MAR. 2000 Ver 1.00

HYUNDAI MICRO ELECTRONICS
8-BIT SINGLE-CHIP MICROCONTROLLERS

GMS81C2012
GMS81C2020
User's Manual

HYUNDAI
MicroElectronics
Semiconductor Group of Hyundai Electronics Industrial Co., Ltd.

HYUNDAI MICRO ELECTRONICS
8-BIT SINGLE-CHIP MICROCONTROLLERS

GMS81C2012
GMS81C2020

User's Manual (Ver. 1.00)

HYUNDAI
MicroElectronics
Semiconductor Group of Hyundai Electronics Industrial Co., Ltd.

Version 1.00

Published by
MCU Application Team

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

HYUNDAI Micro Electronics 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, HYUNDAI Micro Electronics 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

GMS81C2012/GMS81C2020

CMOS Single-Chip 8-Bit Microcontroller with A/D Converter & VFD Driver

1. OVERVIEW

1.1 Description

The GMS81C2012 and GMS81C2020 are advanced CMOS 8-bit microcontroller with 12K/20K bytes of ROM. These are a powerful microcontroller which provides a highly flexible and cost effective solution to many VFD applications. These provide the following standard features: 12K/20K bytes of 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, 7-bit Watch dog Timer, Serial Peripheral Interface, on-chip oscillator and clock circuitry. They also come with high voltage I/O pins that can directly drive a VFD (Vacuum Fluorescent Display). In addition, the GMS81C2012 and GMS81C2020 support power saving modes to reduce power consumption.

Device name	ROM Size	RAM Size	OTP	Package
GMS81C2012	12K bytes	448 bytes	-	64SDIP, 64MQFP, 64LQFP
GMS81C2020	20K bytes		GMS87C2020	

1.2 Features

- **20K/12K bytes ROM(EPROM)**
- **448 Bytes of On-Chip Data RAM (Including STACK Area)**
- **Minimum Instruction Execution time:**
  - 1uS at 4MHz (2cycle NOP Instruction)
- **One 8-bit Basic Interval Timer**
- **One 7-bit Watch Dog Timer**
- **Two 8-bit Timer/Counters**
- **10-bit High Speed PWM Output**
- **One 8-bit Serial Peripheral Interface**
- **Two External Interrupt Ports**
- **One Programmable 6-bit Buzzer Driving Port**
- **60 I/O Lines**
  - 56 Programmable I/O pins (Included 30 high-voltage pins Max. 40V)
  - Three Input Only pins: 1 high-voltage pin
  - One Output Only pin
- **Eight Interrupt Sources**
  - Two External Sources (INT0, INT1)
  - Two Timer/Counter Sources (Timer0, Timer1)
  - Four Functional Sources (SPI,ADC,WDT,BIT)
- **12-Channel 8-bit On-Chip Analog to Digital Converter**
- **Oscillator:**
  - Crystal
  - Ceramic Resonator
  - External R Oscillator
- **Low Power Dissipation Modes**
  - STOP mode
  - Wake-up Timer Mode
  - Standby Mode
  - Watch Mode
  - Sub-active Mode
- **Operating Voltage: 2.7V ~ 5.5V (at 4.5MHz)**
- **Operating Frequency: 1MHz ~ 4.5MHz**
- **Sub-clock: 32.768KHz Crystal Oscillator**
- **Enhanced EMS Improvement Power Fail Processor (Noise Immunity Circuit)**

1.3 Development Tools

The GMS81C20xx are supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Jr.TM and OTP programmers. There are third different type programmers such as emulator add-on board type, single type, gang type. For mode detail, Refer to "21. OTP PROGRAMMING" on page 89. Macro assembler operates under the MS-Windows 95/98TM.

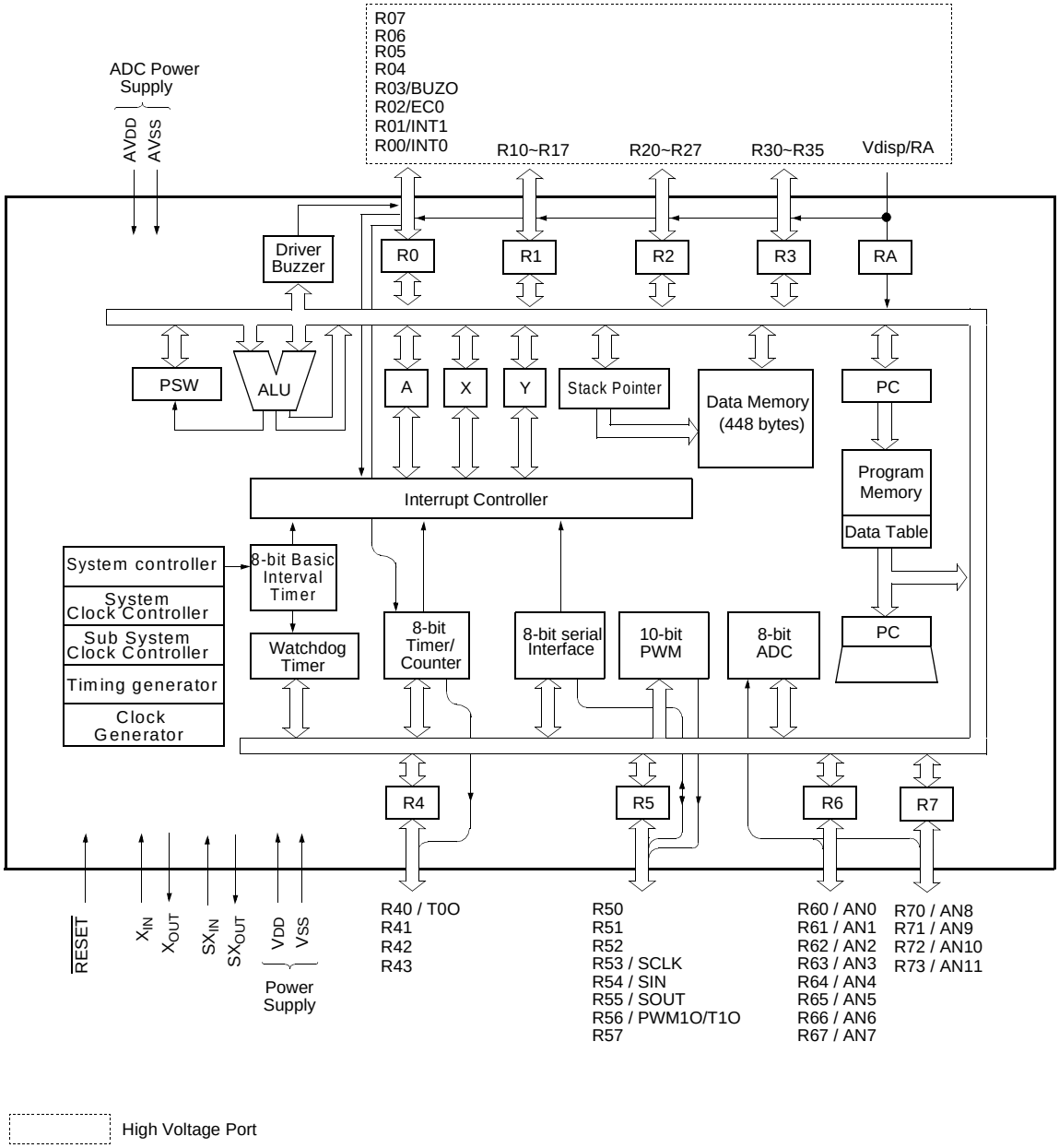
Please contact sales part of Hyundai MicroElectronics.

In Circuit Emulators	CHOICE-Dr.
Socket Adapter for OTP	CHPOD81C20D-64SD (64SDIP) CHPOD81C20D-64QFP (64MQFP) CHPOD81C20D-64LQFP (64LQFP)
Assembler	HME Macro Assembler

1.4 Ordering Information

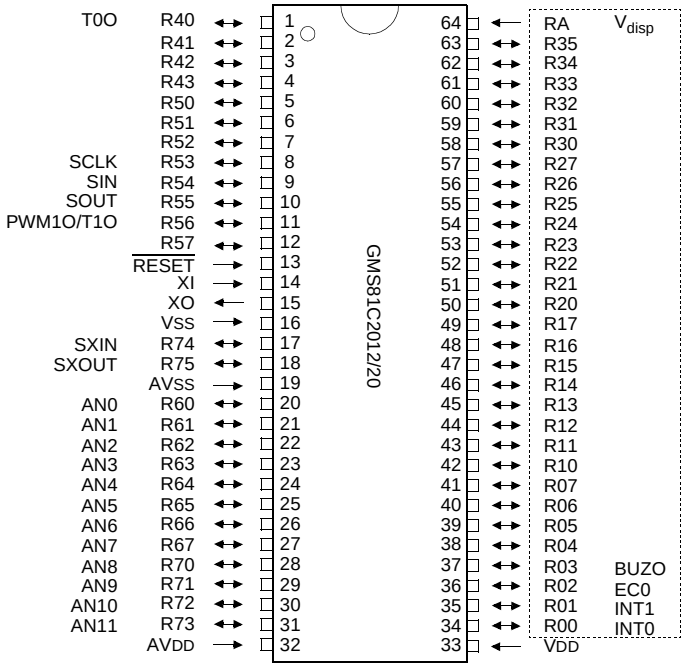
	Device name	ROM Size	RAM size	Package
Mask version	GMS81C2012 K GMS81C2012 Q GMS81C2012 LQ GMS81C2020 K GMS81C2020 Q GMS81C2020 LQ	12K bytes 12K bytes 12K bytes 20K bytes 20K bytes 20K bytes	448 bytes 448 bytes 448 bytes 448 bytes 448 bytes 448 bytes	64SDIP 64MQFP 64LQFP 64SDIP 64MQFP 64LQFP
OTP version	GMS87C2020 K GMS87C2020 Q GMS87C2020 LQ	20K bytes OTP 20K bytes OTP 20K bytes OTP	448 bytes 448 bytes 448 bytes	64SDIP 64MQFP 64LQFP

2. BLOCK DIAGRAM

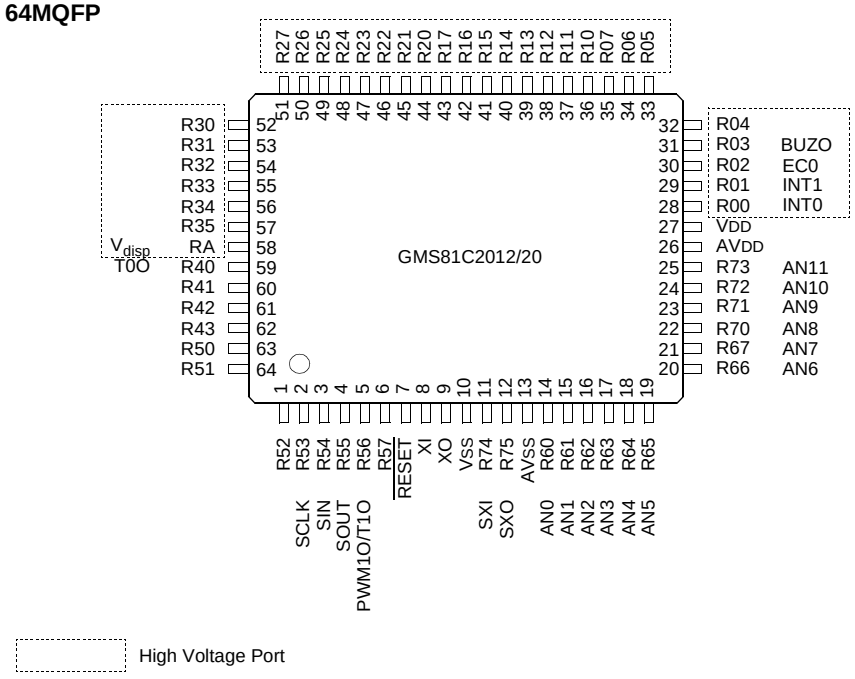


3. PIN ASSIGNMENT

64SDIP

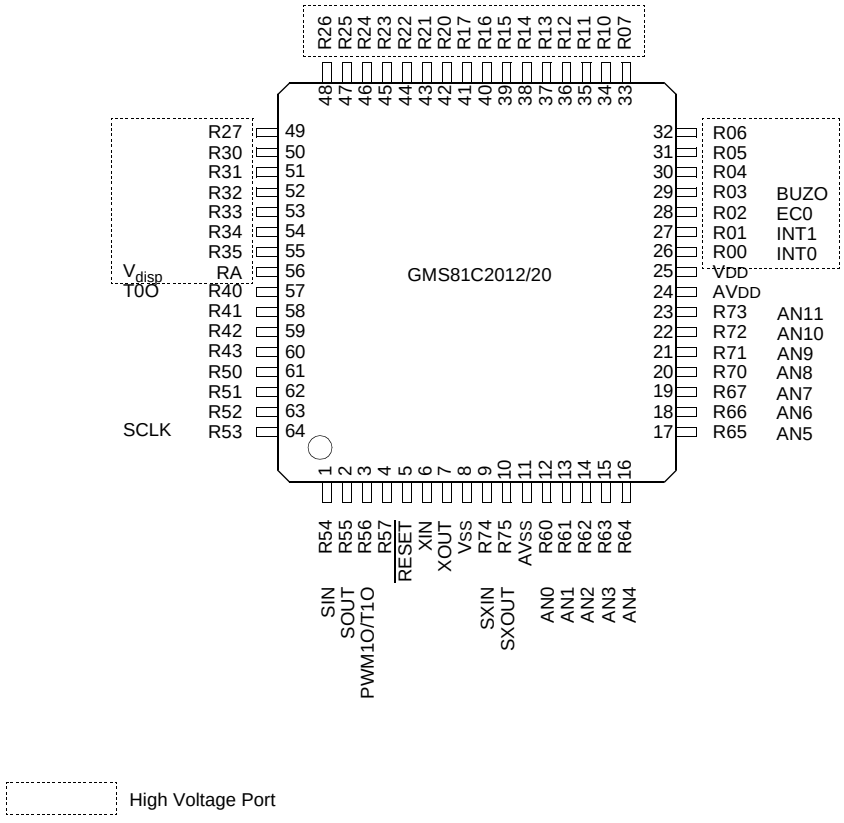


64MQFP



High Voltage Port

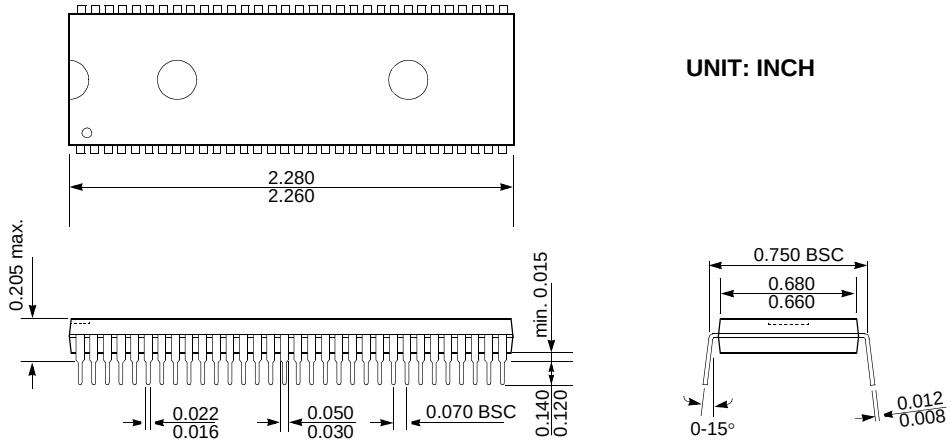
64LQFP



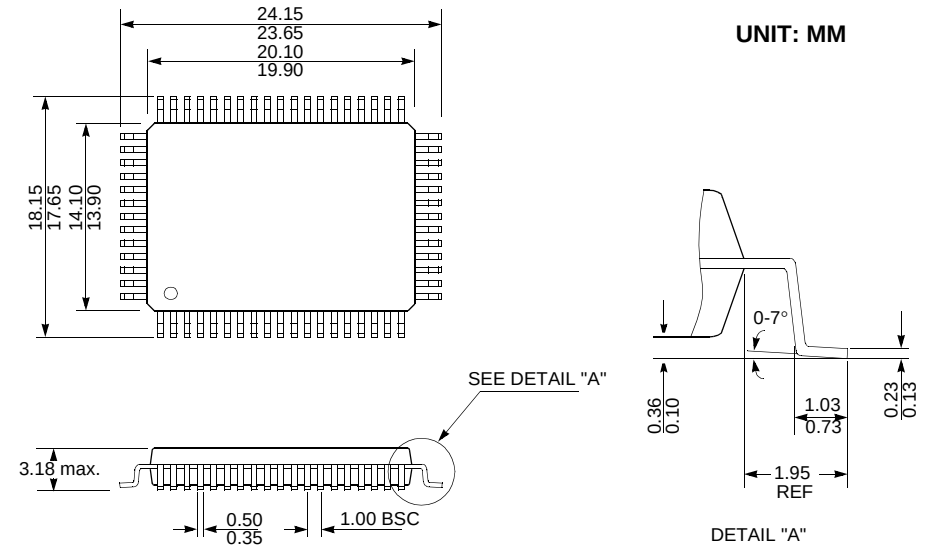
High Voltage Port

4. PACKAGE DIAGRAM

64SDIP

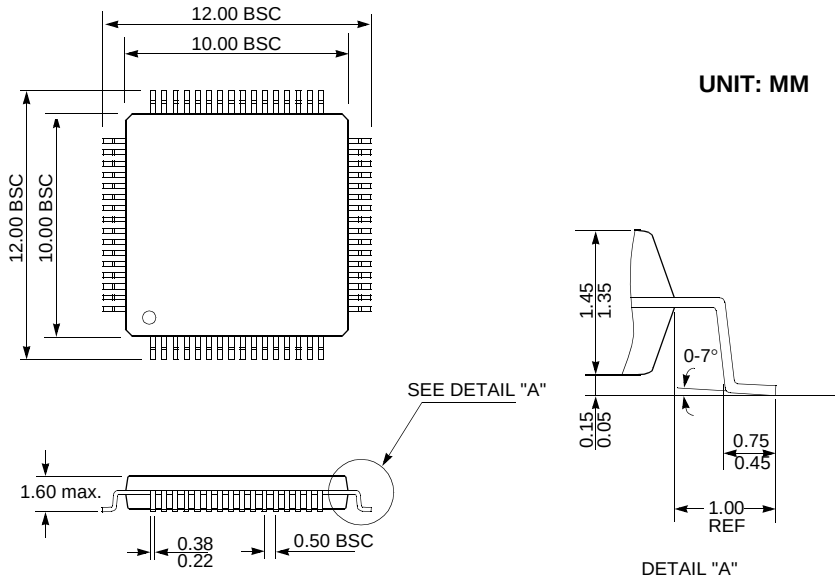


64MQFP



64LQFP

UNIT: MM



5. PIN FUNCTION

VDD: Supply voltage.

VSS: Circuit ground.

AVDD: 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.

AVSS: ADC circuit ground.

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

XIN: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XOUT: Output from the inverting oscillator amplifier.

RA(V_{disp}): RA is one-bit high-voltage input only port pin. In addition, RA serves the functions of the V_{disp} special features. V_{disp} is used as a high-voltage input power supply pin when selected by the mask option.

Port pin	Alternate function
RA	V_{disp} (High-voltage input power supply)

R00~R07: R0 is an 8-bit high-voltage CMOS bidirectional I/O port. R0 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. In addition, R0 serves the functions of the various following special features.

Port pin	Alternate function
R00	INT0 (External interrupt 0)
R01	INT1 (External interrupt 1)
R02	EC0 (Event counter input)
R03	BUZO (Buzzer driver output)

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

R20~R27: R2 is an 8-bit high-voltage CMOS bidirectional I/O port. R2 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

R30~R35: R3 is a 6-bit high-voltage CMOS bidirectional I/O port. R3 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs.

R40~R43: R4 is a 4-bit CMOS bidirectional I/O port. R4 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. In addition, R4 serves the functions of the following special features.

Port pin	Alternate function
R40	T0O (Timer/Counter 0 output)

R50~R57: 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. In addition, R5 serves the functions of the various following special features.

Port pin	Alternate function
R53	SCLK (Serial clock)
R54	SIN (Serial data input)
R55	SOUT (Serial data output)
R56	PWM1O (PWM1 Output)
	T1O (Timer/Counter 1 output)

R60~R67: 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. In addition, R6 is shared with the ADC input.

Port pin	Alternate function
R60	AN0 (Analog Input 0)
R61	AN1 (Analog Input 1)
R62	AN2 (Analog Input 2)
R63	AN3 (Analog Input 3)
R64	AN4 (Analog Input 4)
R66	AN5 (Analog Input 5)
R66	AN6 (Analog Input 6)
R67	AN7 (Analog Input 7)

R70~R73: R7 is a 4-bit CMOS bidirectional I/O port. R6 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. In addition, R7 is shared with the ADC input.

Port pin	Alternate function
R70	AN8 (Analog Input 8)
R71	AN9 (Analog Input 9)
R72	AN10 (Analog Input 10)
R73	AN11 (Analog Input 11)

SXIN: Input to the internal subsystem clock operating circuit. In addition, SXIN serves the R74 pin when selected by the code option. *R74 has a Pull-up circuit.

SXOUT: Output from the inverting subsystem oscillator amplifier. In addition, SXOUT serves the R75 pin when

selected by the code option. *R75 has a Pull-up circuit.

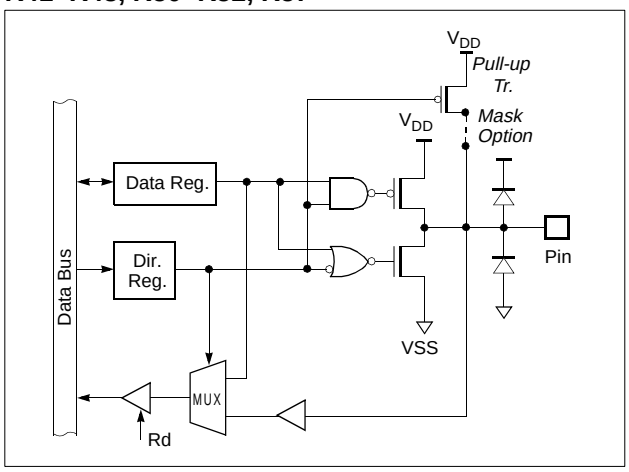
Port pin	Alternate function
SXI SXO	R74(Included Internal Pull-up Resister) R75(Included Internal Pull-up Resister)

PIN NAME	In/Out	Function	
		Basic	Alternate
VDD	-	Supply voltage	
VSS	-	Circuit ground	
RA (V_{disp})	I(I)	1-bit high-voltage Input only port	High-voltage input power supply pin
$\overline{\text{RESET}}$	I	Reset signal input	
XIN	I	Oscillation input	
XOUT	O	Oscillation output	
SXIN(R74)	I	Sub Oscillation input	General I/O ports
SXOUT(R75)	O	Sub Oscillation output	
R00 (INT0)	I/O (I)	8-bit high-voltage I/O ports	External interrupt 0 input
R01 (INT1)	I/O (I)		External interrupt 1 input
R02 (EC0)	I/O (I)		Timer/Counter 0 external input
R03 (BUZO)	I/O (O)		Buzzer driving output
R04~R07	I/O		
R10~R17	I/O	8-bit high-voltage I/O ports	
R20~R27	I/O	8-bit high-voltage I/O ports	
R30~R35	I/O	6-bit high-voltage I/O ports	
R40 (T0O)	I/O (O)	4-bit general I/O ports	Timer/Counter 0 output
R41~R43	I/O		
R50~R52	I/O	8-bit general I/O ports	
R53 (SCLK)	I/O (I/O)		Serial clock source
R54 (SIN)	I/O (I)		Serial data input
R55 (SOUT)	I/O (O)		Serial data output
R56 (PWM1O/T1O)	I/O (O)		PWM 1 pulse output /Timer/Counter 1 output
R57	I/O		
R60~R67 (AN0~AN7)	I/O (I)	8-bit general I/O ports	Analog voltage input
R70~R73 (AN8~AN11)	I/O (I)	4-bit general I/O ports	
AVDD	-	Supply voltage input pin for ADC	
AVSS	-	Ground level input pin for ADC	
VDD	-	Supply voltage	
VSS	-	Circuit ground	

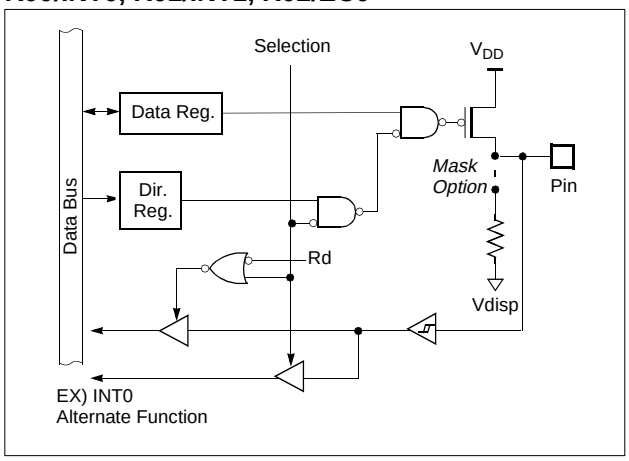
Table 5-1 GMS81C2020 Port Function Description

6. PORT STRUCTURES

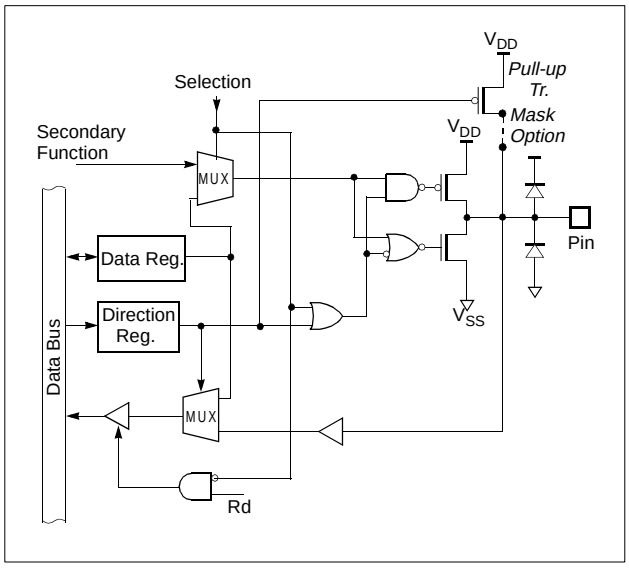
R41~R43, R50~R52, R57



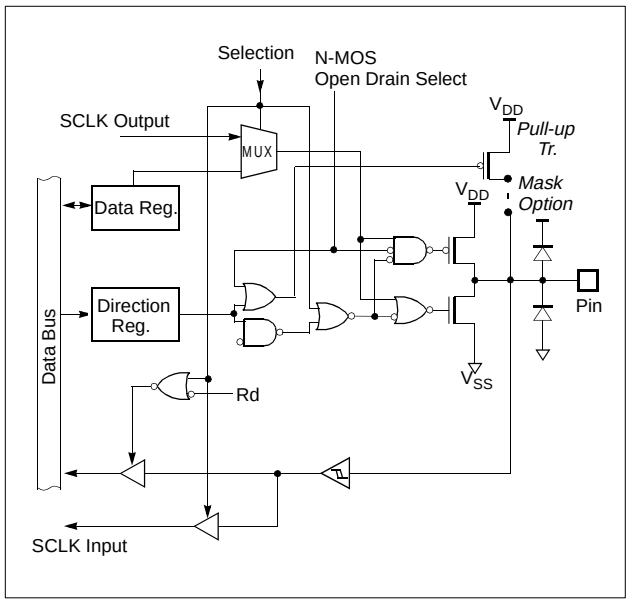
R00/INT0, R01/INT1, R02/EC0



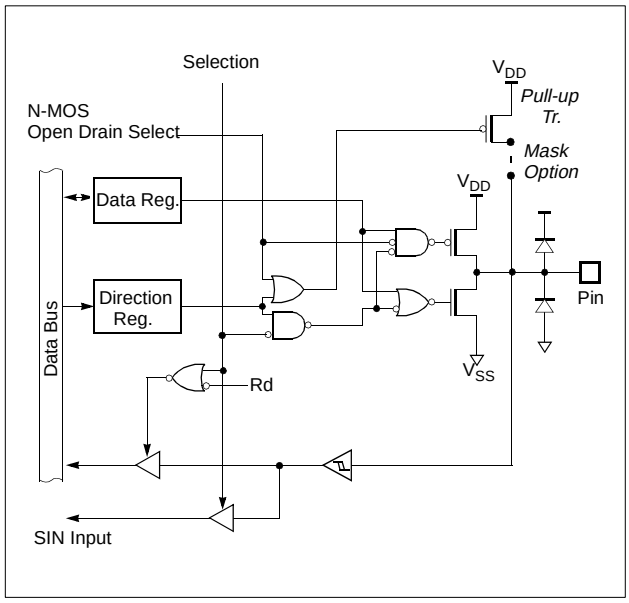
R40/T0O



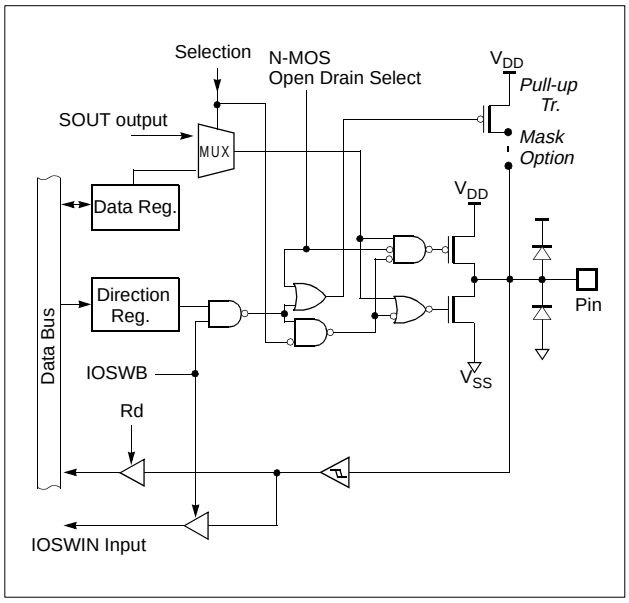
R53/SCLK



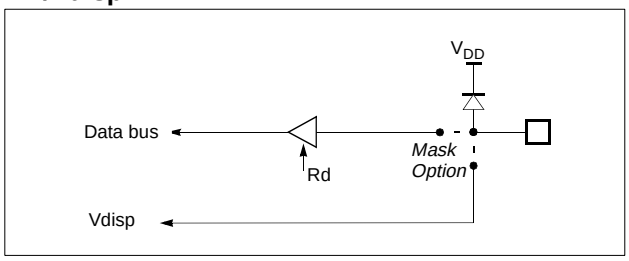
R54/SIN



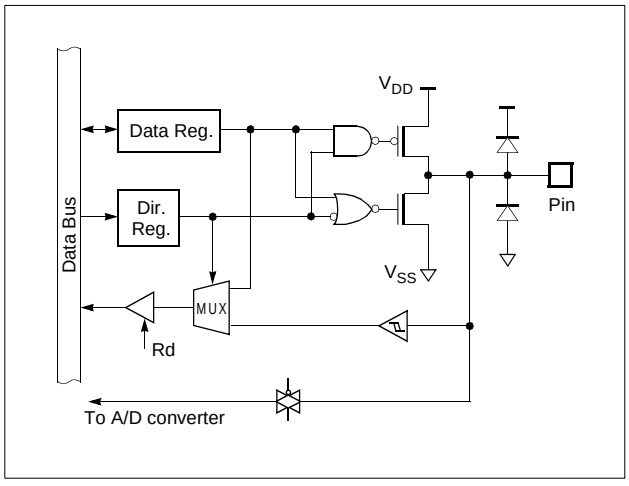
R55/SOUT



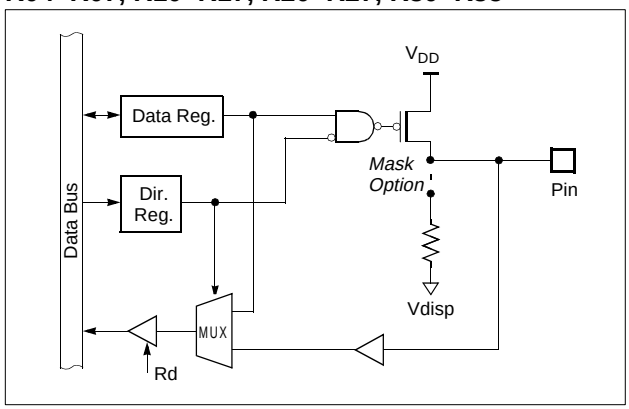
RA/Vdisp



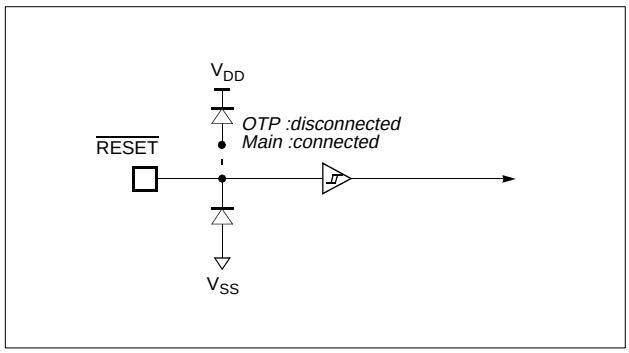
R64/AN7 ~ R67/AN7



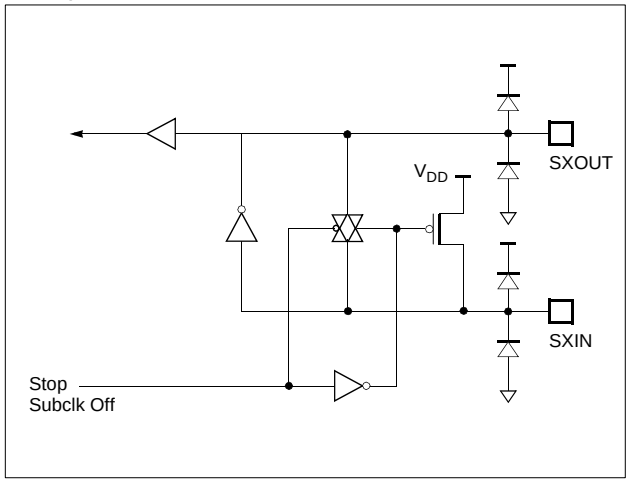
R04~R07, R10~R17, R20~R27, R30~R35



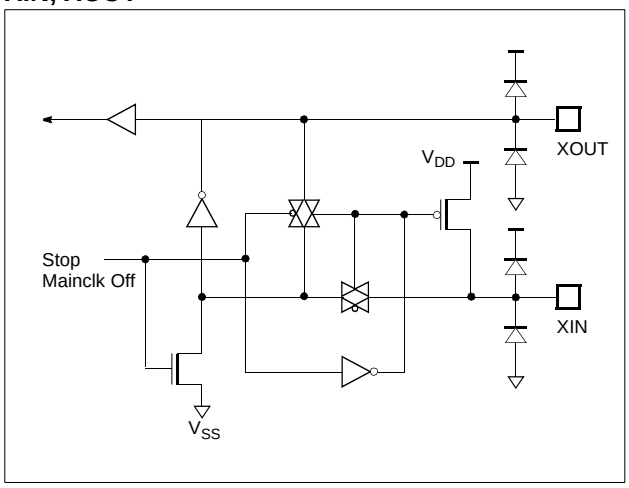
$\overline{\text{RESET}}$



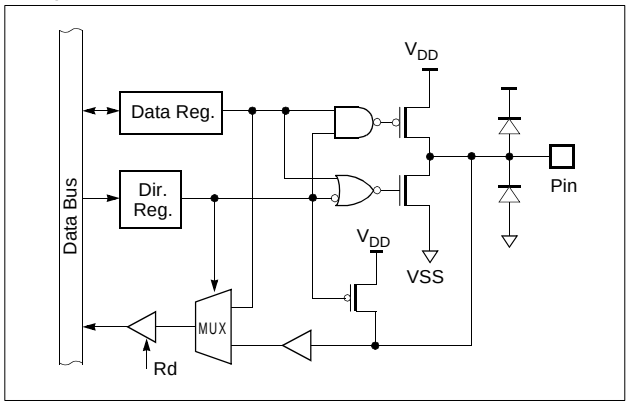
SXIN, SXOUT



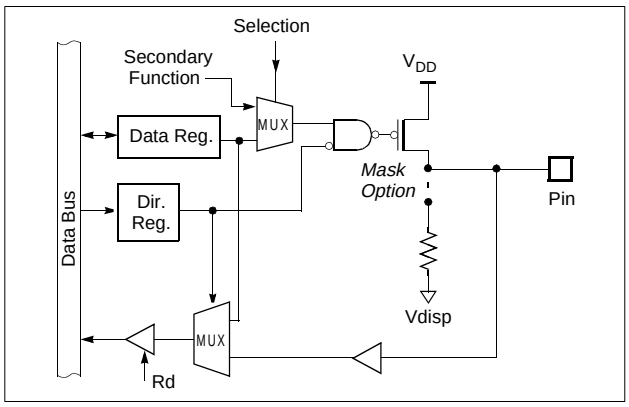
XIN, XOUT



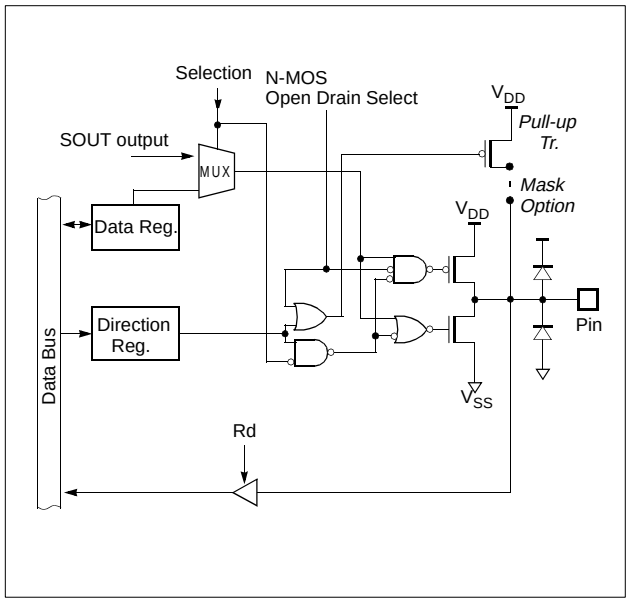
R74, R75



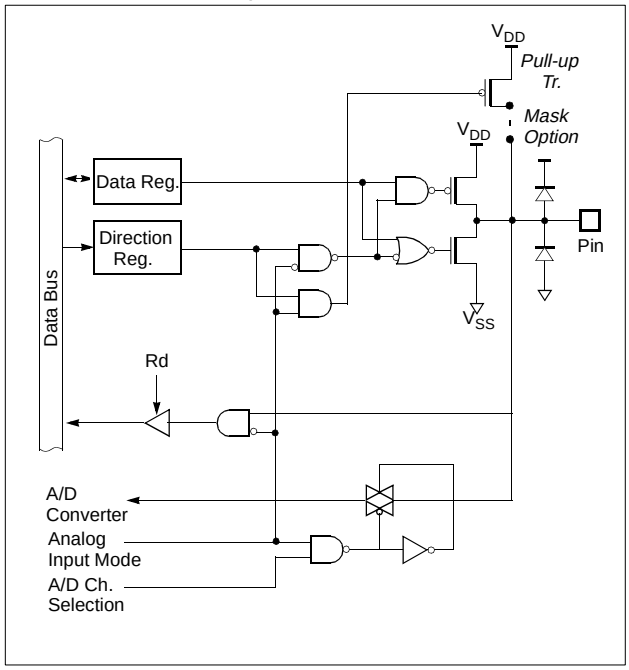
R03/BUZO



R56/PWM1O/T1O



R60~R67/AN0~AN7, R70~R74/AN8~AN11



7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings

Supply voltage ..... -0.3 to +7.0 V

Storage Temperature ..... -40 to +85 °C

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

Voltage on High voltage pin with respect to Ground (V_{SS}) ..... -45V to V_{DD}+0.3 V

Maximum current out of V_{SS} pin ..... 150 mA

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

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

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

Maximum current (ΣI_{OL}) ..... 100 mA

Maximum current (ΣI_{OH}) ..... 50 mA

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

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Specifications		Unit
			Min.	Max.	
Supply Voltage	V_{DD}	f_{XI} = 4.5 MHz	2.7	5.5	V
Operating Frequency	f_{XIN}	V_{DD} = V_{DD}	1	4.5	MHz
Operating Temperature	T_{OPR}		-40	85	°C

7.3 A/D Converter Characteristics

(T_A=25°C, V_{DD}=5V, V_{SS}=0V, AV_{DD}=5.12V, AV_{SS}=0V @f_{XIN} =4MHz)

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.[1]	Max.	
Analog Power Supply Input Voltage Range	AV_{DD}		AV_{SS}	-	AV_{DD}	V
Analog Input Voltage Range	V_{AN}		AV_{SS}-0.3		AV_{DD}+0.3	V
Current Following Between AV_{DD} and AV_{SS}	I_{AVDD}		-	–	200	uA
Overall Accuracy	CA_{IN}		-	±1.5	±2	LSB
Non-Linearity Error	N_{NLE}		-	±1.5	±2	LSB
Differential Non-Linearity Error	N_{DNLE}		-	±0.5	±1	LSB
Zero Offset Error	N_{ZOE}		-	±0.5	±1.5	LSB
Full Scale Error	N_{FSE}		-	±0.5	±1	LSB
Gain Error	N_{NLE}		-	±0.5	±1	LSB
Conversion Time	T_{CONV}	f_{XIN}=4MHz	-	-	20	us

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

7.4 DC Electrical Characteristics for Standard Pins(5V)

(V_{DD} = 5.0V ± 10%, V_{SS} = 0V, T_A = -40 ~ 85°C, f_{XIN} = 4 MHz, Vdisp = V_{DD}-40V to V_{DD}),

Parameter	Pin	Symbol	Test Condition	Specification			Unit
				Min	Typ.[1]	Max	
Input High Voltage	XIN, SXIN	V_{IH1}	External Clock	0.9V_{DD}		V_{DD}+0.3	V
	$\overline{RESET}$,SIN,R55,SCLK, INT0&1,EC0	V_{IH2}		0.8V_{DD}		V_{DD}+0.3	
	R40~R43,R5,R6,R70~R73	V_{IH3}		0.7V_{DD}		V_{DD}+0.3	
Input Low Voltage	XIN, SXIN	V_{IL1}	External Clock	-0.3		0.1V_{DD}	V
	$\overline{RESET}$,SIN,R55,SCLK, INT0&1,EC0	V_{IL2}		-0.3		0.2V_{DD}	
	R40~R43,R5,R6,R70~R73	V_{IL3}		-0.3		0.3V_{DD}	
Output High Voltage	R40~R43,R5,R6,R70~R73 BUZO,T0O,PWM1O/T1O, SCLK,SOUT	V_{OH}	I_{OH} = -0.5mA	V_{DD}-0.5			V
Output Low Voltage	R40~R43,R5,R6,R70~R73 BUZO,T0O,PWM1O/T1O, SCLK,SOUT	V_{OL1} V_{OL2}	I_{OL} = 1.6mA I_{OL} = 10mA			0.4 2	V
Input High Leakage Current	R40~R43,R5,R6,R70~R73	I_{IH1}				1	uA
	XIN	I_{IH2}				1	
Input Low Leakage Current	R40~R43,R5,R6,R70~R73	I_{IL1}		-1			uA
	XIN	I_{IL2}		-1			
Input Pull-up Current(*Option)	R40~R43,R5,R6,R70~R73	I_{PU}		50	100	180	uA
Power Fail Detect Voltage	V_{DD}	V_{PFD}			2.7		V
Current dissipation in active mode	V_{DD}	I_{DD}	f_{XIN}=4.5MHz			8	mA
Current dissipation in standby mode	V_{DD}	I_{STBY}	f_{XIN}=4.5MHz			3	mA
Current dissipation in sub-active mode	V_{DD}	I_{SUB}	f_{XIN} = Off f_{SXIN}=32.7KHz			100	uA
Current dissipation in watch mode	V_{DD}	I_{WTC}	f_{XIN}=Off f_{SXNI}=32.7KHz			20	uA
Current dissipation in stop mode	V_{DD}	I_{STOP}	f_{XIN}=Off f_{SXIN}=32.7KHz			10	uA
Hysteresis	$\overline{RESET}$,SIN,R55,SCLK, INT0,INT1,EC0	V_{T+}~V_{T-}		0.4			V
Internal RC WDT Frequency	XOUT	T_{RCWDT}		8		30	KHz
RC Oscillation Frequency	XOUT	f_{RCOSC}	R= 120KΩ	1.5	2	2.5	MHz

1. Data in “Typ.” column is at 4.5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

7.5 DC Electrical Characteristics for High-Voltage Pins

(V_{DD} = 5.0V ± 10%, V_{SS} = 0V, T_A = -40 ~ 85°C, f_{XIN} = 4 MHz, Vdisp = V_{DD}-40V to V_{DD})

Parameter	Pin	Symbol	Test Condition	Specification			Unit
				Min	Typ.[1]	Max	
Input High Voltage	R0,R1,R2,R30~R35,RA	V_{IH}		$0.7V_{DD}$		V_{DD}+0.3	V
Input Low Voltage	R0,R1,R2,R30~R35,RA	V_{IL}		V_{DD}-40		$0.3V_{DD}$	V
Output High Voltage	R0,R1,R2,R30~R35	V_{OH}	I_{OH} = -15mA I_{OH} = -10mA I_{OH} = - 4mA	V_{DD}-3.0 V_{DD}-2.0 V_{DD}-1.0			V
Output Low Voltage	R0,R1,R2,R30~R35	V_{OL}	Vdisp = V_{DD}-40 150KΩ atV_{DD}-40			V_{DD}-37 V_{DD}-37	V
Input High Leakage Current	R0,R1,R2,R30~R35,RA	I_{IH}	V_{IN}=V_{DD}-40V to V_{DD}			20	uA
Input Pull-down Current(*Option)	R0,R1,R2,R30~R35	I_{PD}	Vdisp=V_{DD}-35V V_{IN}=V_{DD}	200	600	1000	uA
Input High Voltage	R0,R1,R2,R30~R35,RA	V_{IH}		$0.7V_{DD}$		V_{DD}+0.3	V

1. Data in “Typ.” column is at 4.5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

7.6 AC Characteristics

(T_A=-40~ 85°C, V_{DD}=5V±10%, V_{SS}=0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Operating Frequency	f_{CP}	XIN	1	-	8	MHz
External Clock Pulse Width	t_{CPW}	XIN	80	-	-	nS
External Clock Transition Time	t_{RCP},t_{FCP}	XIN	-	-	20	nS
Oscillation Stabilizing Time	t_{ST}	XIN, XOUT	-	-	20	mS
External Input Pulse Width	t_{EPW}	INT0, INT1, EC0	2	-	-	t_{SYS}
External Input Pulse Transition Time	t_{REP},t_{FEP}	INT0, INT1, EC0	-	-	20	nS
RESET Input Width	t_{RST}	$\overline{\text{RESET}}$	8	-	-	t_{SYS}

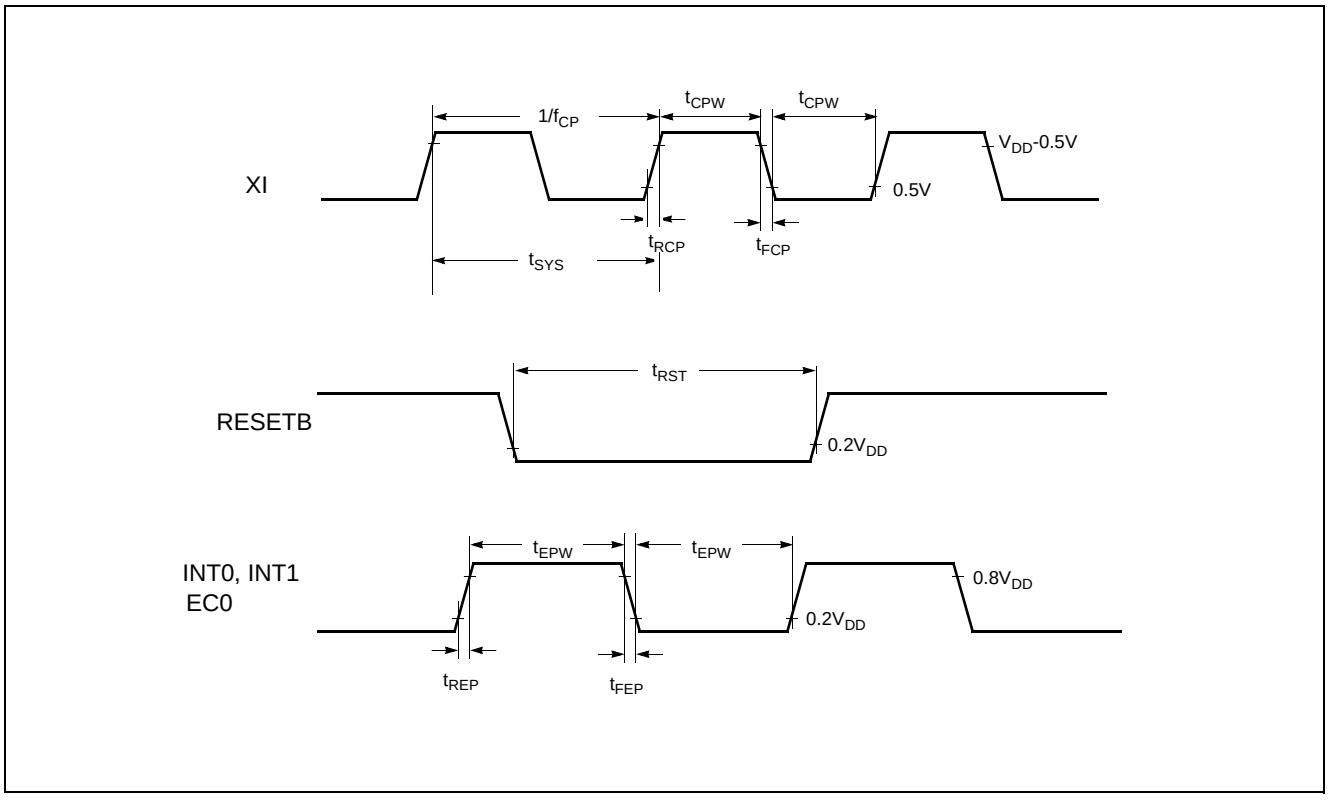


Figure 7-1 Timing Chart

7.7 AC Characteristics

(T_A=-20~+85°C, V_{DD}=5V±10%, V_{SS}=0V, f_{XIN}=4MHz)

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

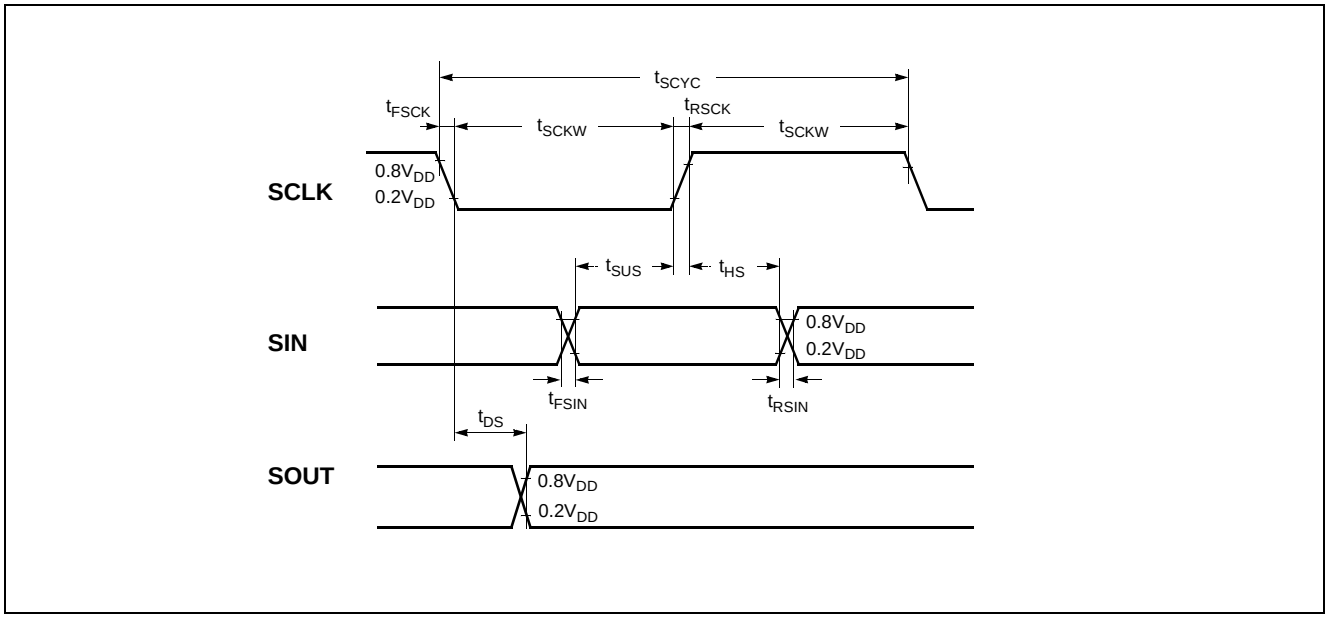


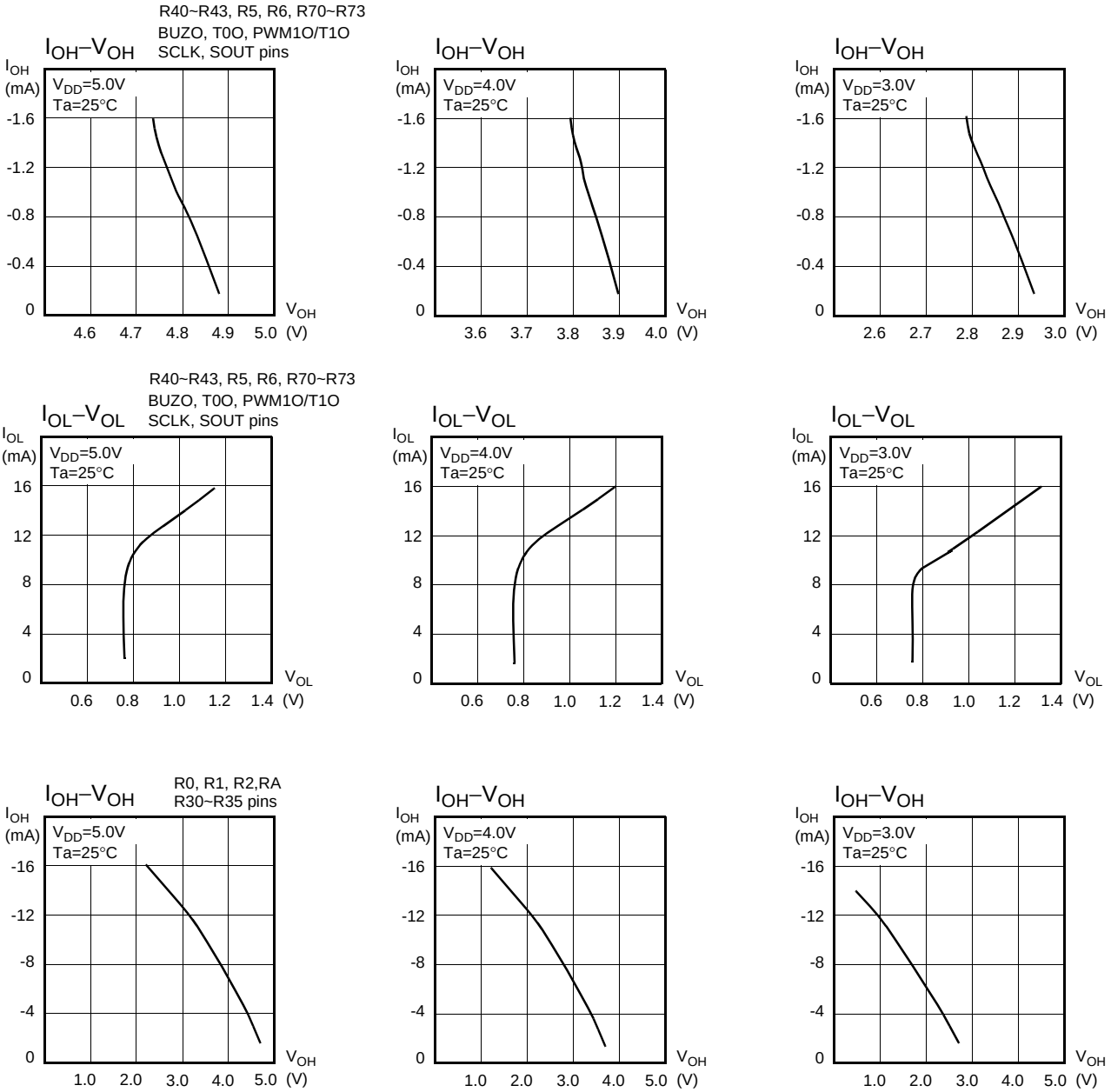
Figure 7-2 Serial I/O Timing Chart

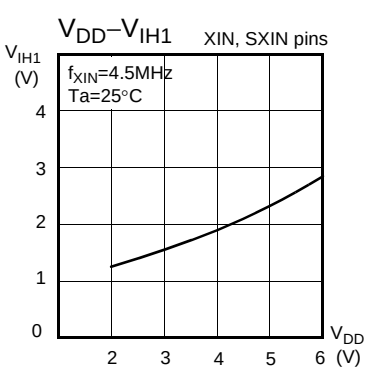
7.8 Typical Characteristics

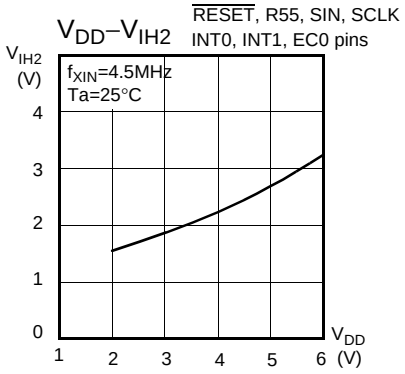
This graphs and tables provided in this section are for design guidance only and are not tested or guaranteed.

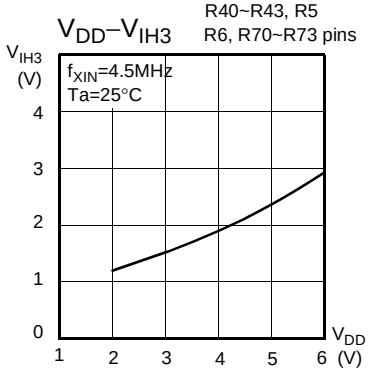
In some graphs or tables the data presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for information only and devices are guaranteed to operate properly only within the specified range.

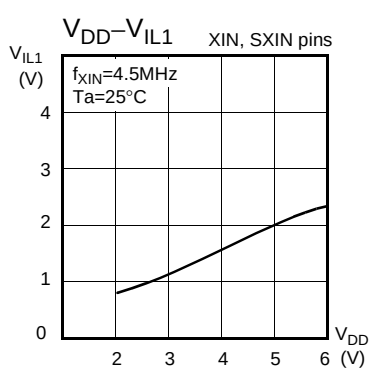
The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean – 3σ) respectively where σ is standard deviation

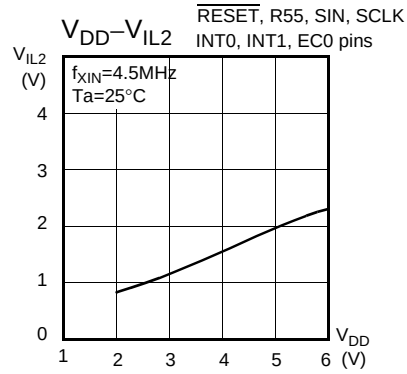


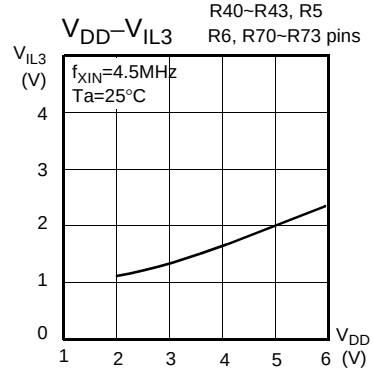

V_{DD}–V_{IH1}
XIN, SXIN pins
V_{IH1} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)

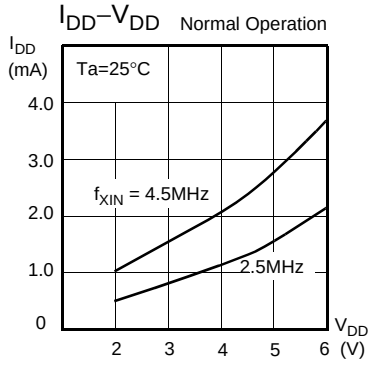

V_{DD}–V_{IH2}
$\overline{RESET}$, R55, SIN, SCLK
INT0, INT1, EC0 pins
V_{IH2} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)

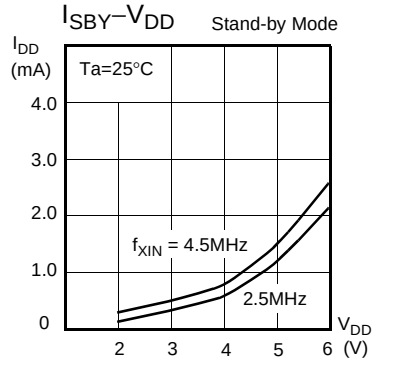

V_{DD}–V_{IH3}
R40~R43, R5
R6, R70~R73 pins
V_{IH3} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)

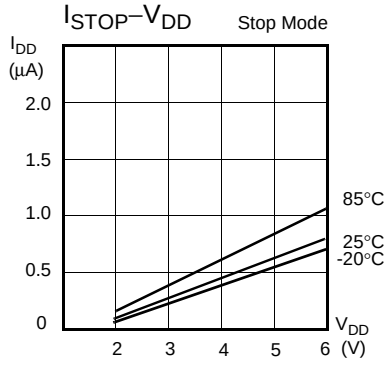

V_{DD}–V_{IL1}
XIN, SXIN pins
V_{IL1} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)


V_{DD}–V_{IL2}
$\overline{RESET}$, R55, SIN, SCLK
INT0, INT1, EC0 pins
V_{IL2} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)


V_{DD}–V_{IL3}
R40~R43, R5
R6, R70~R73 pins
V_{IL3} (V)
f_{XIN}=4.5MHz
Ta=25°C
V_{DD} (V)


I_{DD}–V_{DD}
Normal Operation
I_{DD} (mA)
Ta=25°C
f_{XIN} = 4.5MHz
2.5MHz
V_{DD} (V)


I_{SBY}–V_{DD}
Stand-by Mode
I_{DD} (mA)
Ta=25°C
f_{XIN} = 4.5MHz
2.5MHz
V_{DD} (V)


I_{STOP}–V_{DD}
Stop Mode
I_{DD} (μA)
85°C
25°C
-20°C
V_{DD} (V)

8. MEMORY ORGANIZATION

The GMS81C2012 and GMS81C2020 have separate address spaces for Program memory and Data Memory. Program memory can only be read, not written to. It can be up to 12K/20K bytes of Program memory. Data memory can be read and written to up to 448 bytes including the stack area.

8.1 Registers

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

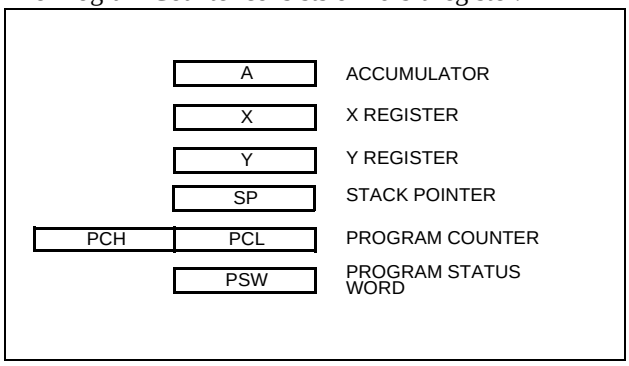


Figure 8-1 Configuration of Registers

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

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

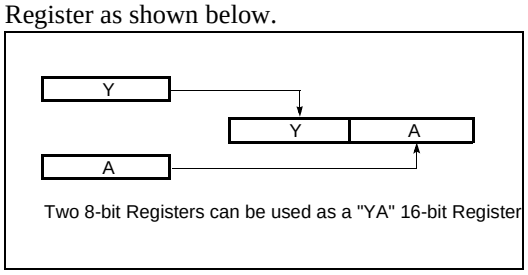


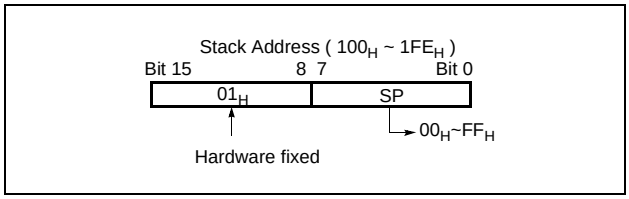
Figure 8-2 Configuration of YA 16-bit Register

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

Stack Pointer: The Stack Pointer is an 8-bit register used for occurrence interrupts and calling out subroutines. Stack Pointer identifies the location in the stack to be 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 excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost.

The stack can be located at any position within 100_H to $1FF_H$ of the internal data memory. The SP is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of "FF_H" is used.



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 ← FF_H

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

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

[Carry flag C]

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

[Zero flag Z]

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

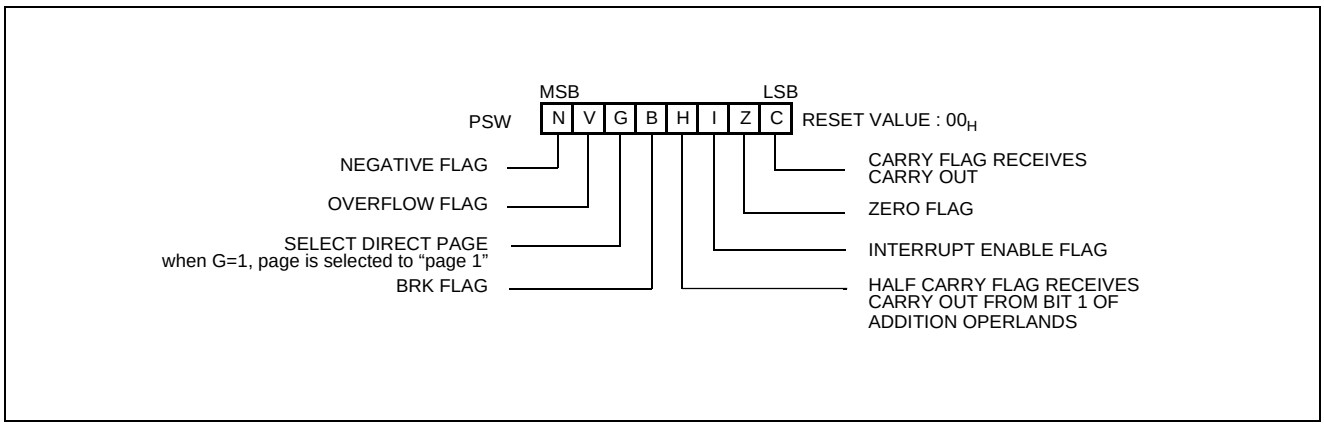


Figure 8-3 PSW (Program Status Word) Register

[Interrupt disable flag I]

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

[Half carry flag H]

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

[Break flag B]

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

[Direct page flag G]

This flag assigns RAM page for direct addressing mode. In the direct addressing mode, addressing area is from zero page 00_H to $0FF_H$ when this flag is "0". If it is set to "1", addressing area is assigned 100_H to $1FF_H$. It is set by SETG instruction and cleared by CLRG.

[Overflow flag V]

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

[Negative flag N]

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

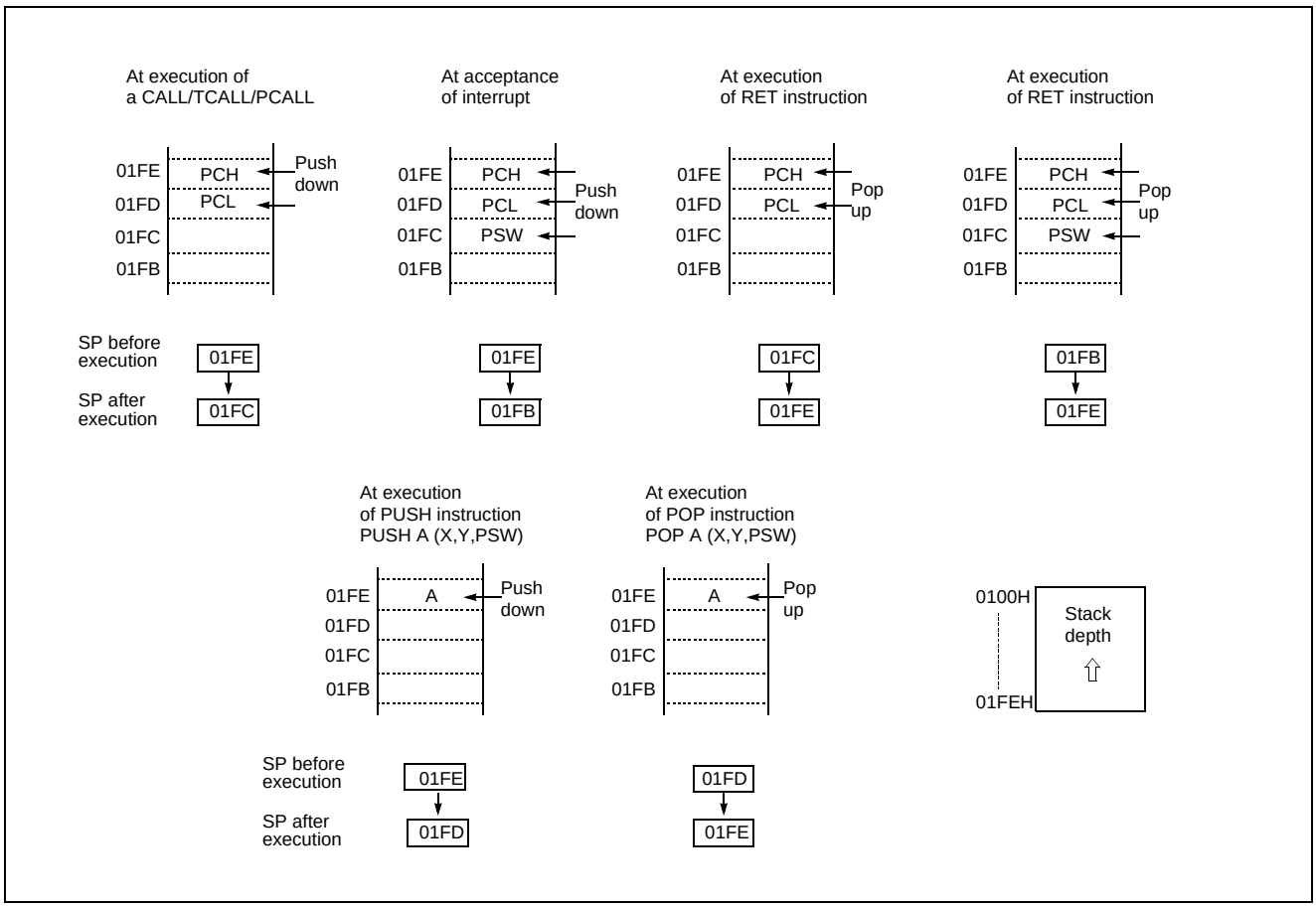


Figure 8-4 Stack Operation

8.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 20K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

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

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

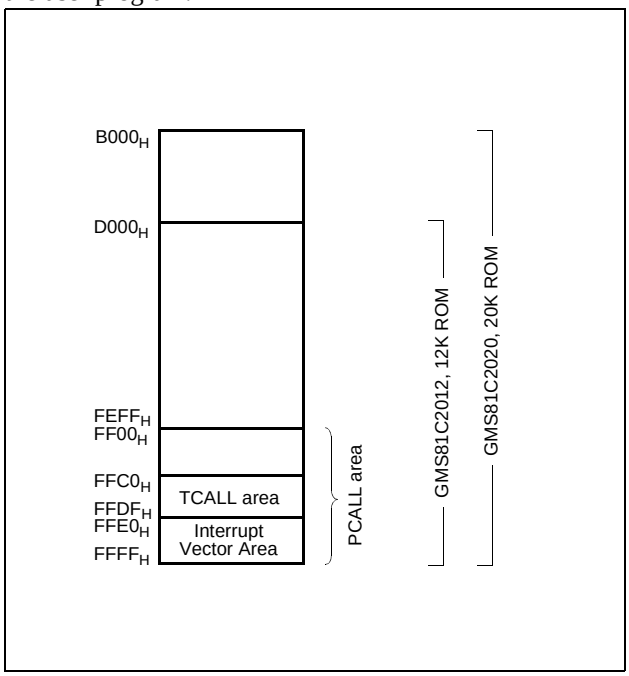


Figure 8-5 Program Memory Map

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

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

Example: Usage of TCALL

```
LDA       #5
          TCALL  0FH          ;1BYTE INSTRUCTION
          :                   ;INSTEAD OF 3 BYTES
          :                   ;NORMAL CALL
;
;TABLE CALL ROUTINE
;
FUNC_A:   LDA    LRG0
          RET
;
FUNC_B:   LDA    LRG1     ②  ①
          RET
;
;TABLE CALL ADD. AREA
;
          ORG    0FFC0H       ;TCALL ADDRESS AREA
          DW     FUNC_A
          DW     FUNC_B
```

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

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

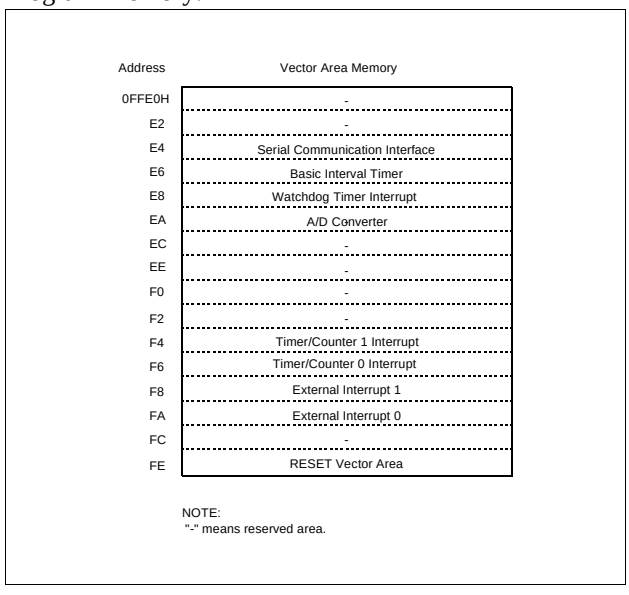


Figure 8-6 Interrupt Vector Area

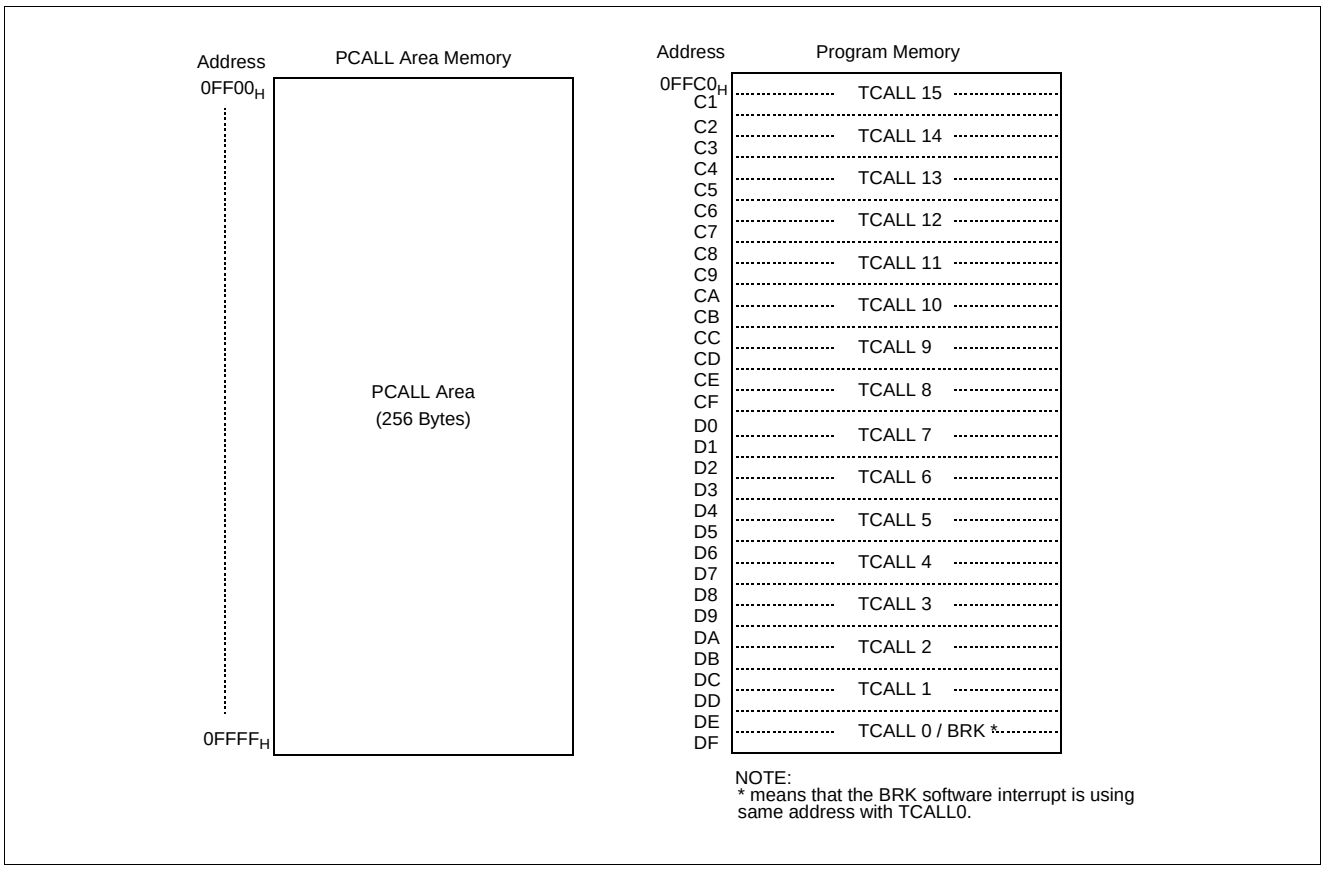
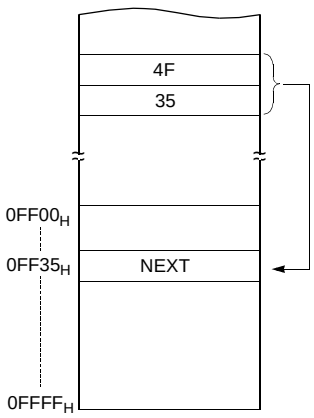


Figure 8-7 PCALL and TCALL Memory Area

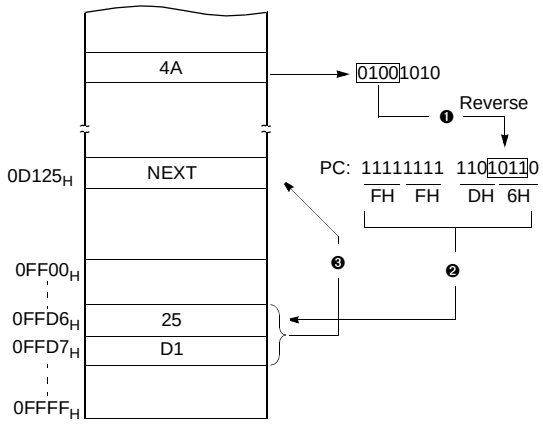
PCALL→ rel

4F35 PCALL 35H



TCALL→ n

4A TCALL 4



Example: The usage software example of Vector address for GMS81C2020.

```
         ORG    0FFE0H

         DW     NOT_USED
         DW     NOT_USED
         DW     SIO          ; Serial Interface
         DW     BIT_TIMER    ; Basic Interval Timer
         DW     WD_TIMER     ; Watchdog Timer
         DW     ADC          ; ADC
         DW     NOT_USED
         DW     NOT_USED
         DW     NOT_USED
         DW     NOT_USED
         DW     TIMER1       ; Timer-1
         DW     TIMER0       ; Timer-0
         DW     INT1         ; Int.1
         DW     INT0         ; Int.0
         DW     NOT_USED     ; -
         DW     RESET        ; Reset


         ORG    0B000H       ; GMS81C2020(20K)ROM Start address
;        ORG    0D000H       ; GMS81C2012(12K)ROM Start address

;*****************************************
;            MAIN     PROGRAM              *
;*****************************************
;
;
RESET:   DI                  ;Disable All Interrupts
         CLRG
         LDX    #0
RAM_CLR: LDA    #0           ;RAM Clear(!0000H->!00BFH)
         STA    {X}+
         CMPX   #0C0H
         BNE    RAM_CLR
;
         LDX    #0FFH        ;Stack Pointer Initialize
         TXSP
;
         LDM    R0, #0       ;Normal Port 0
         LDM    R0IO,#82H    ;Normal Port Direction
         :
         :
         :
         LDM    TDR0,#125    ;8us x 125 = 1mS
         LDM    TM0,#0FH     ;Start Timer0, 8us at 4MHz
         LDM    IRQH,#0
         LDM    IRQL,#0
         LDM    IENH,#0E0H   ;Enable Timer0, INT0, INT1
         LDM    IENL,#0
         LDM    IEDS,#05H    ;Select falling edge detect on INT pin
         LDM    R0FUNC,#03H ;Set external interrupt pin(INT0, INT1)

         EI                  ;Enable master interrupt
         :
         :
         :




         :
         :
NOT_USED:NOP
         RETI
```

8.3 Data Memory

Figure 8-8 shows the internal Data Memory space available. Data Memory is divided into two groups, a user RAM (including Stack) and control registers.

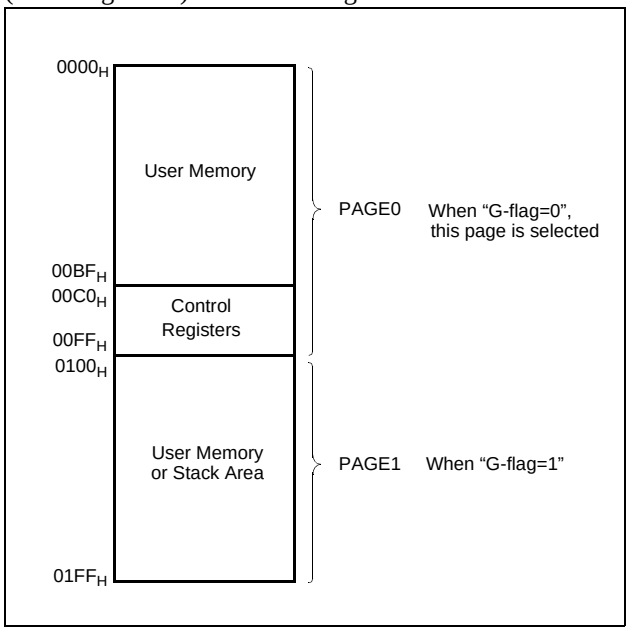


Figure 8-8 Data Memory Map

User Memory

The GMS81C20xx have 448 × 8 bits for the user memory (RAM).

Control Registers

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

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

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

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

Example; To write at CKCTLR

```
LDM    CLCTLR,#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 27.

Address	Symbol	R/W	RESET Value	Addressing mode
0C0H	R0	R/W	Undefined	byte, bit[1]
0C1H	R0IO	W	0000_0000	byte[2]
0C2H	R1	R/W	Undefined	byte, bit
0C3H	R1IO	W	00000000	byte
0C4H	R2	R/W	Undefined	byte, bit
0C5H	R2IO	W	0000_0000	byte
0C6H	R3	R/W	Undefined	byte, bit
0C7H	R3IO	W	--00_0000	byte
0C8H	R4	R/W	Undefined	byte, bit
0C9H	R4IO	W	----_0000	byte
0CAH	R5	R/W	Undefined	byte, bit
0CBH	R5IO	W	0000_0000	byte
0CCH	R6	R/W	Undefined	byte, bit
0CDH	R6IO	W	0000_0000	byte
0CEH	R7	R/W	Undefined	byte, bit
0CFH	R7IO	W	--00_0000	byte
0D0H	TM0	R/W	--00_0000	byte, bit
0D1H	T0	R	0000_0000	byte
0D1H	TDR0	W	1111_1111	byte
0D1H	CDR0	R	0000_0000	byte
0D2H	TM1	R/W	0000_0000	byte, bit
0D3H	TDR1	W	1111_1111	byte
0D3H	T1PPR	W	1111_1111	byte
0D4H	T1	R	0000_0000	byte
0D4H	CDR1	R	0000_0000	byte
0D4H	T1PDR	R/W	0000_0000	byte, bit
0D5H	PWM1HR	W	----_0000	byte
0DEH	BUR	W	1111_1111	byte
0E0H	SIOM	R/W	0000_0001	byte, bit
0E1H	SIOR	R/W	Undefined	byte, bit
0E2H	IENH	R/W	0000_----	byte, bit
0E3H	IENL	R/W	0000_----	byte, bit
0E4H	IRQH	R/W	0000_----	byte, bit
0E5H	IRQL	R/W	0000_----	byte, bit
0E6H	IEDS	R/W	----_0000	byte, bit
0EAH	ADCM	R/W	-000_0001	byte, bit
0EBH	ADCR	R	Undefined	byte
0ECH	BITR	R	0000_0000	byte
0ECH	CKCTLR	W	-001_0111	byte
0EDH	WDTR	R	0000_0000	byte
0EDH	WDTR	W	0111_1111	byte
0EFH	PFDR	R/W	----_-100	byte, bit
0F4H	R0FUNC	W	----_0000	byte
0F5H	R4FUNC	W	----_---0	byte
0F6H	R5FUNC	W	-0--_----	byte
0F7H	R6FUNC	W	0000_0000	byte
0F8H	R7FUNC	W	----_0000	byte
0F9H	R5NODR	W	0000_0000	byte
0FAH	SCMR	R/W	---0_0000	byte
0FBH	RA	R	Undefined	-[3]

Table 8-1 Control Registers

1. "byte, bit" means that register can be addressed by not only bit but byte manipulation instruction.
2. "byte" means that register can be addressed by only byte manipulation instruction. On the other hand, do not use any read-modify-write instruction such as bit manipulation for clearing bit.
3. RA is one-bit high-voltage input only port pin. In addition, RA serves the functions of the Vdisp special features. Vdisp is used as a high-voltage input power supply pin when selected by the mask option.

Note: *Several names are given at same address. Refer to below table.*

Addr.	When read			When write	
	Timer Mode	Capture Mode	PWM Mode	Timer Mode	PWM Mode
D1H	T0	CDR0	-	TDR0	-
D3H	-			TDR1	T1PPR
D4H	T1	CDR1	T1PDR	-	T1PDR
ECH	BITR			CKCTLR	

Table 8-2 Various Register Name in Same Address

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0H	R0	R0 Port Data Register (Bit[7:0])							
C1H	R0IO	R0 Port Direction Register (Bit[7:0])							
C2H	R1	R1 Port Data Register (Bit[7:0])							
C3H	R1IO	R1 Port Direction Register (Bit[7:0])							
C4H	R2	R2 Port Data Register (Bit[7:0])							
C5H	R2IO	R2 Port Direction Register (Bit[7:0])							
C6H	R3	R3 Port Data Register **(Bit[5:0])**							
C7H	R3IO	R3 Port Direction Register **(Bit[5:0])**							
C8H	R4	R4 Port Data Register **(Bit[3:0])**							
C9H	R4IO	R4 Port Direction Register **(Bit[3:0])**							
CAH	R5	R5 Port Data Register (Bit[7:0])							
CBH	R5IO	R5 Port Direction Register (Bit[7:0])							
CCH	R6	R6 Port Data Register (Bit[7:0])							
CDH	R6IO	R6 Port Direction Register (Bit[7:0])							
CEH	R7	R7 Port Data Register **(Bit[5:0])**							
CFH	R7IO	R7 Port Direction Register **(Bit[5:0])**							
D0H	TM0	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
D1H	T0/TDR0/ CDR0	Timer0 Register / Timer0 Data Register / Capture0 Data Register							
D2H	TM1	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
D3H	TDR1/ T1PPR	Timer1 Data Register / PWM1 Period Register							
D4H	T1/CDR1/ T1PDR	Timer1 Register / Capture1 Data Register / PWM1 Duty Register							
D5H	PWM1HR	PWM1 High Register**(Bit[3:0])**							
DEH	BUR	BUCK1	BUCK0	BUR5	BUR4	BUR3	BUR2	BUR1	BUR0
E0H	SIOM	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF
E1H	SIOR	SPI DATA REGISTER							
E2H	IENH	INT0E	INT1E	T0E	T1E				
E3H	IENL	ADE	WDTE	BITE	SPIE	-	-	-	-
E4H	IRQH	INT0IF	INT1IF	T0IF	T1IF				
E5H	IRQL	ADIF	WDTIF	BITIF	SPIIF	-	-	-	-
E6H	IEDS					IED1H	IED1L	IED0H	IED0L
EAH	ADCM	-	ADEN	ADS3	ADS2	ADS1	ADS0	ADST	ADSF
EBH	ADCR	ADC Result Data Register							

Table 8-3 Control Registers of GMS81C2020

These registers of shaded area can not be access by bit manipulation instruction as " SET1, CLR1 ", but should be access by register operation instruction as " LDM dp,#imm ".

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ECH	BITR[1]	Basic Interval Timer Data Register							
ECH	CKCTLR[1]	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0
EDH	WDTR	WDTCL	7-bit Watchdog Counter Register						
EFH	PFDR[2]	-	-	-	-	-	PFDIS	PFDM	PFDS
F4H	R0FUNC	-	-	-	-	BUZO	EC0	INT1	INT0
F5H	R4FUNC	-	-	-	-	-	-	-	T0O
F6H	R5FUNC	-	PWM1O/ T1O	-	-	-	-	-	-
F7H	R6FUNC	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
F8H	R7FUNC	-	-	-	-	AN11	AN10	AN9	AN8
F9H	R5NODR	NODR7	NODR6	NODR5	NODR4	NODR3	NODR2	NODR1	NODR0
FAH	SCMR	-	-	-	CS1	CS0	SUBOFF	CLKSEL	MAINOFF
FBH	RA	-	-	-	-	-	-	-	RA0

Table 8-3 Control Registers of GMS81C2020

These registers of shaded area can not be access by bit manipulation instruction as " SET1, CLR1 ", but should be access by register operation instruction as " LDM dp,#imm ".

1.The register BITR and CKCTLR are located at same address. Address ECH is read as BITR, written to CKCTLR.
2.The register PFDR only be implemented on devices, not on In-circuit Emulator.

8.4 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

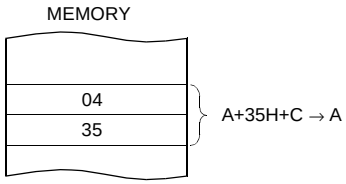
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

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

Example:

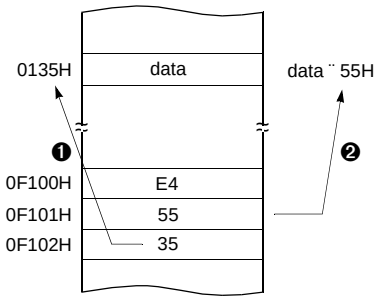
```
0435     ADC    #35H
```



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

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

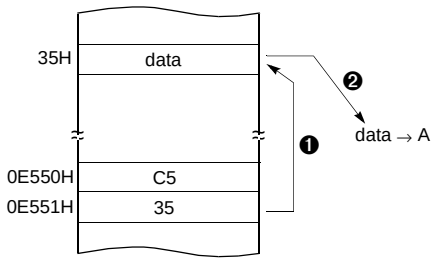


(3) Direct Page Addressing → dp

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

Example; G=0

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



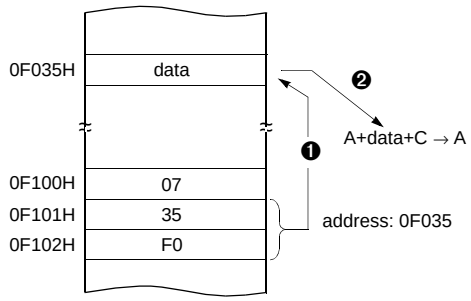
(4) Absolute Addressing → !abs

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

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

Example;

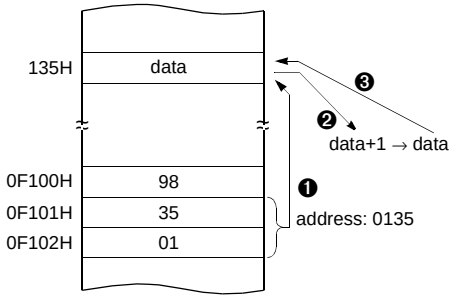
```
0735F0   ADC    !0F035H      ;A ←ROM[0F035H]
```



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

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

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



(5) Indexed Addressing

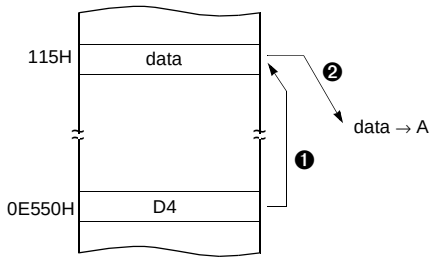
X indexed direct page (no offset) → {X}

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

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

Example; X=15_H, G=1

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



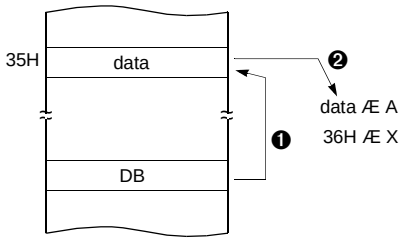
X indexed direct page, auto increment→ {X}+

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

LDA, STA

Example; G=0, X=35_H

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



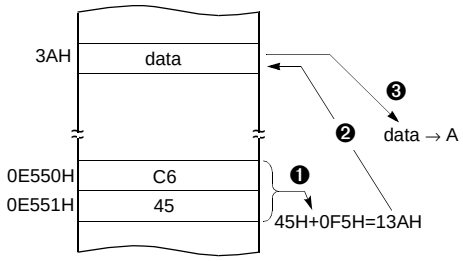
X indexed direct page (8 bit offset) → dp+X

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

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

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

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



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

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

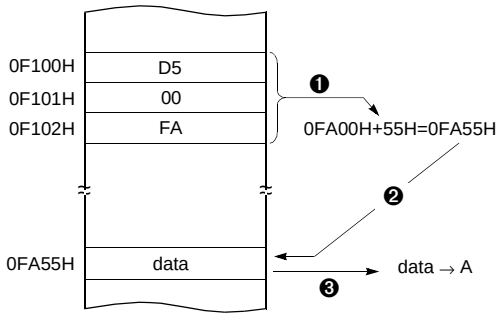
This is same with above (2). Use Y register instead of X.

Y indexed absolute → !abs+Y

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

Example; Y=55_H

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



(6) Indirect Addressing

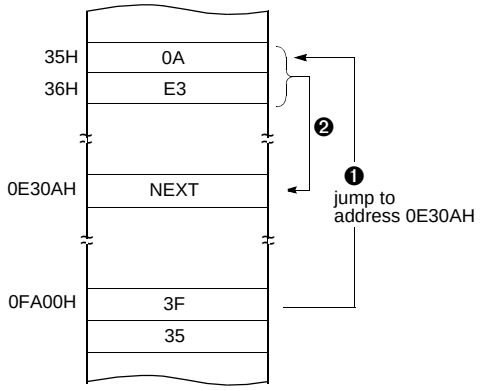
Direct page indirect → [dp]

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

JMP, CALL

Example; G=0

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



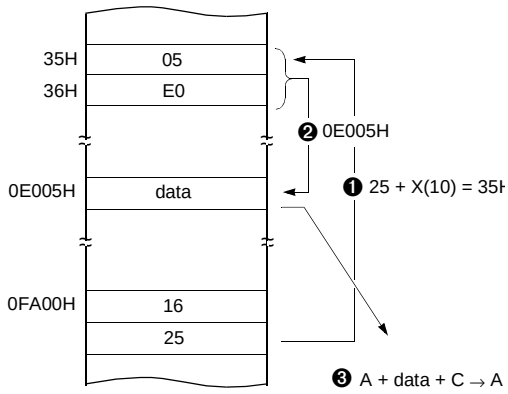
X indexed indirect → [dp+X]

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

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

Example; G=0, X=10_H

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



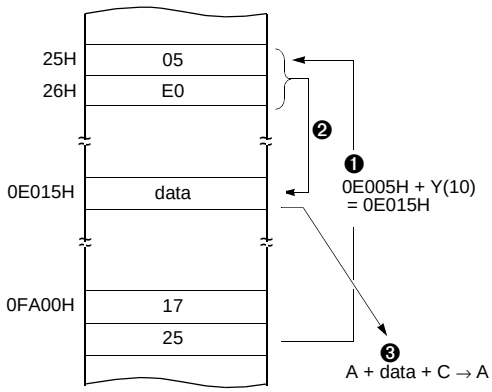
Y indexed indirect → [dp]+Y

Processes 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=10_H

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



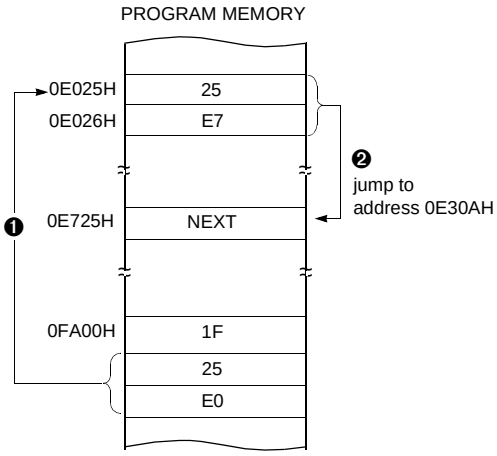
Absolute indirect → [!abs]

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

JMP

Example; G=0

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



9. I/O PORTS

The GMS81C20xx has eight ports (R0, R1, R2, R4, R5, R6 and R7).These ports pins may be multiplexed with an alternate function for the peripheral features on the device.

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 $0C1_H$ (R0 port direction register) during initial setting as shown in Figure 9-1.

All the port direction registers in the GMS81C2020 have 0 written to them by reset function. On the other hand, its initial status is input.

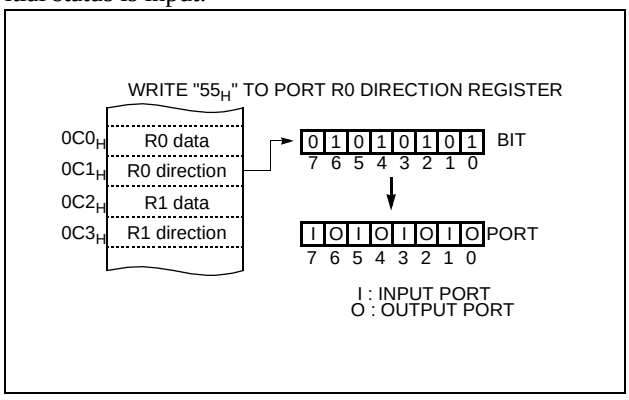
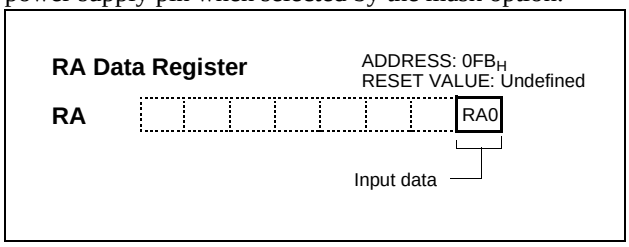


Figure 9-1 Example of Port I/O Assignment

RA(Vdisp) register: RA is one-bit high-voltage input only port pin. In addition, RA serves the functions of the V_{disp} special features. V_{disp} is used as a high-voltage input power supply pin when selected by the mask option.

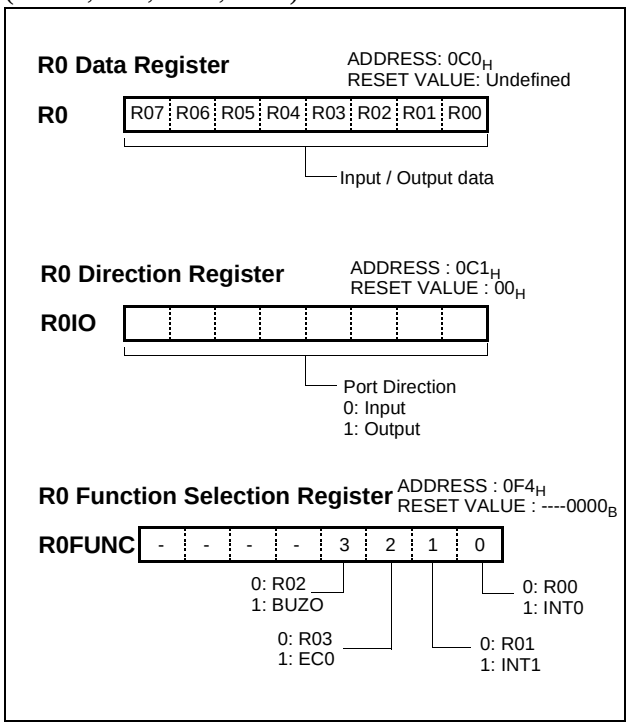


Port pin	Alternate function
RA	V_{disp} (High-voltage input power supply)

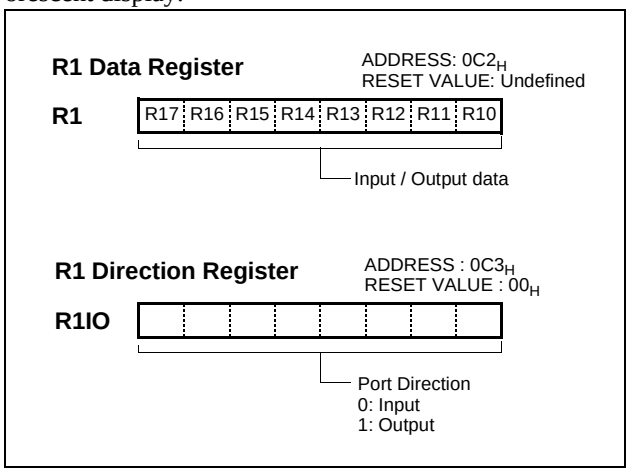
R0 and R0IO register: R0 is an 8-bit high-voltage CMOS bidirectional I/O port (address $0C0_H$). Each port can be set individually as input and output through the R0IO register (address $0C1_H$). Each port can directly drive a vacuum fluorescent display. R03 port is multiplexed with Buzzer Output Port(BUZO), R02 port is multiplexed with Event Counter Input Port (EC0), and R01~R00 are multiplexed with External Interrupt Input Port(INT1, INT0)

Port Pin	Alternate Function
R00	INT0 (External interrupt 0 Input Port)
R01	INT1 (External interrupt 1 Input Port)
R02	EC0 (Event Counter Input Port)
R03	BUZO (Buzzer Output Port)

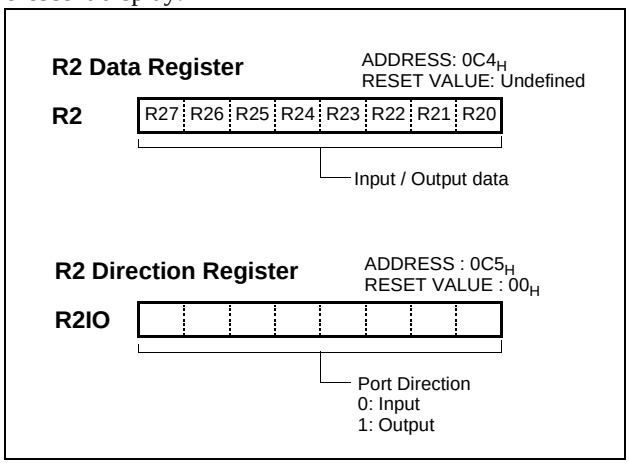
.The control register R0FUNC (address $F4_H$) controls to select alternate function. After reset, this value is "0", port may be used as general I/O ports. To select alternate function such as Buzzer Output, External Event Counter Input and External Interrupt Input, write "1" to the corresponding bit of R0FUNC. Regardless of the direction register R0IO, R0FUNC is selected to use as alternate functions, port pin can be used as a corresponding alternate features (BUZO, EC0, INT1, INT0)



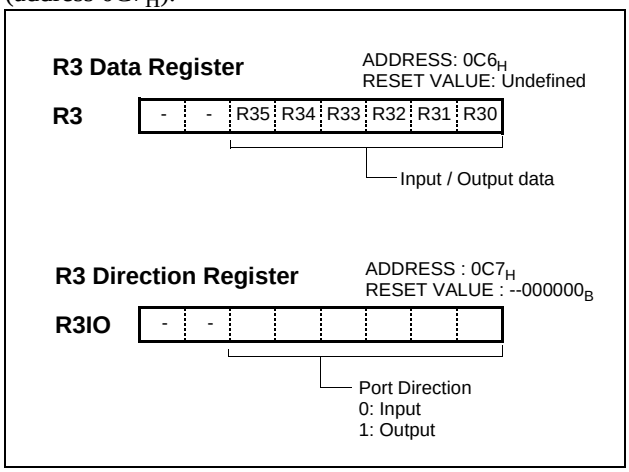
R1 and R0IO register: R1 is an 8-bit high-voltage CMOS bidirectional I/O port (address $0C2_H$). Each port can be set individually as input and output through the R1IO register (address $0C3_H$). Each port can directly drive a vacuum fluorescent display.



R2 and R2IO register: R2 is an 8-bit high-voltage CMOS bidirectional I/O port (address $0C4_H$). Each port can be set individually as input and output through the R2IO register (address $0C5_H$). Each port can directly drive a vacuum fluorescent display.



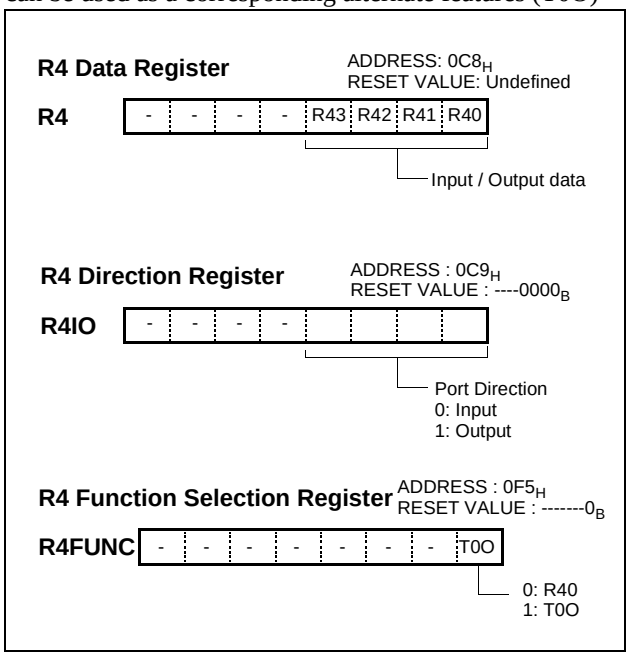
R3 and R3IO register: R3 is a 6-bit high-voltage CMOS bidirectional I/O port (address $0C6_H$). Each port can be set individually as input and output through the R3IO register (address $0C7_H$).



R4 and R4IO register: R4 is a 4-bit bidirectional I/O port (address $0C8_H$). Each port can be set individually as input and output through the R4IO register (address $0C9_H$). R40 port is multiplexed with Timer 0 Output Port (T0O).

Port Pin	Alternate Function
R40	T0O (Timer 0 Compare Output Port)

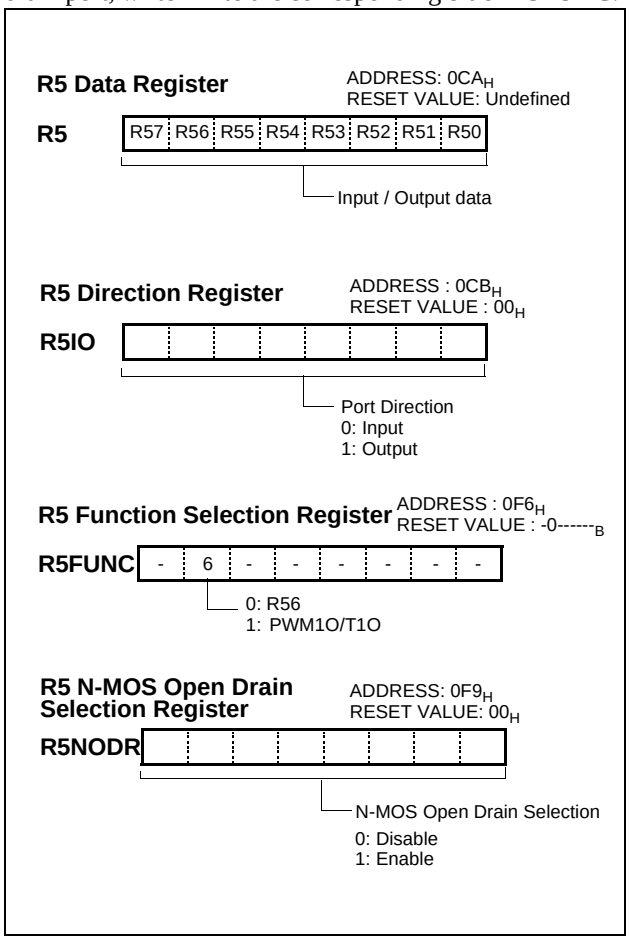
The control register R4FUNC (address $0F5_H$) controls to select alternate function. After reset, this value is "0", port may be used as general I/O ports. To select alternate function such as Timer 0 Output, write "1" to the corresponding bit of R4FUNC. Regardless of the direction register R4IO, R4FUNC is selected to use as alternate functions, port pin can be used as a corresponding alternate features (T0O)



R5 and R5IO register: R5 is an 8-bit bidirectional I/O port (address $0CA_H$). Each pin can be set individually as input and output through the R5IO register (address $0CB_H$).In addition, Port R5 is multiplexed with Pulse Width Modulator (PWM).

Port Pin	Alternate Function
R56	PWM1 Data Output Timer 1 Data Output

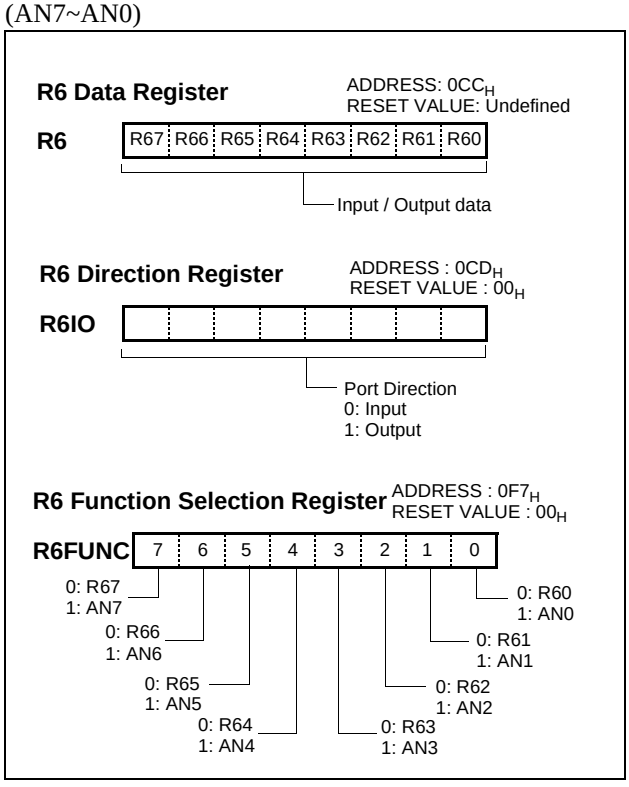
The control register R5FUNC (address $0F6_H$) controls to select PWM function.After reset, the R5IO register value is "0", port may be used as general I/O ports. To select PWM function, write "1" to the corresponding bit of R5FUNC.
The control register R5NODR (address $0F9_H$) controls to select N-MOS open drain port. To select N-MOS open drain port, write "1" to the corresponding bit of R5FUNC.



R6 and R6IO register: R6 is an 8-bit bidirectional I/O port (address $0CC_H$). Each port can be set individually as input and output through the R6IO register (address $0CD_H$). R67~R60 ports are multiplexed with Analog Input Port.

Port Pin	Alternate Function
R60	AN0 (ADC input 0)
R61	AN1 (ADC input 1)
R62	AN2 (ADC input 2)
R63	AN3 (ADC input 3)
R64	AN4 (ADC input 4)
R65	AN5 (ADC input 5)
R66	AN6 (ADC input 6)
R67	AN7 (ADC input 7)

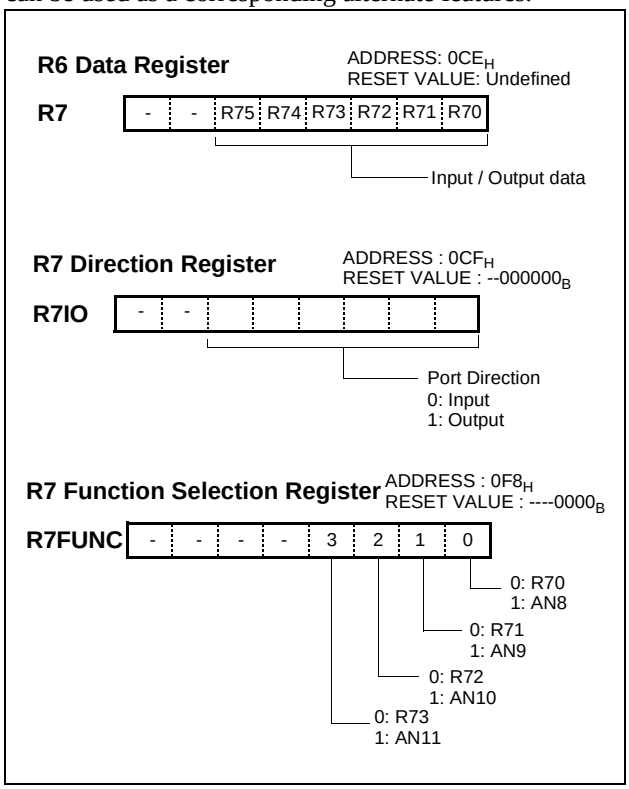
The control register R6FUNC (address $0F7_H$) controls to select alternate function. After reset, this value is "0", port may be used as general I/O ports. To select alternate function such as Analog Input, write "1" to the corresponding bit of R6FUNC. Regardless of the direction register R6IO, R6FUNC is selected to use as alternate functions, port pin can be used as a corresponding alternate features (AN7~AN0)



R7 and R7IO register: R7 is a 4-bit bidirectional I/O port (address $0CE_H$). Each port can be set individually as input and output through the R7IO register (address $0CF_H$). R73~R70 ports are multiplexed with Analog Input Port AN8~AN11). R74, R75 ports are alternate function of SXI, SXO ports. R74, R75 ports can be set individually as input and output through the R7IO register.

Port Pin	Alternate Function
R70	AN8 (ADC input 8)
R71	AN9 (ADC input 9)
R72	AN10 (ADC input 10)
R73	AN11 (ADC input 11)
SXI	R74(included Internal Pull-up Resister)
SXO	R75(included Internal Pull-up Resister)

The control register R7FUNC (address $0F8_H$) controls to select alternate function. After reset, this value is "0", port may be used as general I/O ports. To select alternate function such as Analog Input, write "1" to the corresponding bit of R7FUNC. Regardless of the direction register R7IO, R7FUNC is selected to use as alternate functions, port pin can be used as a corresponding alternate features.



10. BASIC INTERVAL TIMER

The GMS81C20xx has one 8-bit Basic Interval Timer that is free-run, can not stop. Block diagram is shown in Figure 10-1. In addition, the Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt (BITIF).

The 8-bit Basic interval timer register (BITR) is increased every internal count pulse which is divided by prescaler. Since prescaler has divided ratio by 8 to 1024, the count rate is 1/8 to 1/1024 of the oscillator frequency. As the count overflows from FF_H to 00_H, this overflow causes to generate the Basic interval timer interrupt. The BITIF is interrupt request flag of Basic interval timer. The Basic Interval Timer is controlled by the clock control register (CKCTLR) shown in Figure 10-2.

When write "1" to bit BTCL of CKCTLR, BITR register is cleared to "0" and restart to count-up. The bit BTCL becomes "0" after one machine cycle by hardware.

If the STOP instruction executed after writing "1" to bit WAKEUP of CKCTLR, it goes into the wake-up timer mode. In this mode, all of the block is halted except the oscillator, prescaler (only fXIN÷2048) and Timer0.

If the STOP instruction executed after writing "1" to bit RCWDT of CKCTLR, it goes into the internal RC oscillated watchdog timer mode. In this mode, all of the block is halted except the internal RC oscillator, Basic Interval Timer and Watchdog Timer. More detail informations are explained in Power Saving Function. The bit WDTON decides Watchdog Timer or the normal 7-bit timer.

Source clock can be selected by lower 3 bits of CKCTLR. BITR and CKCTLR are located at same address, and address $0EC_H$ is read as a BITR, and written to CKCTLR.

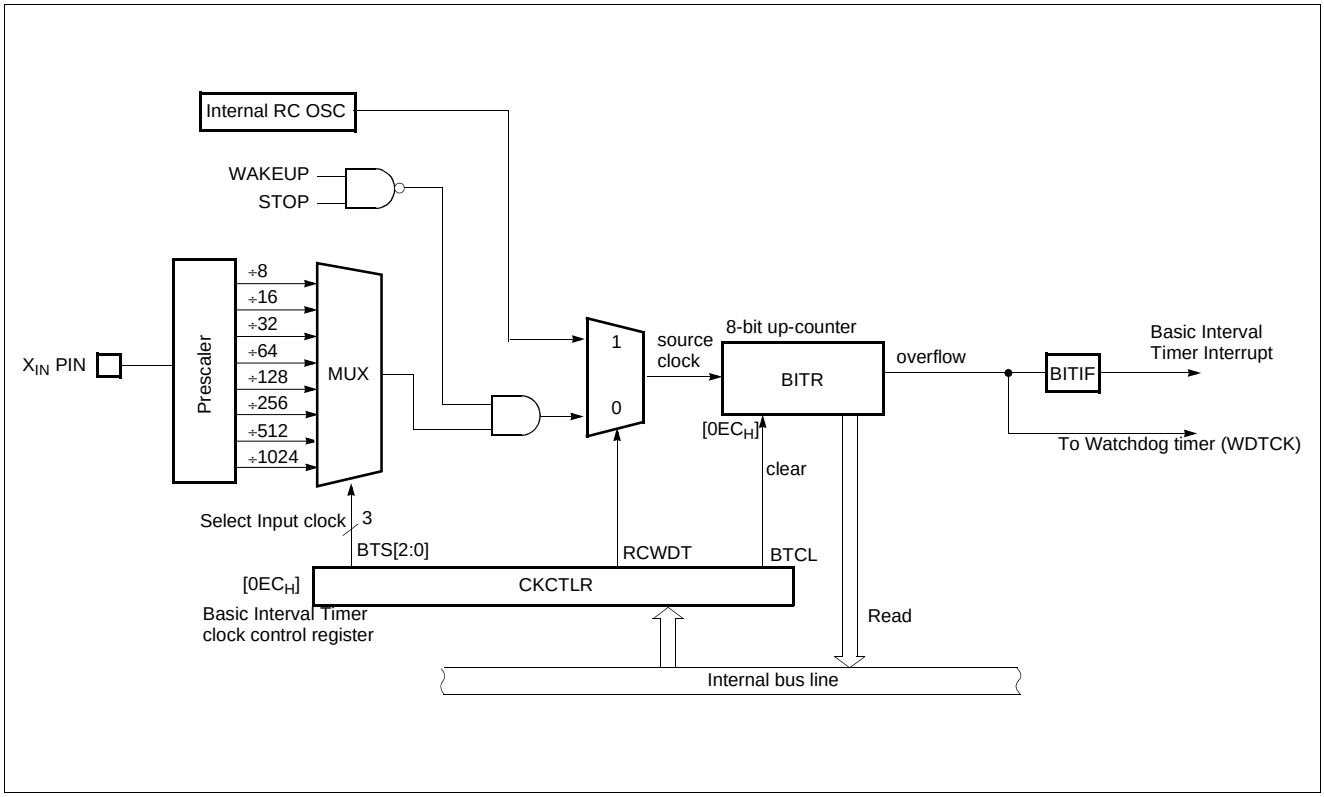


Figure 10-1 Block Diagram of Basic Interval Timer

CKCTLR [2:0]	Source clock	Interrupt (overflow) Period (ms) @ f_{XIN} = 4MHz
000	f_{XIN}÷8	0.512
001	f_{XIN}÷16	1.024
010	f_{XIN}÷32	2.048
011	f_{XIN}÷64	4.096
100	f_{XIN}÷128	8.192
101	f_{XIN}÷256	16.384
110	f_{XIN}÷512	32.768
111	f_{XIN}÷1024	65.536

Table 10-1 Basic Interval Timer Interrupt Time

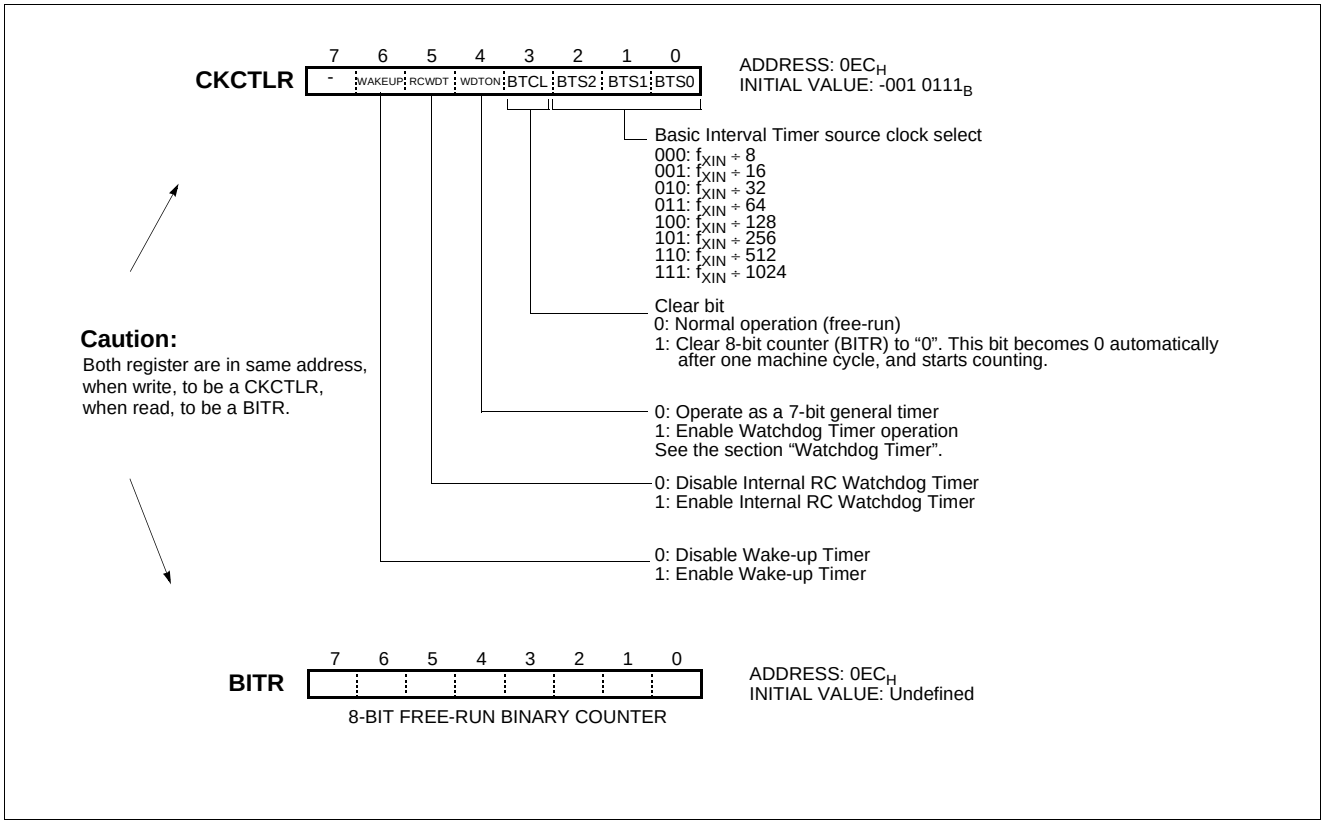


Figure 10-2 BITR: Basic Interval Timer Mode Register

Example 1:

Basic Interval Timer Interrupt request flag is generated every 4.096ms at 4MHz.

```
:
LDM    CKCTLR,#03H
SET1   BITE
EI
:
```

Example 2:

Basic Interval Timer Interrupt request flag is generated every 1.024ms at 4MHz.

```
:
LDM    CKCTLR,#01H
SET1   BITE
EI
:
```

11. WATCHDOG TIMER

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

When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. The purpose of the watchdog timer is to detect the malfunction (runaway) of program due to external noise or other causes and return the operation to the normal condition.

The watchdog timer has two types of clock source.

The first type is an on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the external oscillator of the Xin pin. It means that the watchdog timer will run, even if the clock on the Xin pin of the device has been stopped, for example, by entering the STOP mode.

The other type is a prescaled system clock.

The watchdog timer consists of 7-bit binary counter and the watchdog timer data register. When the value of 7-bit binary counter is equal to the lower 7 bits of WDTR, the interrupt request flag is generated. This can be used as WDT interrupt or reset the CPU in accordance with the bit WDTON.

Note: *Because the watchdog timer counter is enabled after clearing Basic Interval Timer, after the bit WDTON set to "1", maximum error of timer is depend on prescaler ratio of Basic Interval Timer. The 7-bit binary counter is cleared by setting WDTCL(bit7 of WDTR) and the WDTCL is cleared automatically after 1 machine cycle.*

The RC oscillated watchdog timer is activated by setting the bit RCWDT as shown below.

```
LDM     CKCTLR,#3FH; enable the RC-osc WDT
LDM     WDTR,#0FFH; set the WDT period
STOP            ; enter the STOP mode
NOP
NOP             ; RC-osc WDT running
  :
```

The RCWDT oscillation period is vary with temperature, VDD and process variations from part to part (approximately, 40~120uS). The following equation shows the RCWDT oscillated watchdog timer time-out.

$$T_{RCWDT} = CLK_{RCWDT} \times 2^8 \times [WDTR.6\sim0] + (CLK_{RCWDT} \times 2^8)/2$$

$$where, CLK_{RCWDT} = 40\sim120uS$$

In addition, this watchdog timer can be used as a simple 7-bit timer by interrupt WDTIF. The interval of watchdog timer interrupt is decided by Basic Interval Timer. Interval equation is as below.

$$T_{WDT} = [WDTR.6\sim0] \times Interval\ of\ BIT$$

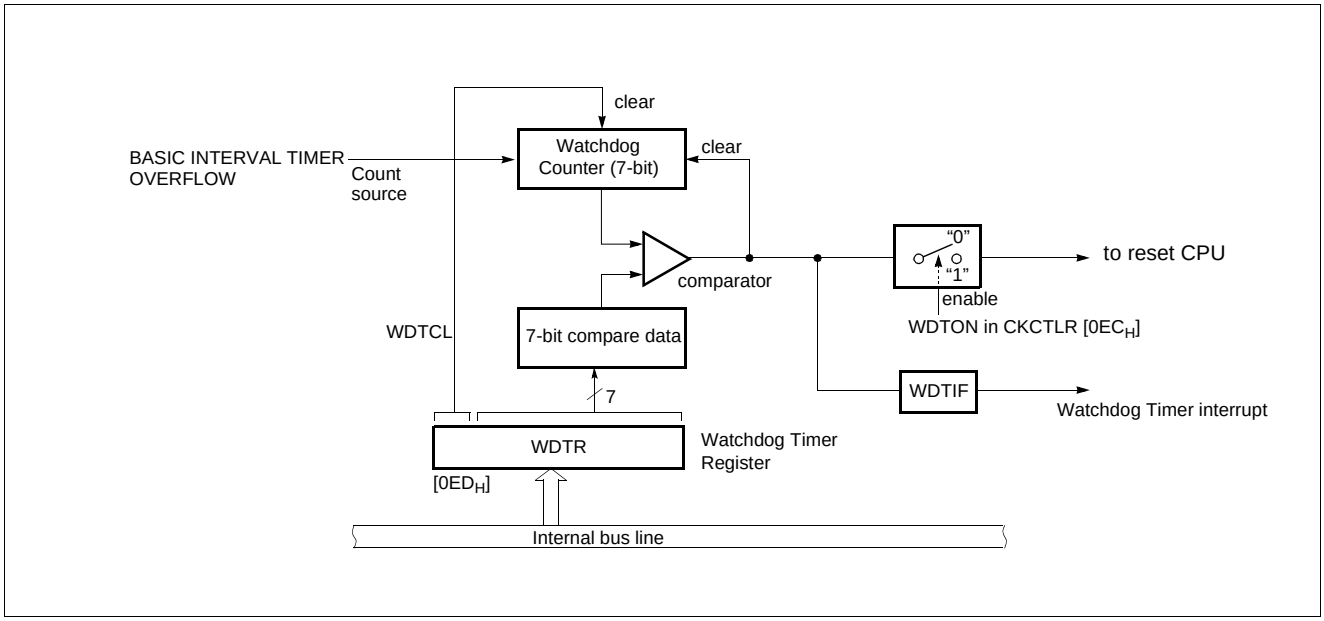


Figure 11-1 Block Diagram of Watchdog Timer

Watchdog Timer Control

Figure 11-2 shows the watchdog timer control register. The watchdog timer is automatically disabled after reset.

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

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

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

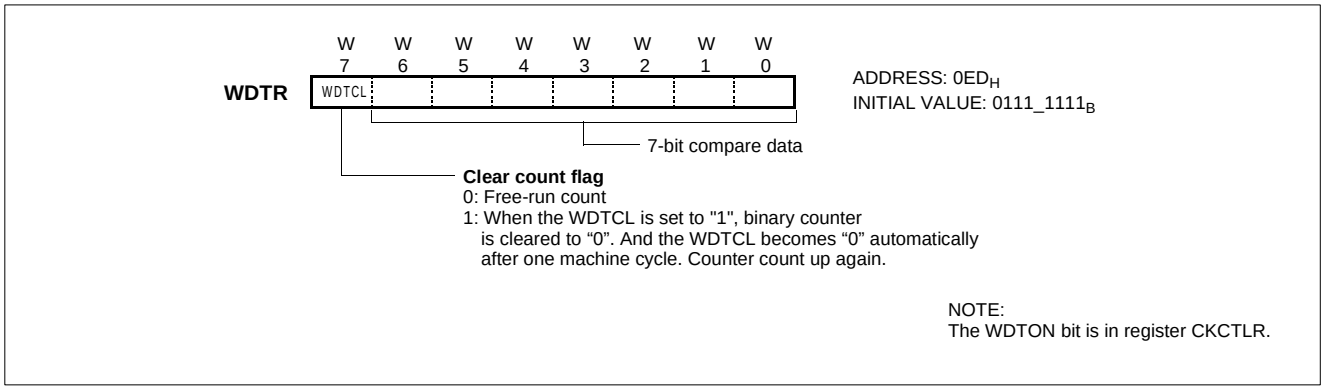


Figure 11-2 WDTR: Watchdog Timer Data Register

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

```
        LDM     CKCTLR,#3FH     ;Select 1/2048 clock source, WDTON ← 1, Clear Counter
        LDM     WDTR,#04FH
        LDM     WDTR,#04FH      ;Clear counter
        :                       Within WDT
        :                       detection time
        :
        LDM     WDTR,#04FH      ;Clear counter
        :                       Within WDT
        :                       detection time
        :
        LDM     WDTR,#04FH      ;Clear counter
```

Enable and Disable Watchdog

Watchdog timer is enabled by setting WDTON (bit 4 in CKCTLR) to “1”. WDTON is initialized to “0” during reset and it should be set to “1” to operate after reset is released.

Example: Enables watchdog timer for Reset

```
:
LDM    CKCTLR,#xx1x_xxxxB;WDTON ← 1
:
:
```

The watchdog timer is disabled by clearing bit 5 (WDTON) of CKCTLR. The watchdog timer is halted in STOP mode and restarts automatically after STOP mode is released.

Watchdog Timer Interrupt

The watchdog timer can be also used as a simple 7-bit timer by clearing bit5 of CKCTLR to “0”. The interval of watchdog timer interrupt is decided by Basic Interval Timer. Interval equation is shown as below.

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

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

Example: 7-bit timer interrupt set up.

```
LDM    CKCTLR,#xx0xxxxxB;WDTON ←0
LDM    WDTR,#7FH    ;WDTCL ←1

:
```

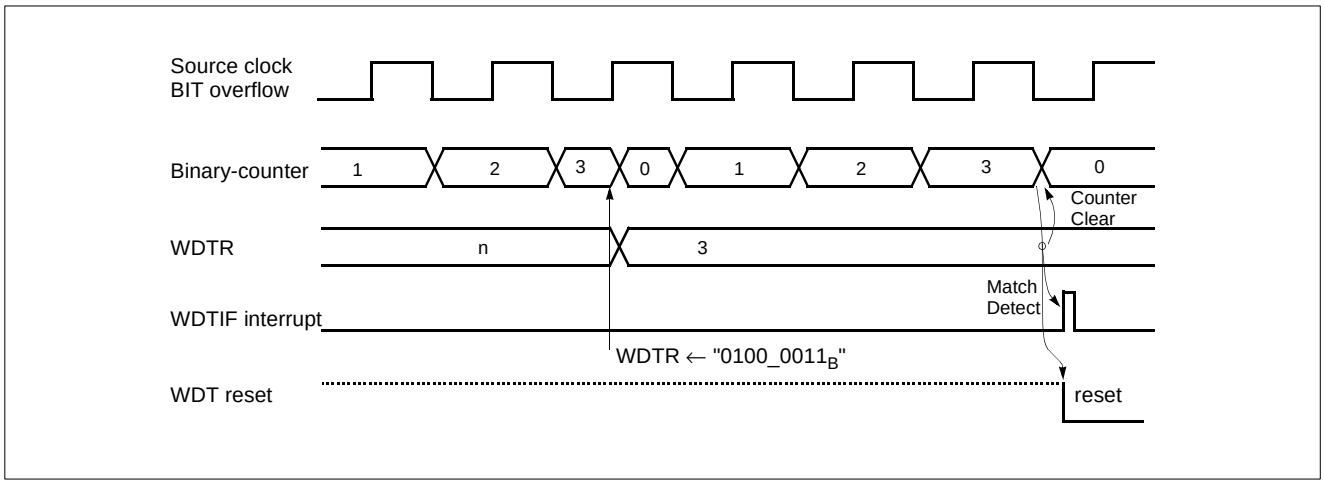


Figure 11-3 Watchdog timer Timing

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

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

12. TIMER/EVENT COUNTER

The GMS81C20xx has two Timer/Counter registers. Each module can generate an interrupt to indicate that an event has occurred (i.e. timer match).

Timer 0 and Timer 1 are can be used either two 8-bit Timer/Counter or one 16-bit Timer/Counter with combine them.

In the "timer" function, the register is increased every internal clock input. Thus, one can think of it as counting internal clock input. Since a least clock consists of 2 and most clock consists of 2048 oscillator periods, the count rate is 1/2 to 1/2048 of the oscillator frequency in Timer0. And Timer1 can use the same clock source too. In addition, Timer1 has more fast clock source (1/1 to 1/8).

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

In addition the “capture” function, the register is increased 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 CDRx.

Timer1 is shared with "PWM" function and "Compare output" function

It has seven operating modes: "8-bit timer/counter", "16-bit timer/counter", "8-bit capture", "16-bit capture", "8-bit compare output", "16-bit compare output" and "10-bit PWM" which are selected by bit in Timer mode register TM0 and TM1 as shown in Figure 12-1 and Table 12-1.

16BIT	CAP0	CAP1	PWM1E	T0CK [2:0]	T1CK [1:0]	PWM1O	TIMER 0	TIMER 1
0	0	0	0	XXX	XX	0	8-bit Timer	8-bit Timer
0	0	1	0	111	XX	0	8-bit Event counter	8-bit Capture
0	1	0	0	XXX	XX	1	8-bit Capture (internal clock)	8-bit Compare Output
0	X	0	1	XXX	XX	1	8-bit Timer/Counter	10-bit PWM
1	0	0	0	XXX	11	0	16-bit Timer	
1	0	0	0	111	11	0	16-bit Event counter	
1	1	X	0	XXX	11	0	16-bit Capture (internal clock)	
1	0	0	0	XXX	11	1	16-bit Compare Output	

Table 12-1 Operating Modes of Timer0 and Timer1

TM0

R/W 5	R/W 4	R/W 3	R/W 2	R/W 1	R/W 0
CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST

ADDRESS: $0D0_H$
INITIAL VALUE: $--000000_B$

Bit Name	Bit Position	Description
CAP0	TM0.5	0: Timer/Counter mode 1: Capture mode selection flag
T0CK2 T0CK1 T0CK0	TM0.4 TM0.3 TM0.2	000: 8-bit Timer, Clock source is $f_{XIN} \div 2$ 001: 8-bit Timer, Clock source is $f_{XIN} \div 4$ 010: 8-bit Timer, Clock source is $f_{XIN} \div 8$ 011: 8-bit Timer, Clock source is $f_{XIN} \div 32$ 100: 8-bit Timer, Clock source is $f_{XIN} \div 128$ 101: 8-bit Timer, Clock source is $f_{XIN} \div 512$ 110: 8-bit Timer, Clock source is $f_{XIN} \div 2048$ 111: $\overline{EC0}$ (External clock)
T0CN	TM0.1	0: Stop the timer 1: A logic 1 starts the timer.
T0ST	TM0.0	0: When cleared, stop the counting. 1: When set, Timer 0 Count Register is cleared and start again.

TM1

R/W 7	R/W 6	R/W 5	R/W 4	R/W 3	R/W 2	R/W 1	R/W 0
POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST

ADDRESS: $0D2_H$
INITIAL VALUE: 00_H

Bit Name	Bit Position	Description
POL	TM1.7	0: PWM Duty Active Low 1: PWM Duty Active High
16BIT	TM1.6	0: 8-bit Mode 1: 16-bit Mode
PWMIE	TM1.5	0: Disable PWM 1: Enable PWM
CAP1	TM1.4	0: Timer/Counter mode 1: Capture mode selection flag
T1CK1 T1CK0	TM1.3 TM1.2	00: 8-bit Timer, Clock source is f_{XIN} 01: 8-bit Timer, Clock source is $f_{XIN} \div 2$ 10: 8-bit Timer, Clock source is $f_{XIN} \div 8$ 11: 8-bit Timer, Clock source is Using the the Timer 0 Clock
T0CN	TM1.1	0: Stop the timer 1: A logic 1 starts the timer.
T0ST	TM1.0	0: When cleared, stop the counting. 1: When set, Timer 0 Count Register is cleared and start again.

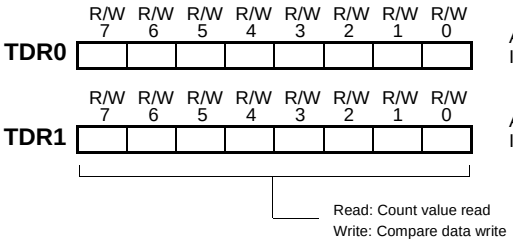


Figure 12-1 TM0, TM1 Registers

12.1 8-bit Timer / Counter Mode

The GMS81C20xx has two 8-bit Timer/Counters, Timer 0, Timer 1 as shown in Figure 12-2.

The "timer" or "counter" function is selected by mode registers TMx as shown in Figure 12-1 and Table 12-1. To use as an 8-bit timer/counter mode, bit CAP0 of TM0 is cleared to "0" and bits 16BIT of TM1 should be cleared to “0”(Table 12-1).

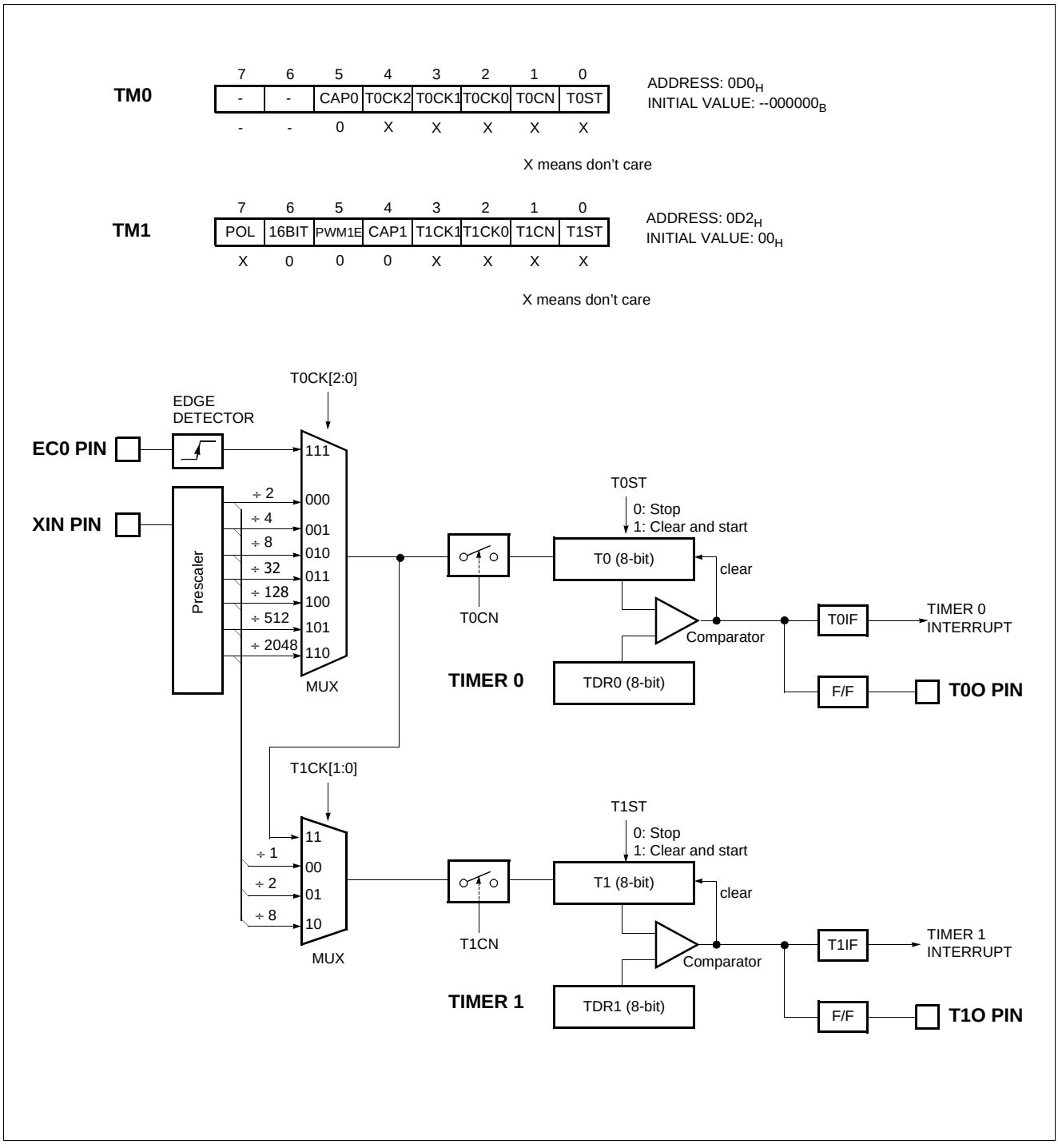


Figure 12-2 8-bit Timer/Counter 0, 1

Example 1:

Timer0 = 2ms 8-bit timer mode at 4MHz
Timer1 = 0.5ms 8-bit timer mode at 4MHz

```
LDM    TDR0,#250
LDM    TDR1,#250
LDM    TM0,#0000_1111B
LDM    TM1,#0000_1011B
SET1   T0E
SET1   T1E
EI
```

Example 2:

Timer0 = 8-bit event counter mode
Timer1 = 0.5ms 8-bit timer mode at 4MHz

```
LDM    TDR0,#250
LDM    TDR1,#250
LDM    TM0,#0001_1111B
LDM    TM1,#0000_1011B
SET1   T0E
SET1   T1E
EI
```

Note: *The contents of Timer data register TDRx should be initialized 1_H~FF_H, not 0_H, because it is undefined after reset.*

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, 4, 8, 32,128, 512, 2048 selected by control bits T0CK[2:0] of register (TM0) and 1, 2, 8 selected by control bits T1CK[1:0] of register (TM1). In the Timer 0, timer register T0 increases from 00_H until it matches TDR0 and then reset to 00_H. 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(1-to-0) (rising & falling edge) transition of EC0 pin. In order to use counter function, the bit EC0 of the R0 Function Selection Register (R0FUNC.2) is set to "1". The Timer 0 can be used as a counter by pin EC0 input, but Timer 1 can not.

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 TDR*n* are compared with the contents of up-counter, T*n*. 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 TDR*n* is changeable by software, time interval is set as you want

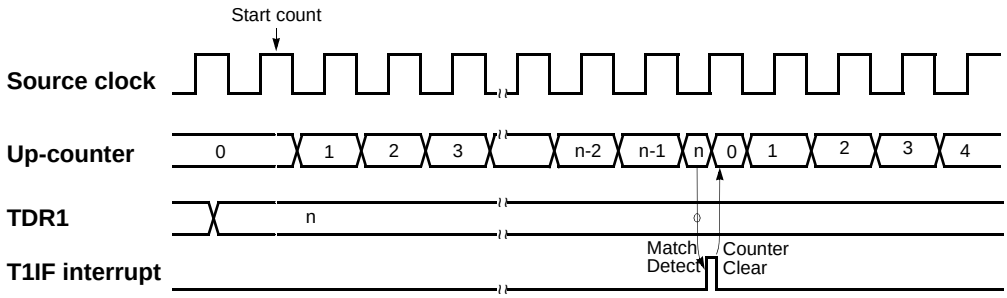


Figure 12-3 Timer Mode Timing Chart

Example: Make 2ms interrupt using by Timer0 at 4MHz

```
LDM   TM0,#0FH    ; divide by 32
LDM   TDR0,#125   ; 8us x 125= 1ms
SET1  T0E         ; Enable Timer 0 Interrupt
EI                ; Enable Master Interrupt
```

When
- TM0 = 0000 1111_B (8-bit Timer mode, Prescaler divide ratio = 32)
- TDR0 = $125_D = 7D_H$
- f_{XIN} = 4 MHz

$$\text{INTERRUPT PERIOD} = \frac{1}{4 \times 10^6 \text{ Hz}} \times 32 \times 125 = 1 \text{ ms}$$

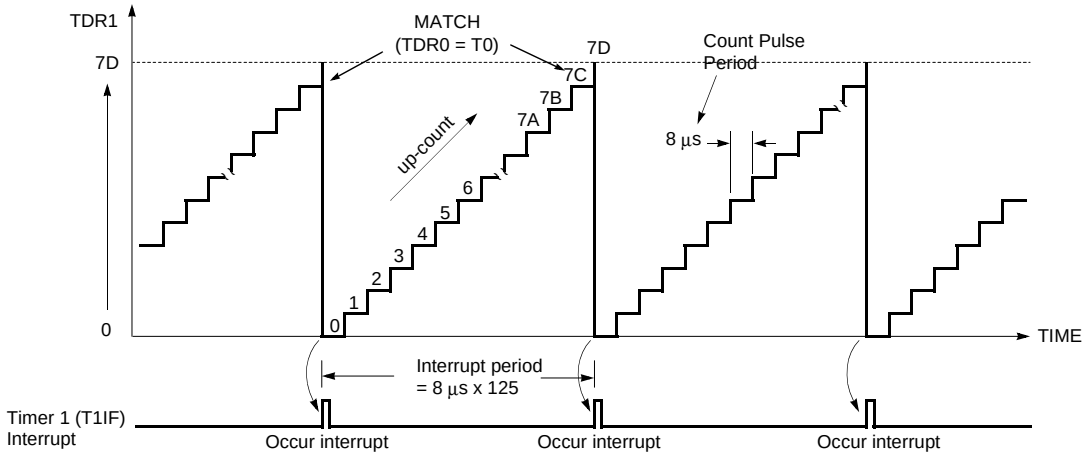


Figure 12-4 Timer Count Example

8-bit Event Counter Mode

In this mode, counting up is started by an external trigger. This trigger means falling edge or rising edge of the EC0 pin input. Source clock is used as an internal clock selected with timer mode register TM0. The contents of timer data register TDR0 is compared with the contents of the up-counter T0. If a match is found, an timer interrupt request flag T0IF is generated, and the counter is cleared to “0”. The counter is restart and count up continuously by every falling edge or rising edge of the EC0 pin input.

The maximum frequency applied to the EC0 pin is $f_{XIN}/2$ [Hz].

In order to use event counter function, the bit 2 of the R5 function register (R5FUNC.2) is required to be set to “1”.

After reset, the value of timer data register TDR0 is undefined, it should be initialized to between 1_H~FF_H, not to "0"The interval period of Timer is calculated as below equation.

$$Period\ (sec) = \frac{1}{f_{XIN}} \times 2 \times Divide\ Ratio \times TDR0$$

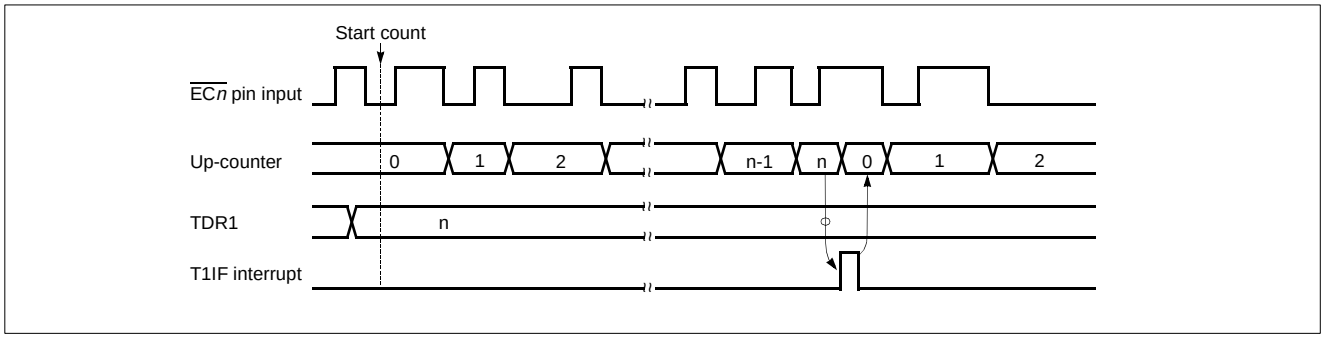


Figure 12-5 Event Counter Mode Timing Chart

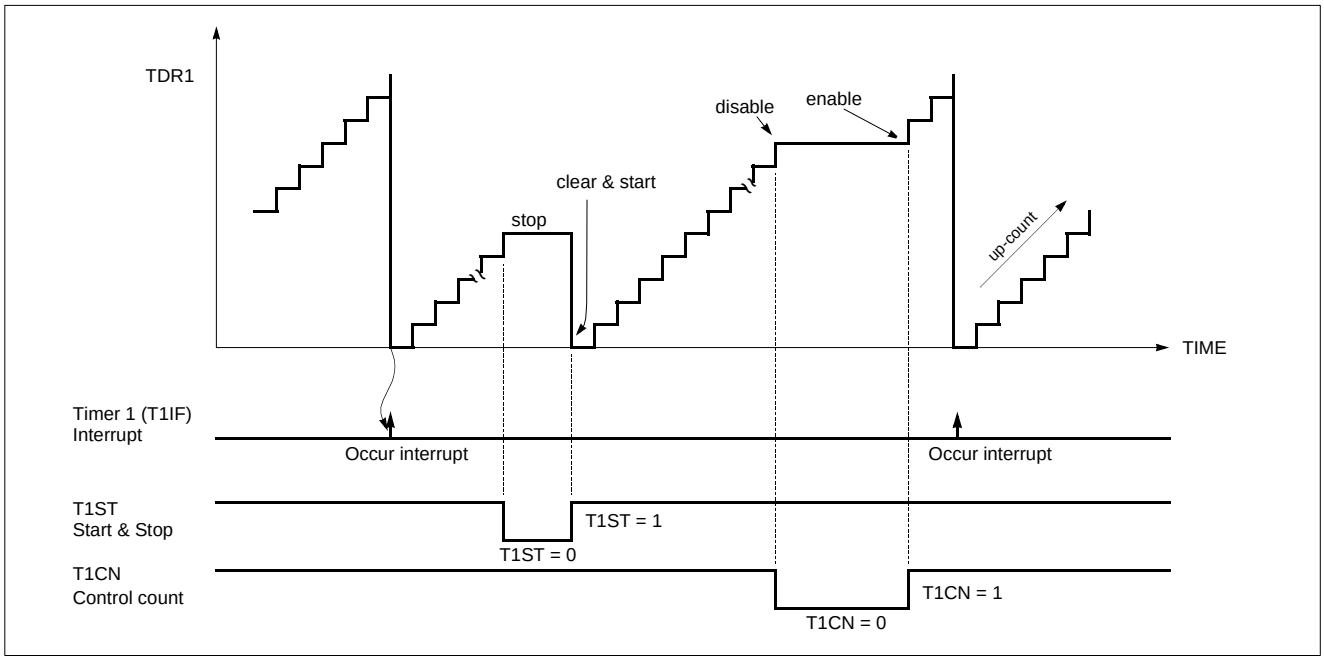


Figure 12-6 Count Operation of Timer / Event counter

12.2 16-bit Timer / Counter Mode

The Timer register is being run with 16 bits. A 16-bit timer/ counter register T0, T1 are increased from 0000_H until it matches TDR0, TDR1 and then resets to 0000_H. The match output generates Timer 0 interrupt not Timer 1 interrupt.

The clock source of the Timer 0 is selected either internal or external clock by bit T0CK[2:0].

In 16-bit mode, the bits T1CK[1:0] and 16BIT of TM1 should be set to "1" respectively.

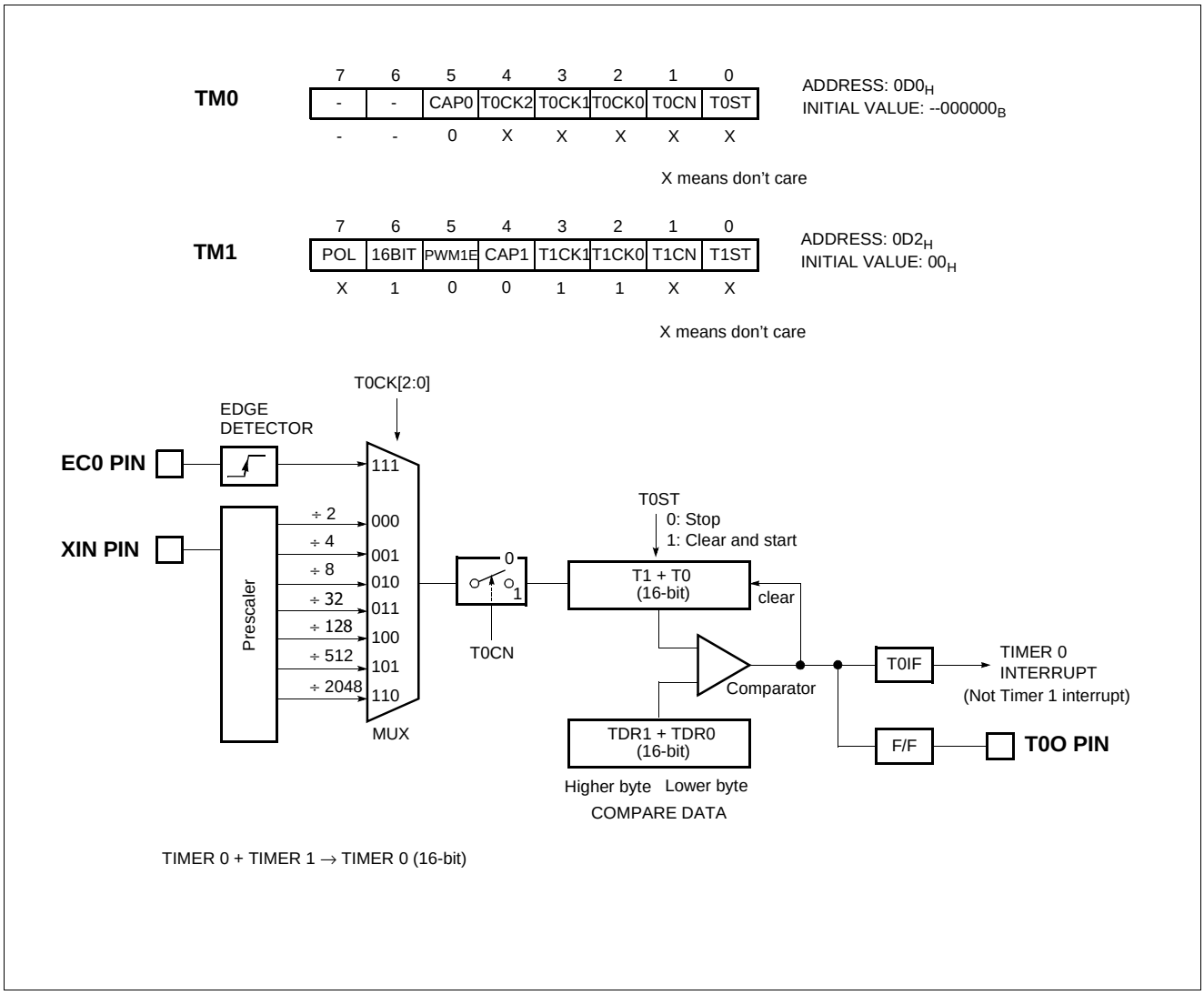


Figure 12-7 16-bit Timer/Counter

12.3 8-bit Compare Output (16-bit)

The GMS81C20xx has a function of Timer Compare Output. To pulse out, the timer match can goes to port pin(T0O, T1O) as shown in Figure 12-2 and Figure 12-7. Thus, pulse out is generated by the timer match. These operation is implemented to pin, T0O, PWM1O/T1O.

In this mode, the bit PWM1O/T1O of R5 function register (R5FUNC.6) should be set to "1", and the bit PWM1E of timer1 mode register (TM1) should be set to "0". In addition, 16-bit Compare output mode is available, also.

This pin output the signal having a 50 : 50 duty square wave, and output frequency is same as below equation.

$$f_{COMP} = \frac{Oscillation\ Frequency}{2 \times Prescaler\ Value \times (TDR + 1)}$$

12.4 8-bit Capture Mode

The Timer 0 capture mode is set by bit CAP0 of timer mode register TM0 (bit CAP1 of timer mode register TM1 for Timer 1) as shown in Figure 12-8.

As mentioned above, not only Timer 0 but Timer 1 can also be used as a capture mode.

The Timer/Counter register is increased in response internal or external input. This counting function is same with normal timer mode, and Timer interrupt is generated when timer register T0 (T1) increases and matches TDR0 (TDR1).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is more wider than the maximum period of Timer.

For example, in Figure 12-10, the pulse width of captured signal is wider than the timer data value (FF_H) over 2 times. When external interrupt is occurred, the captured value (13_H) is more little than wanted value. It can be obtained correct value by counting the number of timer overflow occurrence.

Timer/Counter still does the above, but with the added feature that a edge transition at external input INTx pin causes the current value in the Timer x register (T0,T1), to be captured into registers CDRx (CDR0, CDR1), respectively. After captured, Timer x register is cleared and restarts by hardware.

Note: *The CDRx, TDRx and Tx are in same address. In the capture mode, reading operation is read the CDRx, not Tx because path is opened to the CDRx, and TDRx is only for writing operation.*

It has three transition modes: "falling edge", "rising edge", "both edge" which are selected by interrupt edge selection register IEDS (Refer to External interrupt section). In addition, the transition at INTx pin generate an interrupt.

.

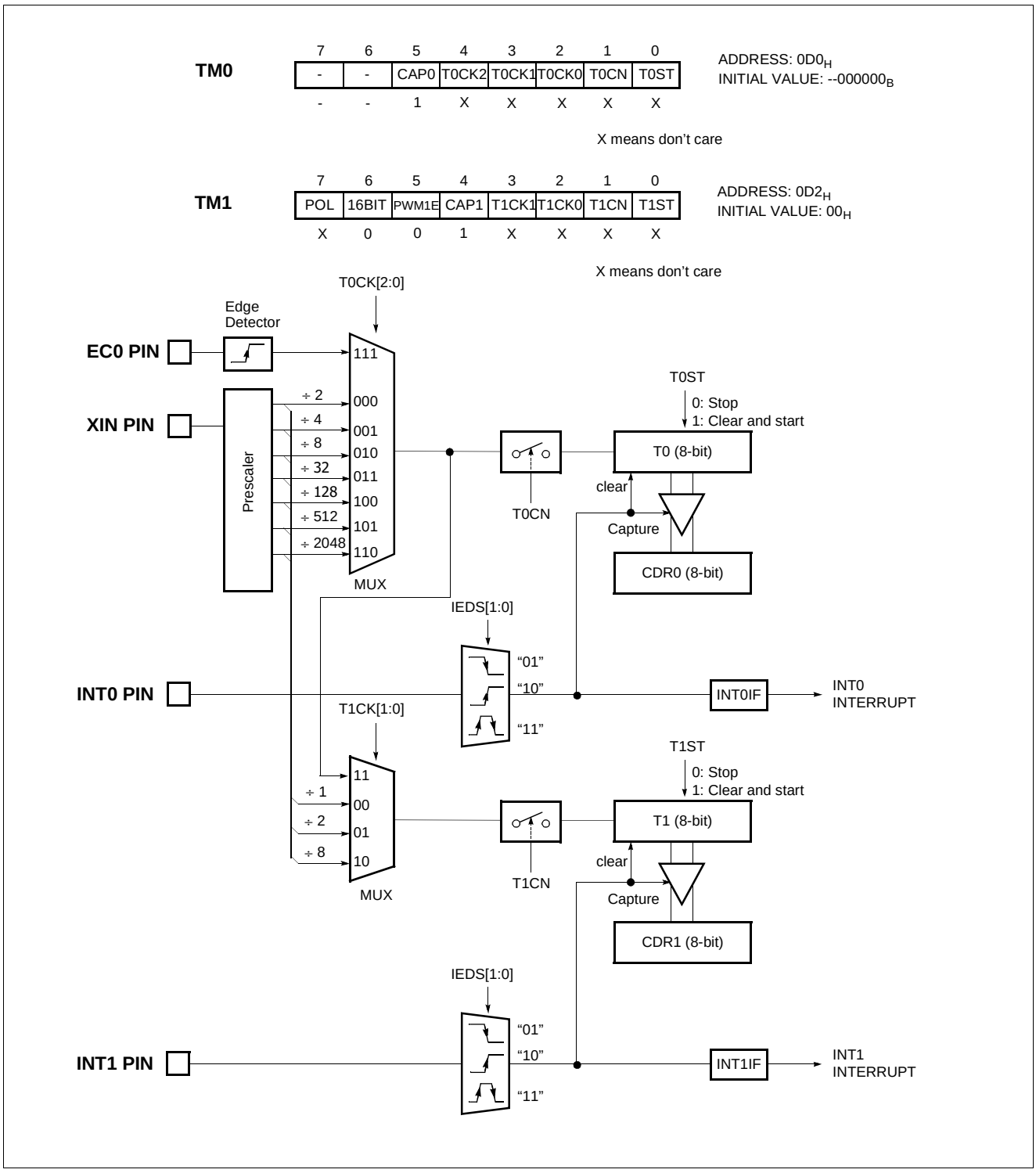


Figure 12-8 8-bit Capture Mode

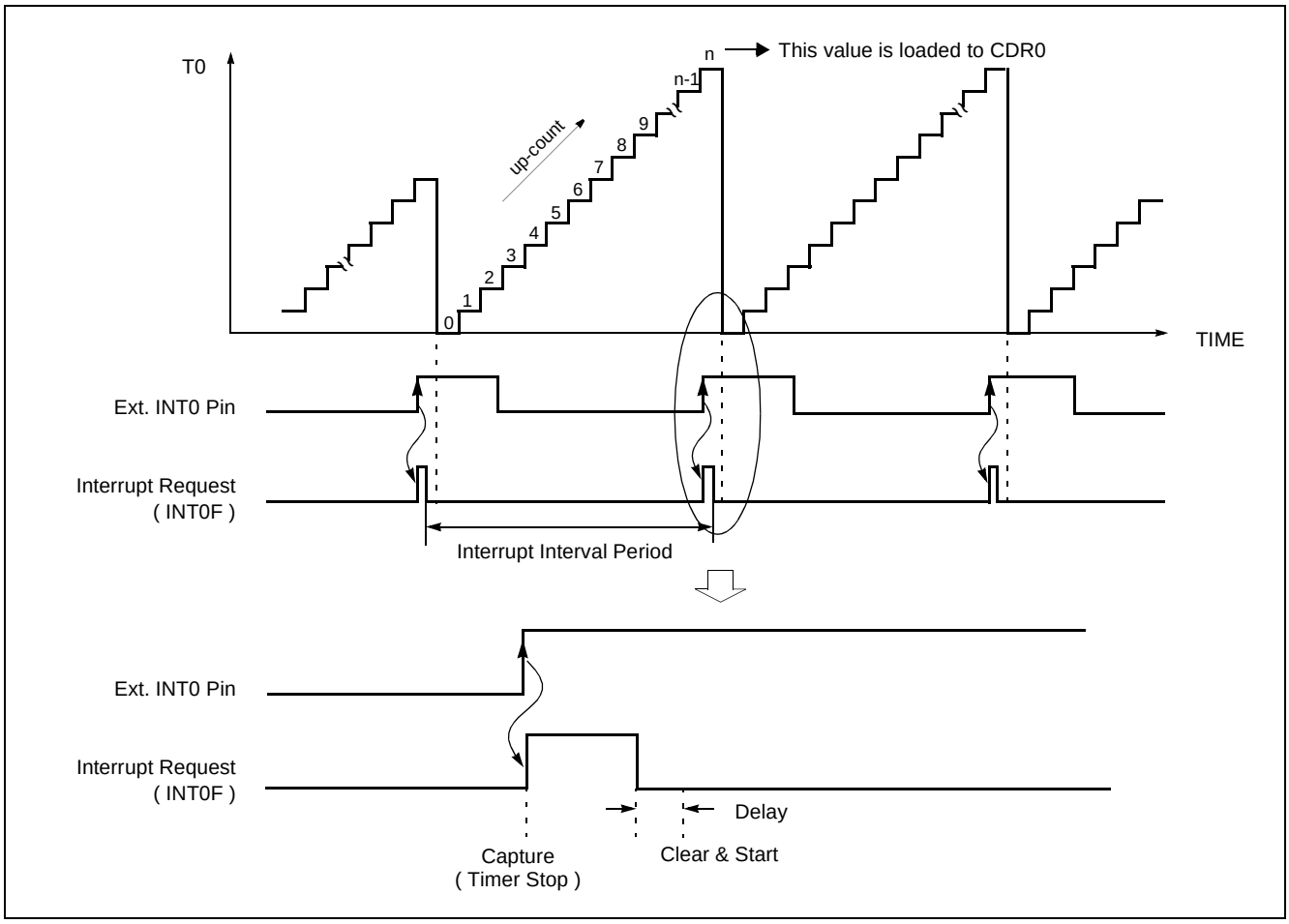


Figure 12-9 Input Capture Operation

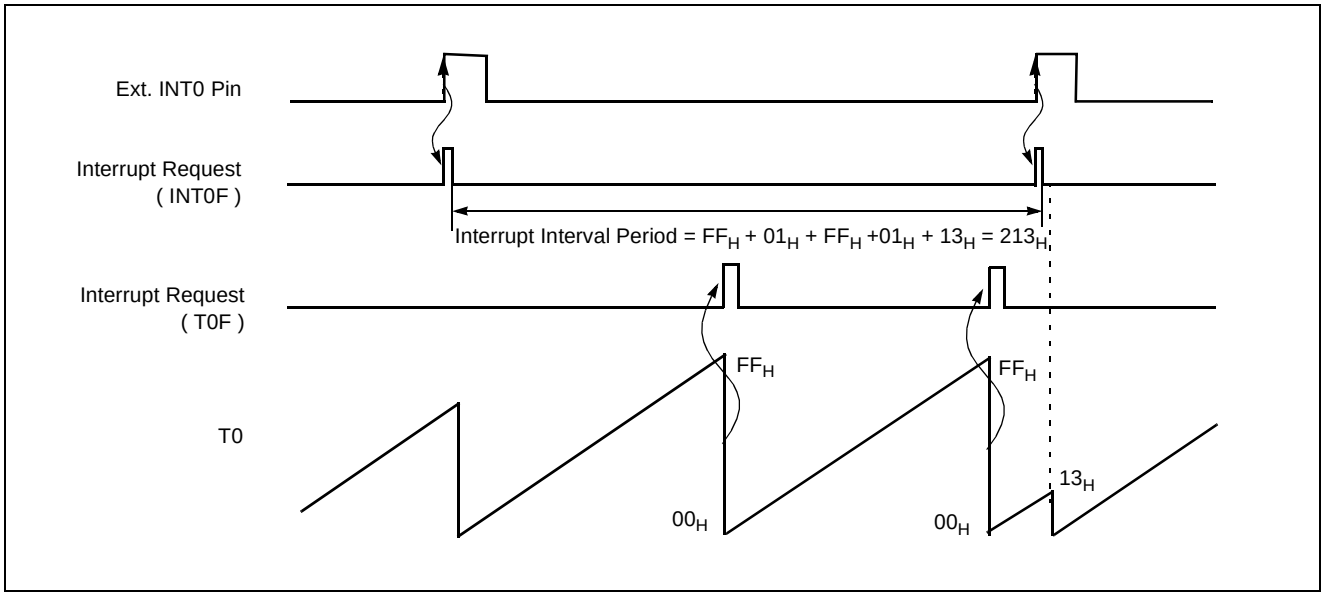


Figure 12-10 Excess Timer Overflow in Capture Mode

12.5 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.

The clock source of the Timer 0 is selected either internal or external clock by bit T0CK2, T0CK1 and T0CK0.

In 16-bit mode, the bits T1CK1,T1CK0 and 16BIT of TM1 should be set to "1" respectively.

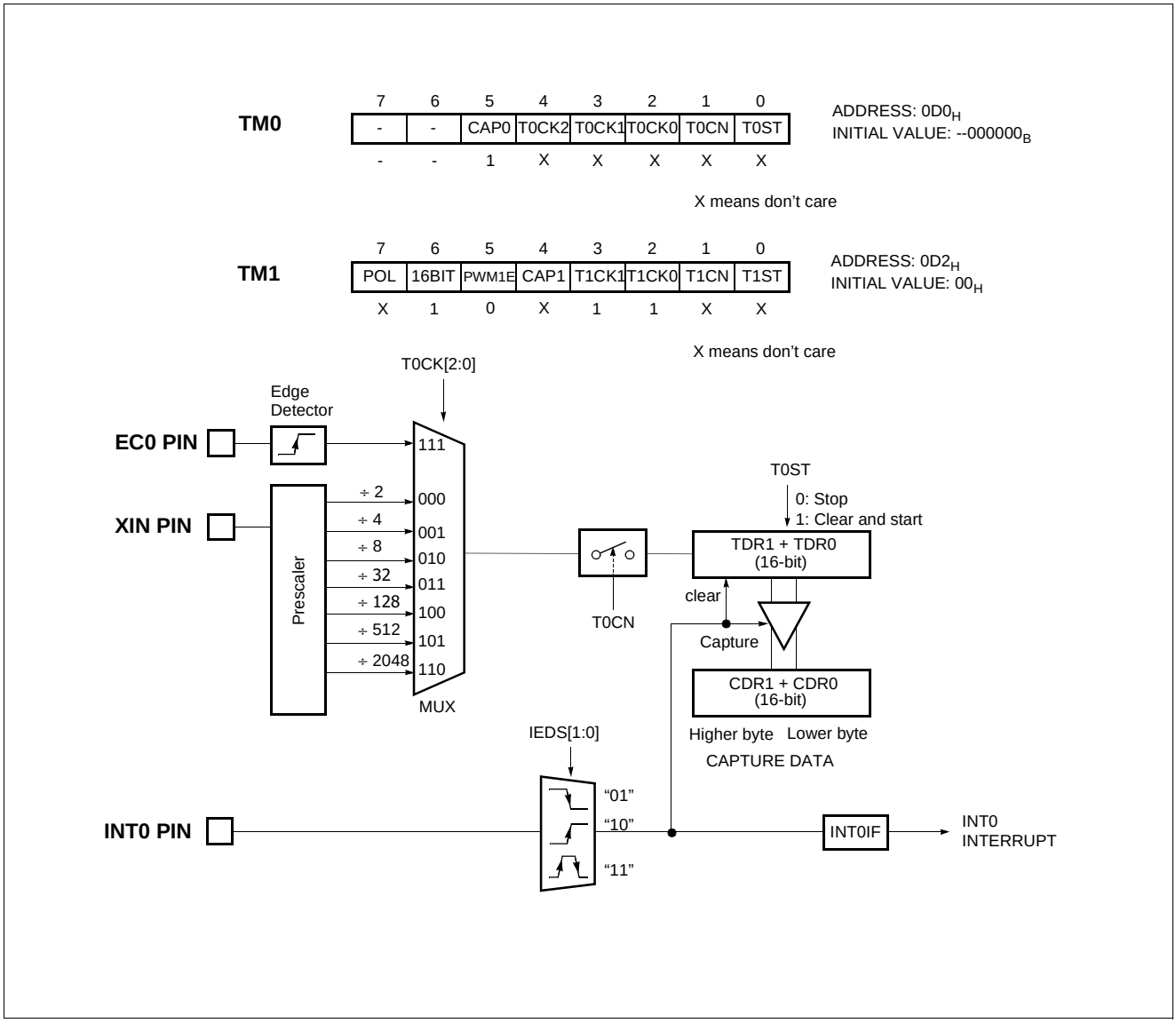


Figure 12-11 16-bit Capture Mode

Example 1:

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

```
LDM   TM0,#0000_1111B;8uS
LDM   TM1,#0100_1100B;16bit Mode
LDM   TDR0,#<62500   ;8uS X 62500
LDM   TDR1,#>62500   ;=0.5s
SET1  T0E
EI
:
:
```

Example 2:

Timer0 = 16-bit event counter mode

```
LDM   R0FUNC,#0000_0100B;EC0 Set
LDM   TM0,#0001_1111B;Counter Mode
LDM   TM1,#0100_1100B;16bit Mode
LDM   TDR0,#<0FFH    ;
LDM   TDR1,#>0FFH    ;
SET1  T0E
EI
:
:
```

Example 3:

Timer0 = 16-bit capture mode

```
LDM   R0FUNC,#0000_0001B;INT0 set
LDM   TM0,#0010_1111B;Capture Mode
LDM   TM1,#0100_1100B;16bit Mode
LDM   TDR0,#<0FFH    ;
LDM   TDR1,#>0FFH    ;
LDM   IEDS,#01H;Falling Edge
SET1  T0E
EI
:
:
```

12.6 PWM Mode

The GMS81C2020 has a high speed PWM (Pulse Width Modulation) functions which shared with Timer1.

In PWM mode, pin R56/PWM1O/T1O outputs up to a 10-bit resolution PWM output. This pin should be configured as a PWM output by setting "1" bit PWM1O in R5FUNC.6 register.

The period of the PWM output is determined by the T1PPR (PWM1 Period Register) and PWM1HR[3:2] (bit3,2 of PWM1 High Register) and the duty of the PWM output is determined by the T1PDR (PWM1 Duty Register) and PWM1HR[1:0] (bit1,0 of PWM1 High Register).

The user writes the lower 8-bit period value to the T1PPR and the higher 2-bit period value to the PWM1HR[3:2]. And writes duty value to the T1PDR and the PWM1HR[1:0] same way.

The T1PDR is configured as a double buffering for glitch-less PWM output. In Figure 12-12, 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)

PWM Period = [PWM1HR[3:2]T1PPR] X Source Clock

PWM Duty = [PWM1HR[1:0]T1PDR] X Source Clock

The relation of frequency and resolution is in inverse proportion. Table 12-2 shows the relation of PWM frequency vs. resolution.

If it needed more higher frequency of PWM, it should be reduced resolution.

Resolution	Frequency		
	T1CK[1:0] = 00(250nS)	T1CK[1:0] = 01(500nS)	T1CK[1:0] = 10(2uS)
10-bit	3.9KHz	0.98KHZ	0.49KHZ
9-bit	7.8KHz	1.95KHz	0.97KHz
8-bit	15.6KHz	3.90KHz	1.95KHz
7-bit	31.2KHz	7.81KHz	3.90KHz

Table 12-2 PWM Frequency vs. Resolution at 4MHz

The bit POL of TM1 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 POL (1: High, 0: Low). And if the duty value is set to "00_H", the PWM output is determined by the bit POL (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 12-14. As it were, the absolute duty time is not changed in varying frequency. But the changed period value must greater than the duty value.

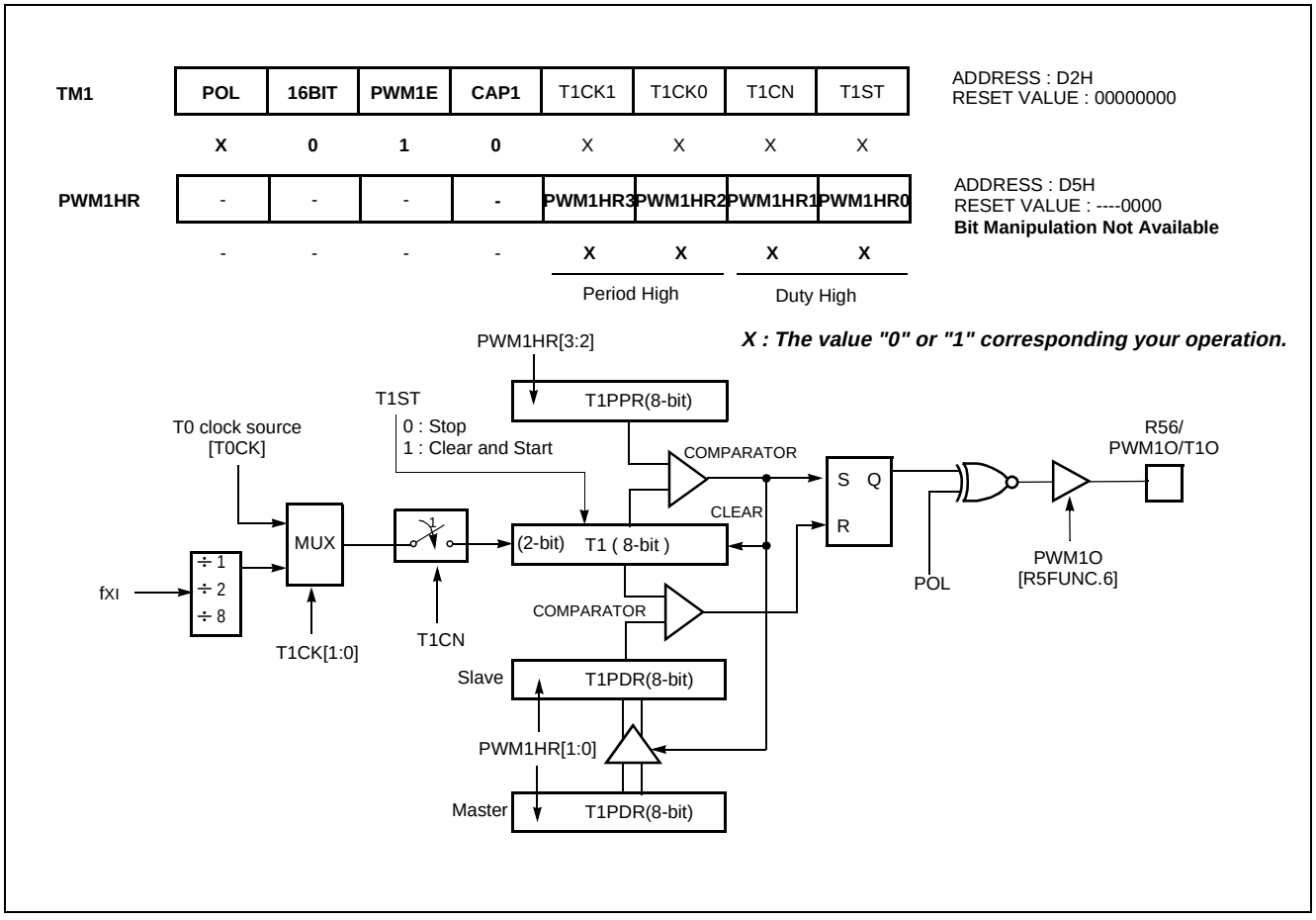


Figure 12-12 PWM Mode

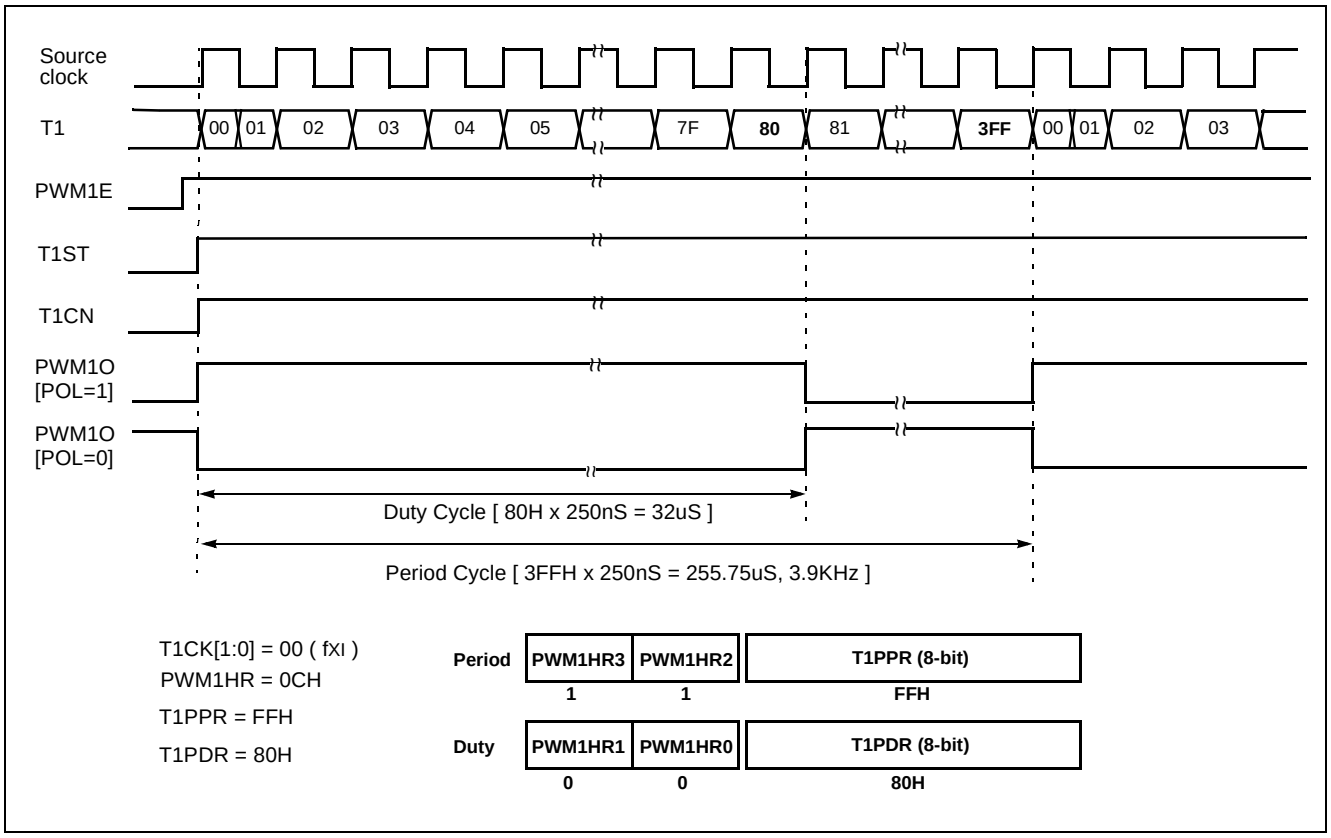


Figure 12-13 Example of PWM at 4MHz

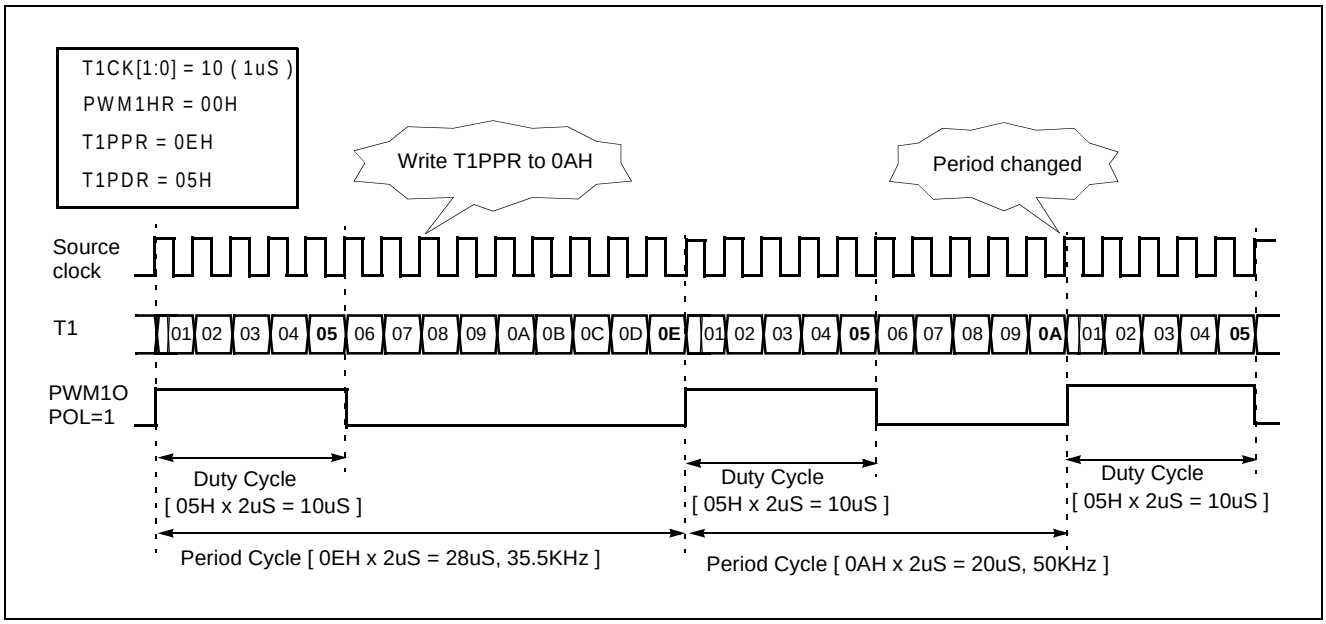


Figure 12-14 Example of Changing the Period in Absolute Duty Cycle (@4MHz)

13. 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 13-1, 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, each port is assigned analog input port by setting the bit ANSEL[7:0] in R6FUNC register. Also it is assigned analog input port by setting the bit ANSEL[11:8] in R7FUNC register. And selected the corresponding channel to be converted by setting ADS[3:0].

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 ADCR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCR, the A/D conversion status bit ADSF is set to "1", and the A/D interrupt flag ADIF is set. The block diagram of the A/D module is shown in Figure 13-2. 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 fXI=4 MHz)

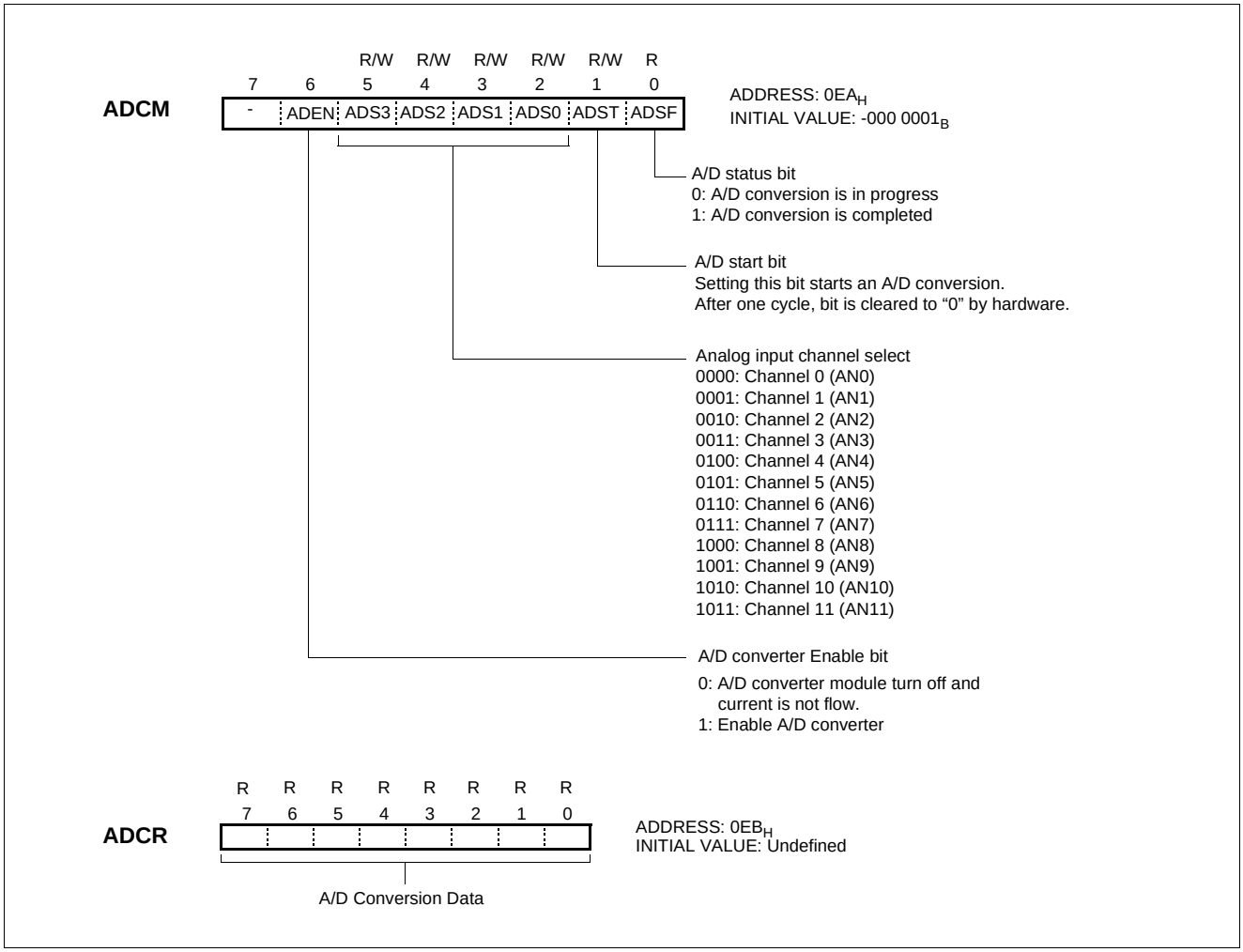


Figure 13-1 A/D Converter Control Register

.

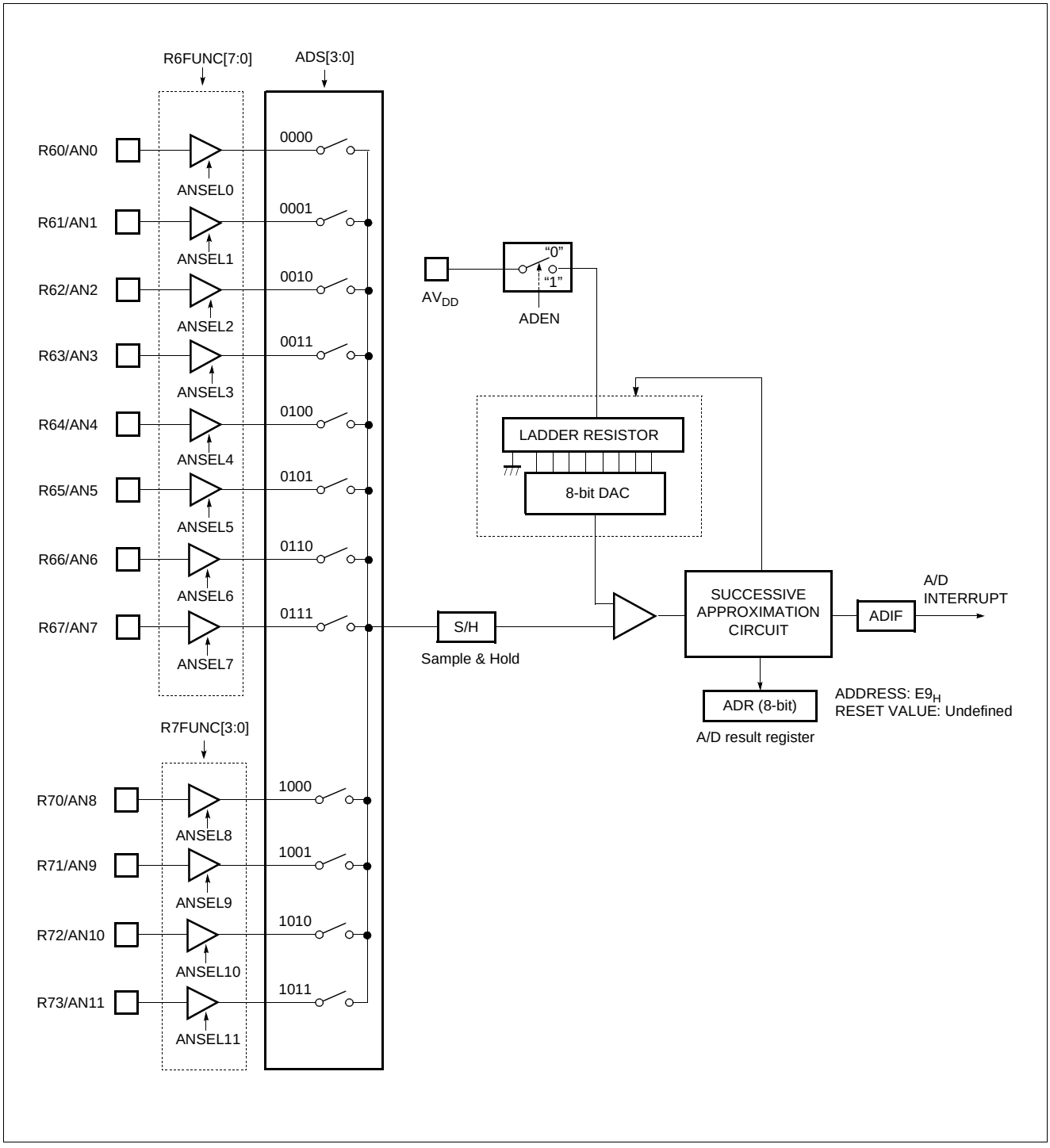


Figure 13-2 A/D Block Diagram

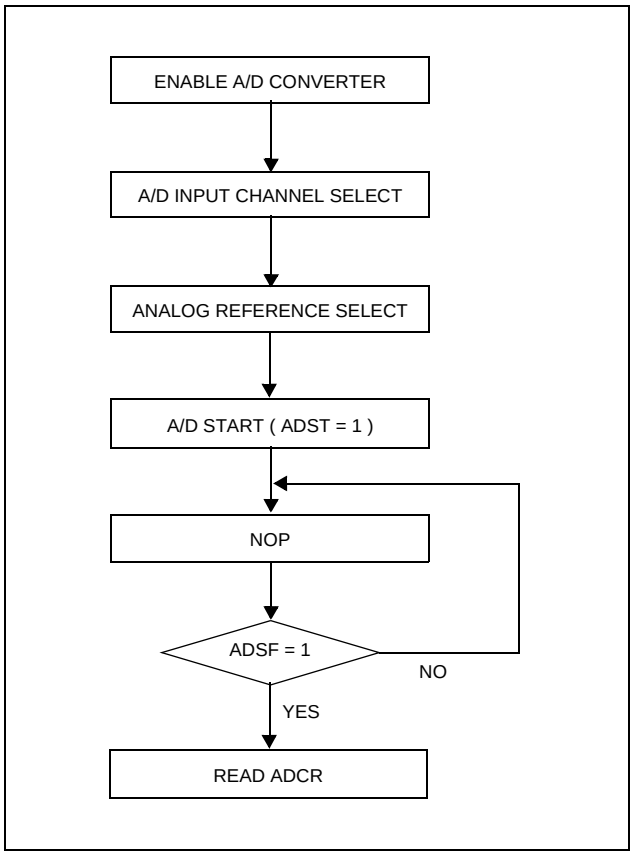


Figure 13-3 A/D Converter Operation Flow

A/D Converter Cautions

(1) Input range of AN11 to AN0

The input voltage of AN11 to AN0 should be within the specification range. In particular, if a voltage above AVDD or below AVSS 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 AVDD and AN11 to AN0. 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 13-4 in order to reduce noise.

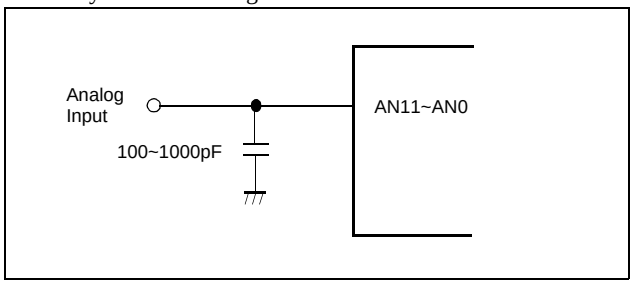


Figure 13-4 Analog Input Pin Connecting Capacitor

(3) Pins AN11/R73 to AN8/R70 and AN7/R67 to AN0/R60

The analog input pins AN11 to AN0 also function as input/output port (PORT R7 and R6) pins. When A/D conversion is performed with any of pins AN11 to AN0 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) AVDD pin input impedance

A series resistor string of approximately 10KΩ is connected between the AVDD pin and the AVSS 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 AVDD pin and the AVSS pin, and there will be a large reference voltage error.

14. SERIAL PERIPHERAL INTERFACE

The Serial Peripheral Interface (SPI) module is a serial interface useful for communicating with other peripheral of microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The Serial Peripheral Interface(SPI) is 8-bit clock synchronous type and consists of serial I/O register, serial I/O mode register, clock selection circuit octal counter and control circuit. The SOUT pin is designed to input and output. So Serial Peripheral Interface(SPI) can be operated with minimum two pin

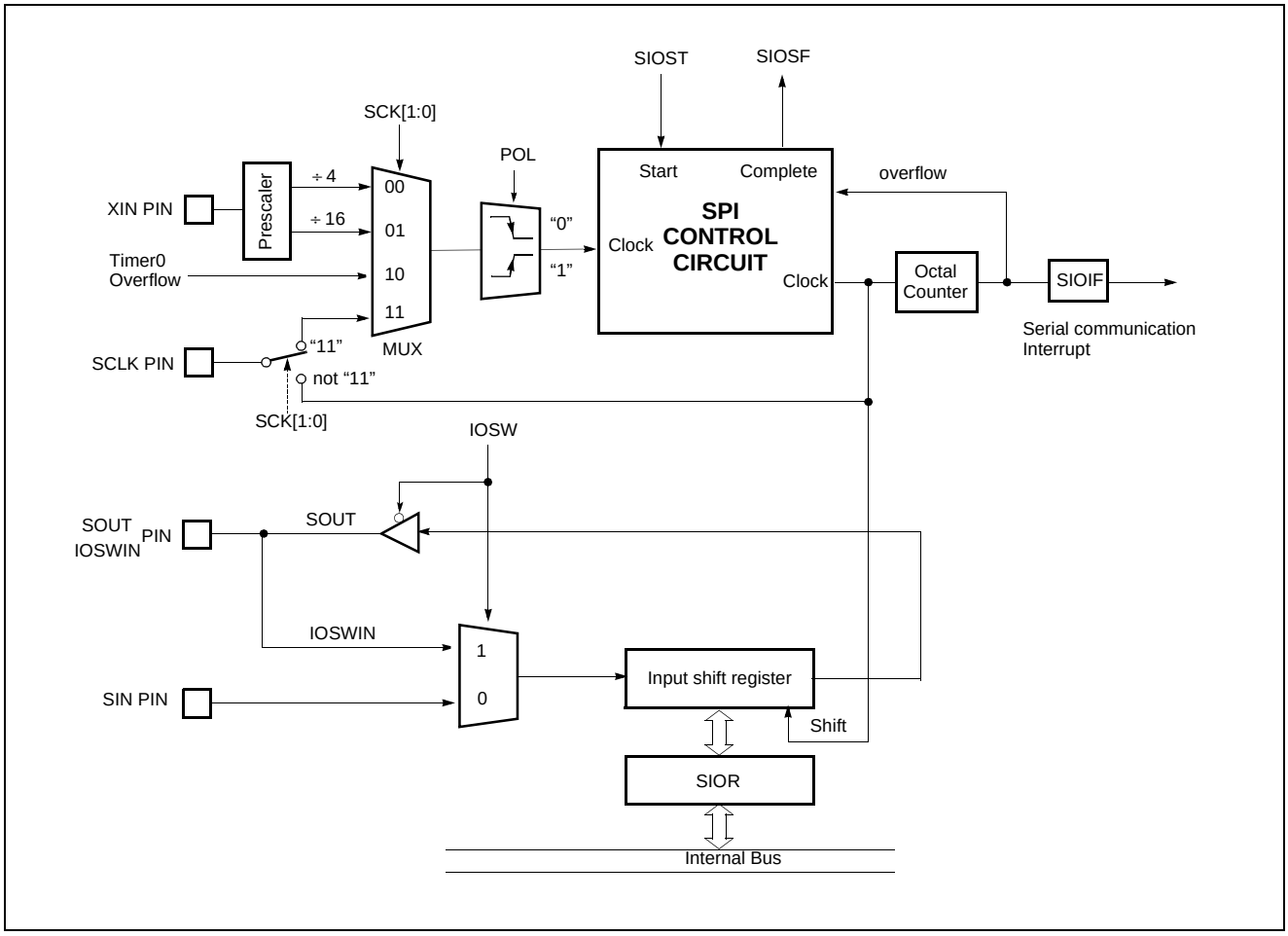


Figure 14-1 SPI Block Diagram

Serial I/O Mode Register(SIOM) controls serial I/O function. According to SCK1 and SCK0, the internal clock or external clock can be selected. The serial transmission operation mode is decided by setting the SM1 and SM0, and the polarity of transfer clock is selected by setting the POL.

Serial I/O Data Register(SIOR) is a 8-bit shift register. First LSB is send or is received. When receiving mode, serial input pin is selected by IOSW. The SPI allows 8-bits of data to be synchronously transmitted and received.

To accomplish communication, typically three pins are used:

- Serial Data In R54/SIN
- Serial Data Out R55/SOUT
- Serial Clock R53/SCLK

.

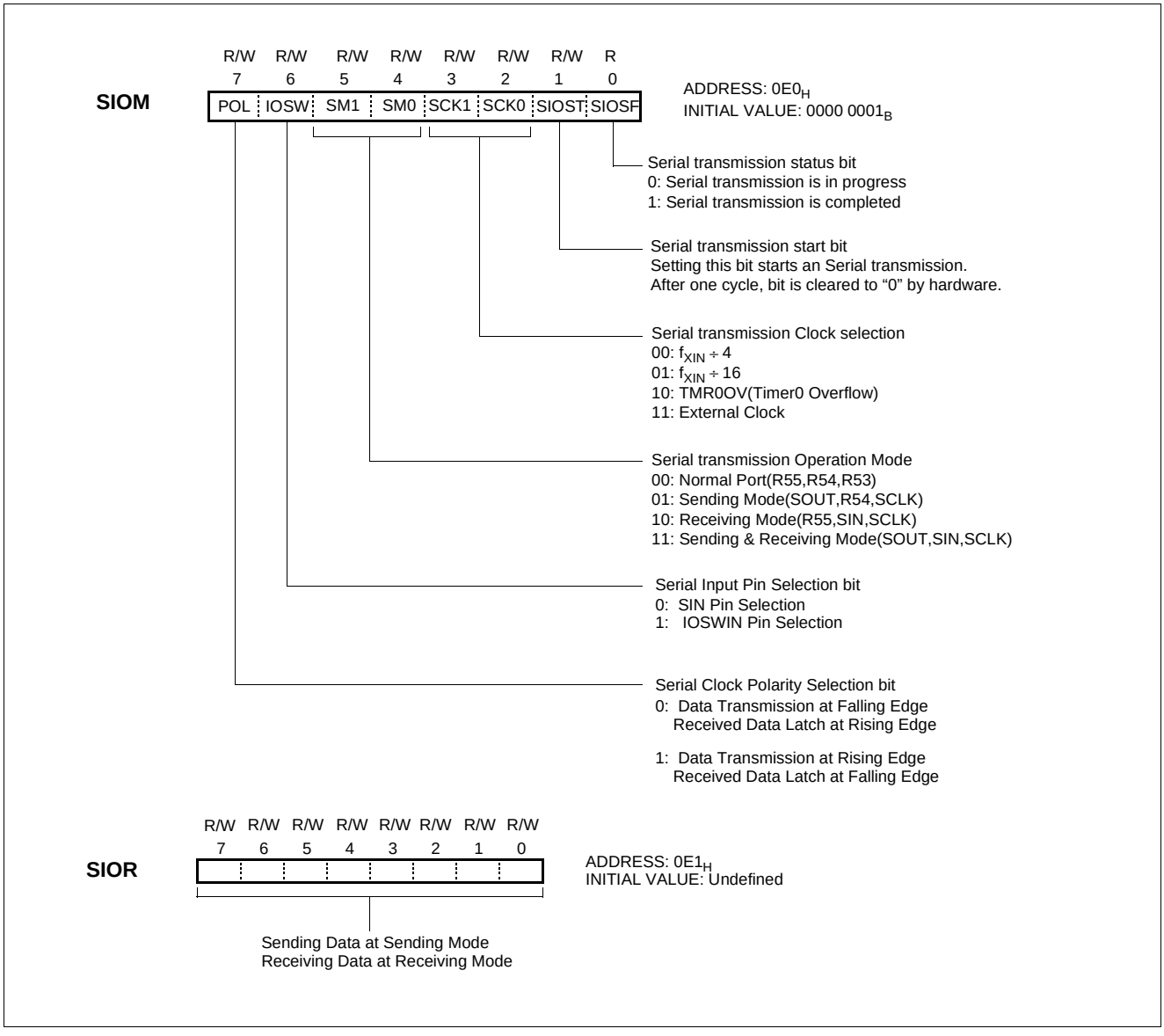


Figure 14-2 SPI Control Register

14.1 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(IFSIO) occurred.

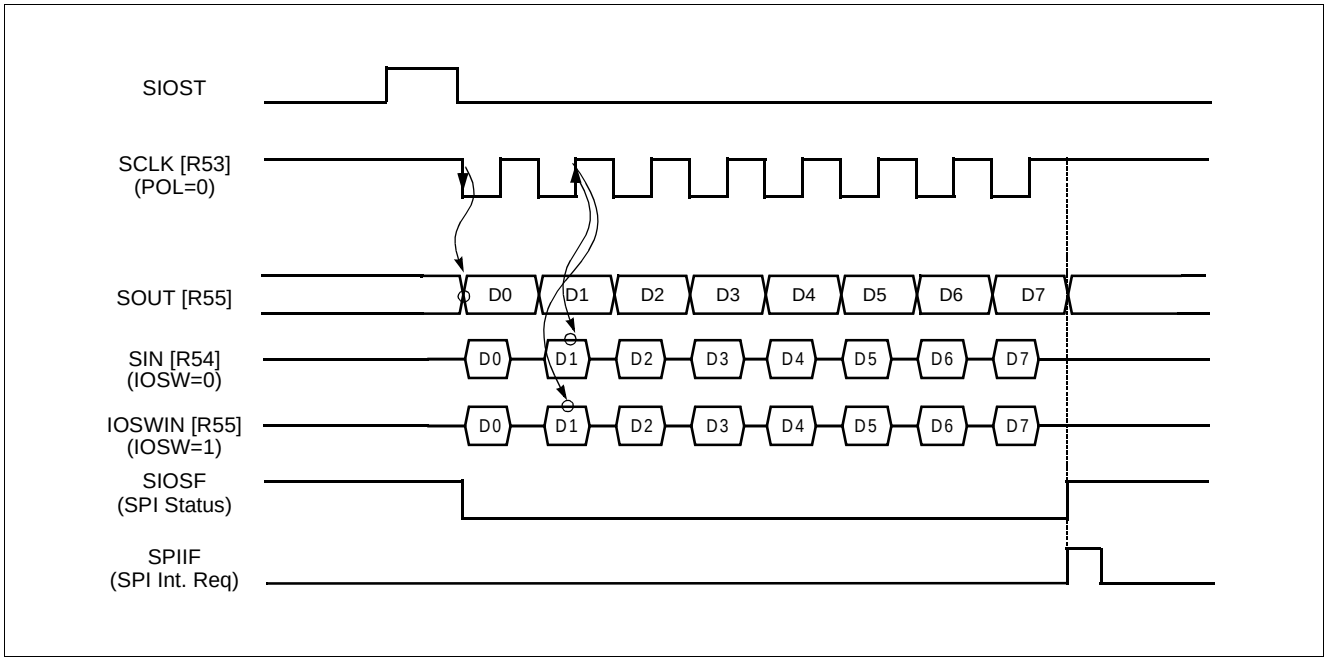


Figure 14-3 SPI Timing Diagram at POL=0

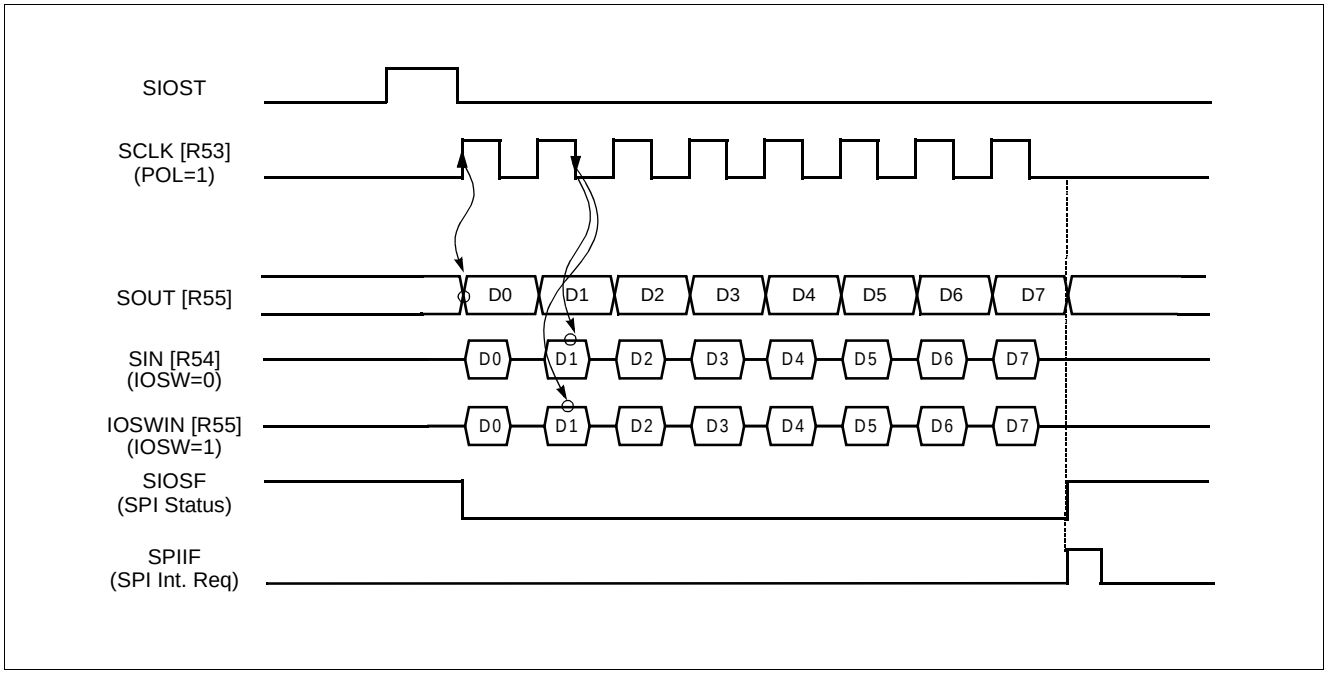


Figure 14-4 SPI Timing Diagram at POL=1

14.2 The method of Serial I/O

① Select transmission/receiving mode

Note: *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.

Note: *If both transmission mode is selected and transmission is performed simultaneously it would be made error.*

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

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

14.3 The Method to Test Correct Transmission

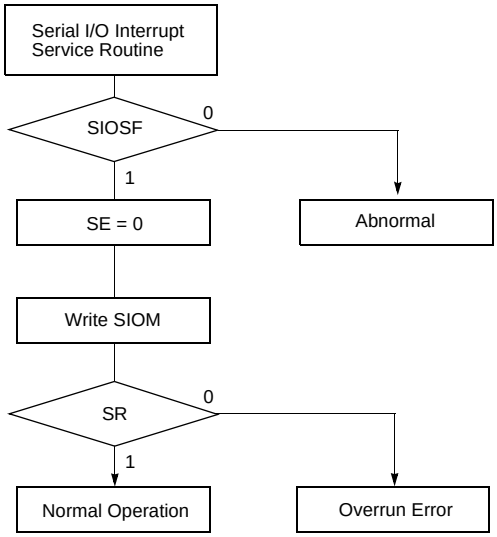


Figure 14-5 Serial Method to Test Transmission

15. BUZZER FUNCTION

The buzzer driver block consists of 6-bit binary counter, buzzer register BUR, and clock source selector. It generates square-wave which has very wide range frequency (480Hz ~ 250kHz at f_{XIN}= 4MHz) by user software.

A 50% duty pulse can be output to R03/BUZO pin to use for piezo-electric buzzer drive. Pin R03 is assigned for output port of Buzzer driver by setting the bit 3 of R0FUNC(address $0F4_H$) to "1". At this time, the pin R03 must be defined as output mode (the bit 3 of R0IO=1).

Example: 5kHz output at 4MHz.

```
LDM    R0IO,#XXXX_1XXXB
LDM    BUR,#0011_0010B

LDM    R0FUNC,#XXXX_1XXXB
```

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.

$$f_{BUZ} = \frac{f_{XIN}}{2 \times DivideRatio \times (BUR + 1)}$$

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

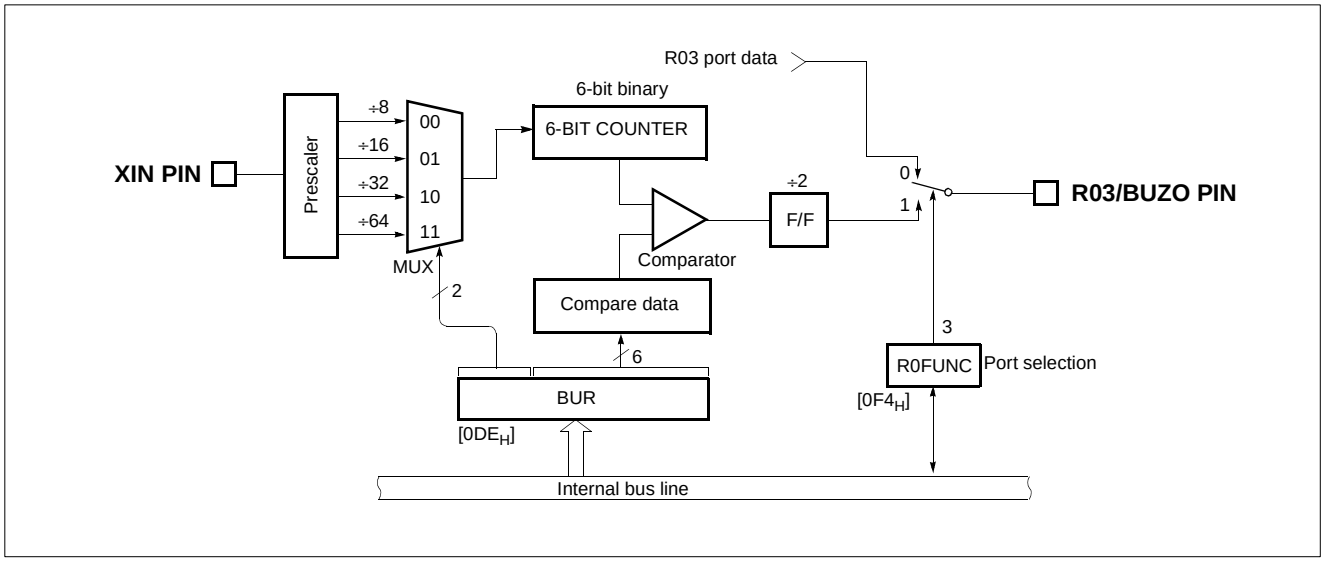


Figure 15-1 Block Diagram of Buzzer Driver

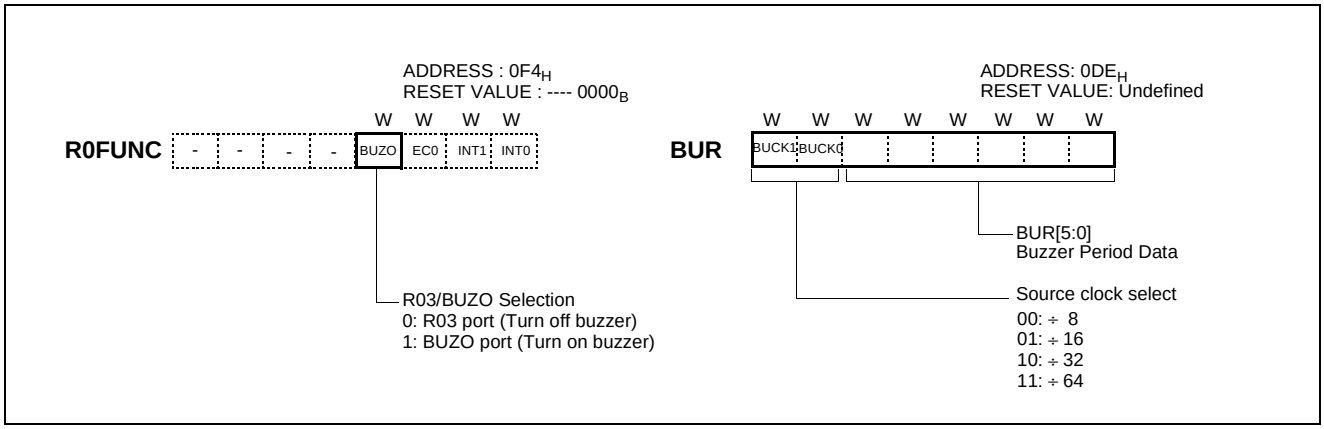


Figure 15-2 R0FUNC and Buzzer Register

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

BUR [5:0]	BUR[7:6]			
	00	01	10	11
00	-	-	-	-
01	250.000	125.000	62.500	31.250
02	125.000	62.500	31.250	15.625
03	83.333	41.667	20.833	10.417
04	62.500	31.250	15.625	7.813
05	50.000	25.000	12.500	6.250
06	41.667	20.833	10.417	5.208
07	35.714	17.857	8.929	4.464
08	31.250	15.625	7.813	3.906
09	27.778	13.889	6.944	3.472
0A	25.000	12.500	6.250	3.125
0B	22.727	11.364	5.682	2.841
0C	20.833	10.417	5.208	2.604
0D	19.231	9.615	4.808	2.404
0E	17.857	8.929	4.464	2.232
0F	16.667	8.333	4.167	2.083
10	15.625	7.813	3.906	1.953
11	14.706	7.353	3.676	1.838
12	13.889	6.944	3.472	1.736
13	13.158	6.579	3.289	1.645
14	12.500	6.250	3.125	1.563
15	11.905	5.952	2.976	1.488
16	11.364	5.682	2.841	1.420
17	10.870	5.435	2.717	1.359
18	10.417	5.208	2.604	1.302
19	10.000	5.000	2.500	1.250
1A	9.615	4.808	2.404	1.202
1B	9.259	4.630	2.315	1.157
1C	8.929	4.464	2.232	1.116
1D	8.621	4.310	2.155	1.078
1E	8.333	4.167	2.083	1.042
1F	8.065	4.032	2.016	1.008

BUR [5:0]	BUR[7:6]			
	00	01	10	11
20	7.813	3.906	1.953	0.977
21	7.576	3.788	1.894	0.947
22	7.353	3.676	1.838	0.919
23	7.143	3.571	1.786	0.893
24	6.944	3.472	1.736	0.868
25	6.757	3.378	1.689	0.845
26	6.579	3.289	1.645	0.822
27	6.410	3.205	1.603	0.801
28	6.250	3.125	1.563	0.781
29	6.098	3.049	1.524	0.762
2A	5.952	2.976	1.488	0.744
2B	5.814	2.907	1.453	0.727
2C	5.682	2.841	1.420	0.710
2D	5.556	2.778	1.389	0.694
2E	5.435	2.717	1.359	0.679
2F	5.319	2.660	1.330	0.665
30	5.208	2.604	1.302	0.651
31	5.102	2.551	1.276	0.638
32	5.000	2.500	1.250	0.625
33	4.902	2.451	1.225	0.613
34	4.808	2.404	1.202	0.601
35	4.717	2.358	1.179	0.590
36	4.630	2.315	1.157	0.579
37	4.545	2.273	1.136	0.568
38	4.464	2.232	1.116	0.558
39	4.386	2.193	1.096	0.548
3A	4.310	2.155	1.078	0.539
3B	4.237	2.119	1.059	0.530
3C	4.167	2.083	1.042	0.521
3D	4.098	2.049	1.025	0.512
3E	4.032	2.016	1.008	0.504
3F	3.968	1.984	0.992	0.496

16. INTERRUPTS

The GMS81C20xx 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). Nine interrupt sources are provided. The configuration of interrupt circuit is shown in Figure 16-2.

The External Interrupts INT0 and INT1 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 INT0F and INT1F in register IRQH. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.

The Timer 0 ~ Timer 1 Interrupts are generated by TxIF which is set by a match in their respective timer/counter register. The Basic Interval Timer Interrupt is generated by BITIF which is set by an overflow in the timer register.

The AD converter Interrupt is generated by ADIF which is set by finishing the analog to digital conversion.
The Watchdog timer Interrupt is generated by WDTIF which set by a match in Watchdog timer register.
The Basic Interval Timer Interrupt is generated by BITIF which are set by a overflow in the timer counter register.

The interrupts are controlled by the interrupt master enable flag I-flag (bit 2 of PSW on page 26), 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	-
External Interrupt 0	INT0	1
External Interrupt 1	INT1	2
Timer/Counter 0	TIMER0	3
Timer/Counter 1	TIMER1	4
-	-	-
-	-	-
-	-	-
-	-	-
ADC Interrupt	ADC	5
Watchdog Timer	WDT	6
Basic Interval Timer	BIT	7
Serial Communication	SCI	8

Vector addresses are shown in Figure 8-6 on page 28. Interrupt enable registers are shown in Figure 16-3. 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.

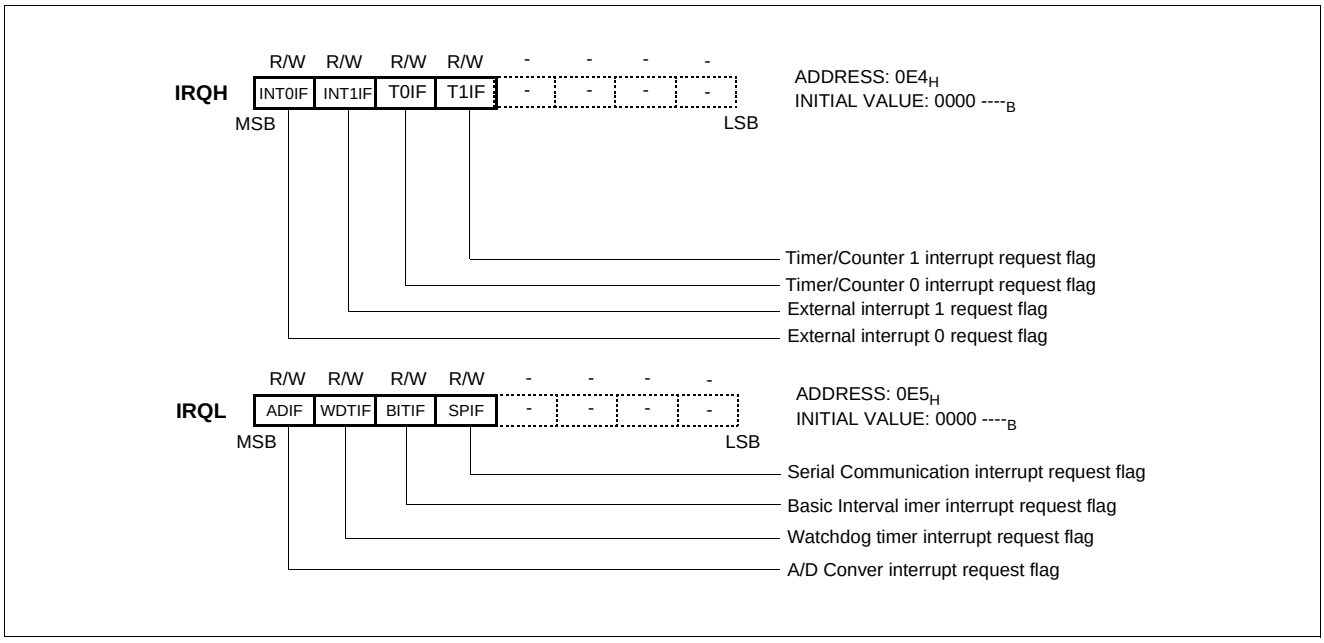


Figure 16-1 Interrupt Request Flag

.

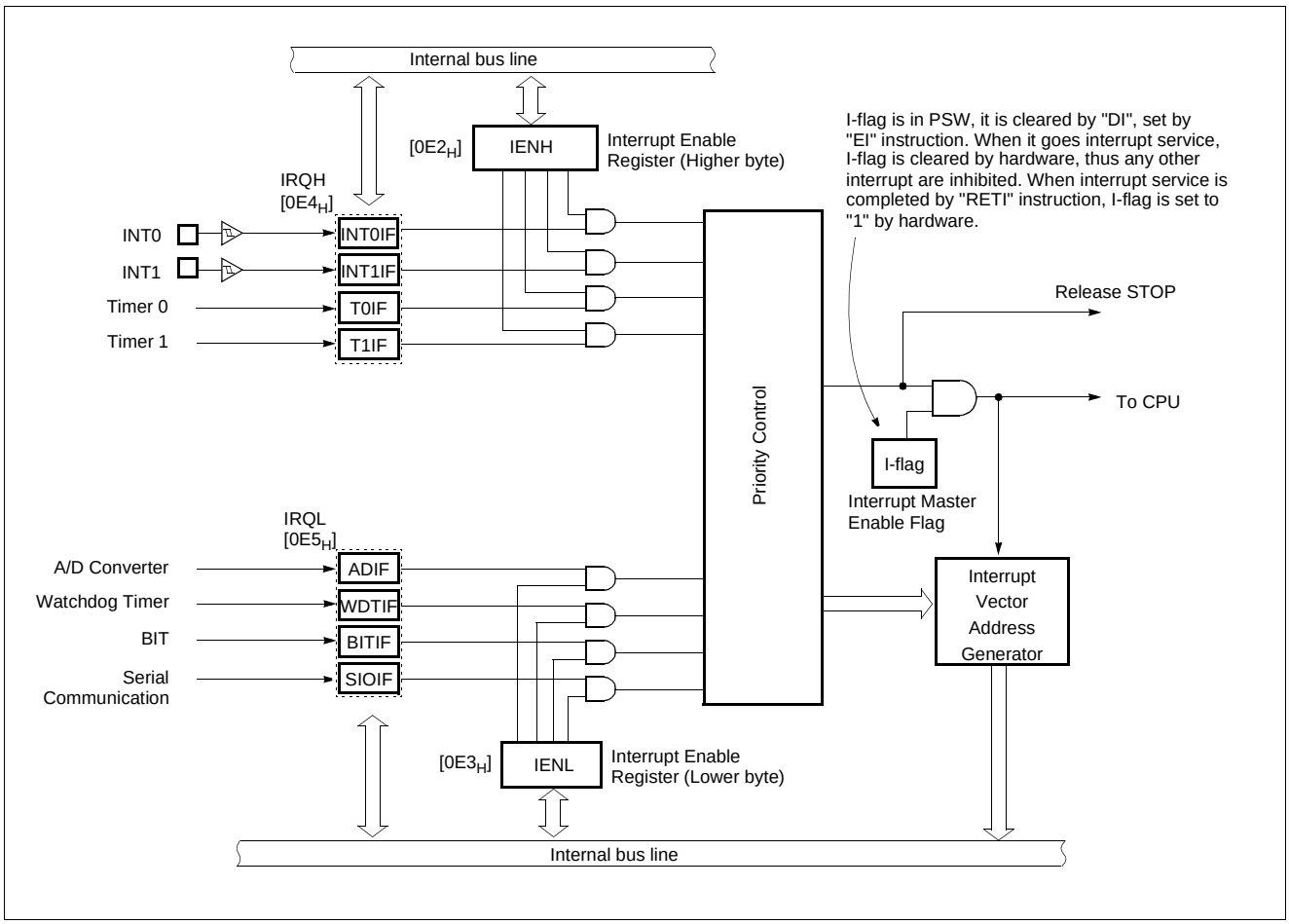


Figure 16-2 Block Diagram of Interrupt

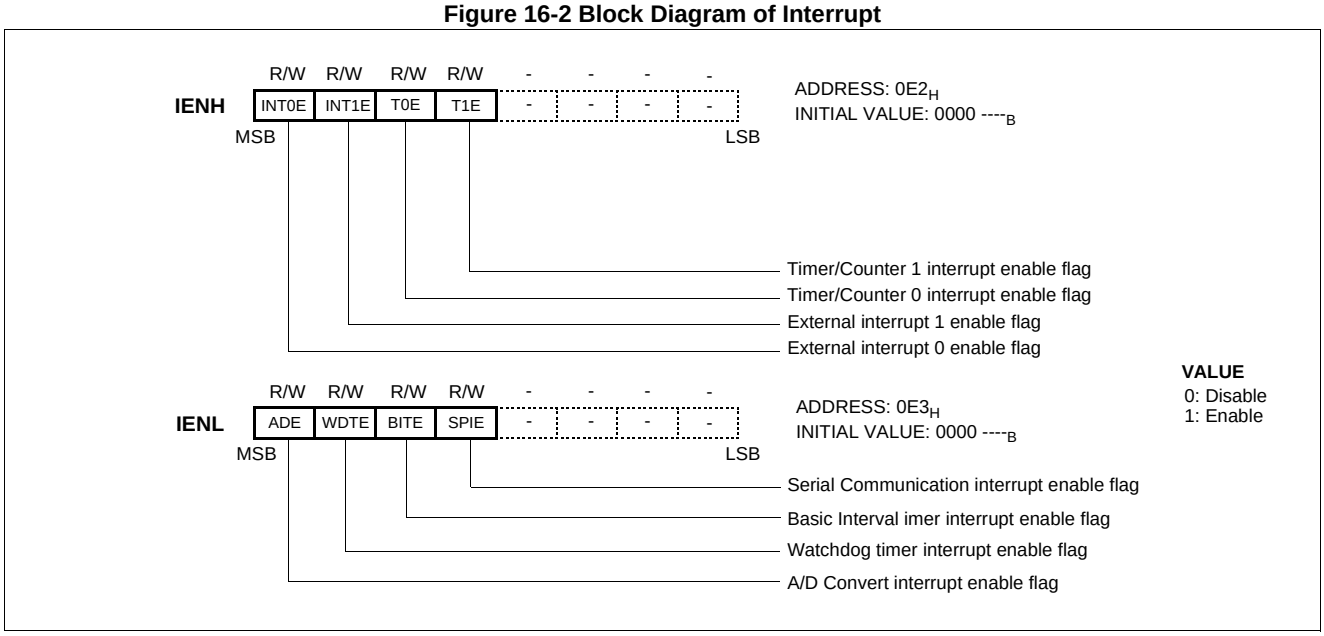


Figure 16-3 Interrupt Enable Flag

16.1 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 µs at f_{MAIN}=4.19MHz) after the completion of the current instruction execution. The interrupt service task is terminated upon execution of an interrupt return instruction [RETI].

Interrupt acceptance

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

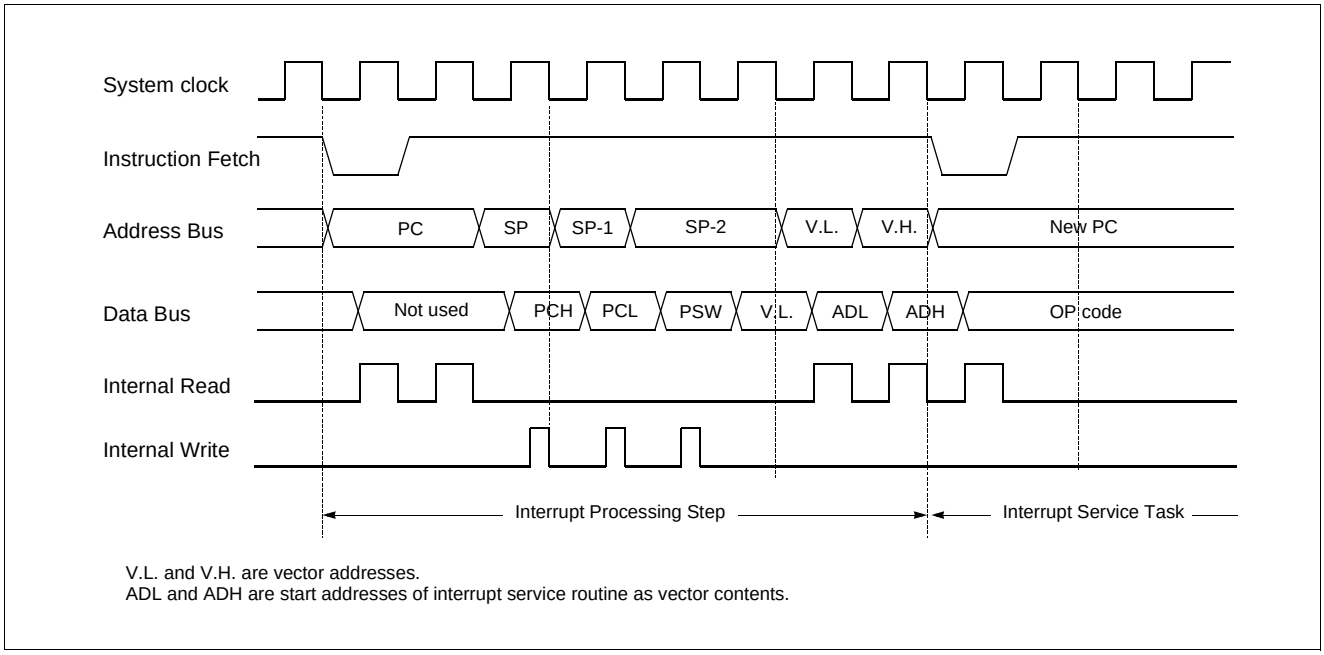
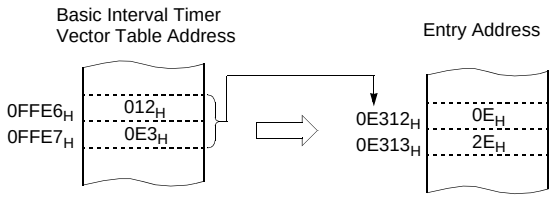


Figure 16-4 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



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

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

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

Saving/Restoring General-purpose Register

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

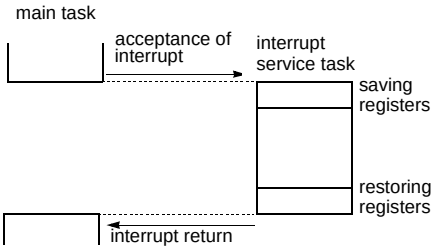
area for saving registers.

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

Example: Register save using push and pop instructions

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



16.2 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 16-5.

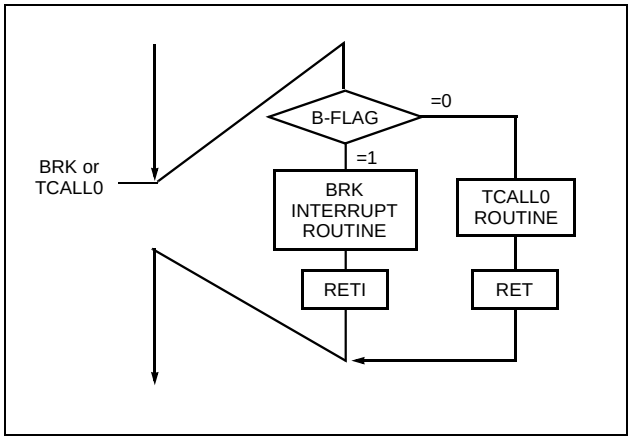


Figure 16-5 Execution of BRK/TCALL0

16.3 Multi Interrupt

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

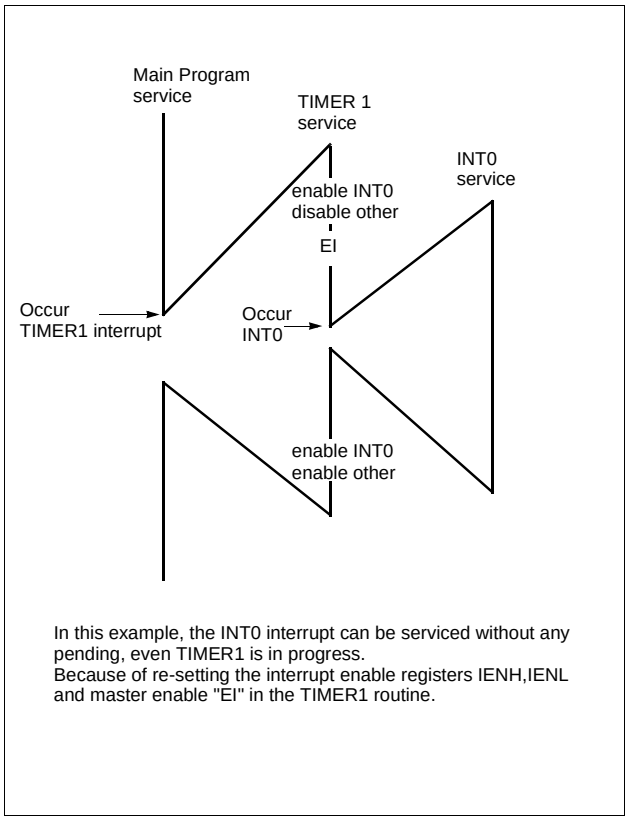


Figure 16-6 Execution of Multi Interrupt

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

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

```
TIMER1:  PUSH   A
         PUSH   X
         PUSH   Y
         LDM    IENH,#80H    ;Enable INT0 only
         LDM    IENL,#0      ;Disable other
         EI                  ;Enable Interrupt
         :
         :
         :

         :
         :
         :
         LDM    IENH,#0F0H   ;Enable all interrupts
         LDM    IENL,#0F0H
         POP    Y
         POP    X
         POP    A
         RETI
```

16.4 External Interrupt

The external interrupt on INT0 and INT1 pins are edge triggered depending on the edge selection register IEDS (address $0F8_H$) as shown in Figure 16-7.

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

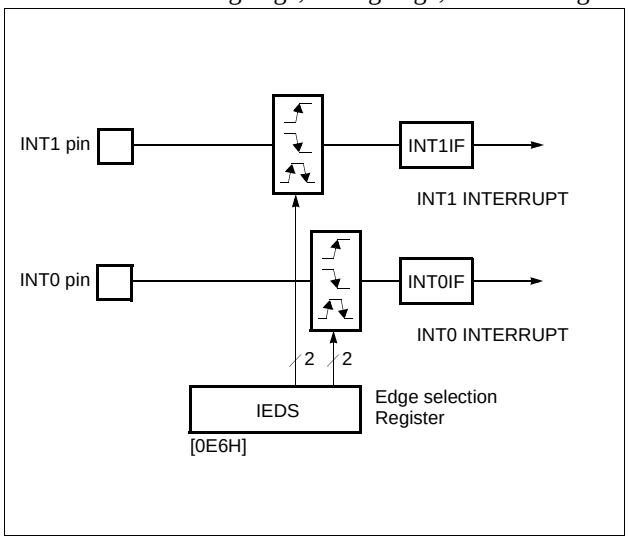


Figure 16-7 External Interrupt Block Diagram

INT0 and INT1 are multiplexed with general I/O ports (R00 and R01). To use as an external interrupt pin, the bit of R4 port mode register R0FUNC should be set to "1" correspondingly.

Example: To use as an INT0 and INT1

```
            :
            :
;**** Set port as an input port R00,R01
            LDM     R0IO,#1111_1100B
            ;
;**** Set port as an interrupt port
            LDM     R0FUNC,#0000_0011B
            ;
;**** Set Falling-edge Detection
            LDM     IEDS,#0000_0101B
            :
            :
            :
```

Response Time

The INT0 and INT1 edge are latched into INT0IF and INT1IF 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 16-8shows interrupt response timings.

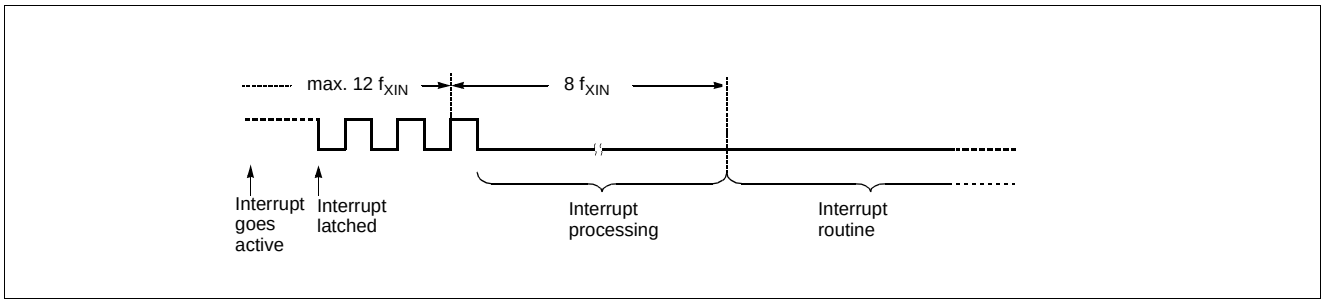


Figure 16-8 Interrupt Response Timing Diagram

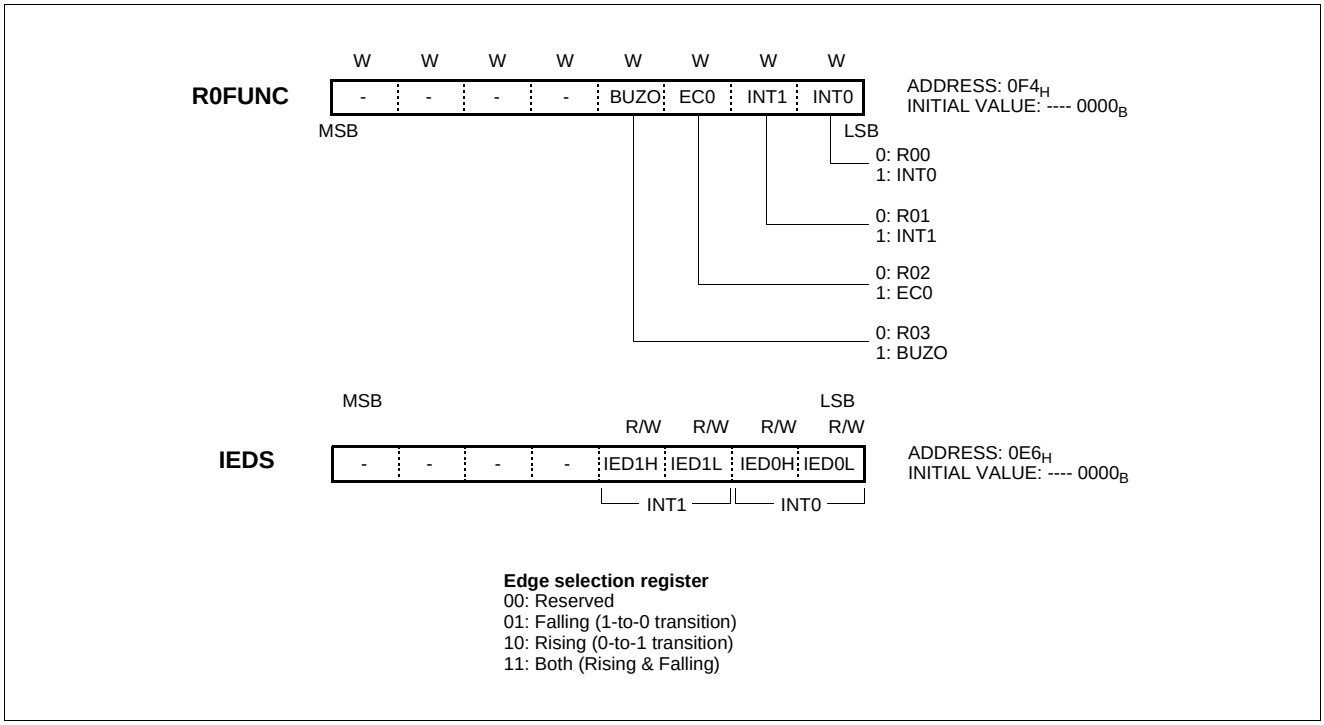


Figure 16-9 R0FUNC and IEDS Registers

17. Power Saving Mode

For applications where power consumption is a critical factor, device provides four kinds of power saving functions, STOP mode, Sub-active mode and Wake-up Timer mode (Stand-by mode, Watch mode). Table 17-1 shows the status of each Power Saving Mode.

Peripheral	STOP Mode	Sub-active Mode	Wake-up Timer Mode	
			Stand-by Mode	Watch Mode
RAM	Retain	Retain	Retain	Retain
Control Registers	Retain	Retain	Retain	Retain
I/O Ports	Retain	Retain	Retain	Retain
CPU	Stop	Operation	Stop	Stop
Timer0	Stop	Operation	Operation	Operation
Oscillation	Stop	Stop	Oscillation	Stop
Sub Oscillation	Stop	Oscillation	Stop	Oscillation
Prescaler	Stop	Operation	÷ 2048 only	÷ 2048 only
Entering Condition [WAKEUP]	0	0	1	1

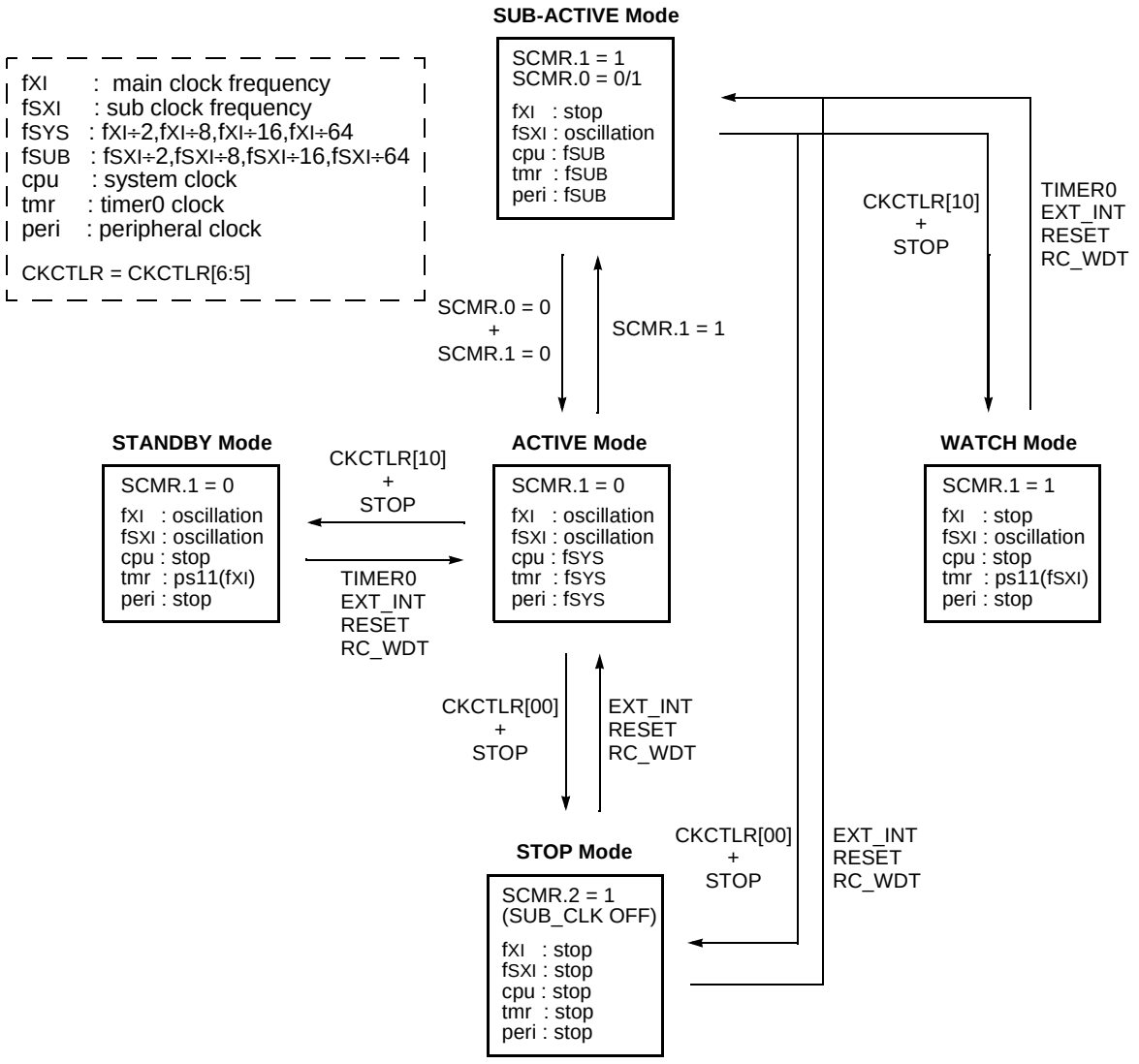
Table 17-1 Power Saving Mode

The power saving function is activated by execution of STOP instruction and by execution of STOP instruction after setting the corresponding status (WAKEUP) of CKCTLR. We shows the release sources from each Power Saving Mode

Release Source	STOP Mode	Sub-active Mode	Wake-up Timer Mode	
			Stand-by Mode	Watch Mode
RESET	O	O	O	O
RCWDT	O	O	O	O
EXT.INT0	O	O	O	O
EXT.INT1				
Timer0	X	X	O	O

Table 17-2 Release Sources from Power Saving Mode

17.1 Operating Mode



System Clock Mode Register

SCMR	-	-	-	CS1	CS0	SUBOFF	CLKSEL	MAINOFF

ADDRESS : FAH
RESET VALUE : ---00000

CS[1:0] Clock selection enable bits
00 : fXI ÷ 2 10 : fXI ÷16
01 : fXI ÷ 8 11 : fXI ÷ 64

CLKSEL Clock selection bit
0 : Main clock selection
1 : Sub clock selection

SUBOFF Sub clock control bit
0: On sub clock
1: Off sub clock

MAINOFF Main clock control bit
0: On main clock
1: Off main clock

17.2 Stop Mode

In the Stop mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Control registers are held. The port pins out the values held by their respective port data register, port direction registers. Oscillator stops and the systems internal operations are all held up.

- The states of the RAM, registers, and latches valid immediately before the system is put in the STOP state are all held.
- The program counter stop the address of the instruction to be executed after the instruction "STOP" which starts the STOP operating mode.

The Stop mode is activated by execution of STOP instruction after clearing the bit WAKEUP of CKCTLR to “0”. (This register should be written by byte operation. If this register is set by bit manipulation instruction, for example "set1" or "clr1" instruction, it may be undesired operation)

In the Stop mode of operation, V_{DD} can be reduced to minimize power consumption. Care must be taken, however, to ensure 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.

Note: *After STOP instruction, at least two or more NOP instruction should be written*

```
Ex)   LDM CKCTLR,#0000_1110B
      STOP
      NOP
      NOP
```

In the STOP operation, the dissipation of the power 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 gets higher than the power voltage level (by approximately 0.3 to 0.5V), 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 to fix the level by pull-up or other means.

Release the STOP mode

The exit from STOP mode is hardware reset or external interrupt. Reset re-defines 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.

If I-flag = 1, the normal interrupt response takes place. If I-flag = 0, the chip will resume execution starting with the instruction following the STOP instruction. It will not vector to interrupt service routine. (refer to Figure 17-1)

When exit from Stop mode by external interrupt, enough oscillation stabilization time is required to normal operation. Figure 17-2 shows the timing diagram. When release the Stop mode, the Basic interval timer is activated on wake-up. It is increased from 00_H until FF_H. The count overflow is set to start normal operation. Therefore, before STOP instruction, user must be set its relevant prescaler divide ratio to have long enough time (more than 20msec). This guarantees that oscillator has started and stabilized.

By reset, exit from Stop mode is shown in Figure .

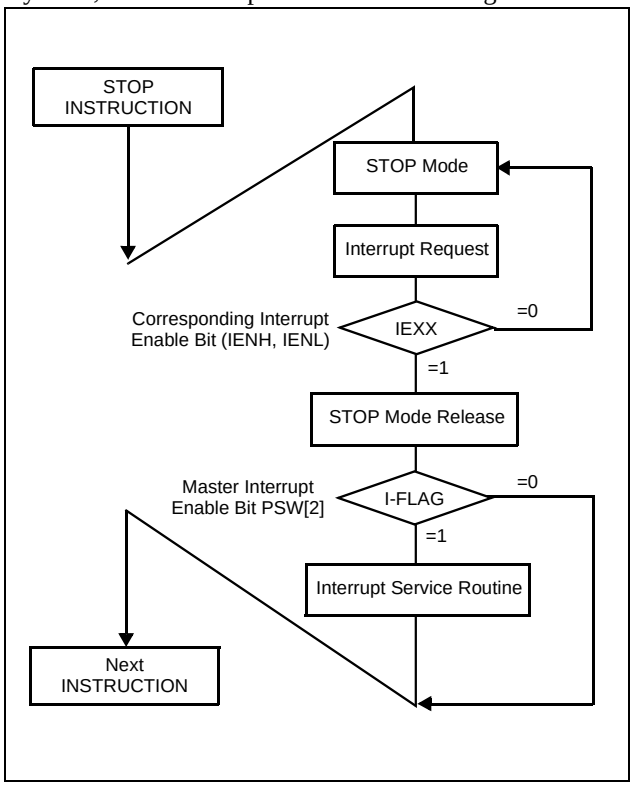


Figure 17-1 STOP Releasing Flow by Interrupts

.

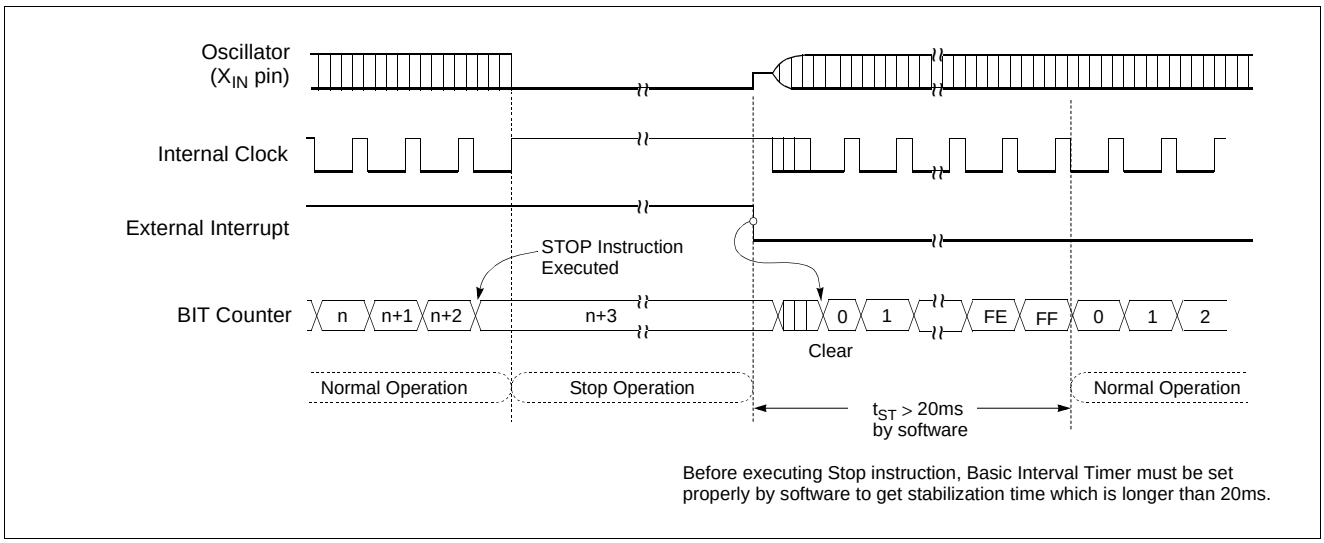


Figure 17-2 STOP Mode Release Timing by External Interrupt

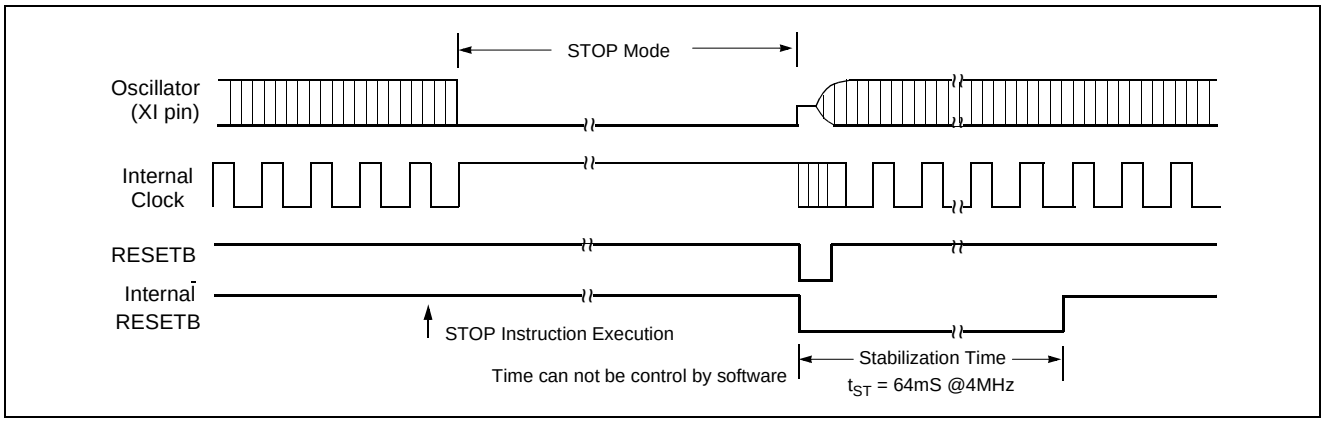


Figure 17-3 Timing of STOP Mode Release by RESET

17.3 Wake-up Timer Mode

In the Wake-up Timer mode, the on-chip oscillator is not stopped. Except the Prescaler(only 2048 divided ratio) and Timer0, all functions are stopped, but the on-chip RAM and Control registers are held. The port pins out the values held by their respective port data register, port direction registers.

The Wake-up Timer mode is activated by execution of STOP instruction after setting the bit WAKEUP of CKCTLR to "1". (This register should be written by byte operation. If this register is set by bit manipulation instruction, for example "set1" or "clr1" instruction, it may be undesired operation)

Note: *After STOP instruction, at least two or more NOP instruction should be written*

```
Ex)     LDM TDR0,#0FFH
        LDM TM0,#0001_1011B
        LDM CKCTLR,#0100_1110B
        STOP
        NOP
        NOP
```

In addition, the clock source of timer0 should be selected to 2048 divided ratio. Otherwise, the wake-up function can not work. And the timer0 can be operated as 16-bit timer with timer1. (refer to timer function)The period of wake-up function is varied by setting the timer data register 0, TDR0.

Release the Wake-up Timer mode

The exit from Wake-up Timer mode is hardware reset, Timer0 overflow or external interrupt. Reset re-defines all the Control registers but does not change the on-chip RAM. External interrupts and Timer0 overflow allow both on-chip RAM and Control registers to retain their values.

If I-flag = 1, the normal interrupt response takes place. If I-flag = 0, the chip will resume execution starting with the instruction following the STOP instruction. It will not vector to interrupt service routine.(refer to Figure 17-1)

When exit from Wake-up Timer mode by external interrupt or timer0 overflow, the oscillation stabilization time is not required to normal operation. Because this mode do not stop the on-chip oscillator shown as Figure 17-4.

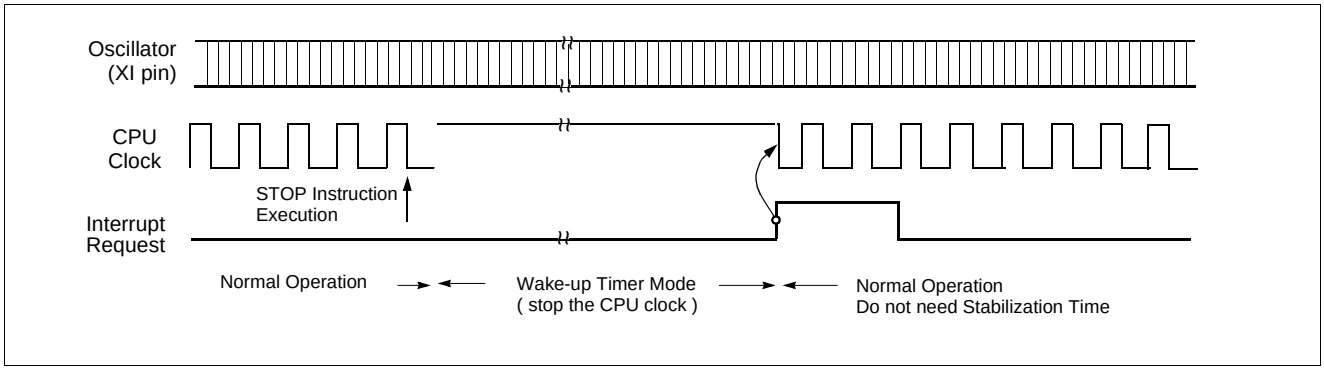


Figure 17-4 Wake-up Timer Mode Releasing by External Interrupt or Timer0 Interrupt

17.4 Internal RC-Oscillated Watchdog Timer Mode

In the Internal RC-Oscillated Watchdog Timer mode, the on-chip oscillator is stopped. But internal RC oscillation circuit is oscillated in this mode. The on-chip RAM and Control registers are held. The port pins out the values held by their respective port data register, port direction registers.

The Internal RC-Oscillated Watchdog Timer mode is activated by execution of STOP instruction after setting the bit WAKEUP and RCWDT of CKCTLR to " 01 ". (This register should be written by byte operation. If this register is set by bit manipulation instruction, for example "set1" or "clr1" instruction, it may be undesired operation)

Note: *Caution: After STOP instruction, at least two or more NOP instruction should be written*

```
Ex)   LDM  WDTR,#1111_1111B
      LDM  CKCTLR,#001_1110B
      STOP
      NOP
      NOP
```

The exit from Internal RC-Oscillated Watchdog Timer mode is hardware reset or external interrupt. Reset re-defines 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.

If I-flag = 1, the normal interrupt response takes place. In this case, if the bit WDTON of CKCTLR is set to "0" and the bit WDTE of IENH is set to "1", the device will execute the watchdog timer interrupt service routine.(Figure 17-5) However, if the bit WDTON of CKCTLR is set to "1", the device will generate the internal RESET signal and execute the reset processing. (Figure 17-6)

If I-flag = 0, the chip will resume execution starting with the instruction following the STOP instruction. It will not vector to interrupt service routine.(refer to Figure 17-1)

When exit from Internal RC-Oscillated Watchdog Timer mode by external interrupt, the oscillation stabilization time is required to normal operation. Figure 17-5 shows the timing diagram. When release the Internal RC-Oscillated Watchdog Timer mode, the basic interval timer is activated on wake-up. It is increased from 00_H until FF_H. The count overflow is set to start normal operation. Therefore, before STOP instruction, user must be set its relevant prescaler divide ratio to have long enough time (more than 20msec). This guarantees that oscillator has started and stabilized.

By reset, exit from internal RC-Oscillated Watchdog Timer mode is shown in Figure 17-6.

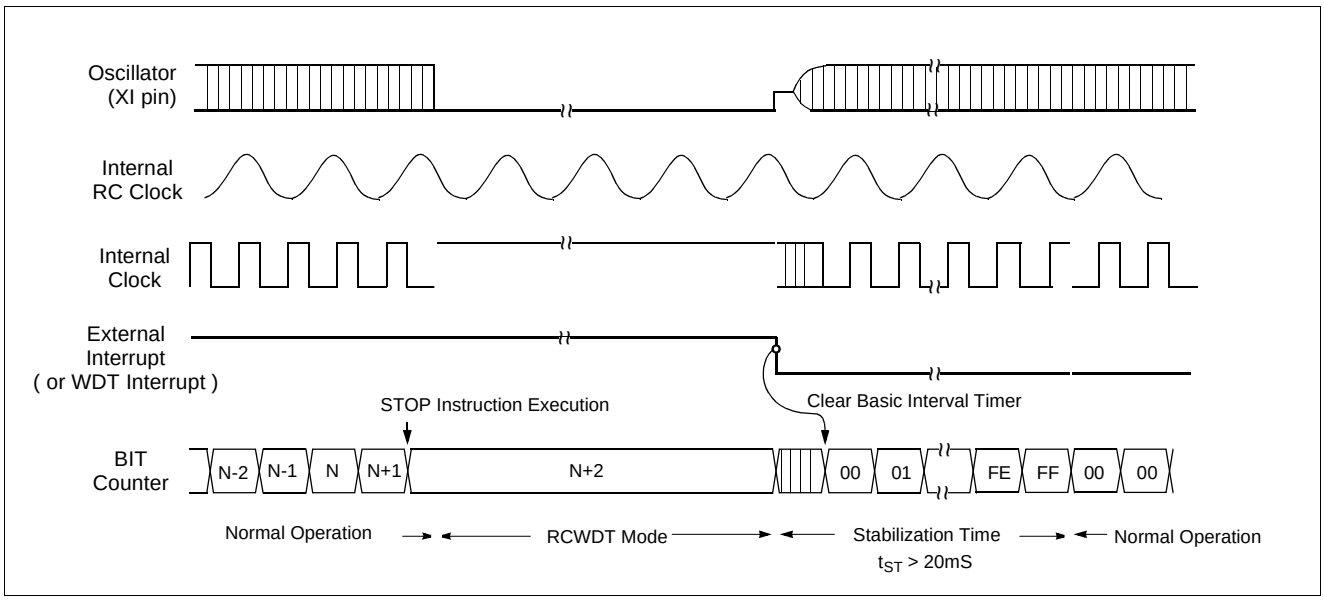


Figure 17-5 Internal RCWDT Mode Releasing by External Interrupt or WDT Interrupt

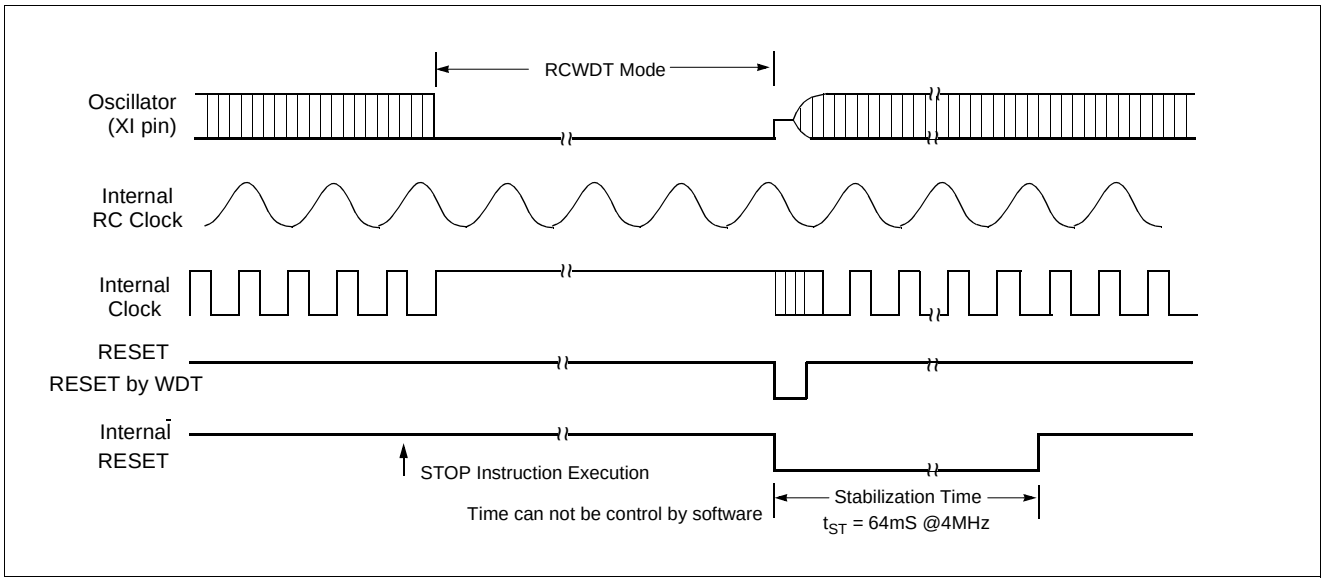


Figure 17-6 Internal RCWDT Mode Releasing by RESET

17.5 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*

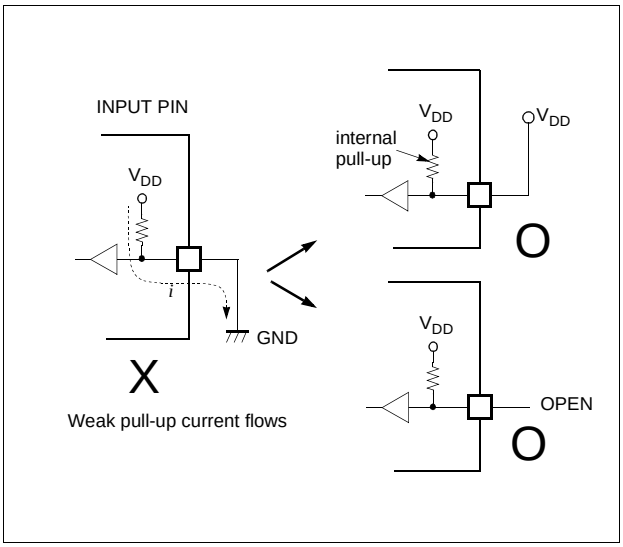
than the power voltage level (by approximately 0.3V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port puts the pin signal into the high-impedance state, a current flow across the ports input transistor, requiring it to fix the level by pull-up or other means.

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

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

But input voltage level should be V_{SS} or V_{DD}. Be careful that if unspecified voltage, i.e. if unfirmed voltage level (not V_{SS}or V_{DD}) is applied to input pin, there can be little current (max. 1mA at around 2V) flow.

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



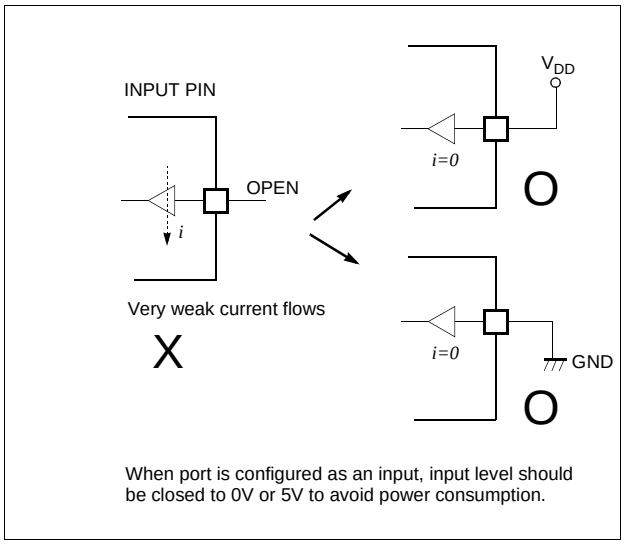
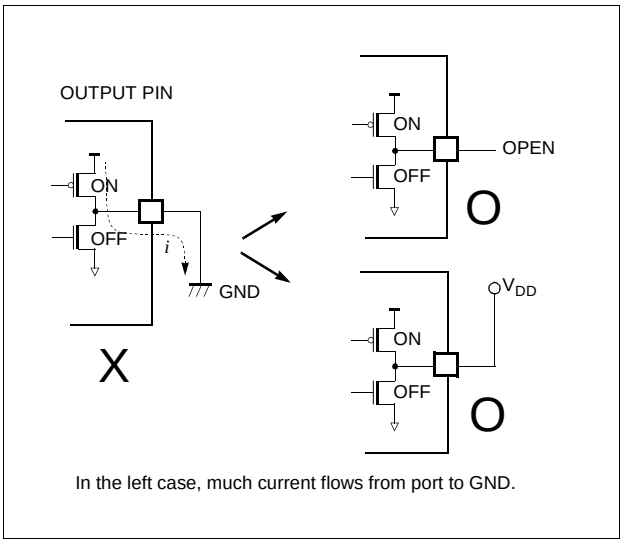


Figure 17-7 Application Example of Unused Input Port



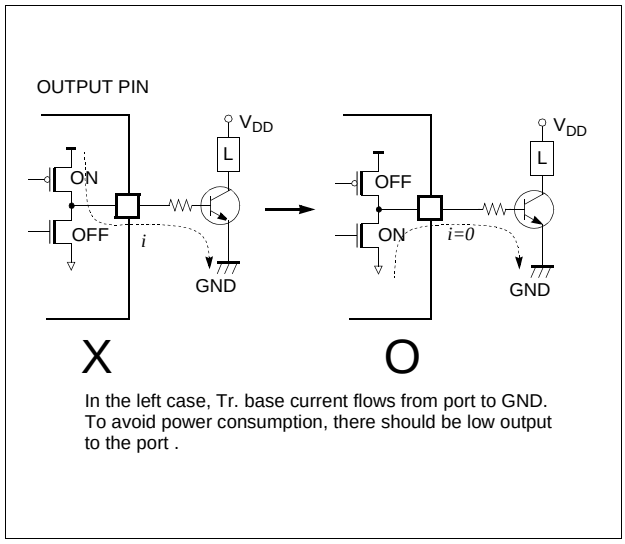


Figure 17-8 Application Example of Unused Output Port

18. OSCILLATOR CIRCUIT

The GMS81C20xx 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 18-1.

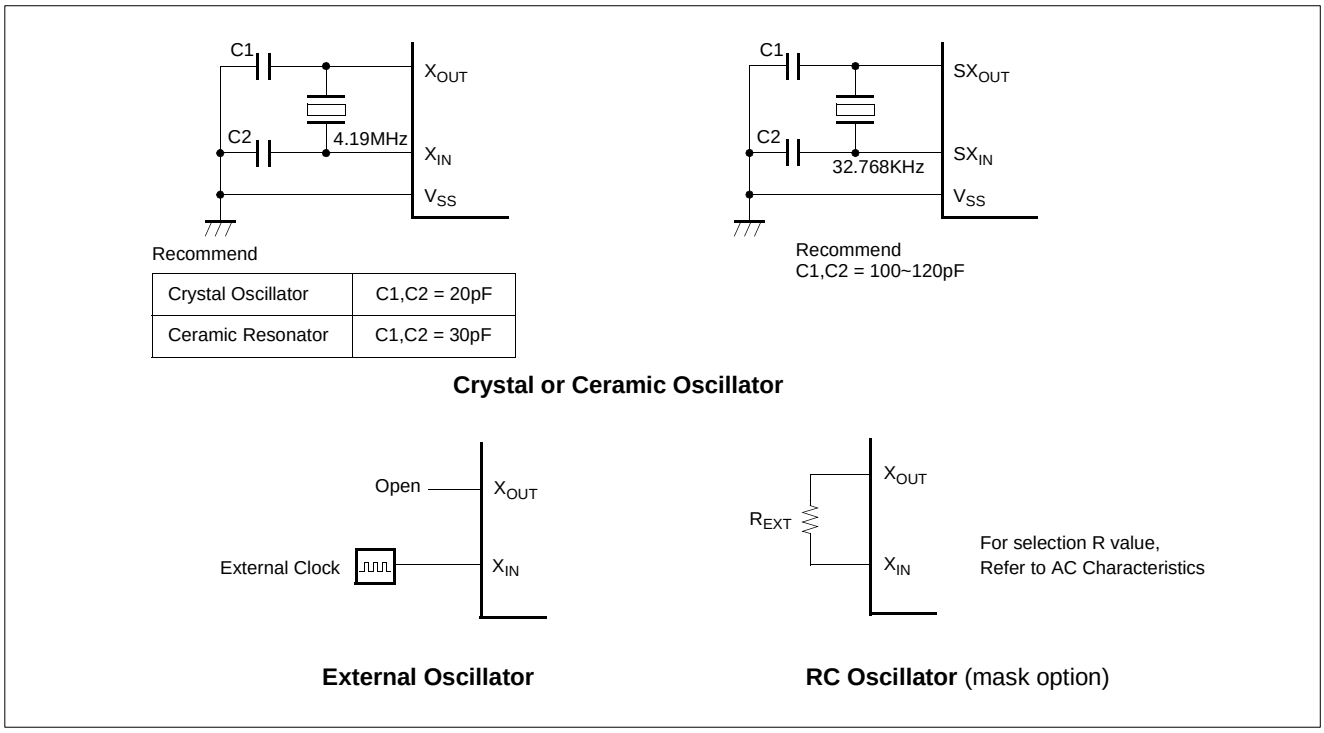


Figure 18-1 Oscillation Circuit

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 18-2 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.*

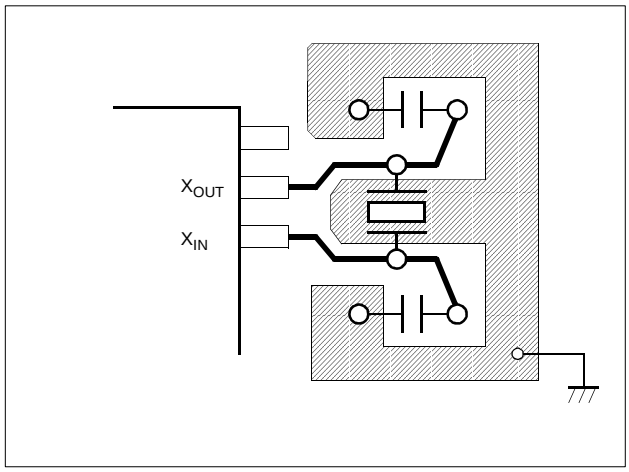


Figure 18-2 Layout of Oscillator PCB circuit

19. RESET

The GMS81C20xx have two types of reset generation procedures; one is an external reset input, the other is a watchdog timer reset. Table 19-1 shows on-chip hardware initialization by reset action.

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	($FFFF_H$) - ($FFFE_H$)	Peripheral clock	Off
RAM page register (RPR)	0	Watchdog timer	Disable
G-flag (G)	0	Control registers	Refer to Table 8-1 on page 32
Operation mode	Main-frequency clock	Power fail detector	Disable

Table 19-1 Initializing Internal Status by Reset Action

19.1 External Reset Input

The reset input is the RESET pin, which is the input to a Schmitt Trigger. A reset in accomplished by holding the RESET pin low for at least 8 oscillator periods, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset, 64ms (at 4 MHz) add with 7 oscillator periods are required to start execution as shown in Figure 19-2.

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 19-1.

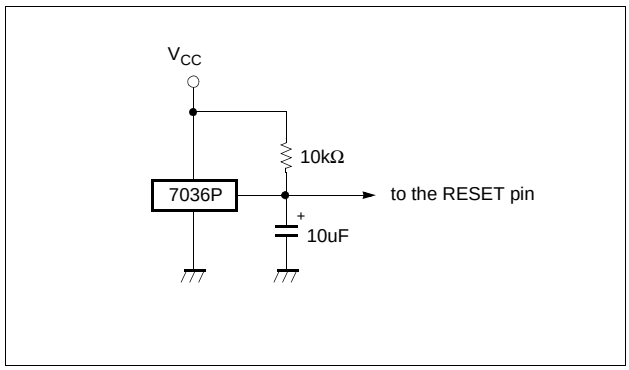


Figure 19-1 Simple Power-on-Reset Circuit

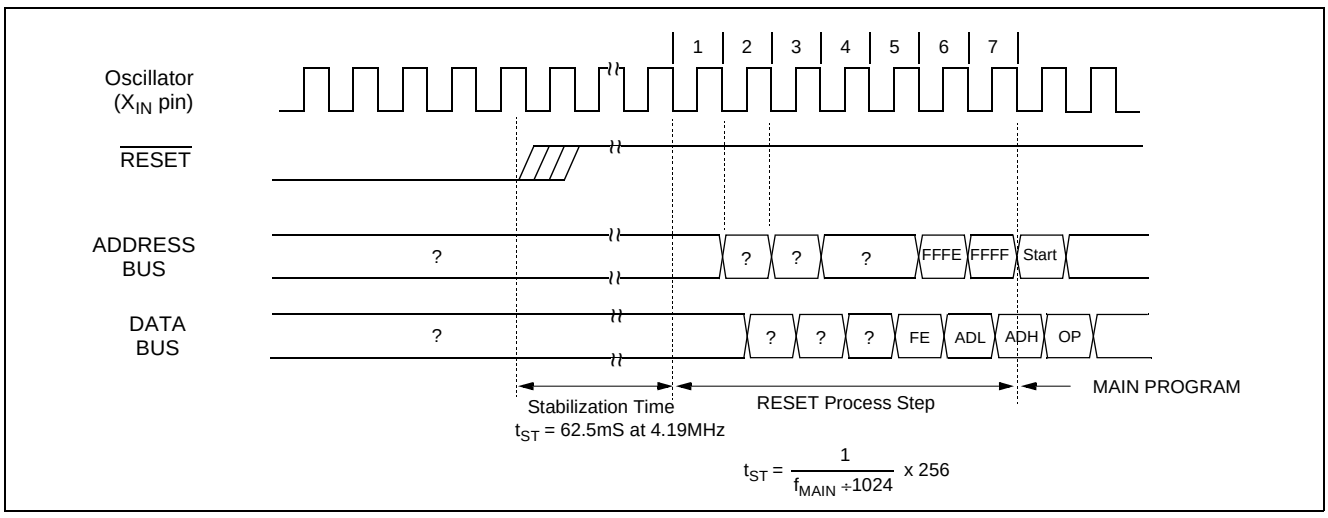


Figure 19-2 Timing Diagram after RESET

19.2 Watchdog Timer Reset

Refer to “11. WATCHDOG TIMER” on page 45.

20. POWER FAIL PROCESSOR

The GMS81C20xx has an on-chip power fail detection circuitry to immunize against power noise. A configuration register, PFDR, can enable or disable the power fail detect circuitry. Whenever V_{DD} falls close to or below power fail voltage for 100ns, the power fail situation may reset or freeze MCU according to PFDM bit of PFDR. Refer to "7.4 DC Electrical Characteristics for Standard Pins(5V)" on page 19.

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

Note: *User can select power fail voltage level according to PFD0, PFD1 bit of CONFIG register($703F_H$) at the OTP (GMS87C20xx) but* ***must select*** *the power fail voltage level to define PFD option of "Mask Order & Verification Sheet" at the mask chip(GMS81C20xx).*
Because the power fail voltage level of mask chip (GMS81C20xx) is determined according to mask option.

Note: *If power fail voltage is selected to 3.0V on 3V operation, MCU is freezed at all the times.*

Power FailFunction	OTP	MASK
Enable/Disable	PFD flag	PFD flag
Level Selection	PFS0 bit PFS1 bit	Mask option

Table 20-1 Power fail processor

.

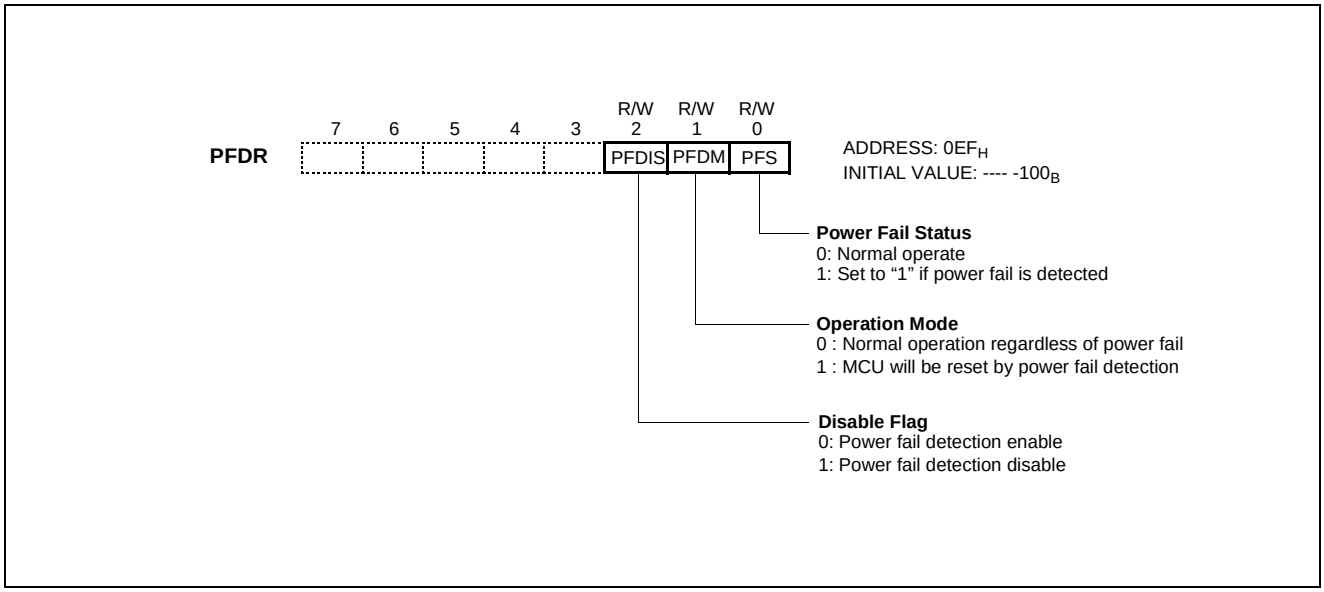


Figure 20-1 Power Fail Voltage Detector Register

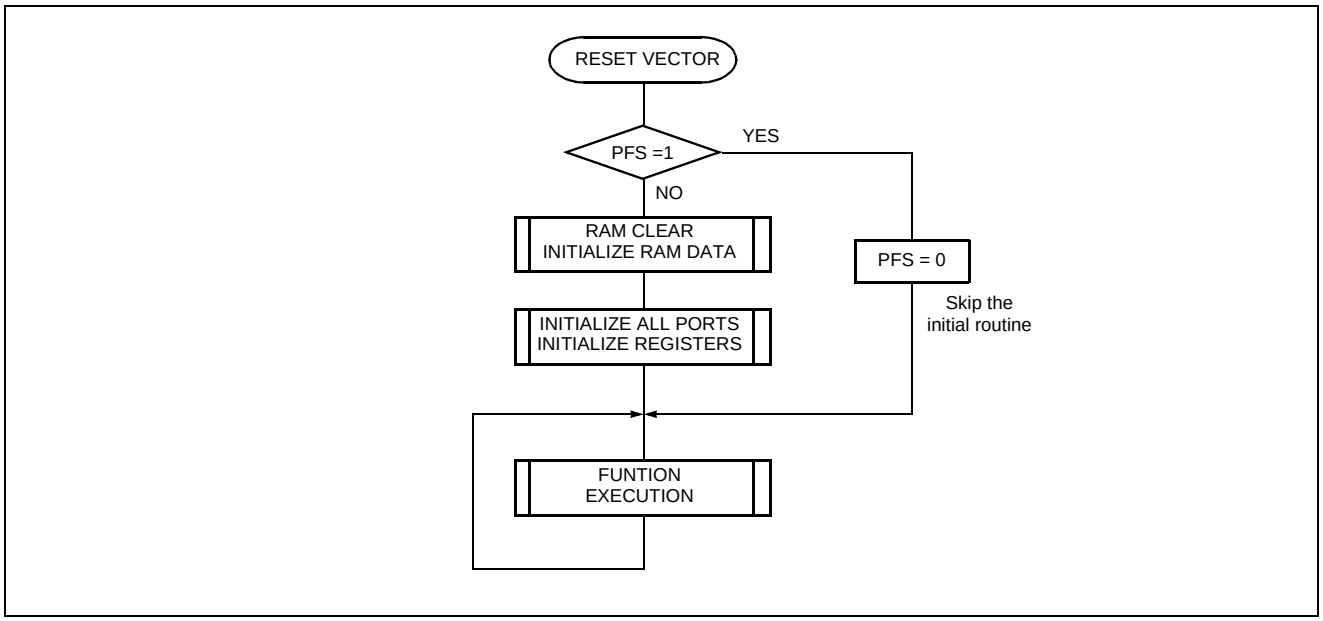


Figure 20-2 Example S/W of RESET flow by Power fail

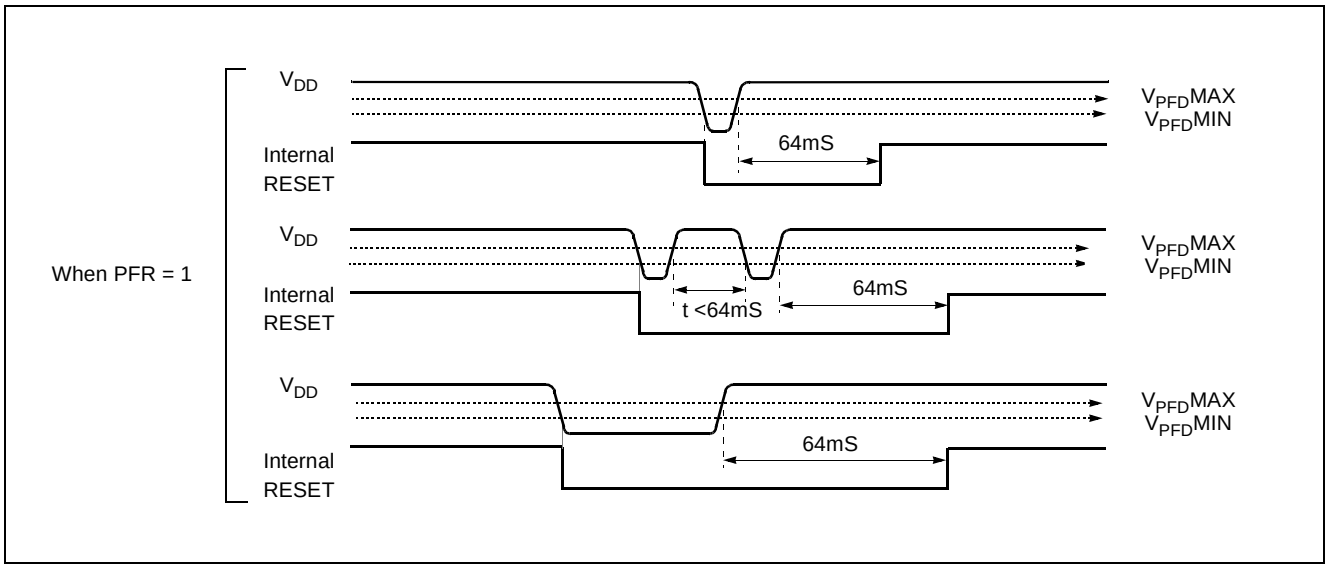


Figure 20-3 Power Fail Processor Situations

21. OTP PROGRAMMING

21.1 DEVICE CONFIGURATION AREA

The Device Configuration Area can be programmed or left unprogrammed to select device configuration such as security bit.

Sixteen memory locations (7030_H ~ $703F_H$) are designated as Customer ID recording locations where the user can store check-sum or other customer identification numbers. This area is not accessible during normal execution but is readable and writable during program / verify.

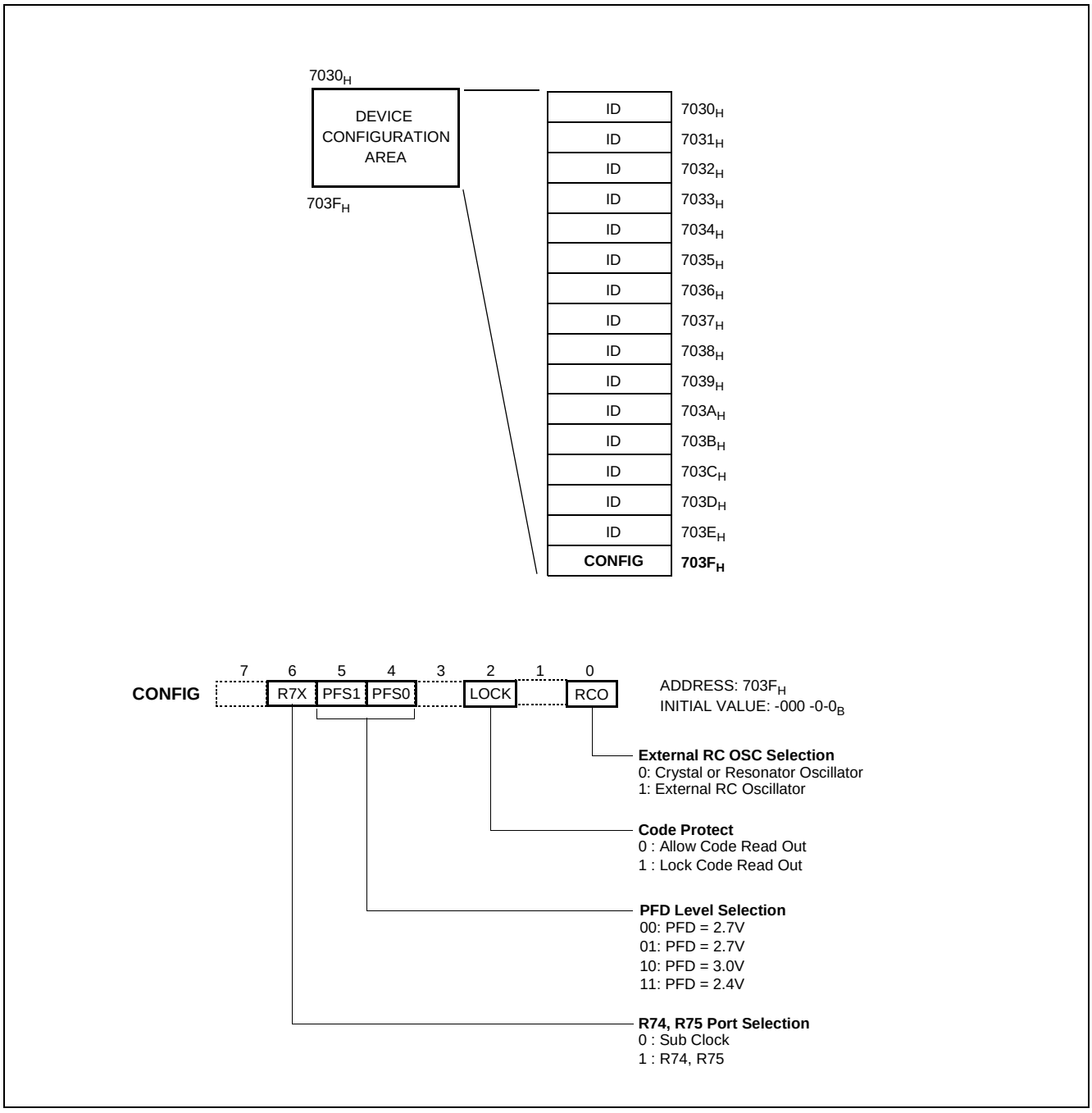


Figure 21-1 Device Configuration Area

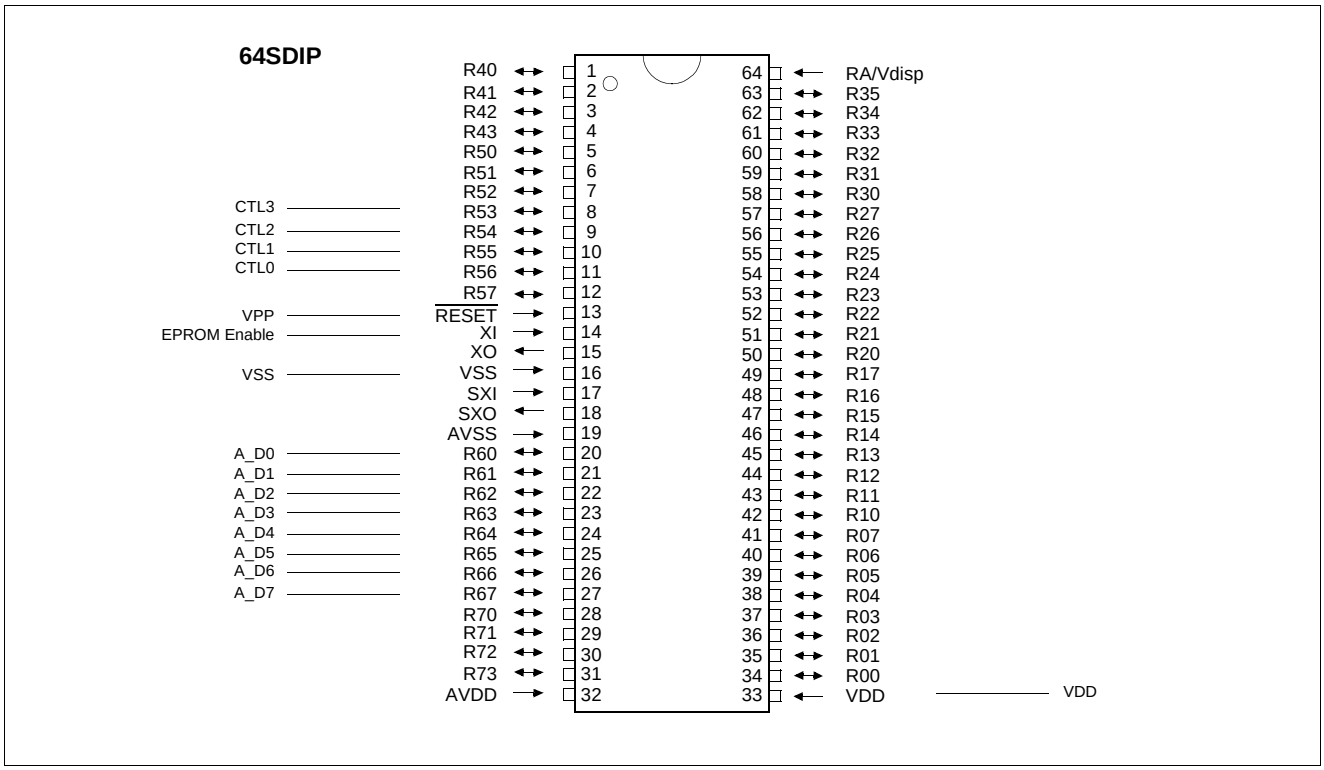


Figure 21-2 Pin Assignment (64SDIP)

Pin No.	User Mode	EPROM MODE				
	Pin Name	Pin Name	Description			
8	R53	CTL3	Read/Write Control			**P_Vb**
9	R54	CTL2	Address/Data Control			**D_Ab**
10	R55	CTL1	Write Control 1			
11	R56	CTL0	Write Control 0			
13	$\overline{\text{RESET}}$	VPP	Programming Power (0V, 12V)			
14	XIN	EPROM Enable	High Active, Latch Address in falling edge			
15	XOUT	NC	No connection			
16	VSS	VSS	Connect to VSS (0V)			
20	R60	A_D0	Address Input Data Input/Output	A8	A0	D0
21	R61	A_D1		A9	A1	D1
22	R62	A_D2		A10	A2	D2
23	R63	A_D3		A11	A3	D3
24	R64	A_D4	Address Input Data Input/Output	A12	A4	D4
25	R65	A_D5		A13	A5	D5
26	R66	A_D6		A14	A6	D6
27	R67	A_D7		A15	A7	D7
33	VDD	VDD	Connect to VDD (6.0V)			

Table 21-1 Pin Description in EPROM Mode (GMS81C2020)

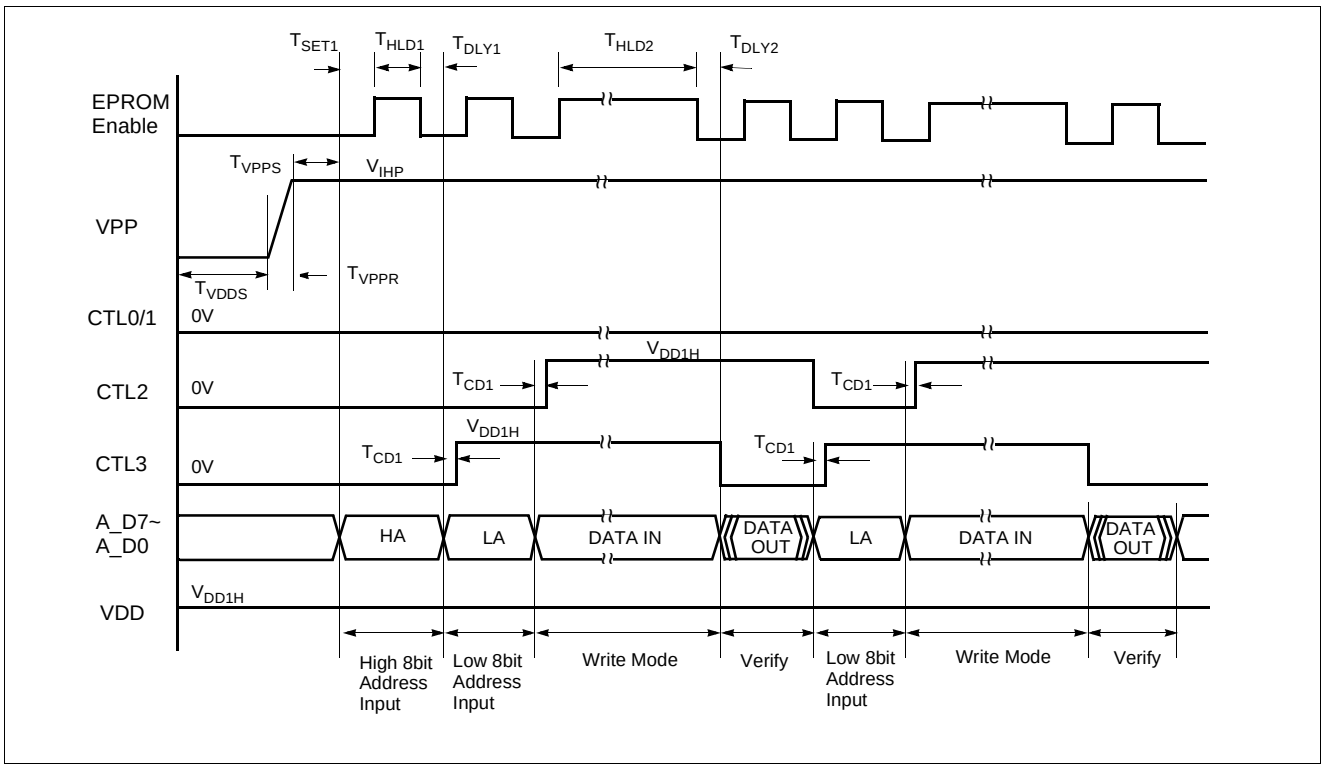


Figure 21-3 Timing Diagram in Program (Write & Verify) Mode

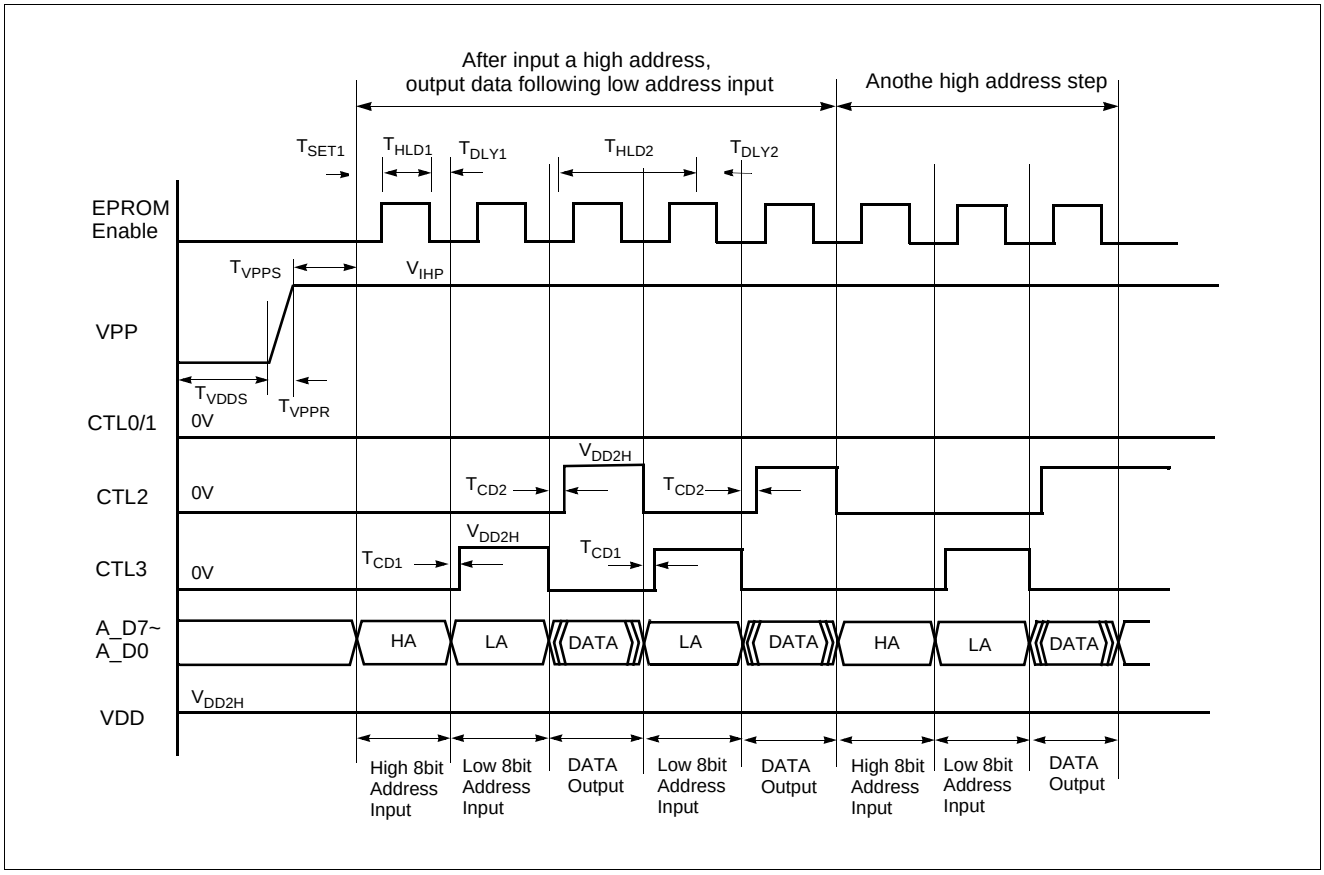


Figure 21-4 Timing Diagram in READ Mode

Parameter	Symbol	MIN	TYP	MAX	Unit
Programming Supply Current	I_{VPP}	-	-	50	mA
Supply Current in EPROM Mode	I_{VDDP}	-	-	20	mA
VPP Level during Programming	V_{IHP}	11.5	12.0	12.5	V
VDD Level in Program Mode	V_{DD1H}	5	6	6.5	V
VDD Level in Read Mode	V_{DD2H}	-	2.7	-	V
CTL3~0 High Level in EPROM Mode	V_{IHC}	$0.8V_{DD}$	-	-	V
CTL3~0 Low Level in EPROM Mode	V_{ILC}	-	-	$0.2V_{DD}$	V
A_D7~A_D0 High Level in EPROM Mode	V_{IHAD}	$0.9V_{DD}$	-	-	V
A_D7~A_D0 Low Level in EPROM Mode	V_{ILAD}	-	-	$0.1V_{DD}$	V
VDD Saturation Time	T_{VDDS}	1	-	-	mS
VPP Setup Time	T_{VPPR}	-	-	1	mS
VPP Saturation Time	T_{VPPS}	1	-	-	mS
EPROM Enable Setup Time after Data Input	T_{SET1}		200		nS
EPROM Enable Hold Time after T_{SET1}	T_{HLD1}		500		nS

Table 21-2 AC/DC Requirements for Program/Read Mode

EPROM Enable Delay Time after T_{HLD1}	T_{DLY1}		200		nS
EPROM Enable Hold Time in Write Mode	T_{HLD2}		100		nS
EPROM Enable Delay Time after T_{HLD2}	T_{DLY2}		200		nS
CTL2,1 Setup Time after Low Address input and Data input	T_{CD1}		100		nS
CTL1 Setup Time before Data output in Read and Verify Mode	T_{CD2}		100		nS

Table 21-2 AC/DC Requirements for Program/Read Mode

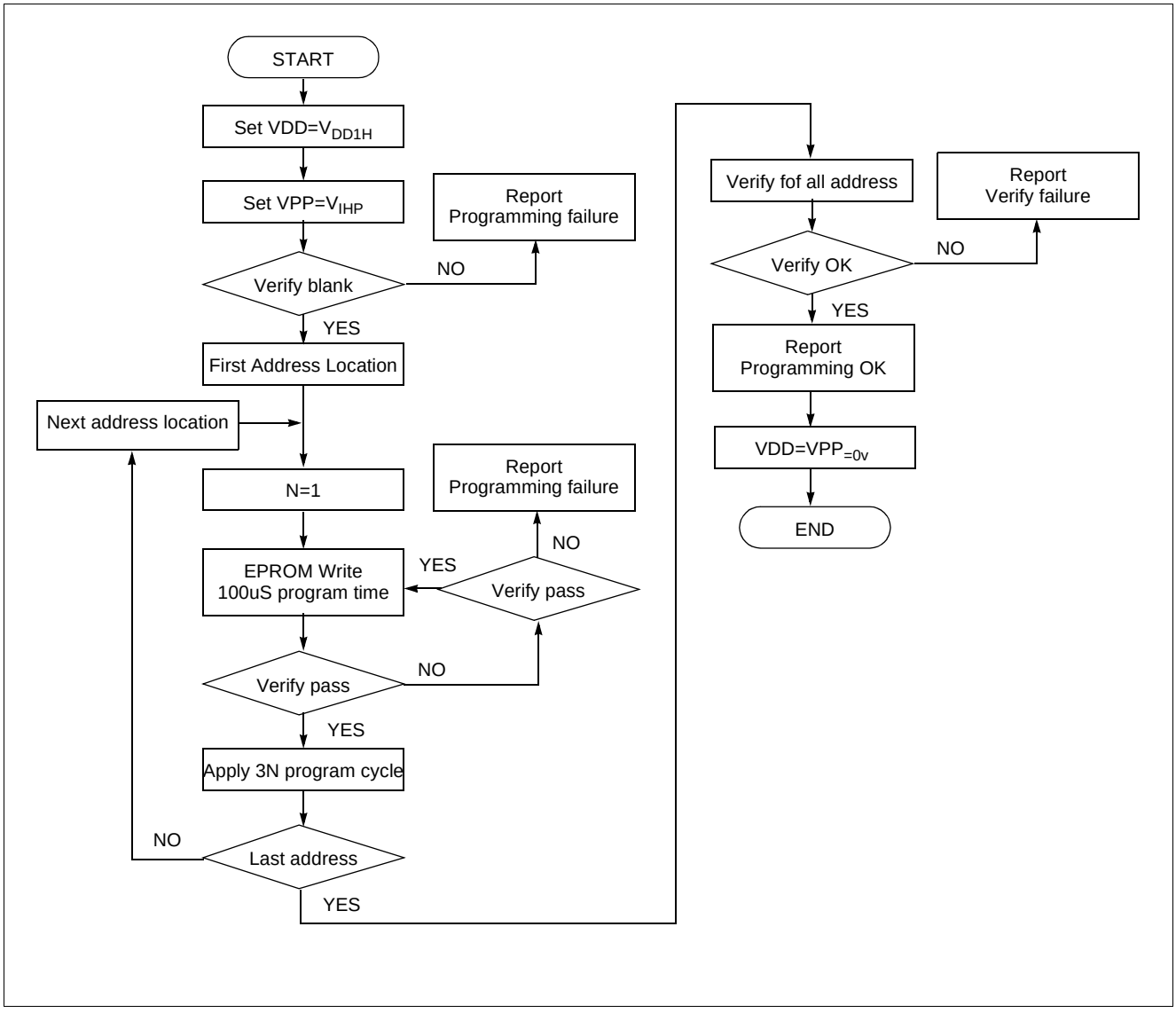
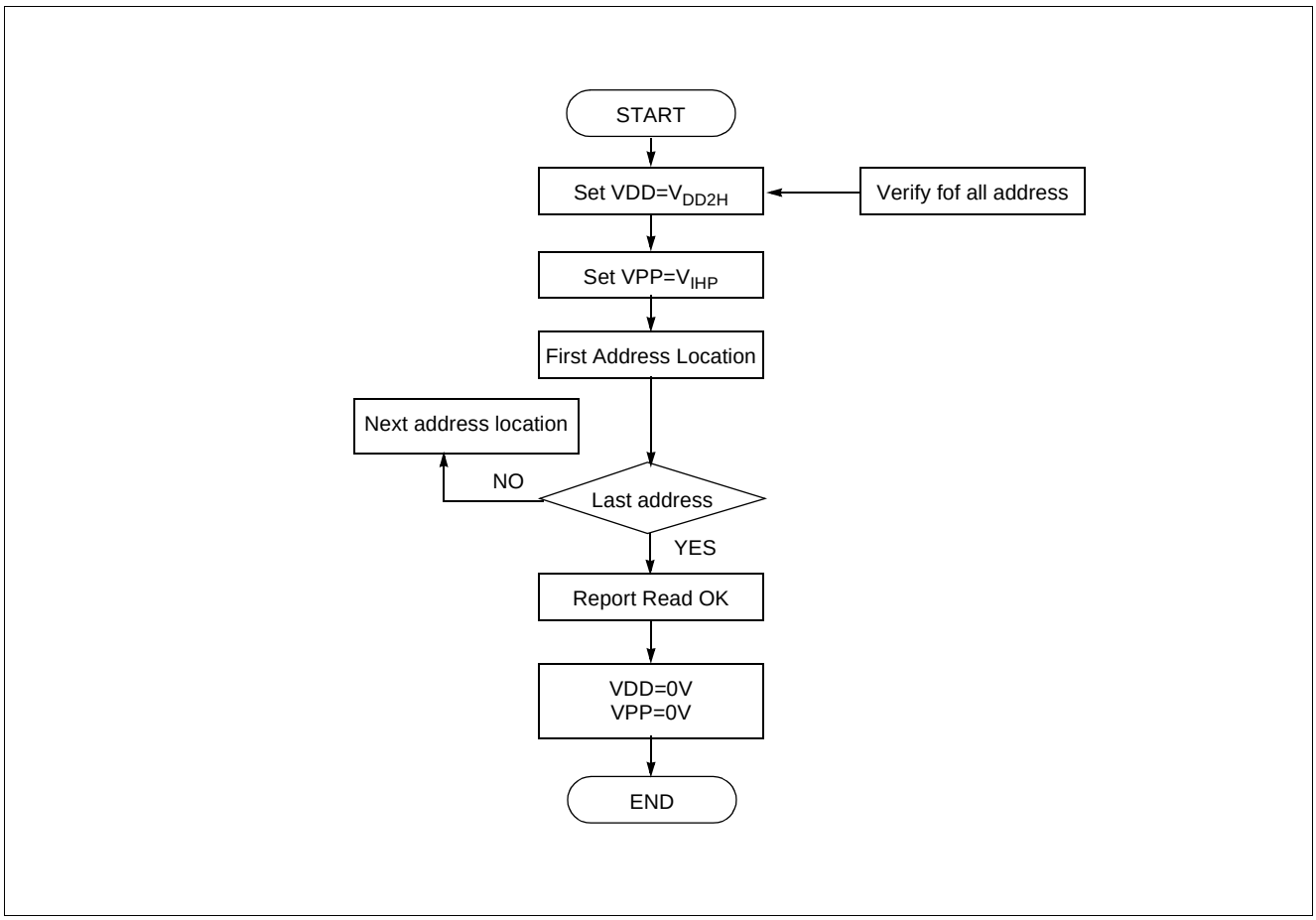


Figure 21-5 Programming Flow Chart


START
Set VDD=V_{DD2H}
Verify fof all address
Set VPP=V_{IHP}
First Address Location
Next address location
NO
Last address
YES
Report Read OK
VDD=0V
VPP=0V
END

APPENDIX

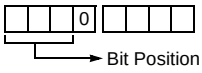
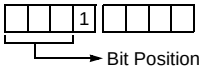
A. CONTROL REGISTER LIST

Address	Register Name	Symbol	R/W	Initial Value								Page
				7	6	5	4	3	2	1	0	
00C0	R0 port data register	R0	R/W	Undefined								39
00C1	R0 port I/O direction register	R0IO	W	0	0	0	0	0	0	0	0	39
00C2	R1 port data register	R1	R/W	Undefined								40
00C3	R1 port I/O direction register	R1IO	W	0	0	0	0	0	0	0	0	40
00C4	R2 port data register	R2	R/W	Undefined								40
00C5	R2 port I/O direction register	R2IO	W	0	0	0	0	0	0	0	0	40
00C6	R3 port data register	R3	R/W	Undefined								40
00C7	R3 port I/O direction register	R3IO	W	-	-	0	0	0	0	0	0	40
00C8	R4 port data register	R4	R/W	Undefined								40
00C9	R4 port I/O direction register	R4IO	W	-	-	-	-	0	0	0	0	40
00CA	R5 port data register	R5	R/W	Undefined								41
00CB	R5 port I/O direction register	R5IO	W	0	0	0	0	0	0	0	0	41
00CC	R6 port data register	R6	R/W	Undefined								41
00CD	R6 port I/O direction register	R6IO	W	0	0	0	0	0	0	0	0	41
00CE	R7 port data register	R7	R/W	Undefined								42
00CF	R7 port I/O direction register	R7IO	W	-	-	0	0	0	0	0	0	42
00D0	Timer mode register 0	TM0	R/W	-	-	0	0	0	0	0	0	49
00D1	Timer 0 register	T0	R	0	0	0	0	0	0	0	0	50
	Timer 0 data register	TDR0	W	1	1	1	1	1	1	1	1	49
	Capture 0 data register	CDR0	R	0	0	0	0	0	0	0	0	56
00D2	Timer mode register 1	TM1	R/W	0	0	0	0	0	0	0	0	49
00D3	Timer 1 data register	TDR1	W	1	1	1	1	1	1	1	1	49
	PWM 1 period register	T1PPR	W	1	1	1	1	1	1	1	1	61
00D4	Timer 1 register	T1	R	0	0	0	0	0	0	0	0	50
	PWM 1 duty register	T1PDR	R/W	0	0	0	0	0	0	0	0	61
	Capture 1 data register	CDR1	R	0	0	0	0	0	0	0	0	56
00D5	PWM 1 High register	PWM1HR	W	-	-	-	-	0	0	0	0	60
00DE	Buzzer driver register	BUR	W	1	1	1	1	1	1	1	1	69
00E0	Serial I/O mode register	SIOM	R/W	0	0	0	0	0	0	0	1	66
00E1	Serial I/O data register	SIOR	R/W	Undefined								66
00E2	Interrupt enable register high	IENH	R/W	0	0	0	0	-	-	-	-	72
00E3	Interrupt enable register low	IENL	R/W	0	0	0	0	-	-	-	-	72
00E4	Interrupt request flag register high	IRQH	R/W	0	0	0	0	-	-	-	-	71
00E5	Interrupt request flag register low	IRQL	R/W	0	0	0	0	-	-	-	-	71
00E6	External interrupt edge selection register	IEDS	R/W	-	-	-	-	0	0	0	0	77
00EA	A/D converter mode register	ADCM	R/W	-	0	0	0	0	0	0	1	62

Address	Register Name	Symbol	R/W	Initial Value								Page
				7	6	5	4	3	2	1	0	
00EB	A/D converter data register	ADCR	R	Undefined								62
00EC	Basic interval timer mode register	BITR	R	0	0	0	0	0	0	0	0	44
	Clock control register	CKCTLR	W	-	0	0	1	0	1	1	1	44
00ED	Watchdog Timer Register	WDTR	R	0	0	0	0	0	0	0	0	46
	Watchdog Timer Register	WDTR	W	0	1	1	1	1	1	1	1	46
00EF	Power fail detection register	PFDR	R/W	-	-	-	-	-	1	0	0	87
00F4	R0 Function selection register	R0FUNC	W	-	-	-	-	0	0	0	0	39
00F5	R4 Function selection register	R4FUNC	W	-	-	-	-	-	-	-	0	40
00F6	R5 Function selection register	R5FUNC	W	-	0	-	-	-	-	-	-	41
00F7	R6 Function selection register	R6FUNC	W	0	0	0	0	0	0	0	0	41
00F8	R7 Function selection register	R7FUNC	W	-	-	-	-	0	0	0	0	42
00F9	R5 N-MOS open drain selection register	R5MPDR	W	0	0	0	0	0	0	0	0	41
00FA	System clock mode register	SCMR	R/W	-	-	-	0	0	0	0	0	79
00FB	RA port data register	RA	R	Undefined								39

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 (000_H~$0FFF_H$)
rel	Relative Addressing Data
upage	U-page ($0FF00_H$~$0FFFF_H$) Offset Address
n	Table CALL Number (0~15)
+	Addition
x	 Upper Nibble Expression in Opcode
y	 Upper Nibble Expression in Opcode
–	Subtraction
×	Multiplication
/	Division
( )	Contents Expression
∧	AND
∨	OR
⊕	Exclusive OR
~	NOT
←	Assignment / Transfer / Shift Left
→	Shift Right
↔	Exchange
=	Equal
≠	Not Equal

B.2 Instruction Map

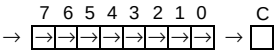
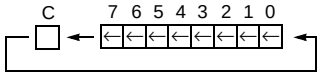
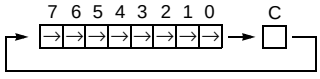
LOW / HIGH	00000 00	00001 01	00010 02	00011 03	00100 04	00101 05	00110 06	00111 07	01000 08	01001 09	01010 0A	01011 0B	01100 0C	01101 0D	01110 0E	01111 0F
000	-	SET1 dp.bit	BBS A.bit,rel	BBS dp.bit,rel	ADC #imm	ADC dp	ADC dp+X	ADC !abs	ASL A	ASL dp	TCALL 0	SETA1 .bit	BIT dp	POP A	PUSH A	BRK
001	CLRC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

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

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.	
2	ADC dp	05	2	3	A ← (A) + (M) + C	
3	ADC dp + X	06	2	4		
4	ADC !abs	07	3	4		NV--H-ZC
5	ADC !abs + Y	15	3	5		
6	ADC [dp + X]	16	2	6		
7	ADC [dp] + Y	17	2	6		
8	ADC { X }	14	1	3		
9	AND #imm	84	2	2	Logical AND	
10	AND dp	85	2	3	A ← (A) ∧ (M)	
11	AND dp + X	86	2	4		
12	AND !abs	87	3	4		N-----Z-
13	AND !abs + Y	95	3	5		
14	AND [dp + X]	96	2	6		
15	AND [dp] + Y	97	2	6		
16	AND { X }	94	1	3		
17	ASL A	08	1	2	Arithmetic shift left	
18	ASL dp	09	2	4	C 7 6 5 4 3 2 1 0 ← “0”	N-----ZC
19	ASL dp + X	19	2	5		
20	ASL !abs	18	3	5		
21	CMP #imm	44	2	2		
22	CMP dp	45	2	3		
23	CMP dp + X	46	2	4		
24	CMP !abs	47	3	4	Compare accumulator contents with memory contents	N-----ZC
25	CMP !abs + Y	55	3	5	(A) - (M)	
26	CMP [dp + X]	56	2	6		
27	CMP [dp] + Y	57	2	6		
28	CMP { X }	54	1	3		
29	CMPX #imm	5E	2	2	Compare X contents with memory contents	
30	CMPX dp	6C	2	3	(X) - (M)	N-----ZC
31	CMPX !abs	7C	3	4		
32	CMPY #imm	7E	2	2	Compare Y contents with memory contents	
33	CMPY dp	8C	2	3	(Y) - (M)	N-----ZC
34	CMPY !abs	9C	3	4		
35	COM dp	2C	2	4	1’S Complement : (dp) ← ~(dp)	N-----Z-
36	DAA	DF	1	3	Decimal adjust for addition	N-----ZC
37	DAS	CF	1	3	Decimal adjust for subtraction	N-----ZC
38	DEC A	A8	1	2	Decrement	N-----Z-
39	DEC dp	A9	2	4	M ← (M) - 1	N-----Z-
40	DEC dp + X	B9	2	5		N-----Z-
41	DEC !abs	B8	3	5		N-----Z-
42	DEC X	AF	1	2		N-----Z-
43	DEC Y	BE	1	2		N-----Z-

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
44	DIV	9B	1	12	Divide : YA / X Q: A, R: Y	NV--H-Z-
45	EOR #imm	A4	2	2	Exclusive OR	
46	EOR dp	A5	2	3	A ← (A) ⊕ (M)	
47	EOR dp + X	A6	2	4		
48	EOR !abs	A7	3	4		N-----Z-
49	EOR !abs + Y	B5	3	5		
50	EOR [dp + X]	B6	2	6		
51	EOR [dp] + Y	B7	2	6		
52	EOR { X }	B4	1	3		
53	INC A	88	1	2	Increment	N-----ZC
54	INC dp	89	2	4	M ← (M) + 1	N-----Z-
55	INC dp + X	99	2	5		N-----Z-
56	INC !abs	98	3	5		N-----Z-
57	INC X	8F	1	2		N-----Z-
58	INC Y	9E	1	2		N-----Z-
59	LSR A	48	1	2	Logical shift right	
60	LSR dp	49	2	4		N-----ZC
61	LSR dp + X	59	2	5		
62	LSR !abs	58	3	5		
63	MUL	5B	1	9	Multiply : YA ← Y × A	N-----Z-
64	OR #imm	64	2	2	Logical OR	
65	OR dp	65	2	3	A ← (A) ∨ (M)	
66	OR dp + X	66	2	4		
67	OR !abs	67	3	4		N-----Z-
68	OR !abs + Y	75	3	5		
69	OR [dp + X]	76	2	6		
70	OR [dp] + Y	77	2	6		
71	OR { X }	74	1	3		
72	ROL A	28	1	2	Rotate left through Carry	
73	ROL dp	29	2	4		N-----ZC
74	ROL dp + X	39	2	5		
75	ROL !abs	38	3	5		
76	ROR A	68	1	2	Rotate right through Carry	
77	ROR dp	69	2	4		N-----ZC
78	ROR dp + X	79	2	5		
79	ROR !abs	78	3	5		
80	SBC #imm	24	2	2	Subtract with Carry	
81	SBC dp	25	2	3	A ← (A) - (M) - ~(C)	
82	SBC dp + X	26	2	4		
83	SBC !abs	27	3	4		NV--HZC
84	SBC !abs + Y	35	3	5		
85	SBC [dp + X]	36	2	6		
86	SBC [dp] + Y	37	2	6		
87	SBC { X }	34	1	3		
88	TST dp	4C	2	3	Test memory contents for negative or zero, (dp) - 00_H	N-----Z-
89	XCN	CE	1	5	Exchange nibbles within the accumulator A_7~A_4 ↔ A_3~A_0	N-----Z-

Register / Memory Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	LDA #imm	C4	2	2	Load accumulator	
2	LDA dp	C5	2	3	A ← (M)	
3	LDA dp + X	C6	2	4		
4	LDA !abs	C7	3	4		
5	LDA !abs + Y	D5	3	5		N-----Z-
6	LDA [dp + X]	D6	2	6		
7	LDA [dp] + Y	D7	2	6		
8	LDA { X }	D4	1	3		
9	LDA { X }+	DB	1	4	X- register auto-increment : A ← (M) , X ← X + 1	
10	LDM dp,#imm	E4	3	5	Load memory with immediate data : (M) ← imm	--------
11	LDX #imm	1E	2	2	Load X-register	
12	LDX dp	CC	2	3	X ← (M)	N-----Z-
13	LDX dp + Y	CD	2	4		
14	LDX !abs	DC	3	4		
15	LDY #imm	3E	2	2	Load Y-register	
16	LDY dp	C9	2	3	Y ← (M)	N-----Z-
17	LDY dp + X	D9	2	4		
18	LDY !abs	D8	3	4		
19	STA dp	E5	2	4	Store accumulator contents in memory	
20	STA dp + X	E6	2	5	(M) ← A	
21	STA !abs	E7	3	5		
22	STA !abs + Y	F5	3	6		--------
23	STA [dp + X]	F6	2	7		
24	STA [dp] + Y	F7	2	7		
25	STA { X }	F4	1	4		
26	STA { X }+	FB	1	4	X- register auto-increment : (M) ← A, X ← X + 1	
27	STX dp	EC	2	4	Store X-register contents in memory	
28	STX dp + Y	ED	2	5	(M) ← X	--------
29	STX !abs	FC	3	5		
30	STY dp	E9	2	4	Store Y-register contents in memory	
31	STY dp + X	F9	2	5	(M) ← Y	--------
32	STY !abs	F8	3	5		
33	TAX	E8	1	2	Transfer accumulator contents to X-register : X ← A	N-----Z-
34	TAY	9F	1	2	Transfer accumulator contents to Y-register : Y ← A	N-----Z-
35	TSPX	AE	1	2	Transfer stack-pointer contents to X-register : X ← sp	N-----Z-
36	TXA	C8	1	2	Transfer X-register contents to accumulator: A ← X	N-----Z-
37	TXSP	8E	1	2	Transfer X-register contents to stack-pointer: sp ← X	N-----Z-
38	TYA	BF	1	2	Transfer Y-register contents to accumulator: A ← Y	N-----Z-
39	XAX	EE	1	4	Exchange X-register contents with accumulator :X ↔ A	--------
40	XAY	DE	1	4	Exchange Y-register contents with accumulator :Y ↔ A	--------
41	XMA dp	BC	2	5	Exchange memory contents with accumulator	
42	XMA dp+X	AD	2	6	(M) ↔ A	N-----Z-
43	XMA {X}	BB	1	5		
44	XYX	FE	1	4	Exchange X-register contents with Y-register : X ↔ Y	--------

16-BIT operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	ADDW dp	1D	2	5	16-Bits add without Carry YA ← (YA) + (dp +1) (dp)	NV--H-ZC
2	CMPW dp	5D	2	4	Compare YA contents with memory pair contents : (YA) – (dp+1)(dp)	N-----ZC
3	DECW dp	BD	2	6	Decrement memory pair (dp+1)(dp) ← (dp+1) (dp) - 1	N-----Z-
4	INCW dp	9D	2	6	Increment memory pair (dp+1) (dp) ← (dp+1) (dp) + 1	N-----Z-
5	LDYA dp	7D	2	5	Load YA YA ← (dp +1) (dp)	N-----Z-
6	STYA dp	DD	2	5	Store YA (dp +1) (dp) ← YA	--------
7	SUBW dp	3D	2	5	16-Bits subtract without carry YA ← (YA) - (dp +1) (dp)	NV--H-ZC

Bit Manipulation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	AND1 M.bit	8B	3	4	Bit AND C-flag : C ← (C) ∧ (M .bit)	-------C
2	AND1B M.bit	8B	3	4	Bit AND C-flag and NOT : C ← (C) ∧ ~(M .bit)	-------C
3	BIT dp	0C	2	4	Bit test A with memory : Z ← (A) ∧ (M) , N ← (M_7) , V ← (M_6)	MM----Z-
4	BIT !abs	1C	3	5		
5	CLR1 dp.bit	y1	2	4	Clear bit : (M.bit) ← “0”	--------
6	CLRA1 A.bit	2B	2	2	Clear A bit : (A.bit) ← “0”	--------
7	CLRC	20	1	2	Clear C-flag : C ← “0”	-------0
8	CLRG	40	1	2	Clear G-flag : G ← “0”	--0-----
9	CLRV	80	1	2	Clear V-flag : V ← “0”	-0--0---
10	EOR1 M.bit	AB	3	5	Bit exclusive-OR C-flag : C ← (C) ⊕ (M .bit)	-------C
11	EOR1B M.bit	AB	3	5	Bit exclusive-OR C-flag and NOT : C ← (C) ⊕ ~(M .bit)	-------C
12	LDC M.bit	CB	3	4	Load C-flag : C ← (M .bit)	-------C
13	LDCB M.bit	CB	3	4	Load C-flag with NOT : C ← ~(M .bit)	-------C
14	NOT1 M.bit	4B	3	5	Bit complement : (M .bit) ← ~(M .bit)	--------
15	OR1 M.bit	6B	3	5	Bit OR C-flag : C ← (C) ∨ (M .bit)	-------C
16	OR1B M.bit	6B	3	5	Bit OR C-flag and NOT : C ← (C) ∨ ~(M .bit)	-------C
17	SET1 dp.bit	x1	2	4	Set bit : (M.bit) ← “1”	--------
18	SETA1 A.bit	0B	2	2	Set A bit : (A.bit) ← “1”	--------
19	SETC	A0	1	2	Set C-flag : C ← “1”	-------1
20	SETG	C0	1	2	Set G-flag : G ← “1”	--1-----
21	STC M.bit	EB	3	6	Store C-flag : (M .bit) ← C	--------
22	TCLR1 !abs	5C	3	6	Test and clear bits with A : A - (M) , (M) ← (M) ∧ ~(A)	N-----Z-
23	TSET1 !abs	3C	3	6	Test and set bits with A : A - (M) , (M) ← (M) ∨ (A)	N-----Z-

Branch / Jump Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	BBC A.bit,rel	y2	2	4/6	Branch if bit clear :	--------
2	BBC dp.bit,rel	y3	3	5/7	if (bit) = 0 , then pc ← (pc) + rel	
3	BBS A.bit,rel	x2	2	4/6	Branch if bit set :	--------
4	BBS dp.bit,rel	x3	3	5/7	if (bit) = 1 , then pc ← (pc) + rel	
5	BCC rel	50	2	2/4	Branch if carry bit clear if (C) = 0 , then pc ← (pc) + rel	--------
6	BCS rel	D0	2	2/4	Branch if carry bit set if (C) = 1 , then pc ← (pc) + rel	--------
7	BEQ rel	F0	2	2/4	Branch if equal if (Z) = 1 , then pc ← (pc) + rel	--------
8	BMI rel	90	2	2/4	Branch if minus if (N) = 1 , then pc ← (pc) + rel	--------
9	BNE rel	70	2	2/4	Branch if not equal if (Z) = 0 , then pc ← (pc) + rel	--------
10	BPL rel	10	2	2/4	Branch if minus if (N) = 0 , then pc ← (pc) + rel	--------
11	BRA rel	2F	2	4	Branch always pc ← (pc) + rel	--------
12	BVC rel	30	2	2/4	Branch if overflow bit clear if (V) = 0 , then pc ← (pc) + rel	--------
13	BVS rel	B0	2	2/4	Branch if overflow bit set if (V) = 1 , then pc ← (pc) + rel	--------
14	CALL !abs	3B	3	8	Subroutine call	
15	CALL [dp]	5F	2	8	M(sp)←(pc_H), sp←sp - 1, M(sp)← (pc_L), sp ←sp - 1, if !abs, pc← abs ; if [dp], pc_L← (dp), pc_H← (dp+1) .	--------
16	CBNE dp,rel	FD	3	5/7	Compare and branch if not equal :	--------
17	CBNE dp+X,rel	8D	3	6/8	if (A) ≠ (M) , then pc ← (pc) + rel.	
18	DBNE dp,rel	AC	3	5/7	Decrement and branch if not equal :	--------
19	DBNE Y,rel	7B	2	4/6	if (M) ≠ 0 , then pc ← (pc) + rel.	
20	JMP !abs	1B	3	3	Unconditional jump	
21	JMP [!abs]	1F	3	5	pc ← jump address	--------
22	JMP [dp]	3F	2	4		
23	PCALL upage	4F	2	6	U-page call M(sp) ←(pc_H), sp ←sp - 1, M(sp) ← (pc_L), sp ← sp - 1, pc_L ← (upage), pc_H ← "0FF_H" .	--------
24	TCALL n	nA	1	8	Table call : (sp) ←(pc_H), sp ← sp - 1, M(sp) ← (pc_L),sp ← sp - 1, pc_L ← (Table vector L), pc_H ← (Table vector H)	--------

Control Operation & Etc.

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

C. MASK ORDER SHEET

MASK ORDER & VERIFICATION SHEET

GMS81C20XX-HI ☐☐☐

Customer should write inside thick line box.

1. Customer Information

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

2. Device Information

Package	☐ 64SDIP ☐ 64MQFP ☐ 64LQFP	
Mask Data	☐ Internet ☐ Hitel ☐ Chollian	
	File Name	( ) .OTP
	ROM Size (bytes)	☐ 12K ☐ 20K
	Check Sum ( )	

(20K) 3000_H
(12K) 5000_H
Set "FF_H" in blanked area
.OTP file data
$7FFF_H$

(Please check mark √ into ☐)

3. Marking Specification

☐ 12 or 20

HME
GMS81C20XX-HI ☐☐☐
☐☐☐☐☐☐☐☐☐☐☐☐☐☐
YYWW KOREA

☐ *Customer's logo*
☐☐☐☐☐☐☐☐☐☐☐☐☐☐
GMS81C20XX-HI ☐☐☐
YYWW KOREA

Customer logo is not required. ☐

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	HME Confirmation
Customer sample	YYYY • MM • DD	pcs	
Risk order	YYYY • MM • DD	pcs	

5. ROM Code Verification

Please confirm out verification data.

Verification date: YYYY • MM • DD
Check sum:
Tel: Fax:
E-mail address:
Name & Signature:

Approval date: YYYY • MM • DD
I agree with your verification data and confirm you to make mask set.
Tel: Fax:
E-mail address:
Name & Signature:

HYUNDAI MicroElectronics
Semiconductor Group of Hyundai Electronics Industries Co., Ltd.

GMS81C20XX MASK OPTION LIST

Customer should write inside thick line box.

1. RA/Vdisp

☐ RA without pull-down resistor
☐ Vdisp

(Please check mark √ into ☐)

2. CONFIG OPTION Check

X				X		X	

CONFIG Default Value : X000X0X0

	7	6	5	4	3	2	1	0
CONFIG		R7X	PFS1	PFS0		LOCK		RCO

ADDRESS: $703F_H$
INITIAL VALUE: -000 -0-0_B

R74, R75 Selection
0 : Sub Clock
1 : R74, R75

PFD Level Selection
00: PFD = 2.7V
01: PFD = 2.7V
10: PFD = 3.0V
11: PFD = 2.4V

External RC OSC Selection
0: Crystal or Resonator Oscillator
1: External RC Oscillator

Code Protect
0 : Allow Code Read Out
1 : Lock Code Read Out

3. H/V Port OPTION Check (Pull-down Option Check)

Port	Option	
	ON	OFF
R00/INT0		
R01/INT1		
R02/EC0		
R03/BUZO		
R04		
R05		
R06		
R07		

Port	Option	
	ON	OFF
R10		
R11		
R12		
R13		
R14		
R15		
R16		
R17		

Port	Option	
	ON	OFF
R20		
R21		
R22		
R23		
R24		
R25		
R26		
R27		

Port	Option	
	ON	OFF
R30		
R31		
R32		
R33		
R34		
R35		

ON : with pull-down resistor
OFF : without pull-down resistor

4. Normal Port OPTION Check (Pull-up Option Check)

Port	Option	
	ON	OFF
R40/T0O		
R41		
R42		
R43		

Port	Option	
	ON	OFF
R50		
R51		
R52		
R53/SCLK		
R54/SIN		
R55/SOUT		
R56/PWM		
R57		

Port	Option	
	ON	OFF
R60/AN0		
R61/AN1		
R62/AN2		
R63/AN3		
R64/AN4		
R65/AN5		
R66/AN6		
R67/AN7		

Port	Option	
	ON	OFF
R70/AN8		
R71/AN9		
R72/AN10		
R73/AN11		

ON : with pull-up resistor
OFF : without pull-up resistor

HYUNDAI MicroElectronics
Semiconductor Group of Hyundai Electronics Industries Co., Ltd.