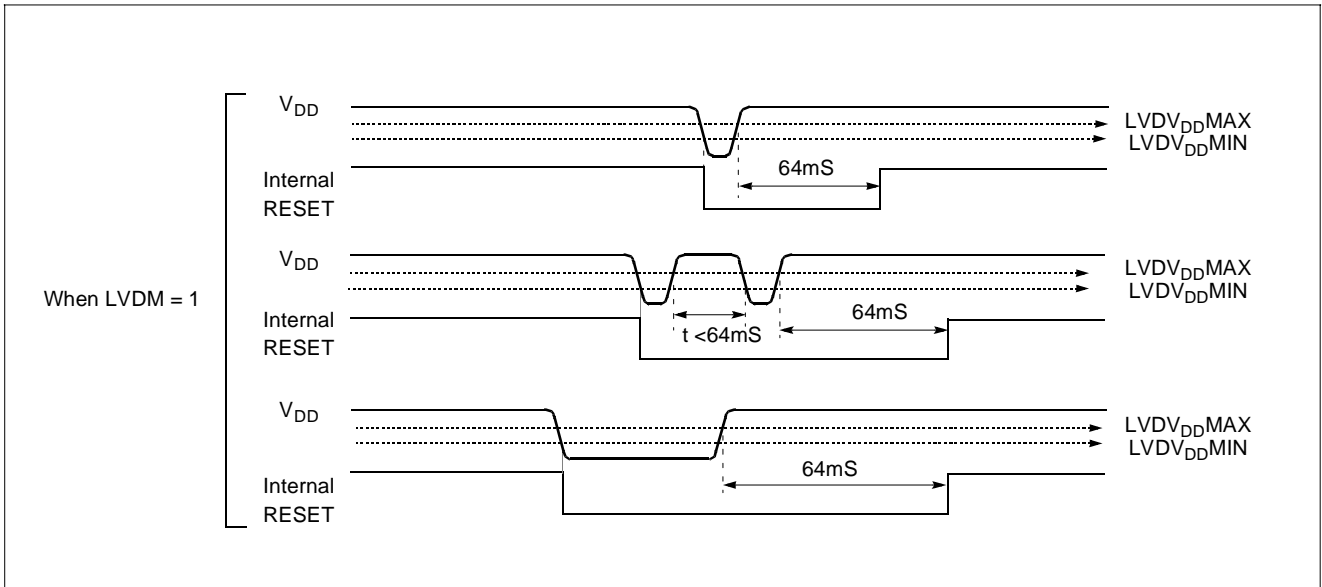


Figure 23-3 Power Fail Processor Situations

24. DEVELOPMENT TOOLS

24.1 OTP Programming

The GMS87C7016 is OTP (One Time Programmable) type microcontrollers. Its internal user memory is constructed with EPROM (Electrically Programmable Read Only Memory).

The OTP microcontroller is generally used for chip evaluation, first production, small amount production, fast mass production, etc.

Blank OTP's internal EPROM is filled by 00_H, not FF_H.

Note: In any case, you have to use the *.OTP file for programming, not the *.HEX file. After assemble the source program, both OTP and HEX file are generated by automatically. The HEX file is used during program emulation on the emulator.



How to Program

To program the OTP devices, user should use HEI own programmer. Ask to HEI sales part for purchasing or more detail.

Programmer: **CHOICE-SIGMA** (Single type)
CHOICE-GNAG4 (4-gang type)
 Socket adapter: 87C70XX-64SD (for 64SDIP)
 87C70XX-64QF (for 64MQFP)

The CHOICE-SIGMA is a HEI Universal Single Programmer for all of HEI OTP devices, also the CHOICE-GANG4 can program four OTPs at once.

Programming Procedure

1. Select device GMS87C7016 as you want.
2. Load the *.OTP file from the PC to Programmer. The file is composed of Motorola-S1 format.
3. Set the programming address range as below table.
4. Mount the socket adapter on the programmer.
5. Set the configuration bytes as your needs.
6. Start program/verify.

Select the option for Program Lock and RC oscillation

Except the user program memory C000_H~FFFF_H, there is configuration byte (address 707F_H) for the selection of program lock and RC oscillation. The configuration byte of OTP is shown as Figure 24-1. It could be served when user use the OTP programmer (Choice-Sigma or Choice-Gang4).

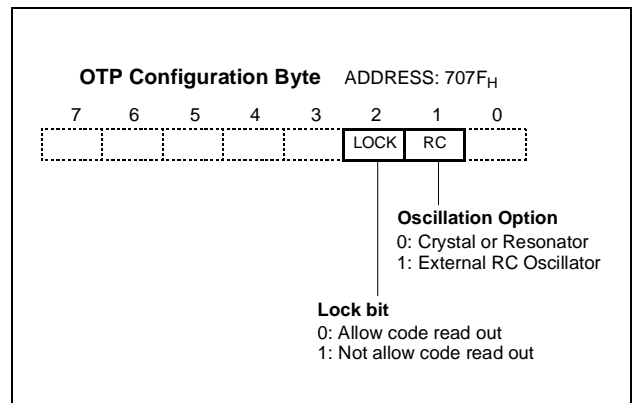
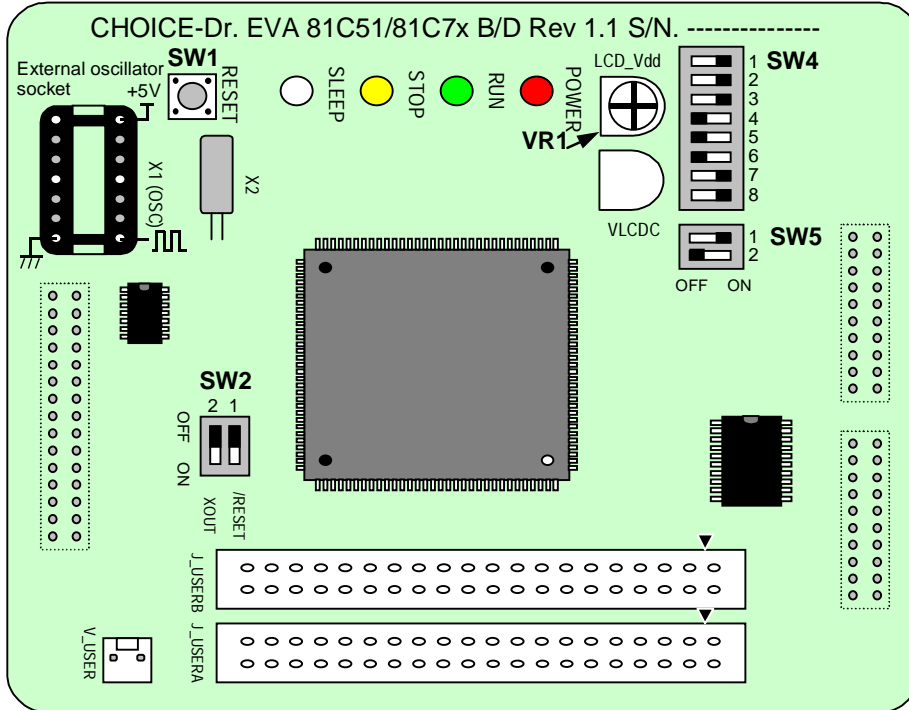
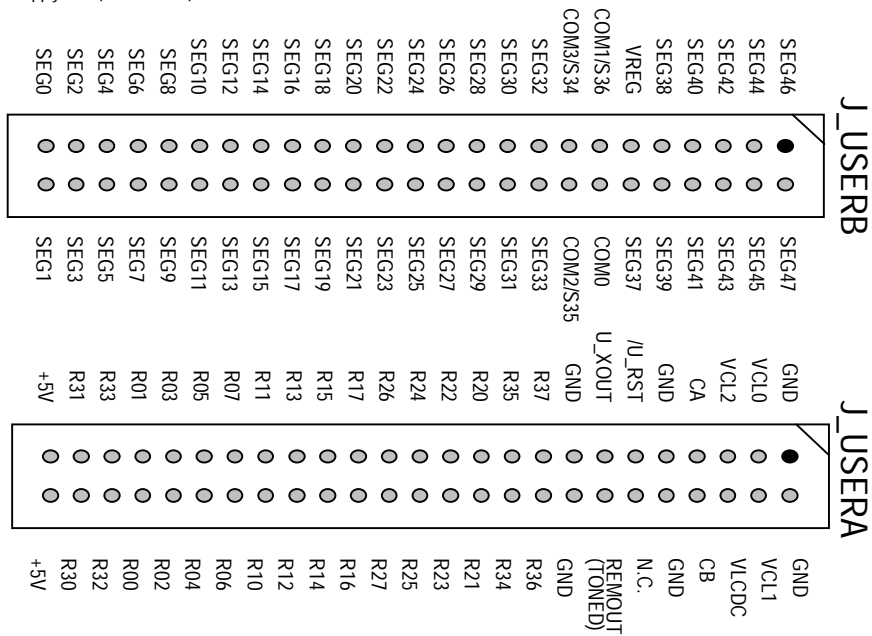


Figure 24-1 The OTP Configuration Byte

24.2 Emulator EVA. Board Setting

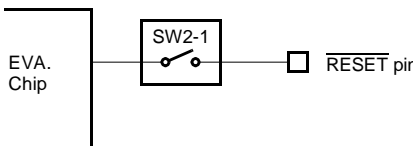
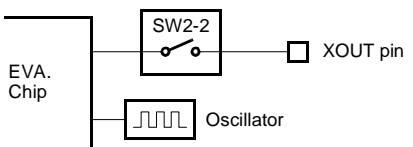
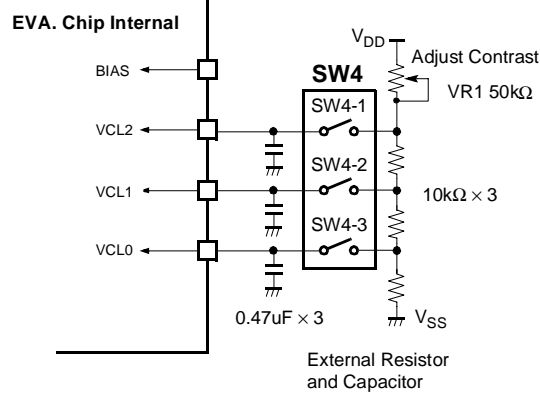
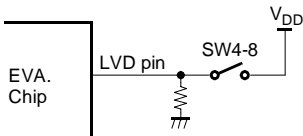


Supply +5V (max. 200mA)



DIP Switch and VR Setting

Before execute the user program, keep in your mind the below configuration

DIP S/W, VR		Description	ON/OFF Setting
SW1	-	Emulator Reset Switch. Reset the Emulator.	Reset the Emulator.
SW2	1	 <p>Pod RESET pin configuration</p>	Normally OFF . EVA. chip can be reset by external user target board. ON : Reset is available by either user target system board or Emulator RESET switch. OFF : Reset the MCU by Emulator RESET switch. Does not work from user target board.
	2	 <p>Pod XOUT pin configuration</p>	Normally OFF . MCU XOUT pin is disconnected internally in the Emulator. Some circumstance user may connect this circuit. ON : Output XOUT signal OFF : Disconnect circuit
SW4	1 2 3	External Bias Resistors Connection  <p>External Resistor and Capacitor</p>	Must be ON position. It serves the external bias resistors. If this switches are turned off, LCD bias voltage does not supplied, floated because there are no internal bias resistors and bias Tr. inside the Emulator.
	4 5 6	LCD Voltage doubling circuit.	Must be OFF position. It is reserved for the GMS81C5108.
	7	Select the Stack Page.	Must be ON position. This switch select the Stack page 0 (off) or page 1 (on). ON : For the 81C7XXX OFF : For the GMS81C5108
	8	 <p>81Cx detect the VDD voltage but Emulator can not do because Emulator can not operate if V_{DD} is below normal opr. voltage (5V), This switch serves LVD environment through the applying 0V to LVD pin of EVA. chip during 5V normal operation.</p>	Position ON during normal operation. ON : Normal operation OFF : Force to detect the LVD, refer to "23. POWER FAIL PROCES-SOR" on page 89.

DIP S/W, VR		Description	ON/OFF Setting
SW5	1	Internal power supply to sub-oscillation circuit.	Must be ON position.
	2	Reserved for other purpose.	Must be OFF position.
VR1	-	<p>Adjust the LCD contrast. It supply bias voltage and adjust the VCL2 voltage.</p> <p style="text-align: center;">External Resistor and Capacitor</p>	Adjust the proper position as well as LCD display good.
VR2	-	Reserved for other purpose.	Don't care.

APPENDIX

A. MASK ORDER SHEET

MASK ORDER & VERIFICATION SHEET

GMS81C7008 -LA
GMS81C7016

Customer should write inside thick line box.

1. Customer Information

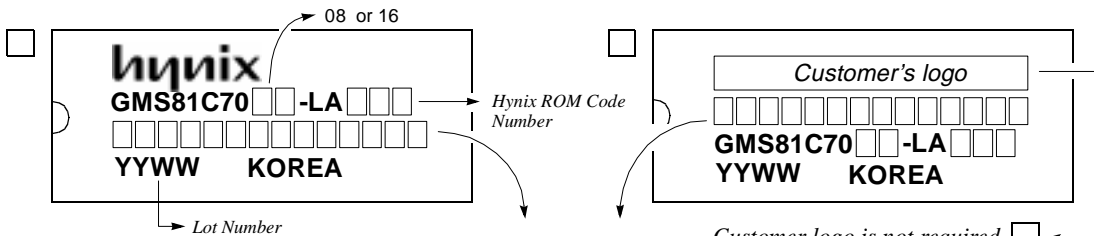
Company Name	
Application	
Order Date	YYYY MM DD • •
Tel:	Fax:
E-mail:	
Name & Signature:	

2. Device Information

Package	<input type="checkbox"/> 64SDIP <input type="checkbox"/> 64MQFP <input type="checkbox"/> 64LQFP
ROM Size	<input type="checkbox"/> 8K <input type="checkbox"/> 16K
RC OSC Opt.	<input type="checkbox"/> Crystal <input type="checkbox"/> RC
Mask Data	File Name: (.OTP) Check Sum: ()
<input type="checkbox"/> Internet	

3. Marking Specification

(Please check mark into)



If the customer logo must be used in the special mark, please submit a clean original of the logo.

Customer's part number

4. Delivery Schedule

	Date	Quantity	Hynix Confirmation
Customer Sample	YYYY MM DD • •	pcs	
Risk Order	YYYY MM DD • •	pcs	

5. ROM Code Verification

This box is written after "5.Verification".

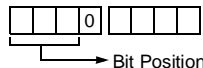
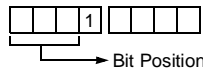
Verification Date:	YYYY MM DD • •
<i>Please confirm our verification data.</i>	
Check Sum:	
Tel:	Fax:
E-mail:	
Name & Signature:	

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



B. INSTRUCTION

B.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 (000H~0FFFH)
rel	Relative Addressing Data
upage	U-page (0FF00H~0FFFFH) Offset Address
n	Table CALL Number (0~15)
+	Addition
x	 <p>Upper Nibble Expression in Opcode</p>
y	 <p>Upper Nibble Expression in Opcode</p>
-	Subtraction
×	Multiplication
/	Division
()	Contents Expression
^	AND
∨	OR
⊕	Exclusive OR
~	NOT
←	Assignment / Transfer / Shift Left
→	Shift Right
↔	Exchange
=	Equal
≠	Not Equal

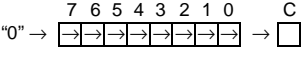
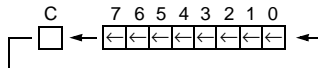
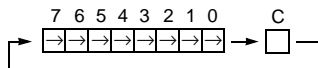
B.2 Instruction Map

LOW HIGH	0000 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	CLR V				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

B.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.	NVGBHIZC
2	ADC dp	05	2	3	$A \leftarrow (A) + (M) + C$	NV--H-ZC
3	ADC dp + X	06	2	4		
4	ADC !abs	07	3	4		
5	ADC !abs + Y	15	3	5		
6	ADC [dp + X]	16	2	6		
7	ADC [dp] + Y	17	2	6		
8	ADC { X }	14	1	3		
9	AND #imm	84	2	2		
10	AND dp	85	2	3		
11	AND dp + X	86	2	4		
12	AND !abs	87	3	4		
13	AND !abs + Y	95	3	5		
14	AND [dp + X]	96	2	6		
15	AND [dp] + Y	97	2	6		
16	AND { X }	94	1	3		
17	ASL A	08	1	2	Arithmetic shift left C 7 6 5 4 3 2 1 0 $\square \leftarrow \left[\leftarrow \leftarrow \leftarrow \leftarrow \leftarrow \leftarrow \leftarrow \leftarrow \right] \leftarrow "0"$	N-----ZC
18	ASL dp	09	2	4		
19	ASL dp + X	19	2	5		
20	ASL !abs	18	3	5		
21	CMP #imm	44	2	2	Compare accumulator contents with memory contents $(A) - (M)$	N-----ZC
22	CMP dp	45	2	3		
23	CMP dp + X	46	2	4		
24	CMP !abs	47	3	4		
25	CMP !abs + Y	55	3	5		
26	CMP [dp + X]	56	2	6		
27	CMP [dp] + Y	57	2	6		
28	CMP { X }	54	1	3		
29	CMPX #imm	5E	2	2	Compare X contents with memory contents $(X) - (M)$	N-----ZC
30	CMPX dp	6C	2	3		
31	CMPX !abs	7C	3	4		
32	CMPY #imm	7E	2	2	Compare Y contents with memory contents $(Y) - (M)$	N-----ZC
33	CMPY dp	8C	2	3		
34	CMPY !abs	9C	3	4		
35	COM dp	2C	2	4	1'S Complement : $(dp) \leftarrow \sim(dp)$	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 $M \leftarrow (M) - 1$	N-----Z-
39	DEC dp	A9	2	4		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 $A \leftarrow (A) \oplus (M)$	N-----Z-
46	EOR dp	A5	2	3		
47	EOR dp + X	A6	2	4		
48	EOR !abs	A7	3	4		
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 $M \leftarrow (M) + 1$	N-----ZC
54	INC dp	89	2	4		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 "0" → 	N-----ZC
60	LSR dp	49	2	4		
61	LSR dp + X	59	2	5		
62	LSR !abs	58	3	5		
63	MUL	5B	1	9	Multiply : $YA \leftarrow Y \times A$	N-----Z-
64	OR #imm	64	2	2	Logical OR $A \leftarrow (A) \vee (M)$	N-----Z-
65	OR dp	65	2	3		
66	OR dp + X	66	2	4		
67	OR !abs	67	3	4		
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 	N-----ZC
73	ROL dp	29	2	4		
74	ROL dp + X	39	2	5		
75	ROL !abs	38	3	5		
76	ROR A	68	1	2	Rotate right through Carry 	N-----ZC
77	ROR dp	69	2	4		
78	ROR dp + X	79	2	5		
79	ROR !abs	78	3	5		
80	SBC #imm	24	2	2	Subtract with Carry $A \leftarrow (A) - (M) - \sim(C)$	NV--HZC
81	SBC dp	25	2	3		
82	SBC dp + X	26	2	4		
83	SBC !abs	27	3	4		
84	SBC !abs + Y	35	3	5		
85	SBC [dp + X]	36	2	6		
86	SBC [dp] + Y	37	2	6		
87	SBC { X }	34	1	3		
88	TST dp	4C	2	3	Test memory contents for negative or zero, (dp) - 00 _H	N-----Z-
89	XCN	CE	1	5	Exchange nibbles within the accumulator $A_7 \sim A_4 \leftrightarrow A_3 \sim A_0$	N-----Z-

Register / Memory Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	LDA #imm	C4	2	2	Load accumulator	N-----Z-
2	LDA dp	C5	2	3	$A \leftarrow (M)$	
3	LDA dp + X	C6	2	4		
4	LDA !abs	C7	3	4		
5	LDA !abs + Y	D5	3	5		
6	LDA [dp + X]	D6	2	6		
7	LDA [dp] + Y	D7	2	6		
8	LDA { X }	D4	1	3		
9	LDA { X }+	DB	1	4	X- register auto-increment : $A \leftarrow (M)$, $X \leftarrow X + 1$	
10	LDM dp,#imm	E4	3	5	Load memory with immediate data : $(M) \leftarrow \text{imm}$	-----
11	LDX #imm	1E	2	2	Load X-register	N-----Z-
12	LDX dp	CC	2	3	$X \leftarrow (M)$	
13	LDX dp + Y	CD	2	4		
14	LDX !abs	DC	3	4		
15	LDY #imm	3E	2	2	Load Y-register	N-----Z-
16	LDY dp	C9	2	3	$Y \leftarrow (M)$	
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	$(M) \leftarrow A$	
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 : $(M) \leftarrow A$, $X \leftarrow X + 1$	
27	STX dp	EC	2	4	Store X-register contents in memory	-----
28	STX dp + Y	ED	2	5	$(M) \leftarrow X$	
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	$(M) \leftarrow Y$	
32	STY !abs	F8	3	5		
33	TAX	E8	1	2	Transfer accumulator contents to X-register : $X \leftarrow A$	N-----Z-
34	TAY	9F	1	2	Transfer accumulator contents to Y-register : $Y \leftarrow A$	N-----Z-
35	TSPX	AE	1	2	Transfer stack-pointer contents to X-register : $X \leftarrow \text{sp}$	N-----Z-
36	TXA	C8	1	2	Transfer X-register contents to accumulator: $A \leftarrow X$	N-----Z-
37	TXSP	8E	1	2	Transfer X-register contents to stack-pointer: $\text{sp} \leftarrow X$	N-----Z-
38	TYA	BF	1	2	Transfer Y-register contents to accumulator: $A \leftarrow Y$	N-----Z-
39	XAX	EE	1	4	Exchange X-register contents with accumulator : $X \leftrightarrow A$	-----
40	XAY	DE	1	4	Exchange Y-register contents with accumulator : $Y \leftrightarrow A$	-----
41	XMA dp	BC	2	5	Exchange memory contents with accumulator	N-----Z-
42	XMA dp+X	AD	2	6	$(M) \leftrightarrow A$	
43	XMA {X}	BB	1	5		
44	XYX	FE	1	4	Exchange X-register contents with Y-register : $X \leftrightarrow Y$	-----

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 $YA \leftarrow (YA) (dp+1) (dp)$	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 $(dp+1)(dp) \leftarrow (dp+1)(dp) - 1$	N-----Z-
4	INCW dp	9D	2	6	Increment memory pair $(dp+1)(dp) \leftarrow (dp+1)(dp) + 1$	N-----Z-
5	LDYA dp	7D	2	5	Load YA $YA \leftarrow (dp+1)(dp)$	N-----Z-
6	STYA dp	DD	2	5	Store YA $(dp+1)(dp) \leftarrow YA$	-----
7	SUBW dp	3D	2	5	16-Bits subtract without carry $YA \leftarrow (YA) - (dp+1)(dp)$	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 : $C \leftarrow (C) \wedge (M.bit)$	-----C
2	AND1B M.bit	8B	3	4	Bit AND C-flag and NOT : $C \leftarrow (C) \wedge \sim(M.bit)$	-----C
3	BIT dp	0C	2	4	Bit test A with memory :	MM----Z-
4	BIT !abs	1C	3	5	$Z \leftarrow (A) \wedge (M), N \leftarrow (M_7), V \leftarrow (M_6)$	
5	CLR1 dp.bit	y1	2	4	Clear bit : $(M.bit) \leftarrow "0"$	-----
6	CLRA1 A.bit	2B	2	2	Clear A bit : $(A.bit) \leftarrow "0"$	-----
7	CLRC	20	1	2	Clear C-flag : $C \leftarrow "0"$	-----0
8	CLRG	40	1	2	Clear G-flag : $G \leftarrow "0"$	--0-----
9	CLRV	80	1	2	Clear V-flag : $V \leftarrow "0"$	-0--0---
10	EOR1 M.bit	AB	3	5	Bit exclusive-OR C-flag : $C \leftarrow (C) \oplus (M.bit)$	-----C
11	EOR1B M.bit	AB	3	5	Bit exclusive-OR C-flag and NOT : $C \leftarrow (C) \oplus \sim(M.bit)$	-----C
12	LDC M.bit	CB	3	4	Load C-flag : $C \leftarrow (M.bit)$	-----C
13	LDCB M.bit	CB	3	4	Load C-flag with NOT : $C \leftarrow \sim(M.bit)$	-----C
14	NOT1 M.bit	4B	3	5	Bit complement : $(M.bit) \leftarrow \sim(M.bit)$	-----
15	OR1 M.bit	6B	3	5	Bit OR C-flag : $C \leftarrow (C) \vee (M.bit)$	-----C
16	OR1B M.bit	6B	3	5	Bit OR C-flag and NOT : $C \leftarrow (C) \vee \sim(M.bit)$	-----C
17	SET1 dp.bit	x1	2	4	Set bit : $(M.bit) \leftarrow "1"$	-----
18	SETA1 A.bit	0B	2	2	Set A bit : $(A.bit) \leftarrow "1"$	-----
19	SETC	A0	1	2	Set C-flag : $C \leftarrow "1"$	-----1
20	SETG	C0	1	2	Set G-flag : $G \leftarrow "1"$	--1-----
21	STC M.bit	EB	3	6	Store C-flag : $(M.bit) \leftarrow C$	-----
22	TCLR1 !abs	5C	3	6	Test and clear bits with A : $A - (M), (M) \leftarrow (M) \wedge \sim(A)$	N-----Z-
23	TSET1 !abs	3C	3	6	Test and set bits with A : $A - (M), (M) \leftarrow (M) \vee (A)$	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 $pc \leftarrow (pc) + rel$	
3	BBS A.bit,rel	x2	2	4/6	Branch if bit set :	-----
4	BBS dp.bit,rel	x3	3	5/7	if (bit) = 1 , then $pc \leftarrow (pc) + rel$	
5	BCC rel	50	2	2/4	Branch if carry bit clear if (C) = 0 , then $pc \leftarrow (pc) + rel$	-----
6	BCS rel	D0	2	2/4	Branch if carry bit set if (C) = 1 , then $pc \leftarrow (pc) + rel$	-----
7	BEQ rel	F0	2	2/4	Branch if equal if (Z) = 1 , then $pc \leftarrow (pc) + rel$	-----
8	BMI rel	90	2	2/4	Branch if minus if (N) = 1 , then $pc \leftarrow (pc) + rel$	-----
9	BNE rel	70	2	2/4	Branch if not equal if (Z) = 0 , then $pc \leftarrow (pc) + rel$	-----
10	BPL rel	10	2	2/4	Branch if plus if (N) = 0 , then $pc \leftarrow (pc) + rel$	-----
11	BRA rel	2F	2	4	Branch always $pc \leftarrow (pc) + rel$	-----
12	BVC rel	30	2	2/4	Branch if overflow bit clear if (V) = 0 , then $pc \leftarrow (pc) + rel$	-----
13	BVS rel	B0	2	2/4	Branch if overflow bit set if (V) = 1 , then $pc \leftarrow (pc) + rel$	-----
14	CALL !abs	3B	3	8	Subroutine call	
15	CALL [dp]	5F	2	8	$M(sp) \leftarrow (pc_H)$, $sp \leftarrow sp - 1$, $M(sp) \leftarrow (pc_L)$, $sp \leftarrow sp - 1$, if !abs, $pc \leftarrow abs$; if [dp], $pc_L \leftarrow (dp)$, $pc_H \leftarrow (dp + 1)$.	-----
16	CBNE dp,rel	FD	3	5/7	Compare and branch if not equal :	-----
17	CBNE dp+X,rel	8D	3	6/8	if (A) \neq (M) , then $pc \leftarrow (pc) + rel$.	
18	DBNE dp,rel	AC	3	5/7	Decrement and branch if not equal :	-----
19	DBNE Y,rel	7B	2	4/6	if (M) \neq 0 , then $pc \leftarrow (pc) + rel$.	
20	JMP !abs	1B	3	3	Unconditional jump	
21	JMP [!abs]	1F	3	5	$pc \leftarrow$ jump address	-----
22	JMP [dp]	3F	2	4		
23	PCALL upage	4F	2	6	U-page call $M(sp) \leftarrow (pc_H)$, $sp \leftarrow sp - 1$, $M(sp) \leftarrow (pc_L)$, $sp \leftarrow sp - 1$, $pc_L \leftarrow (upage)$, $pc_H \leftarrow "0FFH"$.	-----
24	TCALL n	nA	1	8	Table call : $(sp) \leftarrow (pc_H)$, $sp \leftarrow sp - 1$, $M(sp) \leftarrow (pc_L)$, $sp \leftarrow sp - 1$, $pc_L \leftarrow (Table\ vector\ L)$, $pc_H \leftarrow (Table\ vector\ H)$	-----

Control Operation & Etc.

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

C. SOFTWARE EXAMPLE

```

;*****
; Title:          GMS81C7016 (GMS800 Series) Demonstration Program      *
; Company:       Hynix semiconductor Inc.                               *
; Contents:      LCD DISPLAY & DUAL THERMOMETER                         *
;*****
;
;*****  DEFINE    I/O PORT & FUNCTION  REGISTER ADDRESS  *****
;
R0      EQU      0C0H          ;port R0 register
R1      EQU      0C1H          ;port R1 register
R2      EQU      0C2H          ;port R2 register
R3      EQU      0C3H          ;port R3 register
R4      EQU      0C4H          ;port R4 register
R5      EQU      0C5H          ;port R5 register
;
R0DD    EQU      0C8H          ;port R0 data I/O direction register
R1DD    EQU      0C9H          ;port R1 data I/O direction register
R2DD    EQU      0CAH          ;port R2 data I/O direction register
R3DD    EQU      0CBH          ;port R3 data I/O direction register
R4DD    EQU      0CCH          ;port R4 data I/O direction register
R5DD    EQU      0CDH          ;port R5 data I/O direction register
;
R0PU    EQU      0D0H          ;port R0 Pull-up selection register
R1PU    EQU      0D1H          ;port R1 Pull-up selection register
R2PU    EQU      0D2H          ;port R2 Pull-up selection register
R3PU    EQU      0D3H          ;port R3 Pull-up selection register
;
R0CR    EQU      0D4H          ;port R0 Type selection register
R1CR    EQU      0D5H          ;port R1 Type selection register
R2CR    EQU      0D6H          ;port R2 Type selection register
R3CR    EQU      0D7H          ;port R3 Type selection register
;
IEDS    EQU      0D8H          ;External interrupt edge selection register
PMR     EQU      0D9H          ;Alternative port mode register
IENL    EQU      0DAH          ;int. enable register low
IENH    EQU      0DBH          ;int. enable register high
IRQL    EQU      0DCH          ;int. request flag register low
IRQH    EQU      0DDH          ;int. request flag register high

SLPR    EQU      0DEH          ;sleep mode register
WDTR    EQU      0DFH          ;Watchdog timer register

TM0     EQU      0E0H          ;Timer 0 mode register
TDR0    EQU      0E1H          ;Timer 0 data register
TM1     EQU      0E2H          ;Timer 1 mode register
TDR1    EQU      0E3H          ;Timer 1 data register
T1PPR   EQU      0E3H          ;PWM0 period register
T1PDR   EQU      0E4H          ;Timer 1 pulse duty register
PWM0HR  EQU      0E5H          ;PWM0 high register
TM2     EQU      0E6H          ;Timer 2 mode register
TDR2    EQU      0E7H          ;Timer 2 data register
TM3     EQU      0E8H          ;Timer 3 mode register
TDR3    EQU      0E9H          ;Timer 3 data register
T3PPR   EQU      0E9H          ;PWM1 period register
T3PDR   EQU      0EAH          ;Timer 3 pulse duty register
PWM1HR  EQU      0EBH          ;PWM1 high register

ADCM    EQU      0ECH          ;ADC mode register
ADR     EQU      0EDH          ;ADC result data register
WTMR    EQU      0EFH          ;Watch timer mode register

KSMR    EQU      0F0H          ;Key scan mode register
LCDM    EQU      0F1H          ;LCD mode register
LCDPM   EQU      0F2H          ;LCD port mode register
RPR     EQU      0F3H          ;RAM paging register
BITR    EQU      0F4H          ;Basic interval timer data register
CKCTLR  EQU      0F4H          ;Clock control register
SCMR    EQU      0F5H          ;System clock mode register
PFDR    EQU      0FBH          ;Power fail detector

BUR     EQU      0FDH          ;buzzer data register
SMR     EQU      0FEH          ;Serial mode register
SIOD    EQU      0FFH          ;Serial data buffer register
;
;*****  MACRO    DEFINITION  *****
;
R_SAVEMACRO          ;Save Registers to Stacks

```

```

PUSH      A
          PUSH      X
PUSH      Y
ENDM
;
R_RSTRMACRO                ;Restore Register from Stacks
POP        Y
POP        X
POP        A
ENDM
;
;***** CONSTANT DEFINITION *****
;
;
;*****
;
;***** RAM ALLOCATION *****
;*****
TEMP0      DS      1
TEMP1      DS      1
TEMP2      DS      1
FLAG1      DS      1

RPTEN      EQU     1,FLAG1          ;SET RPTEN(REPEAT KEY ENABLE) AFTER 1 SEC.
KEYONF     EQU     2,FLAG1          ;KEYSCAN
ACTKEY     EQU     3,FLAG1          ;AT ONCE, KEY VALID
TOGMO3     EQU     4,FLAG1          ;MODE 3 (PORT TOGGLE)
DUAL_T     EQU     5,FLAG1          ;INSIDE & OUTSIDE TEMP. DUAL DISPLAY
OUTSIDE    EQU     6,FLAG1          ;INSIDE TEMP or OUTSIDE TEMP.

FLAG2      DS      1
F200MS    EQU     0,FLAG2
F20MS     EQU     1,FLAG2
F_1MIN    EQU     2,FLAG2          ;WTIMER
LPM       EQU     3,FLAG2          ;LEFT TIME PM FLAG
RPM       EQU     4,FLAG2          ;RIGHT TIME PM FLAG

STATUS     DS      1
RPTKEY     EQU     7,STATUS
F_CLOCK    EQU     6,STATUS
F_ON       EQU     0,STATUS

DISPSIGN   DS      1
DISPRAM    DS      1          ;TEMP.
DISPRAM1   DS      4          ;LEFT TIME, RIGHT TIME

ONDO       DS      2
LHOUR      DS      1          ;LEFT WATCH COUNT
LMINUTE    DS      1
RHOUR      DS      1
RMINUTE    DS      1          ;RIGHT WATCH COUNT BUF.
TIMESSET   DS      4          ;WATCH SET BUFFER
TSFLAG     DS      1
TSLPM     EQU     0,TSFLAG        ;TIME SET LEFT PM
TSRPM     EQU     1,TSFLAG        ;TIME SET RIGHT PM
BLINKCNT   DS      1          ;BLINK COUNTER 0~250 LOOP
;
NEWKY      DS      1
OLDKY      DS      1
PORTDT     DS      1
KEYNM      DS      1
KEYDT     DS      1
TOTLKY     DS      1
CHATFL     DS      1
ROBUF     DS      1

DGTCNT     DS      1
MODE       DS      1
SUBMODE    DS      1
BSCTIME    DS      1

TEMCNT     DS      1
HZCNT     DS      1

PWMF       DS      1
PERIOD     EQU     0,PWMF
;
;*****
;***** INTERRUPT VECTOR TABLE *****
;*****

```

```

;
        ORG    0FFE0H
        DW    NOT_USED           ; Timer-3
        DW    NOT_USED           ; Timer-2
        DW    WTIMER             ; Watch Timer
        DW    INT_AD             ; A/D CON.
        DW    NOT_USED           ; Serial I/O
        DW    NOT_USED           ; Not used
        DW    NOT_USED           ; Not used
        DW    NOT_USED           ; Int.2
        DW    TIMER1             ; Timer-1
        DW    TIMER0             ; Timer-0
        DW    INT1               ; Int.1
        DW    INT0               ; Int.0
        DW    NOT_USED; Watch Dog Timer
        DW    NOT_USED; BIT
        DW    INT_KEY            ; Key Scan
        DW    RESET              ; Reset
;
;*****
;          MAIN      PROGRAM      *
;*****
;
        ORG    0C000H           ;Program Start Address
;ORG    0E000H           ; 8K ROM VERSION
;
RESET:   LDM    WDTR,#0
        LDM    RPR,#1
        ;
        CLRG
RAMCLR:  LDA    #0
        STA    {X}+           ;RAM Clear(!0000H->!00BFH)
        CMPX   #0C0H           ;M(X) <- A, then X <- X+1
        BNE   RAMCLR           ;X = #0C0H ?
        SETG
RAMCLR1: LDA    #0
        STA    {X}+           ;RAM Clear(!0100H->!011AH)
        CMPX   #1BH           ;M(X) <- A, then X <- X+1
        BNE   RAMCLR1          ;X = #01BH ?
        CLRG
        ;
        LDX    #0FFH           ;Stack Pointer Initial
        TXSP                    ;SP. <- #0FFH
;
;***** USER RAM INITIALIZE *****
;
;   LDM    MODE,#4
;   LDM    SUBMODE,#1
;   SET1   LPM                 ;KST PM 12:00 JUST NOON
;   LDM    LHOURL,#12H
;   LDM    LMINUTE,#00H
;   LDM    RHOURL,#03H         ;UTC AM 03:00
;   LDM    RMINUTE,#00H
;   SET1   OUTSIDE
;   SET1   F_ON                ;POWER ON
;
;***** PORT INITIALIZE *****
;
        LDM    LCDPM,#0        ;SEG0~SEG23 are used
        LDM    R0,#0           ;I/O Port Data Clea
        LDM    R1,#0           ;I/O Port Data Clear
        LDM    R2,#0
        LDM    R3,#0
        LDM    R0DD,#1111_0001B ;R05,R06,R07: output for Keyscan
        LDM    R1DD,#0000_0000B
        LDM    R2DD,#0000_0000B ;R20~R23: input for keyscan
        LDM    R3DD,#0000_0100B
        LDM    R2PU,#0000_1111B ;R20~R23 pull-up active
;
;***** CONTROL REGISTER INITIALIZE *****
;
        LDM    CKCTLR,#0       ;WAKE UP TIME = 0.0625 sec
        ;(1/32768)*8*256 = 0.0625sec
        LDM    TDR0,#249       ;8us x (249+1) = 2ms
        LDM    TM0,#0000_1111B ;8BIT Timer,8us,Start Count-up
        LDM    TDR1,#249       ;2us x (249+1) = 500us
        LDM    TM1,#0000_1111B ;Timer1(8bit),32us,Start Count-up
        LDM    TM3,#1010_1011B

```

```

LDM    T3PPR,#99
LDM    T3PDR,#50
LDM    PWM1HR,#00H
LDM    PMR,#80H

LDM    IRQH,#0           ;Clear All Interrupts Requeat Flags
LDM    IRQL,#0
LDM    IENL,#1111_1111B ;INT2,ADC,WT,T2,T3
LDM    IENH,#1111_1111B ;BIT,WDT,INT0,INT1,T0,T1
LDM    IEDS,#0001_0101B ;External Int. Falling edge select
LDM    KSMR,#0000_0001B ;R10 KEY INTERRUPT
LDM    WTMR,#48H        ;ENABLE WT COUNTER, 2Hz, SELECT SUBCLOCK
LDM    LCDM,#70H        ;CLK=fsub/64, 1/4duty, internal Bias
LDM    SCMR,#0          ;1/2, MAIN OSC.

EI           ;Enable Interrupts
;
LOOP:      BBC    KEYONF,EXE1 ;TEST IF KEY IS PRESSED
          CALL   KEYDECODE
          CLR1   KEYONF      ;CLEAR KEY FLAG
EXE1:
          BBC    F20MS,NEXT1
          CLR1   F20MS
          ;
          ;*****EVERY 20MS*****
          ;
          CALL   MODEEXE     ;SETTING DISPLAY MEMORY
          CALL   MODE1EXE    ;DURING CLOCK,
          CALL   MODE3EXE
          CALL   LCDDGT      ;7-Segments Display
          CALL   LCDDOT      ;Dot Display
          CALL   ADCEXE      ;ADC execution
          CALL   LKEYSCAN

NEXT1:
          BBC    F200MS,ELOOP
          CLR1   F200MS
          ;
          ;*****EVERY 200MS*****
          ;
          CALL   WIND

ELOOP:
          BBS    F_ON,EXE2
          CLR1   R0.7        ;FOR WAKE-UP BY NEXT KEY
          CLR1   R0.6        ;FOR WAKE-UP BY NEXT KEY
          CLR1   R0.5        ;FOR WAKE-UP BY NEXT KEY
          CLR1   R0.4        ;FOR WAKE-UP BY NEXT KEY
          STOP
          NOP
          NOP
          IF     [F_1MIN]
            CLR1   F_1MIN
            CALL   MODEEXE
            CALL   LCDDGT      ;7-Segments Display
            CALL   LCDDOT      ;Dot Display
          ENDIF
          CALL   LKEYSCAN
EXE2:
          JMP    LOOP
;
;*****
;          TIMER0_INTERRUPT ROUTINE(2ms)
;*****
;
TIMER0:   R_SAVE           ;Save Registers to Stacks
          CLRG
          CALL   MAKE10MS    ;SET every 10ms
          R_RSTR           ;Restore Registers from Stacks
          RETI
;
;*****
;          TIMER1
;*****
;
TIMER1:   R_SAVE
          CLRG

          R_RSTR

```

```

        RETI
;
;*****
;          WATCH TIMER 4Hz
;*****
;
WTIMER:  R_SAVE
         CLRG
         NOT1  R0.0

         INC   HZCNT
         LDA   HZCNT
         CMP   #120
         BNE   WT5
         LDM   HZCNT,#0
         SET1  F_1MIN
         CALL  INC1MIN

WT5:     R_RSTR
         RETI
;
;*****
;          PORT INTERRUPT
;*****
;
INT_KEY: R_SAVE
         CLRG
         BBS   CHATFL.7,IK8
         BBS   F_ON,IK8
         LDX   #3
         LDM   KSMR,#0           ;MAKE R10 TO BE NORMAL INPUT

WW:      LDY   #2                 ;24ms wait
WW2:     LDA   #8
WW3:     DEC   A
         BNE   WW3
         DEC   Y
         BNE   WW2

         LDA   R1                 ;READ R10
         ROR   A
         BCS   IK8
         DEC   X
         BNE   WW

         LDM   SCMR,#0           ;MAIN OSC.
         SET1  F_ON
         SET1  CHATFL.7
         LDM   OLDKY,#0CH
IK8:     LDM   KSMR,#1
         R_RSTR
         RETI
;
;*****
;          EXTERNAL INTERRUPT 0
;*****
;
INT0:    R_SAVE
         CLRG
         R_RSTR
         RETI
;
;*****
;          EXTERNAL INTERRUPT 1
;*****
;
INT1:    CLRG
         RETI
;
;*****
;          ADC INTERRUPT
;*****
;
INT_AD:  RETI
;
;*****

```

```

; Subject:   LCDDGT
;           LCD 7-SEG. DIGIT DISPLAY (TMPE,LTIME,RTIME)
;*****
; Entry:    DGCNT (DIGIT COUNTER)
;           X (START ADDRESS)
; Output:   Output SEG_PORT (SEG0~SEG23)
;           Output COM_PORT (COM0~COM3)
;*****
; EXAMPLE)
; DGCNT=9
; X=LMINUTE
;           [---] [---] [---] [---]
;           [---] [---] [---] [---]
;           LMINUTE+1 LMINUTE
;*****
;
LCDDGT: LDM   DGCNT,#9
        LDX   #DISPRAM
GOLCD:  LDA   {X}
        PUSH  X
        if [DGCNT.0]
            AND   #0F0H
            XCNT
            CALL  LCDDSP
            POP   X
            ;WHEN DIGIT IS EVEN NUMBER,
            ;WHEN DIGIT IS ODD NUMBER,
            ;HIGHER 4 NIBBLE IS DISPLAYED
        else
            AND   #0FH
            CALL  LCDDSP
            POP   X
            ;LOWER 4 NIBBLE IS DISPLAYED
        endif
        DEC   DGCNT
        BPL   GOLCD
        RET

;
;***** ONE DIGIT DISPLAY *****
;
LCDDSP: TAY
        ;
        ;***** ZERO SURPRESS TO BLANK *****
        ;
        BNE   GOCONT
        LDA   DGCNT
        CMP   #9
        BEQ   BLNK
        CMP   #7
        BEQ   BLNK
        CMP   #3
        BEQ   BLNK
        BRA   GOCONT
BLNK:   LDY   #0AH
        ;
GOCONT: LDA   !FONT+Y
        STA   TEMP0
        LDM   TEMP2,#7
        LDY   DGCNT
        LDA   #14
        MUL
        TAY
DPL1:  LDA   !FONTD0+Y
        TAX
        INC   Y
        LDA   !FONTD0+Y
        STA   TEMP1
        ROR   TEMP0
        BCS   DPL3
        LDA   #0FFH
        ROL   A
        DEC   TEMP1
        BPL   $-3
        SETG
        AND   {X}
        BRA   DPL5
DPL3:  LDA   #00H
        ROL   A
        DEC   TEMP1
        BPL   $-3
        SETG
        OR   {X}
DPL5:  STA   {X}

```

```

        CLRG
        INC     Y
        DBNE   TEMP2,DPL1
        RET

FONTD0  DB    13H,1H,13H,2H,13H,0H,13H,3H,0CH,3H,0CH,2H,0CH,0H    ;RMINUTE0
FONTD1  DB    12H,1H,12H,2H,12H,0H,12H,3H,05H,3H,05H,2H,05H,0H    ;RMINUTE1
FONTD2  DB    06H,1H,06H,2H,06H,0H,06H,3H,01H,3H,01H,2H,01H,0H    ;RHOUR0
FONTD3  DB    80H,0H,01H,1H,01H,1H,80H,0H,80H,0H,80H,0H,80H,0H    ;RHOUR1
FONTD4  DB    02H,1H,02H,2H,02H,0H,02H,3H,15H,3H,15H,2H,15H,0H    ;LMINUTE0
FONTD5  DB    09H,1H,15H,1H,09H,0H,09H,3H,16H,0H,16H,1H,09H,2H    ;LMINUTE1
FONTD6  DB    14H,1H,14H,2H,14H,0H,14H,3H,00H,3H,00H,2H,00H,0H    ;LHOUR0
FONTD7  DB    80H,0H,08H,2H,08H,2H,80H,0H,80H,0H,80H,0H,80H,0H    ;LHOUR1
FONTD8  DB    0BH,2H,0BH,0H,0BH,3H,0BH,1H,17H,1H,17H,0H,17H,3H    ;ONDO0
FONTD9  DB    0FH,2H,0FH,0H,0FH,3H,0FH,1H,10H,1H,10H,0H,10H,3H    ;ONDO1
;
;*****
;          7-SEGMENT PATTERN DATA
;
;          a
;          f | g | b
;          ---
;          e | d | c
;          .h
;*****
;
;          Segment:      hgfe dcba          To be displayed Digit Number
FONT      DB    0011_1111B          ; 0  "0"
          DB    0000_0110B          ; 1
          DB    0101_1011B          ; 2
          DB    0100_1111B          ; 3
          DB    0110_0110B          ; 4
          DB    0110_1101B          ; 5
          DB    0111_1101B          ; 6
          DB    0000_0111B          ; 7
          DB    0111_1111B          ; 8  "8"
          DB    0110_1111B          ; 9  "9"
          DB    0000_0000B          ; A  "BLANK"
          DB    0100_0000B          ; B  "BAR"

_LCOLON  EQU    2,116H
_RCOLON  EQU    2,10EH
_ONDO    EQU    2,107H
_C       EQU    0,111H
_RAM     EQU    1,10EH
_RPM     EQU    0,10EH
_LAM     EQU    1,108H
_LPM     EQU    3,108H
_OUTSIDE EQU    1,104H
_INSIDE  EQU    0,107H
_S1      EQU    2,10AH
_SNOW    EQU    3,10AH
_SAVE    EQU    3,104H
;
LCDDOT:  SETC
          STC   _LCOLON
          STC   _S1
          STC   _ONDO
          STC   _C

          LDCB  F_ON
          STC   _SAVE
          LDCB  DUAL_T
          STC   _RCOLON

          LDC   LPM
          STC   _LPM
          LDCB  LPM
          STC   _LAM

          IF    [DUAL_T]==0
              ldc   RPM          ;AM,PM SETTING
              stc   _RPM
              ldcb  RPM
              stc   _RAM
          ELSE
              LDCB  DUAL_T      ;TURN OFF THE AM, PM
              STC   _RPM
    
```

```

        STC      _RAM
    ENDIF

    LDC      OUTSIDE
    STC      _OUTSIDE
    LDCB     OUTSIDE
    STC      _INSIDE

    RET

;
;*****
; Subject:   ANY EXECUTION *
;*****
; DESCRIPTION: EVERY 20MS *
; *
;*****
;
MODEEXE:  IF      [OUTSIDE]
          LDX     #0
          ELSE
          LDX     #1
          ENDIF

          LDA     ONDO+X          ;COPY ONDO DATA TO DISPRAM
          STA     DISPRAM
          LDA     SIGN+X
          STA     DISPSIGN

          IF      [DISPSIGN.0]   ;IF MINUS ONDO, THEN "-" DISPLAY
          IF      [DISPRAM] < #10
          LDA     #0B0H
          OR      DISPRAM
          STA     DISPRAM
          CLRC
          STC     _SNOW
          ELSE
          SETC
          STC     _SNOW
          ENDIF
          ELSE
          CLRC
          STC     _SNOW
          ENDIF

          LDX     #3              ;MOVE TIME_BUF. TO DISP_BUF.
          LDA     LHOURL+X
          STA     DISPRAM1+X
          DEC     X
          BPL     MX1

          BBC     DUAL_T,MX2      ;IF SINGLE TEMP. MODE, SKIP
          LDA     #0AAH          ;MAKE ERASE DISP BUF. WITCH
          STA     DISPRAM1+2      ;WILL BE DISPLAYED TEMP.

          IF      [OUTSIDE]      ;IF DUAL TEMP. MODE
          LDX     #1              ;IF MAIN=OUSIDE, THEN SELECT INSIDE
          ELSE
          LDX     #0              ;IF MAIN=INSIDE, THEN SELECT OUTSIDE
          ENDIF

          LDA     ONDO+X
          STA     DISPRAM1+3

          LDA     SIGN+X          ;GET BIT0 OF SIGN
          ROR     A                ;COPY SIGN TO CARRY

          IF      C                ;IF MINUS ONDO, THEN "-" DISPLAY
          IF      [DISPRAM1+3] < #10
          LDA     #0B0H          ;EXE) BB-4
          OR      DISPRAM1+3
          STA     DISPRAM1+3
          ELSE
          LDM     DISPRAM1+2,#0ABH ;EXE) B-14
          ENDIF
          ELSE
          IF      [DISPRAM1+3] < #10
          LDA     #0A0H          ;EXE) BB-4
          OR      DISPRAM1+3
          STA     DISPRAM1+3
          ENDIF

```

```

ENDIF

MX2:      RET

;
;*****
; Subject:  MODE 1 EXECUTION *
;*****
; DESCRIPTION: CLOCK SET *
;
;*****
;
MODE1EXE: LDA    MODE
          AND    #0F0H
          CMP    #10H          ;IF MODE=1x
          BNE    MB3
          LDX    #3
MB1:      LDA    TIMESET+X      ;TIMESET BUF. COPIED TO DISP BUF.
          STA    DISPRAM1+X    ;4BYTE & 2 BIT
          DEC    X
          BPL    MB1
          LDC    TSLPM
          STC    LPM
          LDC    TSRPM
          STC    RPM
          ;
          LDA    MODE
          CMP    #10H          ;TEST IF LEFT TIME SET MODE ?
          BEQ    MO10
          CMP    #11H
          BEQ    MO11          ;TEST IF RIGHT TIME SET MODE ?
          BRA    MB3

MO10:     LDA    BLINKCNT
          CMP    #125          ;IF LESS THAN 124, OFF
          BCS    MB3
          LDA    #0AAH
          STA    DISPRAM1
          STA    DISPRAM1+1
MB3:      RET

MO11:     LDA    BLINKCNT
          CMP    #125          ;IF LESS THAN 124, OFF
          BCS    MB3
          LDA    #0AAH
          STA    DISPRAM1+2
          STA    DISPRAM1+3
          BRA    MB3

;
;*****
; Subject:  MODE 3 EXECUTION *
;*****
; DESCRIPTION: All pin goes low and high *
; repeatedly every 20ms, rectangle wave output *
;
;*****
;
MODE3EXE: LDA    MODE
          CMP    #3
          BNE    MO2
          LDA    SUBMODE
          DEC    A          ;BECAUSE INITIAL NO.=1
          ROL    A          ;EIGHT TIMES
          ROL    A
          ROL    A
          NOTL   TOGMO3
          BBC    TOGMO3,MO1
          CLRC
          ADC    #4          ;ADD OFFSET
MO1:      TAY
          LDA    !PPORT+Y
          AND    #0001_1111B
          OR     R0BUF
          STA    R0BUF
          STA    R0
          LDA    !PPORT+1+Y
          STA    R1
          LDA    !PPORT+2+Y
          STA    R2

```

```

        LDA    !PPORT+3+Y
        STA    R3
MO2:    RET

PPORT   DB    00H,00H,00H,00H
        DB    00H,00H,00H,00H

        DB    0FFH,0FFH,0FFH,0FFH
        DB    0FFH,0FFH,0FFH,0FFH

        DB    00H,00H,00H,00H
        DB    0FFH,0FFH,0FFH,0FFH

        DB    00H,00H,00H,00H
        DB    0FFH,00H,0FFH,00H

        DB    00H,0FFH,00H,0FFH
        DB    00H,00H,00H,00H

        DB    00H,0FFH,00H,0FFH
        DB    0FFH,00H,0FFH,00H

        DB    55H,55H,55H,55H
        DB    0AAH,0AAH,0AAH,0AAH
;
;*****
; Subject:   Set falg at every 20ms      *
;*****
;
MAKE10MS: SETC
          LDA    #0
          ADC    BSCTIME
          DAA
          STA    BSCTIME
          BNE    $+4
          SET1   F200MS           ;SET F200MS EVERY 200ms
          AND    #0FH
          BNE    $+4
          SET1   F20MS           ;SET F20MS EVERY 20ms
          ;
          INC    BLINKCNT        ;USED IN MODE0(CLOCK SET)
          LDA    BLINKCNT
          CMP    #250
          BNE    MZ1
          LDM    BLINKCNT,#0
MZ1:    RET
;
;*****
; Subject:   Analog to Digital Conversion *
;*****
; It is called in main routine every 20ms
ADCNT   DS    2
ADR_AVR DS    2
ADTTL   DS    4
ADFLAG  DS    1
AD_CH   EQU   0,ADFLAG
SIGN    DS    2
DIVISOR EQU   250
;
;          :-----: :-----:
;          :ADR_AVR: :ADR_AVR:
;          :      : :      :
;          :OUTSIDE: :INSIDE :
;          :CH4   : :CH5   :
;          :-----: :-----:
;
ADCEXE: IF    [AD_CH]== 0
          LDM    ADCM,#52H           ;AD START CH4
          LDX    #0                  ;SET TO 0 INDEX POINTER
        ELSE
          LDM    ADCM,#56H           ;AD START CH5
          LDX    #1                  ;SET TO 1 INDEX POINTER
        ENDIF

ADWAIT: LDY    #20                   ;WAIT ADC END
        DEC    Y
        BBS    ADCM.0,GOGET
        CMPLY  #0
        BNE    ADWAIT

```

```

GOGET:    CLRC                                ;UP8    LO8
          LDA    ADR                          ;ADTTTL2|ADTTTL0 = CH4 DATA
          ADC    ADTTL+X                      ;ADTTTL3|ADTTTL1 = CH5 DATA
          STA    ADTTL+X
          LDA    #0
          ADC    ADTTL+2+X
          STA    ADTTL+2+X
          ;
          INC    ADCNT+X
          LDA    ADCNT+X
          IF    A == #DIVISOR                 ;GET AVERAGE VALUE
            LDA    #0
            STA    ADCNT+X
            LDY    ADTTL+2+X
            LDA    ADTTL+X
            PUSH  X
            LDX    #DIVISOR                   ;DIVIDE BY DIVISOR
            DIV
            POP   X
            STA    ADR_AVR+X
            LDA    #0                           ;CLEAR SUM BUF.
            STA    ADTTL+X
            STA    ADTTL+2+X

            LDA    ADR_AVR+X
            IF    A < #65                       ;IGNORE BELOW 65
              LDA    #65
            ENDIF
            IF    A > #240                       ;MAX. 240
              LDA    #240
            ENDIF
            CMP    #181                           ;MAKE SIGN
            ROL    SIGN+X                         ;COPY TO MINUS OR PLUS

            SETC
            SBC    #65
            TAY
            LDA    !ADTABLE1+Y
            STA    ONDO+X
          ENDIF

          NOT1  AD_CH
ADCQUIT:  RET
;
;
ADTABLE  DB          50H,49H,49H,48H,48H,47H ; 65~ 70 65->+50'C
          DB    47H,46H,46H,45H,45H,44H,44H,43H,43H,42H ; 71~ 80
          DB    41H,41H,40H,40H,40H,39H,39H,38H,38H,37H ; 81~ 90 83->+40'C
          DB    37H,36H,36H,35H,35H,34H,34H,33H,33H,32H ; 91~100
          DB    32H,31H,31H,30H,30H,30H,29H,29H,28H,28H ;101~110 105->+30'C
          DB    27H,27H,26H,26H,25H,25H,24H,24H,24H,23H ;111~120
          DB    23H,22H,22H,22H,21H,21H,20H,20H,20H,20H ;121~130 129->+20'C
          DB    19H,19H,18H,18H,17H,17H,16H,16H,15H,15H ;131~140
          DB    15H,14H,14H,14H,13H,13H,13H,12H,12H,12H ;141~150
          DB    11H,11H,11H,10H,10H,10H,09H,09H,09H,08H ;151~160 154->+10'C
          DB    08H,07H,07H,07H,06H,05H,05H,04H,04H,04H ;161~170
          DB    03H,03H,02H,02H,01H,01H,00H,00H,00H,01H ;171~180 178-> 0'C
          DB    01H,02H,02H,03H,03H,04H,04H,05H,05H,06H ;181~190
          DB    06H,07H,07H,08H,08H,09H,09H,10H,10H,11H ;191~200 199->-10'C
          DB    11H,12H,12H,13H,13H,14H,15H,15H,16H,17H ;201~210
          DB    17H,18H,18H,19H,19H,20H,20H,21H,21H,22H ;211~220 217->-20'C
          DB    23H,23H,24H,24H,25H,25H,26H,27H,28H,29H ;221~230
          DB    30H,31H,32H,33H,34H,35H,36H,37H,38H,39H ;231~240 231->-30'C
          DB    40H,41H,42H

ADTABLE1 DB          50H,50H,50H,49H,49H,48H ; 65~ 70 65->+50'C
          DB    48H,47H,47H,46H,46H,45H,45H,44H,44H,43H ; 71~ 80
          DB    43H,42H,41H,40H,39H,38H,37H,36H,35H,34H ; 81~ 90 83->+40'C
          DB    35H,35H,34H,34H,33H,33H,32H,32H,31H,31H ; 91~100
          DB    30H,30H,29H,29H,28H,28H,27H,27H,26H,26H ;101~110 105->+30'C
          DB    26H,25H,25H,25H,24H,24H,24H,23H,23H,23H ;111~120
          DB    22H,22H,22H,21H,21H,21H,20H,20H,20H,20H ;121~130 129->+20'C
          DB    19H,18H,18H,18H,17H,17H,17H,17H,16H,16H,16H ;131~140
          DB    15H,15H,15H,14H,14H,14H,13H,13H,13H,12H ;141~150
          DB    12H,11H,11H,10H,10H,09H,09H,08H,08H,08H ;151~160 154->+10'C
          DB    07H,07H,06H,06H,05H,05H,04H,04H,04H,03H ;161~170
          DB    03H,03H,02H,02H,02H,01H,01H,01H,00H,00H ;171~180 178-> 0'C
          DB    01H,01H,02H,02H,03H,03H,04H,04H,05H,05H ;181~190
          DB    06H,06H,07H,07H,08H,08H,09H,09H,10H,10H ;191~200 199->-10'C
          DB    11H,11H,12H,12H,13H,13H,14H,15H,15H,16H ;201~210
          DB    16H,16H,17H,18H,18H,19H,19H,20H,20H,21H ;211~220 217->-20'C

```

```

DB      21H, 22H, 23H, 23H, 24H, 24H, 25H, 25H, 26H, 27H ;221~230
DB      28H, 29H, 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H ;231->-30'C
DB      38H, 39H, 40H
;
;*****
; Subject:   KEYDECODE
;*****
;
;*****
;
REPEAT EQU #1000_0000B
CLOCK  EQU #0100_0000B
PWRON  EQU #0000_0001B

KEYDECODE: LDA  KEYDT
           LDY  #3
           MUL
           TAY
           LDA  !KEY+Y
           STA  TEMP0
           LDA  !KEY+1+Y
           STA  TEMP1
           LDA  !KEY+2+Y
           STA  TEMP2
           CALL CONDICHK
           BCC  QUIT
           JMP  [TEMP0]
;
KEY:      DW   NOKEY           ;0
           DB   0
           DW   NOKEY           ;1
           DB   0
           DW   NOKEY           ;2
           DB   0
           DW   NOKEY           ;3
           DB   0
           DW   NOKEY           ;4
           DB   0
           DW   NOKEY           ;5
           DB   0
           DW   NOKEY           ;6
           DB   0
           DW   DOWNKEY         ;7
           DB   PWRON+REPEAT
           DW   NOKEY           ;8
           DB   0
           DW   DUALKEY         ;9
           DB   PWRON
           DW   SWAPKEY         ;A
           DB   PWRON
           DW   NOKEY           ;B
           DB   0
           DW   POWERKEY        ;C
           DB   PWRON
           DW   CLOCKKEY        ;D
           DB   PWRON+CLOCK
           DW   HOURKEY         ;E
           DB   PWRON+REPEAT+CLOCK
           DW   MINUTEKEY        ;F
           DB   PWRON+REPEAT+CLOCK
           DW   NOKEY           ;10
           DB   0
           DW   UPKEY           ;11
           DB   PWRON+REPEAT
           DW   NOKEY           ;12
           DB   0

QUIT:
NOKEY:  RET

CONDICHK: LDA  TEMP2
          OR   STATUS
          SBC  TEMP2
          BEQ  CDC9
          BCS  CDC10

CDC9:   SETC           ;PASS
          RET

CDC10:  CLRC           ;SKIP
          RET
;

```

```

;*****
;      DISPLAY SWAP KEY (TEMP. DISPLAY SWAP)      *
;*****
;
SWAPKEY:  NOT1   OUTSIDE
          RET
;
;*****
;      DUAL KEY                                  *
;*****
;
DUALKEY:  NOT1   DUAL_T
          RET
;
;*****
;      POWER KEY                                  *
;*****
;
POWERKEY: CLR1   F_ON
          IF     [F_ON]
          ELSE
            LDM   SCMR,#2
            CLR1  DUAL_T
            LDM   MODE,#0
            SET1  F20MS
          ENDIF
          RET
;
;*****
;      CLOCK KEY                                  *
;*****
;
CLOCKKEY: SET1   F_CLOCK
          LDM   BLINKCNT,#0
          LDA   MODE                ; 10->11
          CMP   #10H                 ; 11->00
          BNE   CL1                  ; ETC. -> 10
          LDM   MODE,#11H
          BRA   QUIT
CL1:      CMP   #11H
          BNE   CL2
          LDM   MODE,#0
          CLR1  F_CLOCK
          CALL  SETTO_CNT
          LDC   TSLPM
          STC   LPM
          LDC   TSRPM
          STC   RPM
          LDM   HZCNT,#0
          CLR1  F_1MIN
          BRA   CLQ
CL2:      LDM   MODE,#10H
          CLR1  DUAL_T
          CALL  CNTTO_SET
          LDC   LPM
          STC   TSLPM
          LDC   RPM
          STC   TSRPM
CLQ:      RET
;
SETTO_CNT: LDX   #3
CL11:     LDA   TIMESET+X
          STA   LHOURL+X
          DEC   X
          BPL  CL11
          RET
;
CNTTO_SET: LDX   #3
CL3:      LDA   LHOURL+X
          STA   TIMESET+X
          DEC   X
          BPL  CL3
          RET
;
;*****
;      HOUR/MINUTE KEY                          *
;*****
;
HOURKEY:  LDA   MODE

```

```

        AND    #0F0H
        CMP    #10H
        BNE    HO1
        LDM    BLINKCNT, #125
        LDA    MODE
        CMP    #10H
        BNE    HO2
        SETC
        LDA    #0                ;IF MODE=10H, THEN LEFT TIME SET
        ADC    TIMESET          ;INC. LEFT HOUR 1UP
        DAA
        IF     A==#12H
            NOT1 TSLPM          ;ADJUST AM,PM FLAG
        ENDIF
        IF     A==#13H
            LDA    #1
        ENDIF
HO1:    STA    TIMESET
        RET
HO2:    CMP    #11H
        BNE    HO1
        SETC                ;INC. RIGHT HOUR 1UP
        LDA    #0
        ADC    TIMESET+2
        DAA
        IF     A==#12H
            NOT1 TSRPM          ;ADJUST AM,PM FLAG
        ENDIF
        IF     A==#13H
            LDA    #1
        ENDIF
        STA    TIMESET+2
        BRA    HO1

MINUTEKEY: LDA    MODE
          AND    #0F0H
          CMP    #10H
          BNE    MT3

          LDM    BLINKCNT, #125
          LDX    #3
          LDA    MODE
          CMP    #10H
          BNE    MT1
          LDX    #1
MT1:    SETC
        LDA    #0
        ADC    TIMESET+X
        DAA
        CMP    #60H
        BNE    MT2
        LDA    #0
MT2:    STA    TIMESET+X
MT3:    RET
;
;*****
;          UP /DOWN KEY          *
;*****
;
UPKEY:  BBS    PERIOD, PRU
        LDA    PWM1HR
        AND    #0000_0011B
        CMP    #3
        BNE    UPK1
        LDA    T3PDR
        CMP    #0FFH
        BNE    UPK1
UPK0:   RET

UPK1:   INC    T3PDR
        BNE    UPK0
        INC    PWM1HR
        BRA    UPK0

PRU:

DOWNKEY: BBS    PERIOD, PRD
        LDA    PWM1HR
        AND    #0000_0011B
        CMP    #0

```

```

        BNE    DNK1
        LDA    T3PDR
        CMP    #0
        BEQ    UPK0
DNK1:   DEC    T3PDR
        LDA    T3PDR
        CMP    #0FFH
        BNE    DNK2
        DEC    PWM1HR
DNK2:   RET

PRD:

PWMMODE:
;
;*****
;          PLUS KEY                               *
;
; When MODE=3, PRESS PULS KEY, SUBMODE IS INCREASED *
; When MODE=3, PRESS MINUS KEY, SUBMODE IS DECREASED *
;
;*****
;
;*****
; Subject:   KEYSKAN                               *
;*****
; STROBE OUT: R05,R06,R07                          *
; READ PORT : R20,R21,R22,R23                      *
;
;*****
;
LKEYSCAN: BBS    KEYONF,KS7
          LDM    KEYNM,#1
          LDM    TOTLKY,#0
          LDM    NEWKY,#0
          LDY    #3                                ;INITIALIZE STROBE LINE

KS1:     CMPY   #3
          BNE   $+4
          CLR1  R0.4                                ;OUTPUT STROBE SIGNAL
          CMPY  #2
          BNE   $+4
          CLR1  R0.5                                ;OUTPUT STROBE SIGNAL
          CMPY  #1
          BNE   $+4
          CLR1  R0.6                                ;OUTPUT STROBE SIGNAL
          CMPY  #0
          BNE   $+4
          CLR1  R0.7                                ;OUTPUT STROBE SIGNAL
          ;
          NOP
          NOP
          LDA   R2
          STA   PORTDT                               ;READ KEY IN PORT
          AND   #0FFH
          CMP   #0FFH                               ;IF KEY IS PRESSED ?
          BNE   KS2
          CLRC  ;KEYNM + 4 -> KEYNM
          LDA   #4
          ADC   KEYNM
          STA   KEYNM
          BRA   KS5
          ;
KS2:     LDX   #3                                ;INITIALIZE SHIFT COUNTER
KS3:     ROR   PORTDT
          BCS   KS4
          INC   TOTLKY                               ;IF TOTLKY IS ABOVE 2, THEN QUIT
          LDA   TOTLKY
          CMP   #20
          BEQ   KS7
          LDA   KEYNM                               ;KEYNM -> NEWKY
          STA   NEWKY
KS4:     INC   KEYNM
          DEC   X
          BPL  KS3
KS5:     SET1  R0.4
          SET1  R0.5

```

```

        SET1   R0.6
        SET1   R0.7
        DEC    Y                ;TEST NEXT LINE
        BPL    KS1
        LDA    NEWKY
        CMP    #0                ;WHEN NO KEY IS PRESSED,
        BNE    KS8                ;INITIALIZE NEWKY,OLDKY,CHATFL
KS6:    LDA    NEWKY
        STA    OLDKY
        LDM    CHATFL,#0
        CLR1   RPTKEY
        CLR1   ACTKEY
        CLR1   RPTEN
KS7:    RET
KS8:    LDA    NEWKY
        CMP    OLDKY
        BNE    KS6
        BBS    CHATFL.7,KS10
        LDA    CHATFL
        AND    #0111_1111B
        CMP    #5
        BCC    KS9
        LDA    NEWKY
        STA    KEYDT
        SET1   ACTKEY
KS81:   LDM    CHATFL,#80H        ;SET1 CHATFL.7 & SET TO 0
        SET1   KEYONF
        BRA    KS7
KS9:    INC    CHATFL
        BRA    KS7
KS10:   LDA    CHATFL            ;REPEAT KEY
        AND    #0111_1111B
        BBS    RPTEN,KS11
        CMP    #25
        BCC    KS9
        SET1   RPTEN
        BRA    KS81
KS11:   CMP    #3
        BCC    KS9
        BBC    ACTKEY,KS7
        SET1   RPTKEY
        BRA    KS81
;
;*****
; Subject:   Increase 1 minute      *
;*****
;
INCLMIN: LDX    #LMINUTE
        CALL   MIN1UP
        LDX    #RMINUTE
        CALL   MIN1UP
        RET
;
MIN1UP: SETC
        LDA    #0                ; LMINUTE <- LMINUTE + 1
        ADC    {X}
        DAA
        IF    A ==#60H
            SETC
            LDA    #0
        ENDIF
        STA    {X}
        BCC    INCL
        DEC    X
        LDA    #0
        ADC    {X}
        DAA
        IF    A==#12H
            IF    X==#LHOURL
                NOT1   LPM
            ELSE
                NOT1   RPM
            ENDIF
        ENDIF
        IF    A==#13H
            LDA    #1
        ENDIF
        STA    {X}
INCL:   RET

```

```

;
;*****
; Subject:   WIND DISPLAY *
;*****
;
WIND:      LDA      TEMPCNT
           CLRC
           STC      10DH.0
           STC      10DH.1
           STC      10DH.2
           STC      10DH.3
           CMP      #0
           BEQ      LLL3
           CMP      #1
           BEQ      LLL2
           CMP      #2
           BEQ      LLL1
           CMP      #3
           BEQ      LLL0
           CMP      #4
           BEQ      LLL1
           CMP      #5
           BEQ      LLL2
           CMP      #6
           BEQ      LLL3
           CMP      #7
           BEQ      LLL4
LLL0:      STC      10DH.1
LLL1:      STC      10DH.2
LLL2:      STC      10DH.3
LLL3:      STC      10DH.0
LLL4:      STC      111H.1
           INC      TEMPCNT
           IF      [TEMPCNT]==#8
               LDM      TEMPCNT,#0
           ENDIF
           RET
;
;*****
;
NOT_USED:  nop                ;Discard Unexpected Interrupts
           reti
;
           END                ;Notice Program End

```