GMS81C2020 / GMS81C2120

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

1. OVERVIEW

1.1 Description

The GMS81C2020 and GMS81C2120 are an advanced CMOS 8-bit microcontroller with 20K/12K 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: 20K/12K 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, 8-bit, 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 GMS81C2020 and GMS81C2120 support power saving modes to reduce power consumption.

This document is only explained for the base of GMS81C2020(GMS81C2120), the eliminated functions are same as below.

Device name	ROM Size	RAM Size	Ports	Package
GMS81C2020	20Kbytes	448bytes	R0,R1,R2,R3,R4,R5,R6,R7	64 SDIP, 64MQFP, 64LQFP, 64TQFP
GMS81C2012	12Kbytes	448bytes	R0,R2,R3,R5,R6	64SDIP, 64MQFP, 64LQFP, 64TQFP
*GMS87C2020	20Kbytes (EPROM)	448bytes	R0,R1,R2,R3,R4,R5,R6,R7	64SDIP, 64MQFP, 64LQFP, 64TQFP
GMS81C2120	20Kbytes	448bytes	R0,R1,R2,R3,R4,R5,R6,R7	42SDIP, 44MQFP, 40PDIP
GMS81C2112	12Kbytes	448bytes	R0,R2,R3,R5,R6	42SDIP, 44MQFP, 40PDIP
*GMS87C2120	20Kbytes (EPROM)	448bytes	R0,R1,R2,R3,R4,R5,R6,R7	42SDIP, 44MQFP, 40PDIP

[The * Mark Devices are OTP Version]

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 :
  30 high-voltage pins (40V,max)
  - 3 Input Only pins : 1 high-voltage pin
  - 1 Output Only pin
- Eight Interrupt Sources
  - 2 By External Sources (INT0, INT1)
  - 2 By Timer/Counter Sources (Timer0, Timer1)
  - 4 By Functional Sources (SPI,ADC,WDT,BIT)
- 12-Channel 8-Bit On-Chip Analog to Digital Converter
- Oscillatior :
  - Crystal
  - Ceramic Resonator
  - External RC Oscillator
  - Internal RCWDT Oscillatior
- Low Power Dissipation Modes
  - STOP mode
  - Wake-up Timer Mode
    - Standby Mode
    - Watch Mode
  - Subactive Mode
- Operating Voltage : 4.0V ~ 5.5V (at 4.5MHz)
- Operating Frequency : 0.4MHz ~ 4.5MHz
- Subclock : 32.768KHz Crystal Oscillator
- Enhanced EMS Improvement
  Power Fail Processor
  (Noise Immunity Circuit)

Device name	Total I/O	Normal I/O	High Voltage I/O	Input Only	Output Only
GMS81C2020	60 pins	26 pins	30 pins	3 pins	1 pins
GMS81C2012	60 pins	26 pins	30 pins	3 pins	1 pins
GMS81C2120	38 pins	13 pins	21 pins	3 pins	1 pins
GMS81C2112	38 pins	13 pins	21 pins	3 pins	1 pins

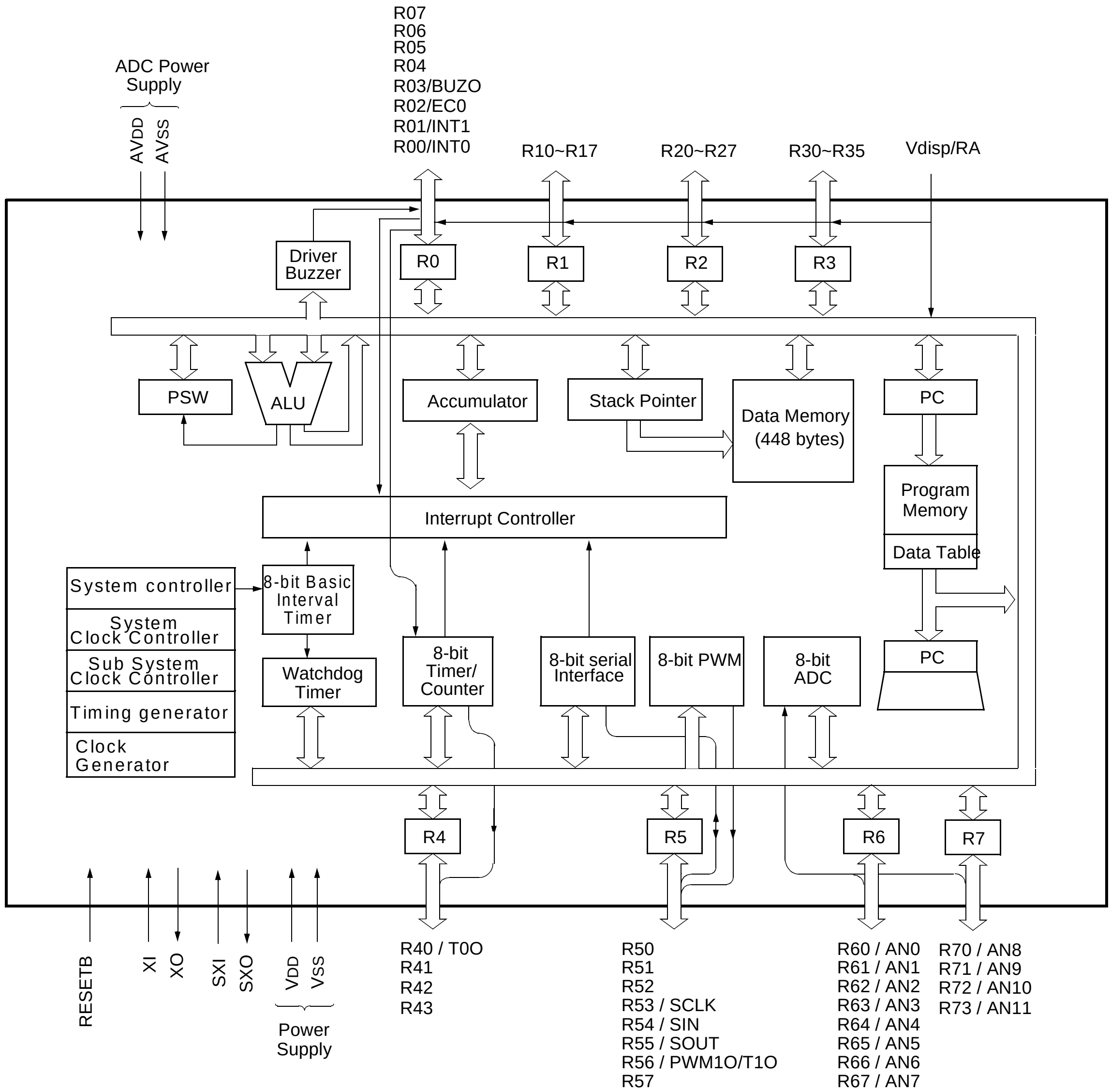
**where, Total I/O is all ports except power and ground ports*

Development Tools

The GMS800 family is supported by a full-featured macro assembler, an in-circuit emulators CHOICE-Dr.™, and add-on board type OTP writer Dr.Writer™ .

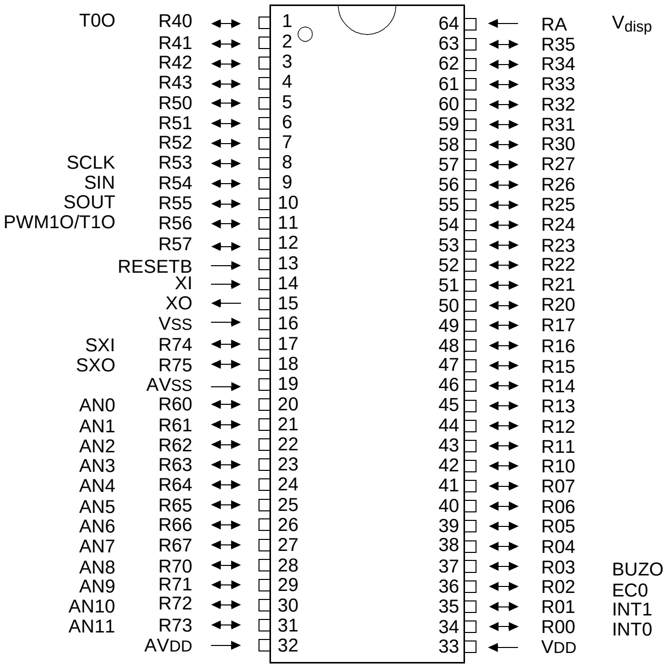
In Circuit Emulator	CHOICE-Dr.
Assembler	HME Macro Assembler
OTP Writer	Dr.Writer

2. BLOCK DIAGRAM (GMS81C2020)

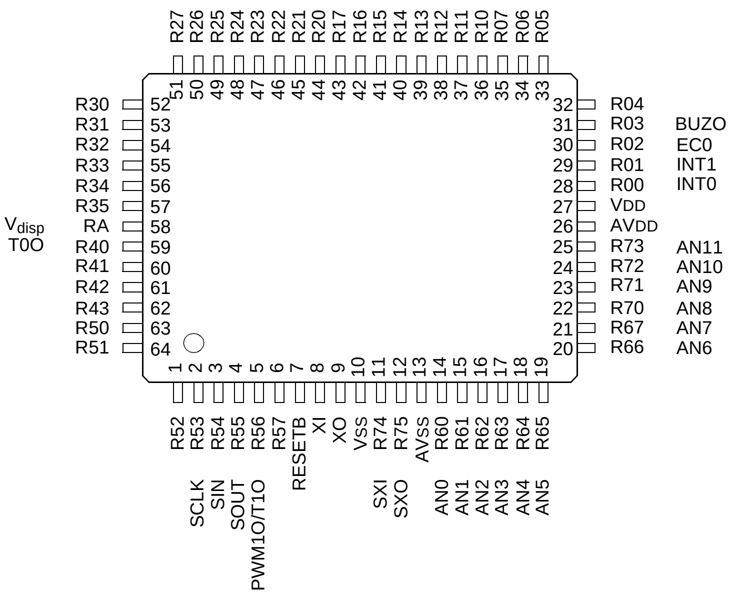


3. PIN ASSIGNMENT (GMS81C2020)

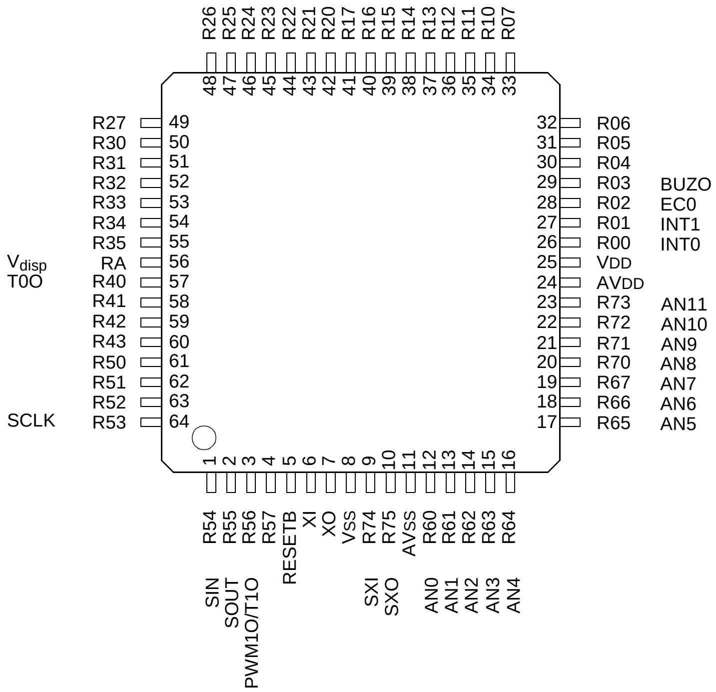
64SDIP



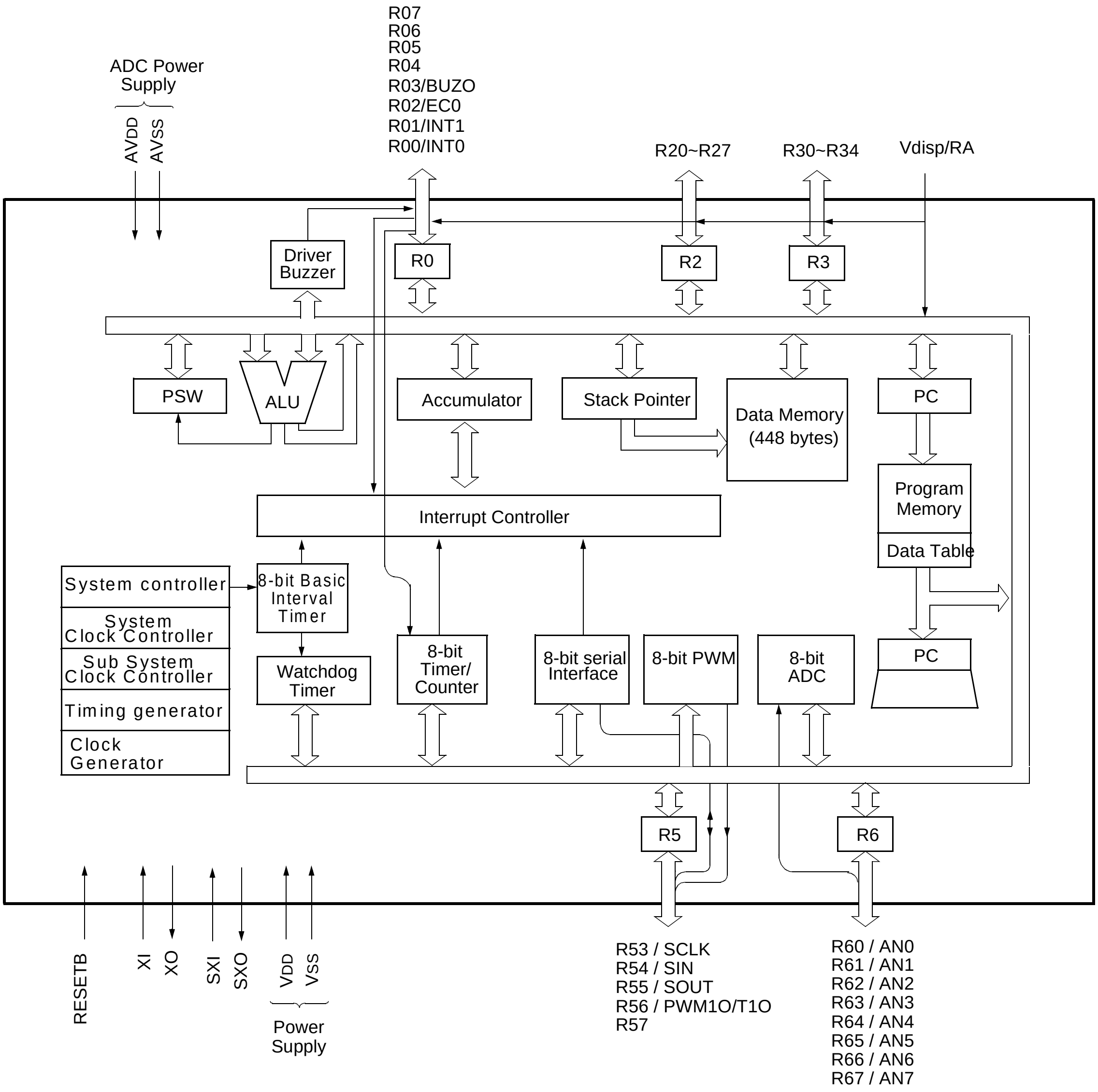
64MQFP



64LQFP

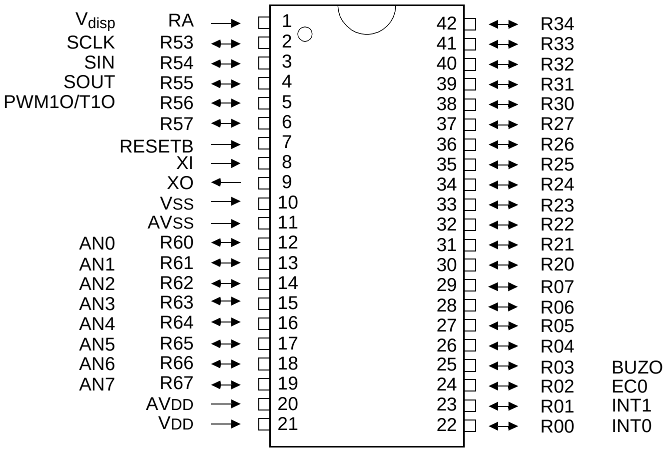

R26 R25 R24 R23 R22 R21 R20 R17 R16 R15 R14 R13 R12 R11 R10 R07
48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
R27 49
R30 50
R31 51
R32 52
R33 53
R34 54
R35 55
Vdisp RA 56
T0O R40 57
R41 58
R42 59
R43 60
R50 61
R51 62
R52 63
SCLK R53 64
32 R06
31 R05
30 R04
29 R03 BUZO
28 R02 EC0
27 R01 INT1
26 R00 INT0
25 VDD
24 AVDD
23 R73 AN11
22 R72 AN10
21 R71 AN9
20 R70 AN8
19 R67 AN7
18 R66 AN6
17 R65 AN5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
R54 R55 R56 R57 RESETB XI XO VSS R74 R75 AVSS R60 R61 R62 R63 R64
SIN SOUT PWM1O/T1O SXI SXO AN0 AN1 AN2 AN3 AN4

4. BLOCK DIAGRAM (GMS81C2120)

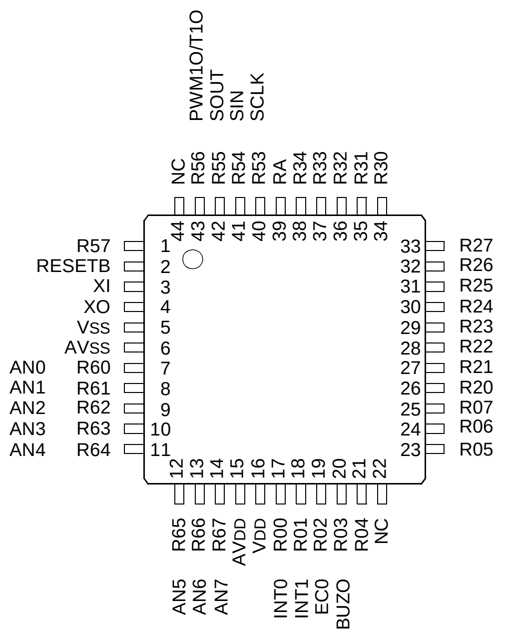


5. PIN ASSIGNMENT (GMS81C2120)

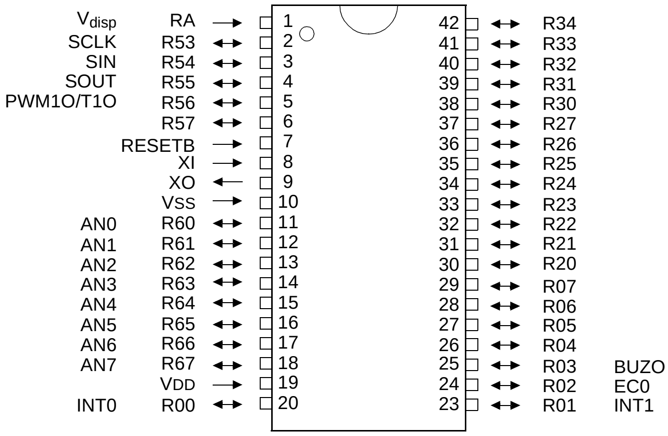
42PDIP



44MQFP



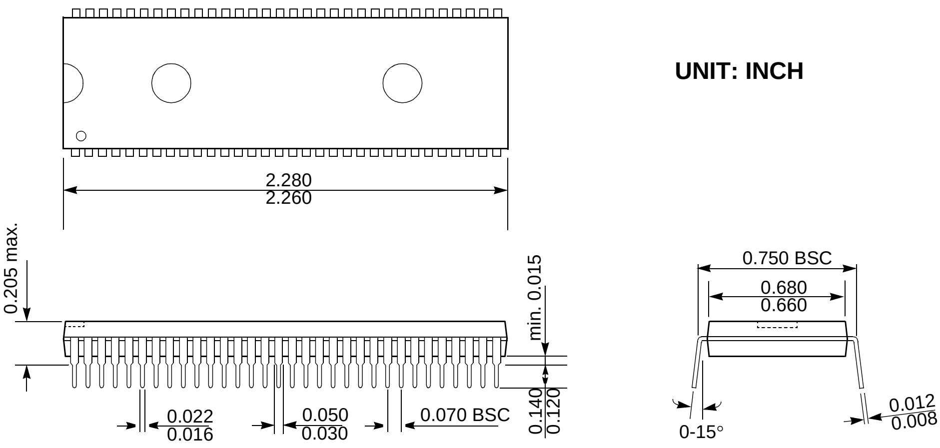
40PDIP


Vdisp RA 1
SCLK R53 2
SIN R54 3
SOUT R55 4
PWM1O/T1O R56 5
R57 6
RESETB 7
XI 8
XO 9
VSS 10
AN0 R60 11
AN1 R61 12
AN2 R62 13
AN3 R63 14
AN4 R64 15
AN5 R65 16
AN6 R66 17
AN7 R67 18
VDD 19
INT0 R00 20
42 R34
41 R33
40 R32
39 R31
38 R30
37 R27
36 R26
35 R25
34 R24
33 R23
32 R22
31 R21
30 R20
29 R07
28 R06
27 R05
26 R04
25 R03 BUZO
24 R02 EC0
23 R01 INT1

6. PACKAGE DIMENSION

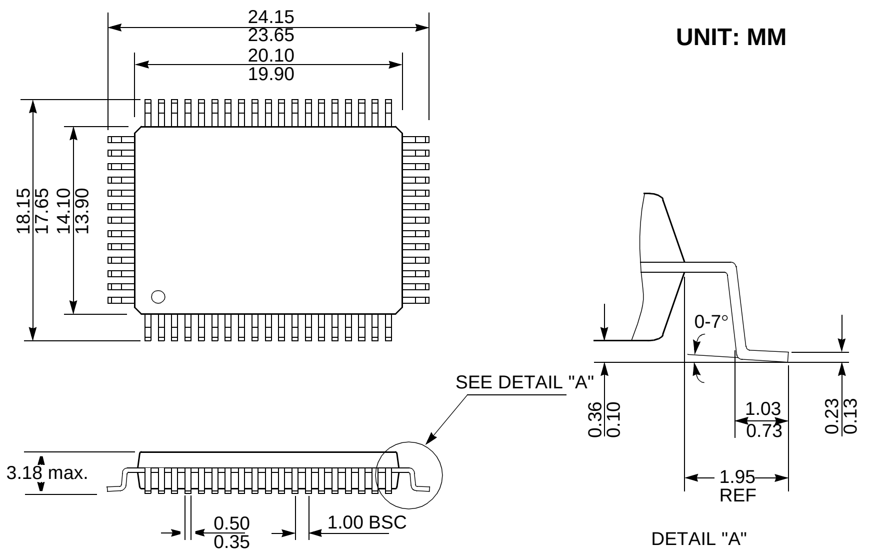
64SDIP

UNIT: INCH



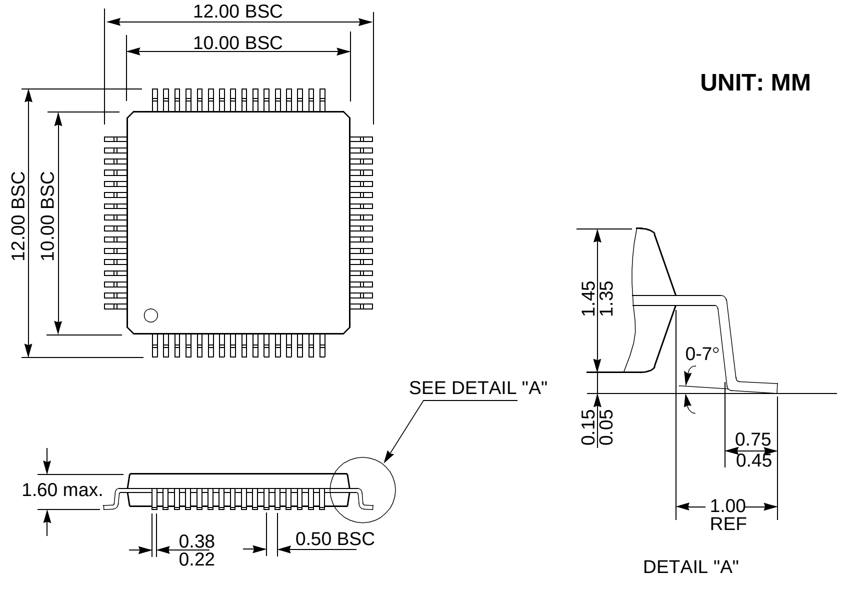
64MQFP

UNIT: MM



64LQFP

UNIT: MM



7. PIN DESCRIPTIONS (GMS81C2020)

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.

RESETB: Reset the MCU.

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

XO: Output from the inverting oscillator amplifier.

SXI: Input to the internal subsystem clock operating circuit. In addition, SXI serves the R74 pin when selected by the code option.

SXO: Output from the inverting subsystem oscillator amplifier. In addition, SXO serves the R75 pin when selected by the code option.

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 an 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 an 8-bit CMOS bidirectional I/O port. R4 pins 1 or 0 written to the Port Direction Register can be used as outputs or 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 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, 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)

PIN NAME	In/Out	Function	
VDD	-	Supply voltage	
VSS	-	Circuit ground	
RA (V_{disp})	I(I)	1-bit high-voltage Input only port	High-voltage input power supply pin
RESETB	I	Reset signal input	
XI	I	Oscillation input	
XO	O	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	

8. PIN DESCRIPTIONS (GMS81C2120)

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.

RESETB: Reset the MCU.

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

XO: 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)

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.

R53~R57: R5 is an 5-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)

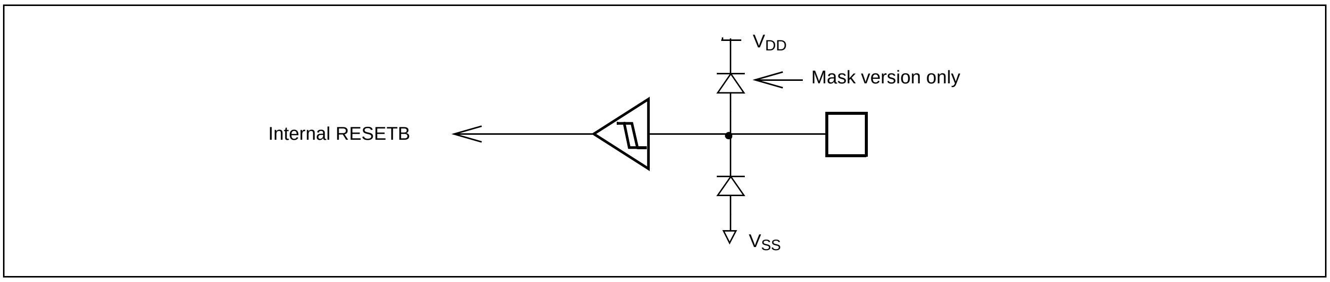
R60~R67: R6 is an 8-bit CMOS bidirectional I/O port. R6 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. In addition, R6 is shared with the ADC input.

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)

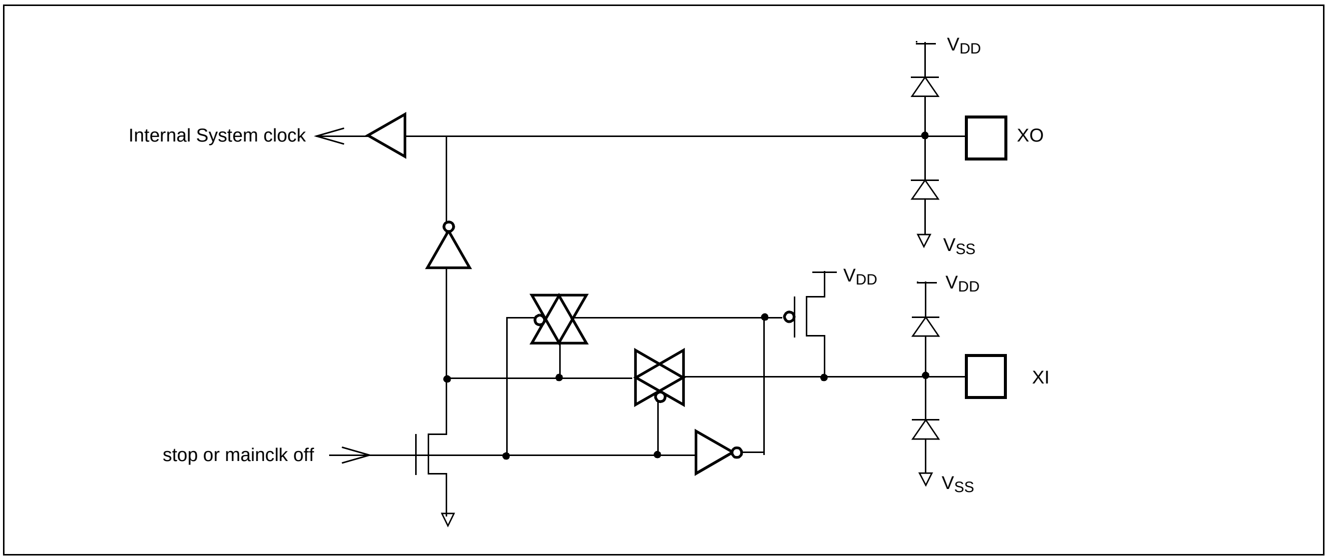
PIN DESCRIPTIONS (GMS81C2120)			
PIN NAME	**In/Out**	**Function**	
VDD	-	Supply voltage	
VSS	-	Circuit ground	
RA (V_{disp})	I(I)	1-bit high-voltage Input only port	High-voltage input power supply pin
RESETB	I	Reset signal input	
XI	I	Oscillation input	
XO	O	Oscillation output	
R00 (INT0)	I/O (I)	8-bit *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		
R20~R27	I/O	8-bit *high-voltage* I/O ports	
R30~R34	I/O	5-bit *high-voltage* I/O ports	
R53 (SCLK)	I/O (I/O)	5-bit general I/O ports	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
AVDD	-	Supply voltage input pin for ADC	
AVSS	-	Ground level input pin for ADC	

9. PORT STRUCTURES

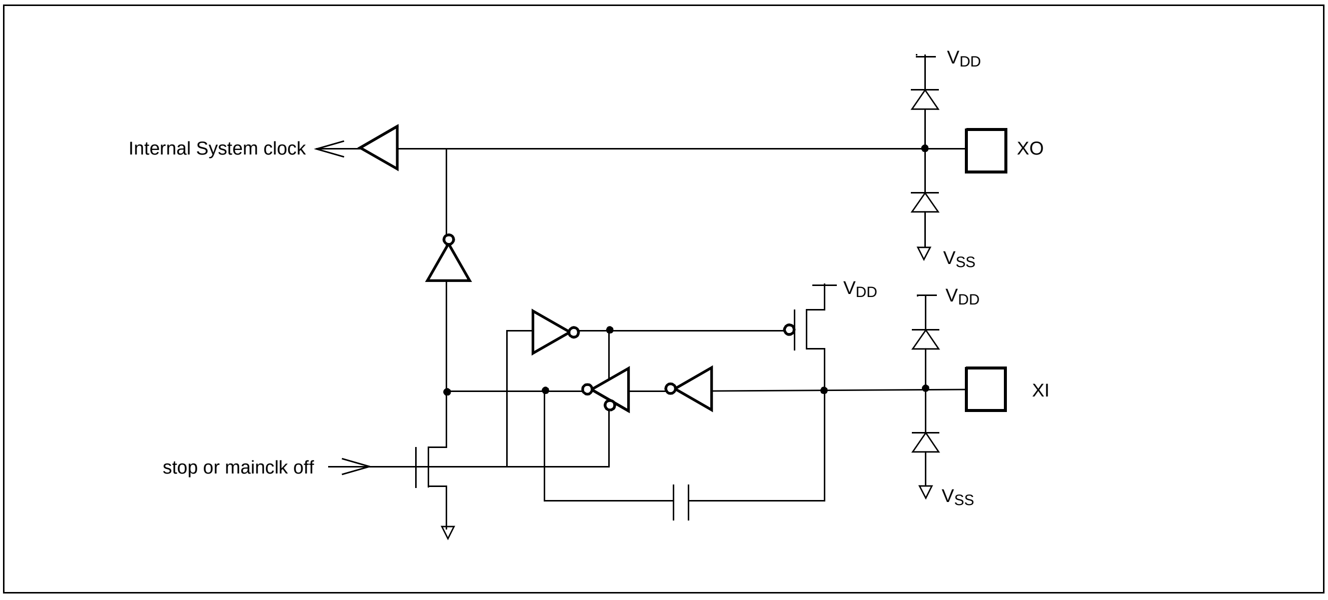
- **RESETB**



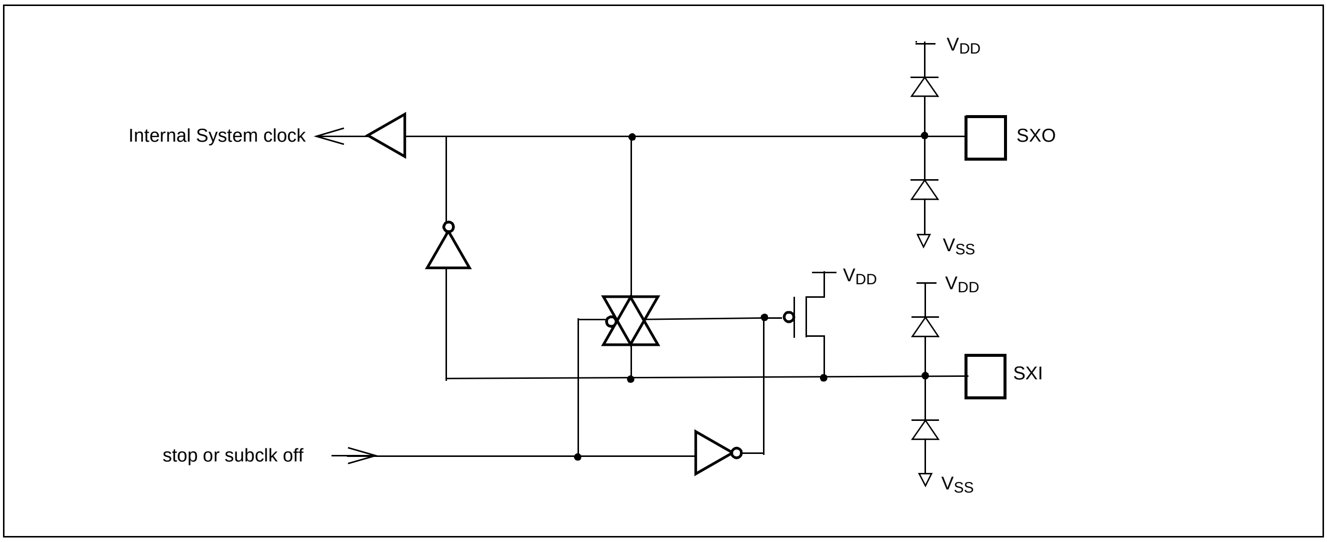
- **XI, XO (Crystal Oscillator)**



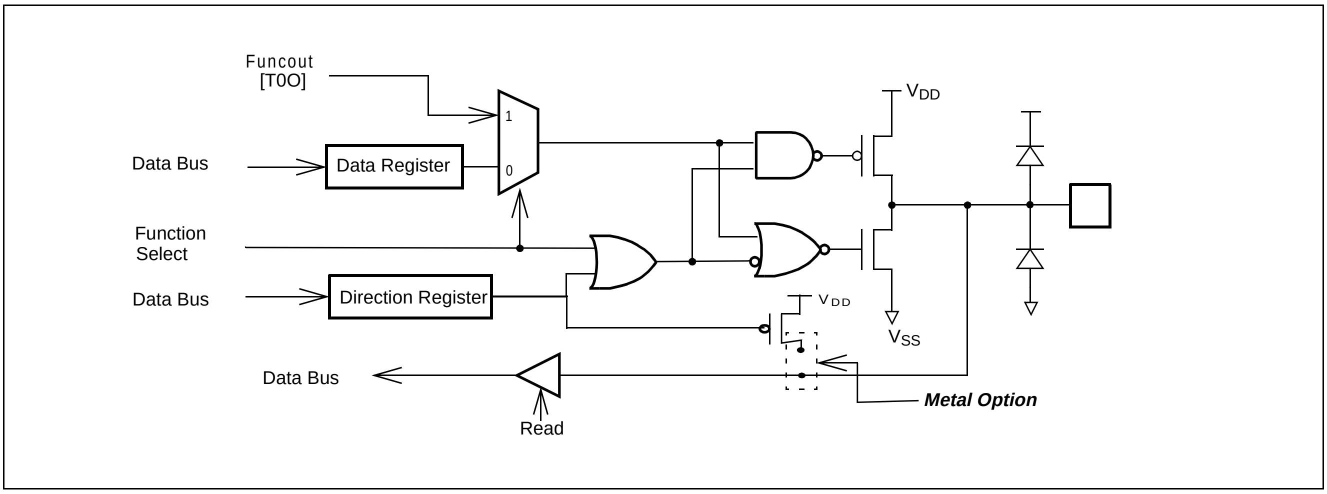
- **XI, XO (RC Oscillator)**



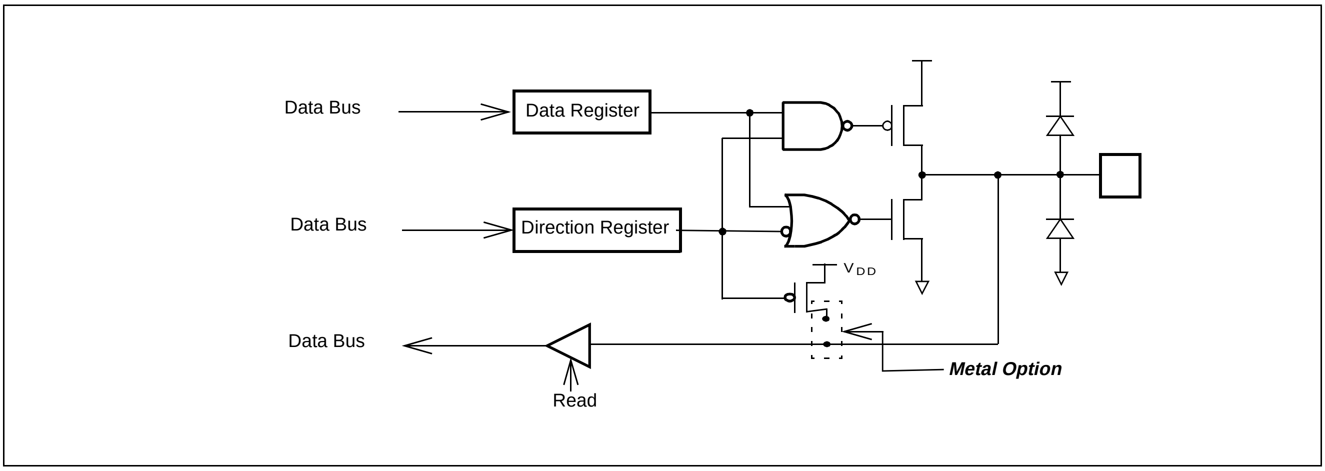
- **SXI, SXO (Sub Oscillator)**



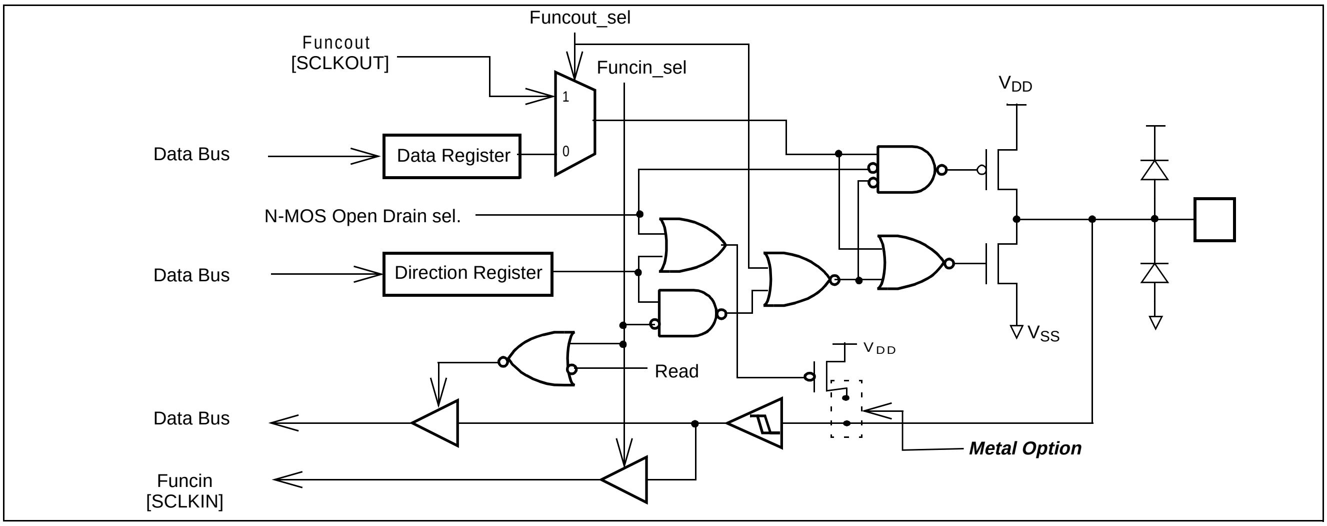
- **R40 / T0O**



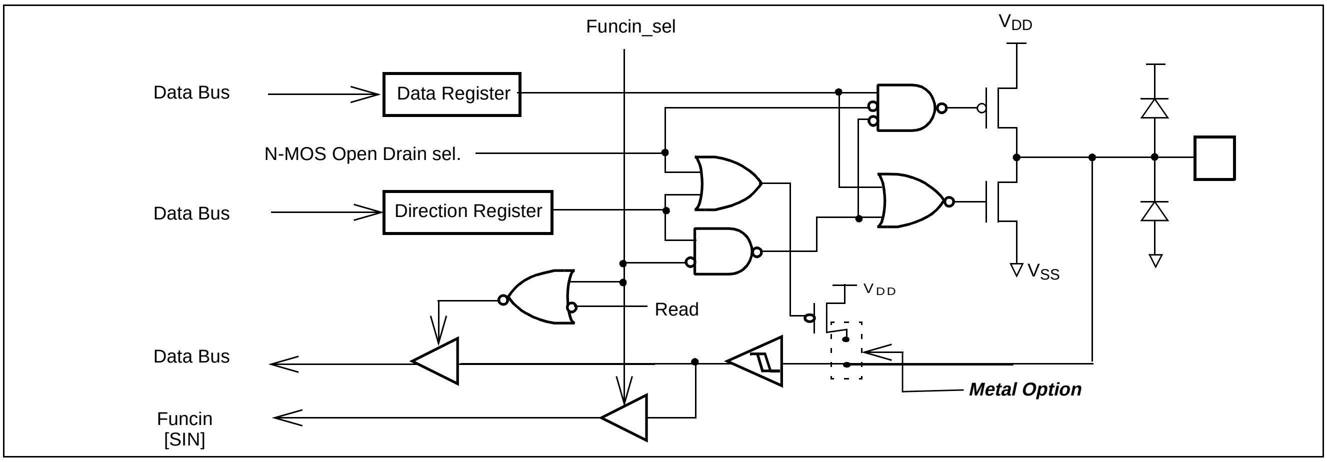
- **R41~R43, R50~R52, R57**



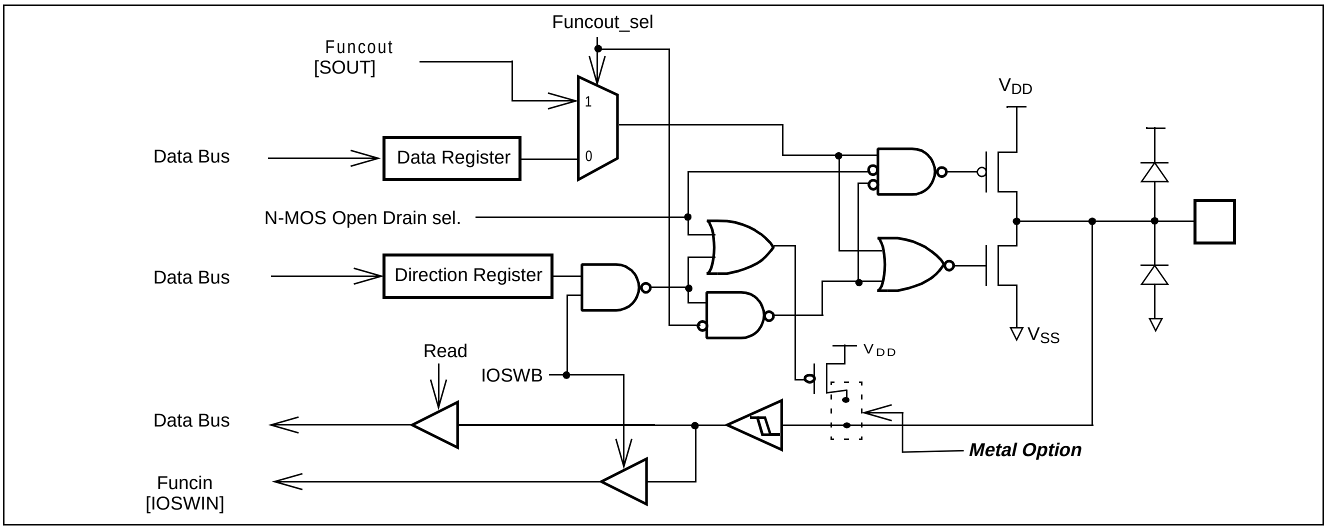
- **R53 / SCLK**



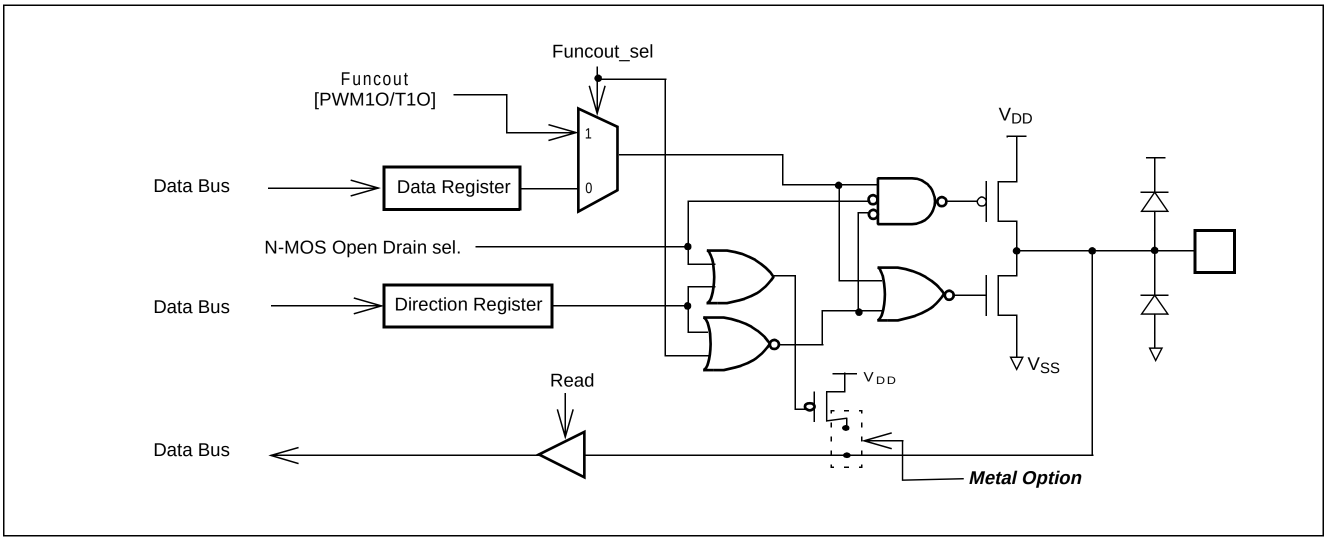
- **R54 / SIN**



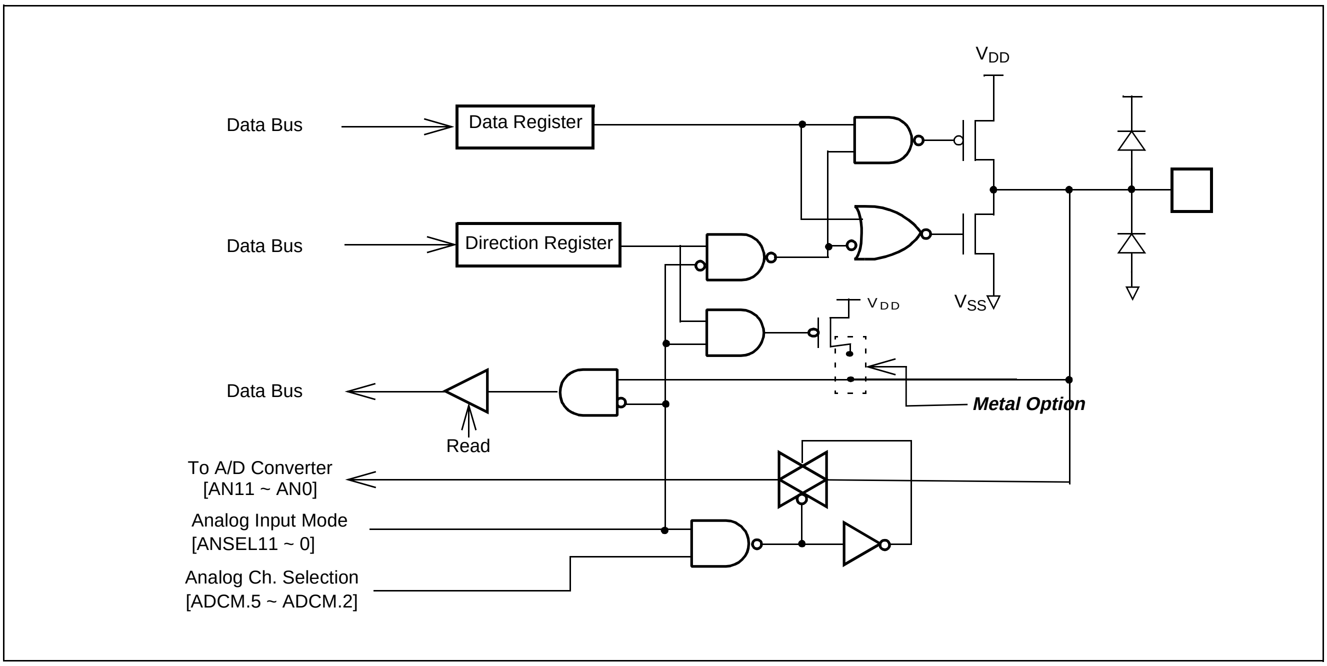
- **R55 / SOUT**



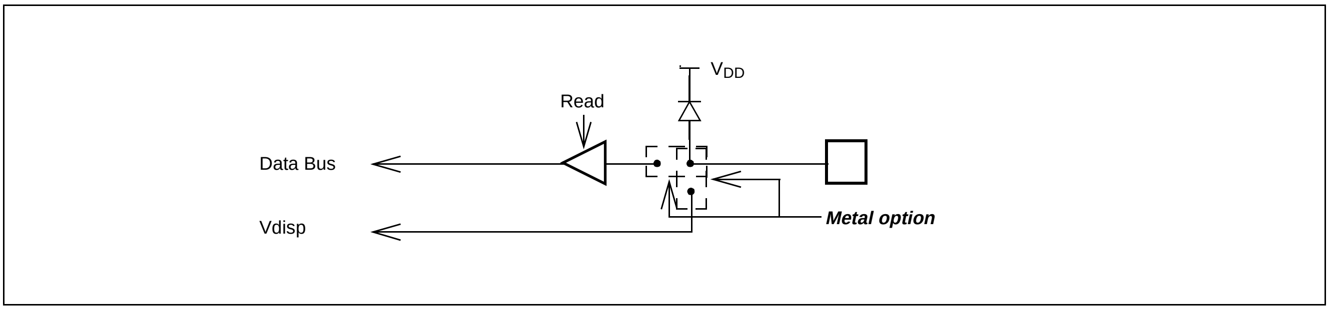
• **R56 / PWM1O / T1O**



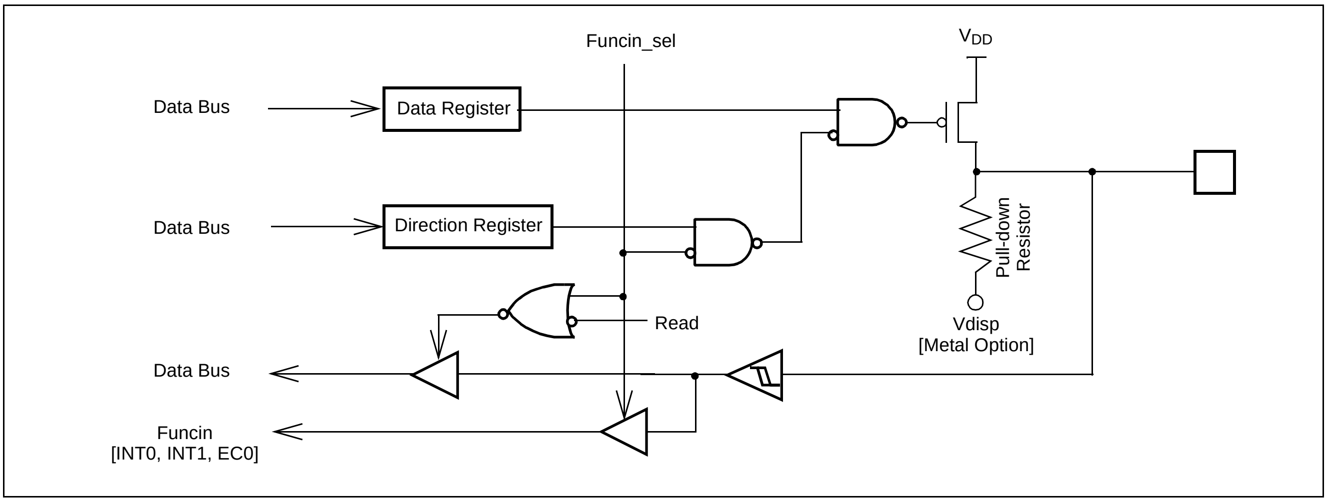
• **R60~R67 [AN0 ~ AN7], R70~R74 [AN8 ~ AN11]**



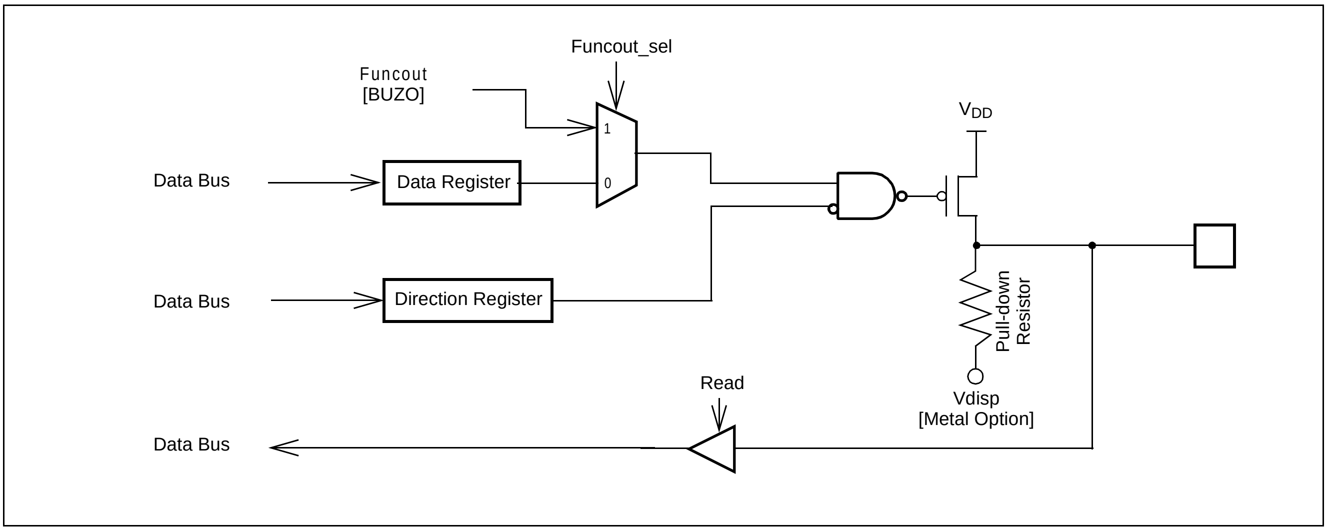
• **RA / Vdisp**



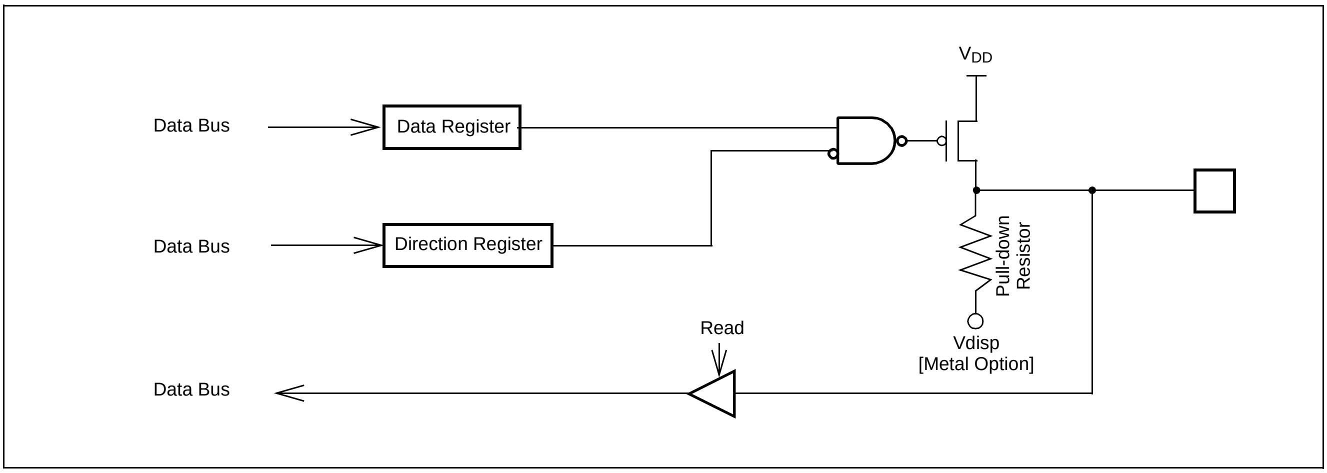
- **R00 / INT0, R01 / INT1, R02 / EC0**



- **R03 / BUZO**



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



10. ELECTRICAL CHARACTERISTICS

- **Absolute Maximum Ratings**

Supply Voltage : V_{DD} . . . . - 0.3 to + 7.0V

Storage Temperature : T_{STG} . . . . -40 to + 125 °C

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

I_{OL} per I/O Pin . . . . 20 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 these of 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.*

Recommended Operating Conditions

Parameter	Symbol	Condition	Specification		Unit
			Min	Max	
Supply Voltage	V_{DD}	f_{XI} = 4.5 MHz	4.0	5.5	V
Operating Frequency	f_{XI}	V_{DD} = V_{DD}	0.4	4.5	MHz
Operating Temperature	T_{OPR}		-40	125	°C

10.1 A/D Converter Characteristics

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

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	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}		-	±1.0	±1.5	LSB
Non-Linearity Error	N_{NLE}		-	±1.0	±1.5	LSB
Differential Non-Linearity Error	N_{DNLE}		-	±1.0	±1.5	LSB
Zero Offset Error	N_{ZOE}		-	±0.5	±1.5	LSB
Full Scale Error	N_{FSE}		-	±0.25	±0.5	LSB
Gain Error	N_{NLE}		-	±1.0	±1.5	LSB
Conversion Time	T_{CONV}	f_{XI}=4MHz	-	-	20	us

DC Characteristics for Standard Pins(5V)

(V_{DD} = 5.0V ± 10%, V_{SS} = 0V, T_A = -40 ~ 125°C, f_{XI} = 4 MHz, Vdisp=V_{DD}-40V to V_{DD})

Parameter	Pin	Symbol	Test Condition	Specification			Unit
				Min	Typ	Max	
Input High Voltage	XI, SXI	V_{IH1}		$0.9V_{DD}$		V_{DD}+0.3	V
	RESETB,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	XI, SXI	V_{IL1}		-0.3		$0.1V_{DD}$	V
	RESETB,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
	XI	I_{IH2}				1	
Input Low Leakage Current	R40~R43,R5,R6,R70~R73	I_{IL1}		-1			uA
	XI	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_{XI}=4.2MHz			5	mA
Current dissipation in standby mode	V_{DD}	I_{STBY}	f_{XI}=4.2MHz			2	mA
Current dissipation in subactive mode	V_{DD}	I_{SUB}	f_{XI}=Off f_{SXI}=32.7KHz			100	uA
Current dissipation in watch mode	V_{DD}	I_{WTC}	f_{XI}=Off f_{SXI}=32.7KHz			20	uA
Current dissipation in stop mode	V_{DD}	I_{STOP}	f_{XI}=Off f_{SXI}=32.7KHz			10	uA
Hysteresis	RESETB,SIN,R55,SCLK, INT0,INT1,EC0	V_{T+}~V_{T-}		0.4			V
Internal RC WDT Frequency	XO	T_{RCWDT}		10		25	MHz
RC Oscillation Frequency	XO	f_{RCOSC}	R= 60KΩ	1.5	2	2.5	MHz

DC Characteristics for High-Voltage Pins

(V_{DD} = 5.0V ± 10%, V_{SS} = 0V, T_A = -40 ~ 125°C, f_{XI} = 4 MHz, Vdisp=V_{DD}-40V to V_{DD})

Parameter	Pin	Symbol	Test Condition	Specification			Unit
				Min	Typ	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

10.2 AC Characteristics

(T_A=-40~ 125°C, V_{DD}=5V±10%, V_{SS}=0V)

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

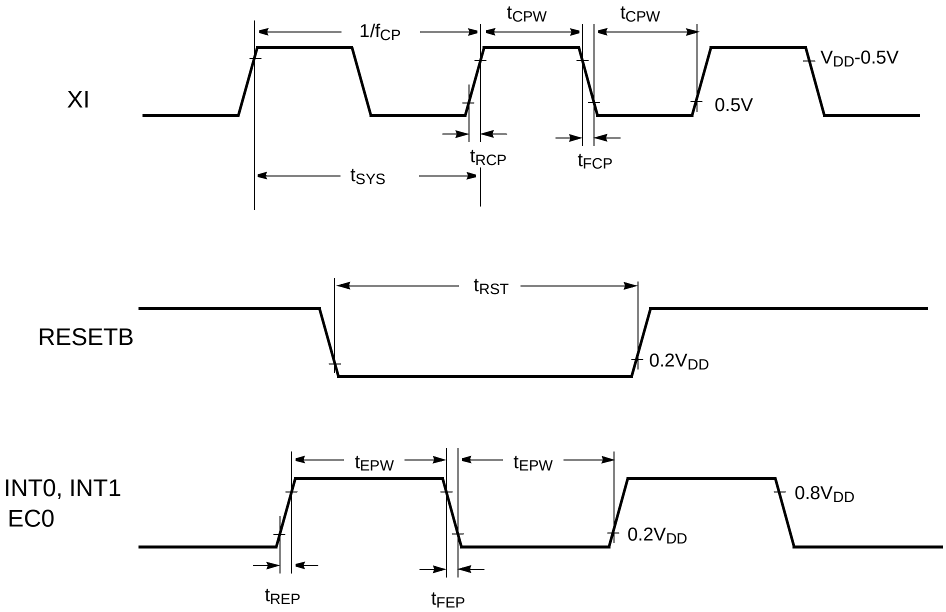


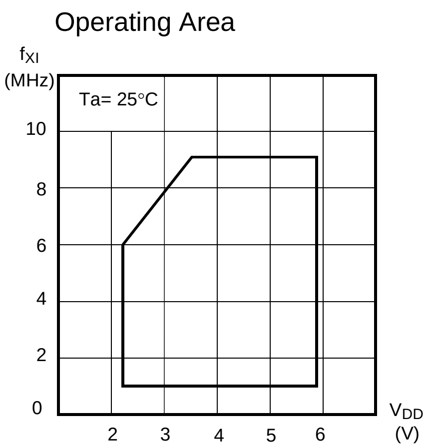
Figure 10-1 Timing Chart

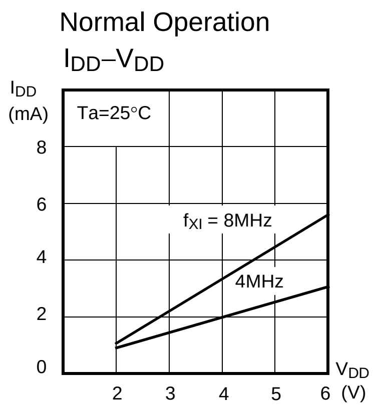
10.3 Typical Characteristics

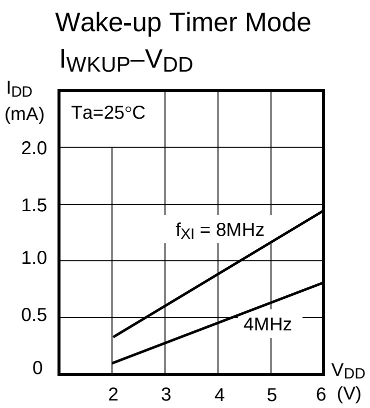
This graphs and tables provided in this section are for design guidance only and are not tested or guranteed.

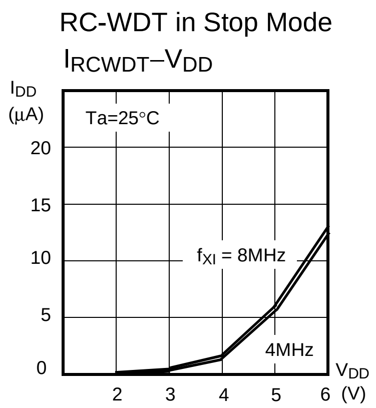
In some graphs or tables the data presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for imformation only and divices are guranteed to operate properly only within the specified range.

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



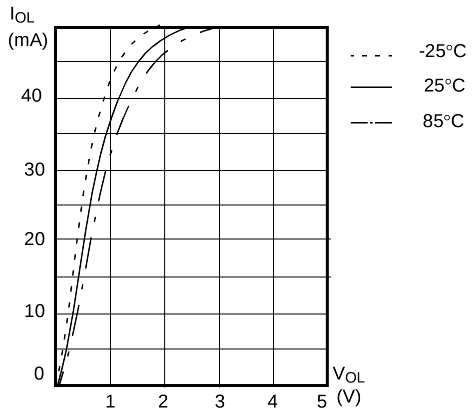




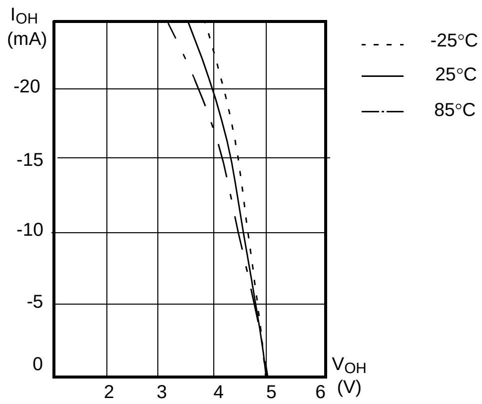


**** FOR MODIFIED ****

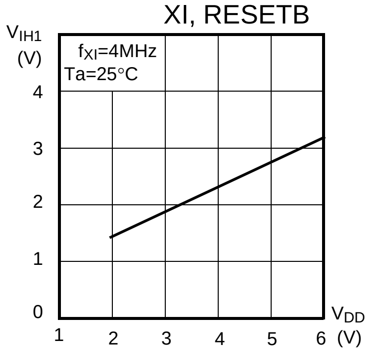
I_{OL}–V_{OL}, V_{DD}=5V



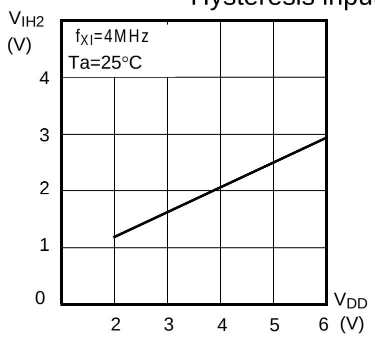
I_{OH}–V_{OH}, V_{DD}=5V



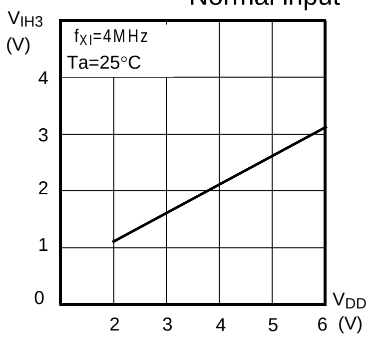
V_{DD}–V_{IH1} XI, RESETB



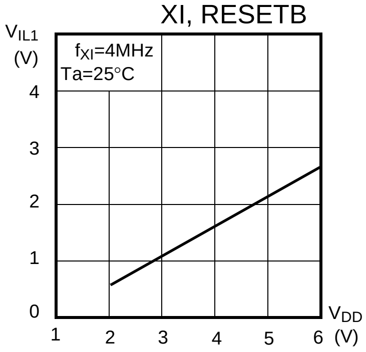
V_{DD}–V_{IH2} Hysteresis input



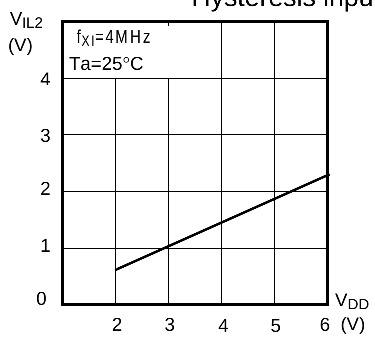
V_{DD}–V_{IH3} Normal input



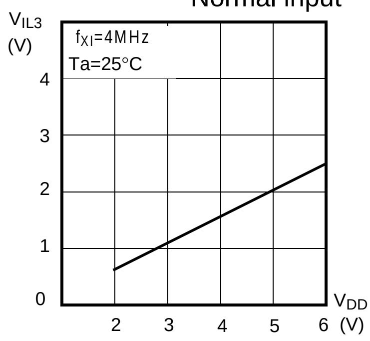
V_{DD}–V_{IL1} XI, RESETB



V_{DD}–V_{IL2} Hysteresis input



V_{DD}–V_{IL3} Normal input



**** FOR MODIFIED ****

11. MEMORY ORGANIZATION

The GMS81C2020 and GMS81C2120 have separate address spaces for Program memory and Data Memory. Program memory can only be read, not written to. It can be up to 20K/12K bytes of Program memory. Data memory can be read and written to up to 448 bytes including the stack area.

11.1 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.

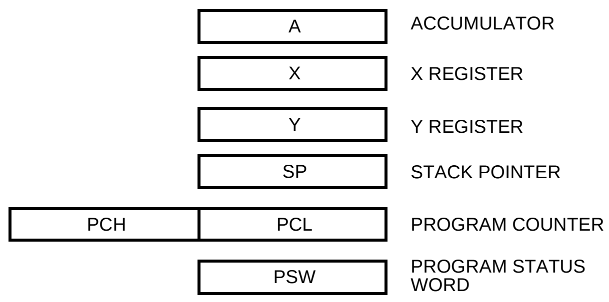


Figure 11-1 Configuration of Registers

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.

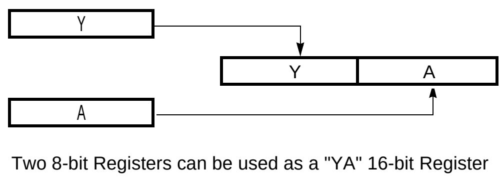


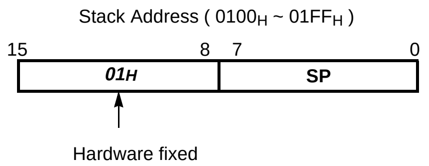
Figure 11-2 Configuration of YA 16-bit Register

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

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

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

The stack can be located at any position within 00_H to FF_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 11-3 . 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.

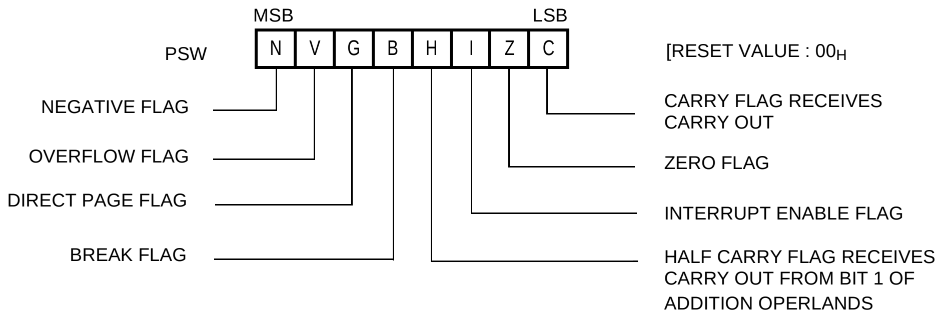


Figure 11-3 PSW (Program Status Word) Register

[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 assign direct page(0-page, 1-page) for direct addressing mode. When G-flag is "0", the direct addressing space is in 0-page(0000h ~ 00FFH). When G-flag is "1", the direct addressing space is in 1-page(0100h ~ 01FFH). It is set and clreared by SETG, CLRG instruction.

[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.

11.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but these devices have 20K/12K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

Figure 11-4 , 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 11-5 .

As shown in Figure 11-4 , each area is assigned a fixed location in Program Memory. Program Memory area contains the user program.

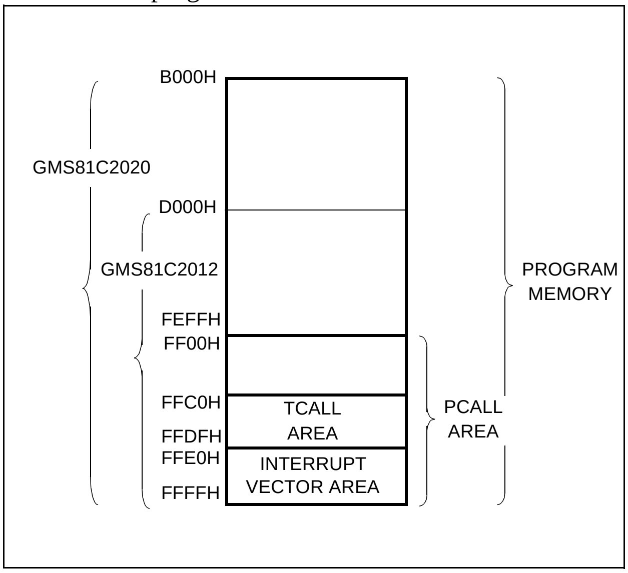


Figure 11-4 Program Memory Map

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 11-6 .

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.

As for the area from $0FF00_H$ to $0FFFF_H$, if any area of them is not going to be used, its service location is available as general purpose Program Memory.

Address	Vector Area Memory
$0FFE0_H$	-
E2	-
E4	Serial Peripheral Interface Interrupt Vector Area
E6	Basic Interval Interrupt Vector Area
E8	Watchdog Timer Interrupt Vector Area
EA	A/D Converter Interrupt Vector Area
EC	-
EE	-
F0	-
F2	-
F4	Timer/Counter 1 Interrupt Vector Area
F6	Timer/Counter 0 Interrupt Vector Area
F8	External Interrupt 1 Vector Area
FA	External Interrupt 0 Vector Area
FC	-
FE	RESET Vector Area

NOTE:
"-" means reserved area.

Figure 11-5 Interrupt Vector Area

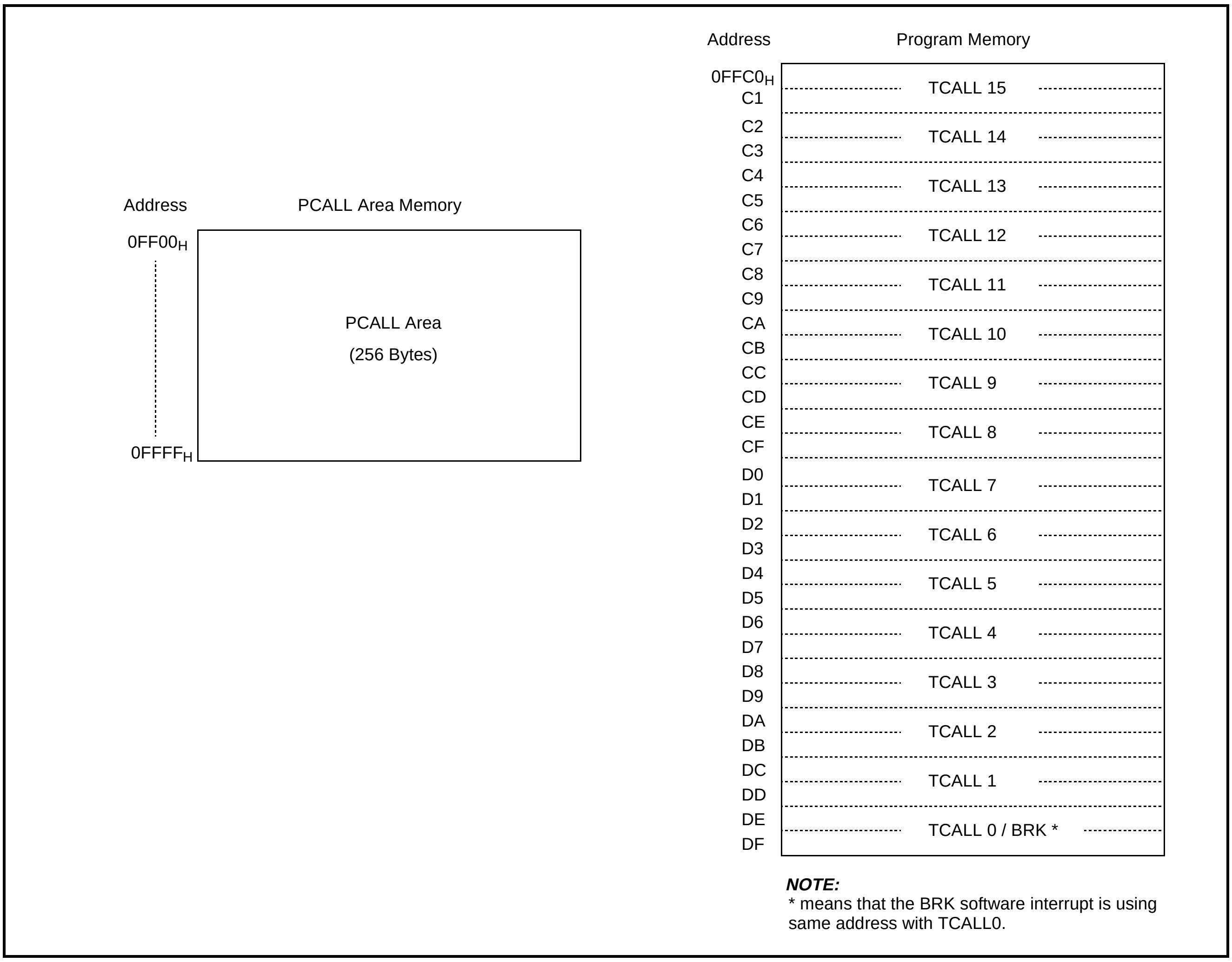
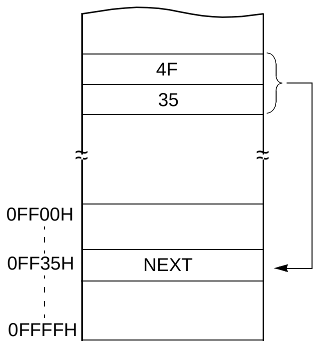


Figure 11-6 PCALL and TCALL Memory Area

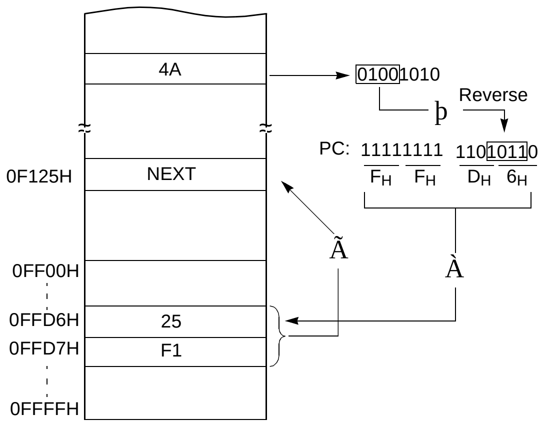
PCALL→ rel

```
4F35     PCALL  35H
```



TCALL→ n

```
4A       TCALL  4
```



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

```
        ORG     0FFE0H

        DW      NOT_USED; (0FFE0)
        DW      NOT_USED; (0FFE2)
        DW      SPI_INT; (0FFE4) Serial Peripheral Interface
        DW      BIT_INT; (0FFE6) Basic Interval Timer
        DW      WDT_INT; (0FFE8) Watchdog Timer
        DW      AD_INT; (0FFEA) A/D Converter
        DW      NOT_USED; (0FFEC)
        DW      NOT_USED; (0FFEE)
        DW      NOT_USED; (0FFF0)
        DW      NOT_USED; (0FFF2)
        DW      TMR1_INT; (0FFF4) Timer-1
        DW      TMR0_INT; (0FFF6) Timer-0
        DW      INT1; (0FFF8) Int.1
        DW      INT0; (0FFFA) Int.0
        DW      NOT_USED; (0FFFC)
        DW      RESET; (0FFFE) Reset


        ORG     0F000H

;*********************************************
;           MAIN     PROGRAM  *
;********************************************
;
;
RESET:  DI       ;Disable All Interrupts
        LDX     #0
RAM_CLR:LDA     #0;RAM Clear(!0000H->!00BFH)
        STA     {X}+
        CMPX    #0C0H
        BNE     RAM_CLR
;
        LDX     #01FFH;Stack Pointer Initialize
        TXSP
;
        CALL    INITIAL;
;
        LDM     R0, #0;Normal Port 0
        LDM     R0IO,#1000_0010B;Normal Port Direction
        LDM     R1, #0;Normal Port 1
        LDM     R1IO,#1000_0010B;Normal Port Direction
        :
        :
        LDM     PFDR,#0;Enable Power Fail Detector
        :
        :
```

11.3 Data Memory (GMS81C2020)

Figure 11-7 shows the internal Data Memory space available. Data Memory is divided into two groups, a user RAM(including Stack) and control registers.

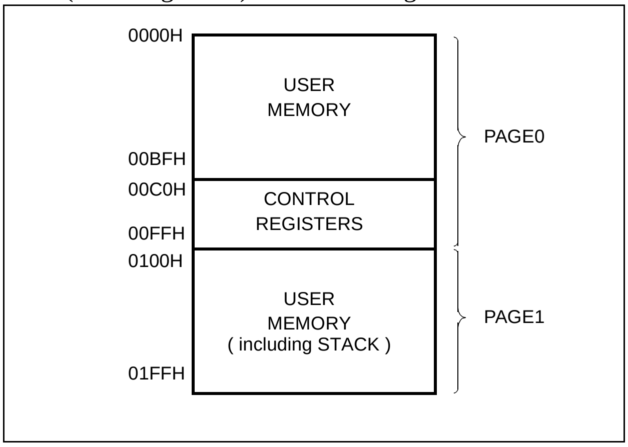


Figure 11-7 Data Memory Map

User Memory

The GMS81C2020 has 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 converter, basic interval timer, serial peripheral interface, watchdog timer, buzzer driver and I/O ports. The control registers are in address range of $0C0_H$ to $0FF_H$.

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

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

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

Example; To write at CKCTLR

```
LDM   CKCTLR,#09H ;Divide ratio ÷16
```

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	----_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	----_--00	byte
0F6H	R5FUNC	W	0000_0000	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	-

Table 11-1 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.

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

<table>
<tr><th rowspan="2">Addr.</th><th colspan="3">When read</th><th colspan="2">When write</th></tr>
<tr><th>Timer Mode</th><th>Capture Mode</th><th>PWM Mode</th><th>Timer Mode</th><th>PWM Mode</th></tr>
<tr><td>D1H</td><td>T0</td><td>CDR0</td><td>-</td><td>TDR0</td><td>-</td></tr>
<tr><td>D3H</td><td colspan="3">-</td><td>TDR1</td><td>T1PPR</td></tr>
<tr><td>D4H</td><td>T1</td><td>CDR1</td><td>T1PDR</td><td>-</td><td>T1PDR</td></tr>
<tr><td>ECH</td><td colspan="3">BITR</td><td colspan="2">CKCTLR</td></tr>
</table>

Table 11-2 Various Register Name in Same Address

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.

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							
ECH	BITR[1]	Basic Interval Timer Data Register							
ECH	CKCTLR[1]	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0

Table 11-3 Control Registers of GMS81C2020

These registers of shaded area can not be accessed by bit manipulation instruction as " SET1, CLR1 ", but should be accessed 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
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	SOUT	SIN	SCLK	-	-	-
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 11-3 Control Registers of GMS81C2020

These registers of shaded area can not be accessed by bit manipulation instruction as " SET1, CLR1 ", but should be accessed 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.

11.4 Data Memory (GMS81C2120)

Figure 11-8 shows the internal Data Memory space available. Data Memory is divided into two groups, a user RAM(including Stack) and control registers.

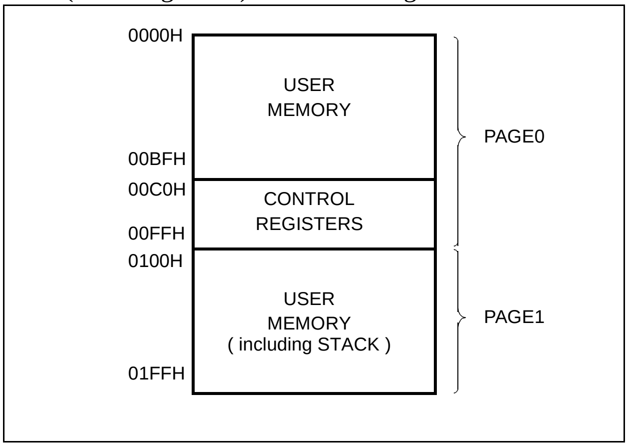


Figure 11-8 Data Memory Map

User Memory

The GMS81C2120 has 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 converter, basic interval timer, serial peripheral interface, watchdog timer, buzzer driver and I/O ports. The control registers are in address range of $0C0_H$ to $0FF_H$.

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

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

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

Example; To write at CKCTLR

```
LDM   CKCTLR,#09H ;Divide ratio ÷16
```

Address	Symbol	R/W	RESET Value	Addressing mode
0C0H	R0	R/W	Undefined	byte, bit[1]
0C1H	R0IO	W	0000_0000	byte[2]
0C4H	R2	R/W	Undefined	byte, bit
0C5H	R2IO	W	0000_0000	byte
0C6H	R3	R/W	Undefined	byte, bit
0C7H	R3IO	W	---0_0000	byte
0CAH	R5	R/W	Undefined	byte, bit
0CBH	R5IO	W	0000_0---	byte
0CCH	R6	R/W	Undefined	byte, bit
0CDH	R6IO	W	0000_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
0F6H	R5FUNC	W	0000_0---	byte
0F7H	R6FUNC	W	0000_0000	byte
0F9H	R5NODR	W	0000_0---	byte
0FAH	SCMR	R/W	---0_0000	byte
0FBH	RA	R	Undefined	-

Table 11-4 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.

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 11-5 Various Register Name in Same Address

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.

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])							
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[4:0])**							
C7H	R3IO	R3 Port Direction Register **(Bit[4:0])**							
CAH	R5	R5 Port Data Register (Bit[7:3])							
CBH	R5IO	R5 Port Direction Register (Bit[7:3])							
CCH	R6	R6 Port Data Register (Bit[7:0])							
CDH	R6IO	R6 Port Direction Register (Bit[7:0])							
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							
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	SOUT	SIN	SCLK	-	-	-

Table 11-6 Control Registers of GMS81C2120

These registers of shaded area can not be accessed by bit manipulation instruction as " SET1, CLR1 ", but should be accessed 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
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 11-6 Control Registers of GMS81C2120

These registers of shaded area can not be accessed by bit manipulation instruction as " SET1, CLR1 ", but should be accessed 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.

11.5 Addressing Mode

The GMS87C1404 and GMS87C1408 uses six addressing modes;

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

(1) Register Addressing

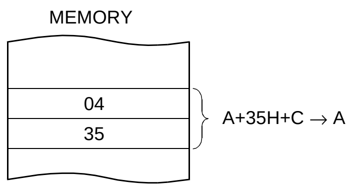
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

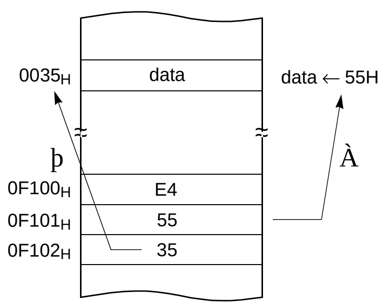
In this mode, second byte (operand) is accessed as a data immediately.

Example:

```
0435     ADC    #35H
```




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

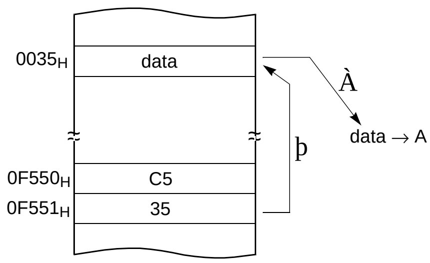


(3) Direct Page Addressing → dp

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

Example;

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



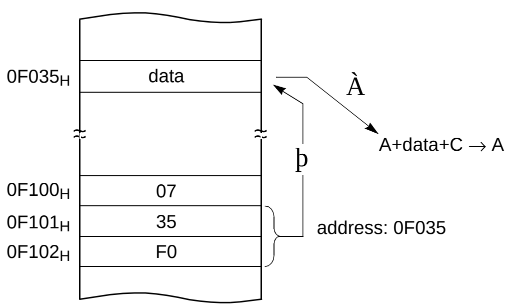
(4) Absolute Addressing → !abs

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;

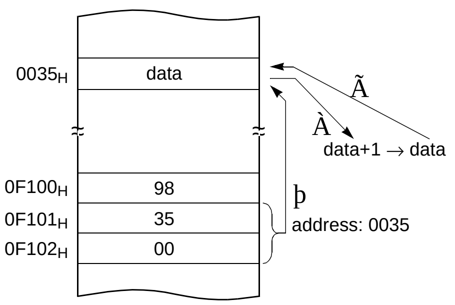
```
0735F0   ADC    !0F035H     ;A ←ROM[0F035H]
```



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

Example; Addressing accesses the address 0135_H .

```
983500   INC    !0035H      ;A ←RAM[035H]
```



(5) Indexed Addressing

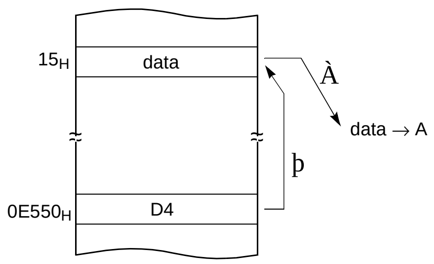
X indexed direct page (no offset) → {X}

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

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

Example; $X=15_H$

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



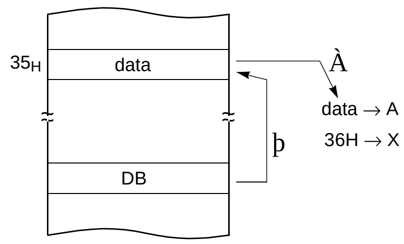
X indexed direct page, auto increment→ {X}+

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; $X=35_H$

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



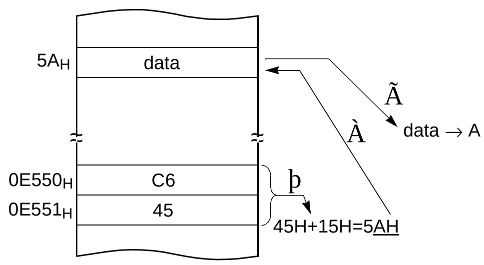
X indexed direct page (8 bit offset) → dp+X

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; $X=015_H$

```
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.

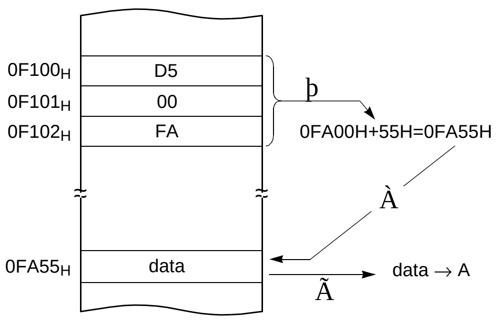
This is same with above (2). Use Y register instead of X.

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

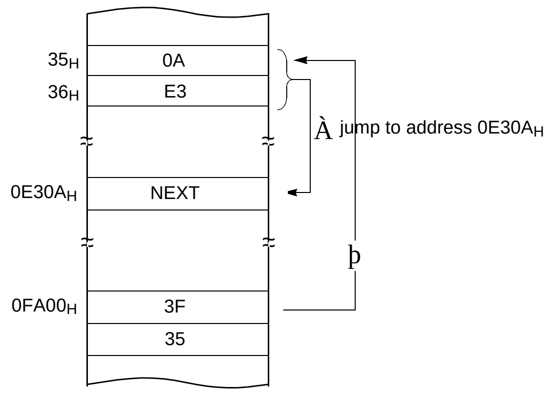
Direct page indirect → [dp]

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;

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



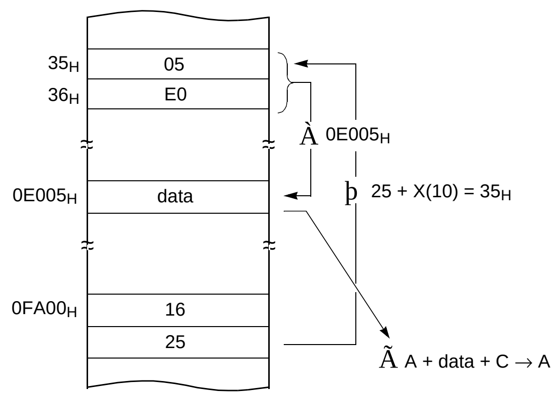
X indexed indirect → [dp+X]

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; $X=10_H$

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



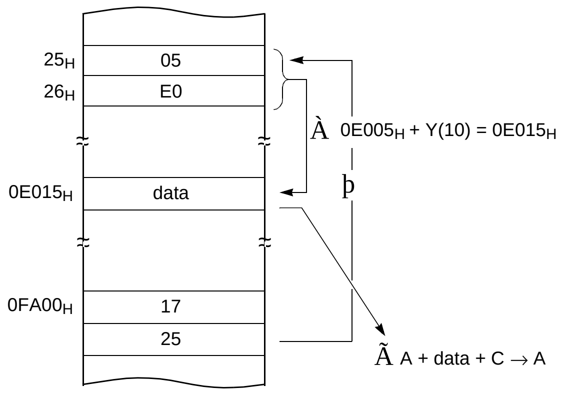
Y indexed indirect → [dp]+Y

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

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

Example; Y=10_H

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



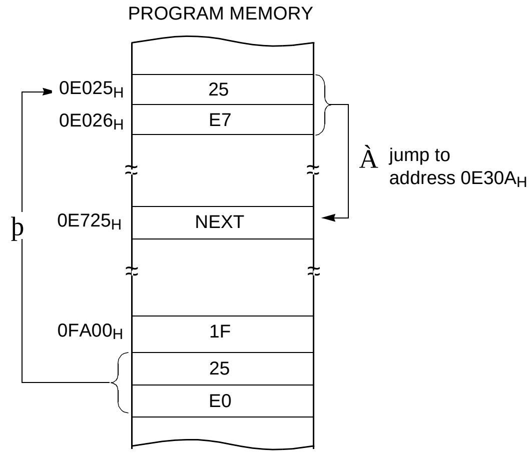
Absolute indirect → [!abs]

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

JMP

Example;

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



12. I/O PORTS

The GMS81C2020 has eight ports, R0, R1, R2, R3, R4, R5, R6 and R7. The GMS81C2120 has five ports, R0, R2, R3, R5 and R6. These ports pins may be multiplexed with an alternate function for the peripheral features on the device. In general, when a initial reset state, all ports are used as a general purpose input port.

All pins have data direction registers which can set these ports as output or input. A "1" in the port direction register defines the corresponding port pin as output. Conversely, write "0" to the corresponding bit to specify as an 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 $C1_H$ (R0 direction register) during initial setting as shown in Figure 12-1 .

Reading data register reads the status of the pins whereas writing to it will write to the port latch..

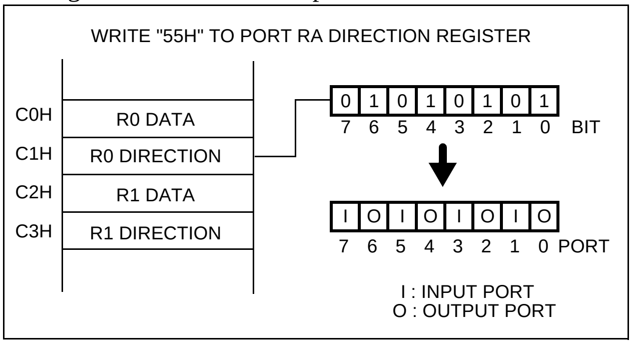
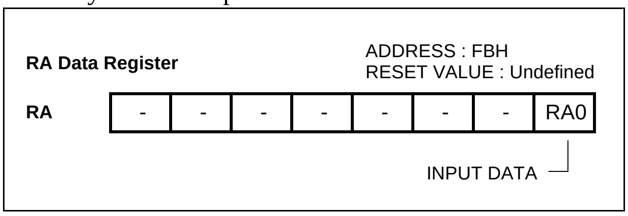


Figure 12-1 Example of port I/O assignment

12.1 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)

12.2 R0 and R0IO registers

R0 is an 8-bit *high-voltage* CMOS bidirectional I/O port (address $C0_H$). Each port can be set individually as input and output through the R0IO register (address $C1_H$). 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).

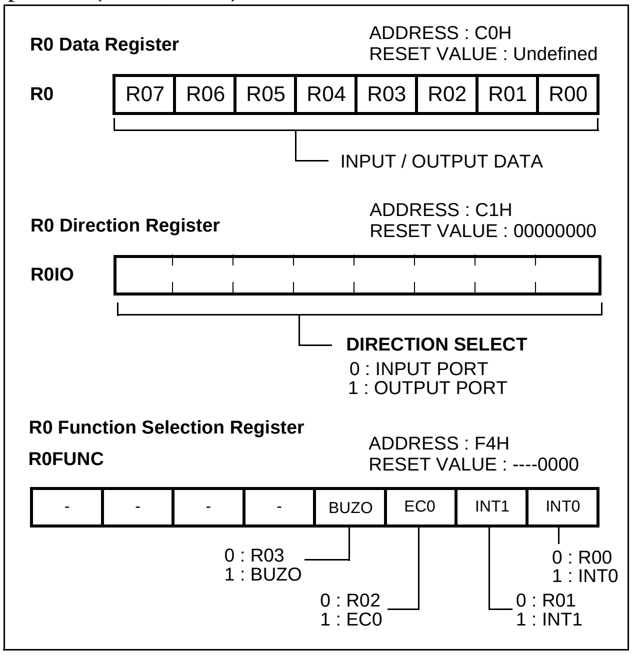


Figure 12-2 Registers of Port R0

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)

PORT	R0FUNC [3:0]	Description
R03/ BUZO	0	R00 (Normal I/O Port)
	1	BUZO (Buzzer Output Port)
R02/ EC0	0	R01 (Normal I/O Port)
	1	EC0 (Event Counter Input Port)
R01/ INT1	0	R01 (Normal I/O Port)
	1	INT1 (External interrupt 1 Input Port)

R00/ INT0	0	R00 (Normal I/O Port)
	1	INT0 (External interrupt 0 Input Port)

12.3 R1 and R1IO registers

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

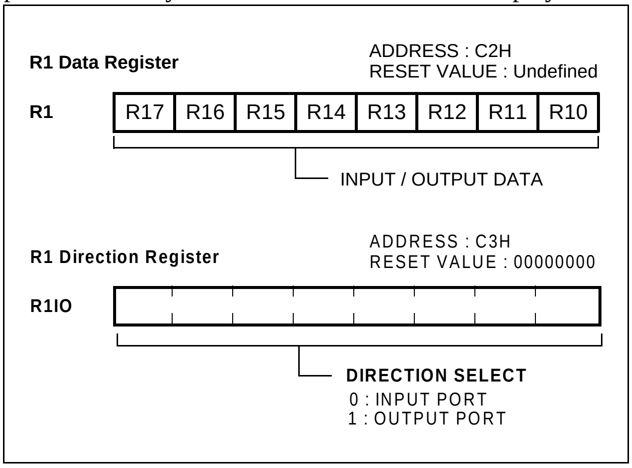


Figure 12-3 Registers of Port R1

12.4 R2 and R2IO registers

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

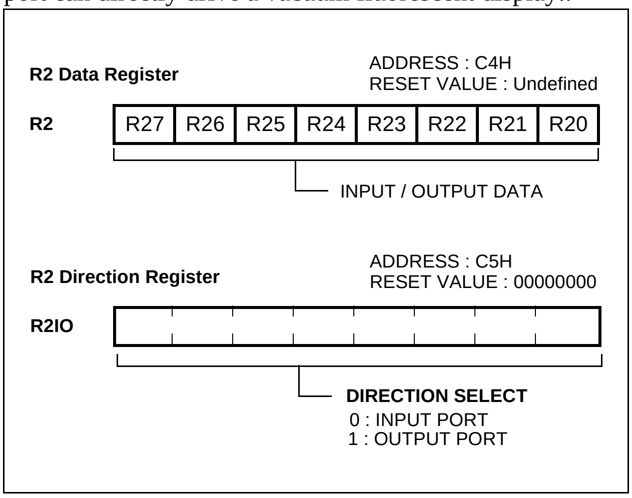


Figure 12-4 Registers of Port R2

12.5 R3 and R3IO registers

R1 is an 6-bit *high-voltage* CMOS bidirectional I/O port (address $C6_H$). Each port can be set individually as input and output through the R3IO register (address $C7_H$).

Each port can directly drive a vacuum fluorescent display..

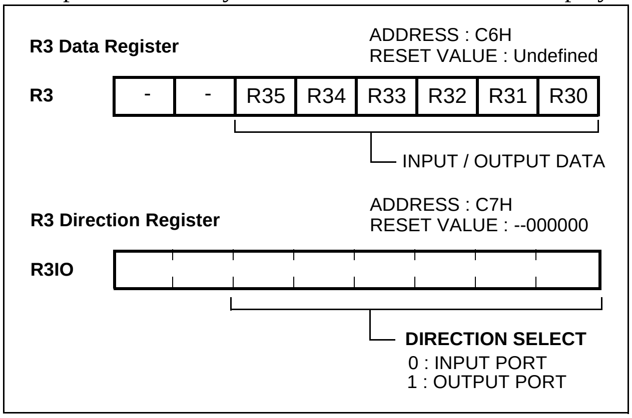


Figure 12-5 Registers of Port R3

12.6 R4 and R4IO registers

R4 is an 4-bit bidirectional I/O port (address $C8_H$). Each port can be set individually as input and output through the R4IO register (address $C9_H$).

R40 port is multiplexed with Timer 0 Output Port(T0O), r

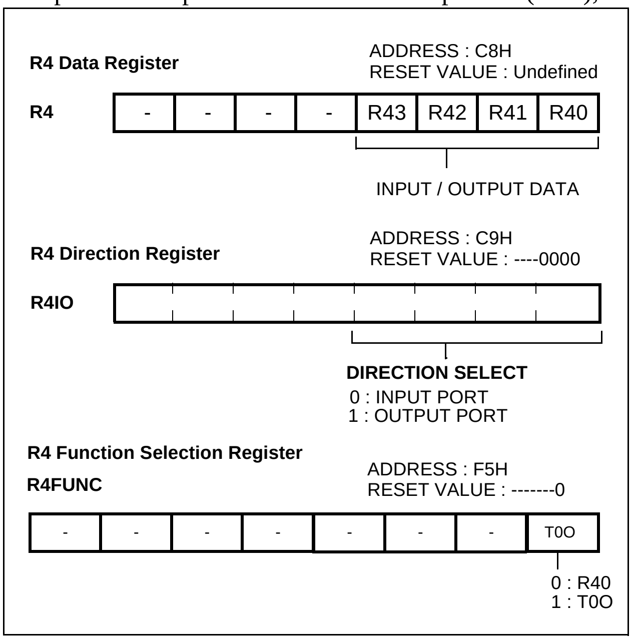


Figure 12-6 Registers of Port R4

The control register R4FUNC (address $F5_H$) 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)

PORT	R4FUNC [0]	Description
R40/ T0O	0	R40 (Normal I/O Port)
	1	T0O (Timer 0 Compare Output Port)

12.7 R5 and R5IO registers

R5 is an 8-bit bidirectional I/O port (address CA_H). Each pin can be set individually as input and output through the R5IO register (address CB_H).In addition, Port R5 is multiplexed with Serial Peripheral Interface (SPI). The control register R5FUNC (address $F6_H$) controls to select Serial Peripheral Interface function.After reset, the R5IO register value is "0", port may be used as general I/O ports. To select Serial Peripheral Interface function, write "1" to the corresponding bit of R5FUNC.

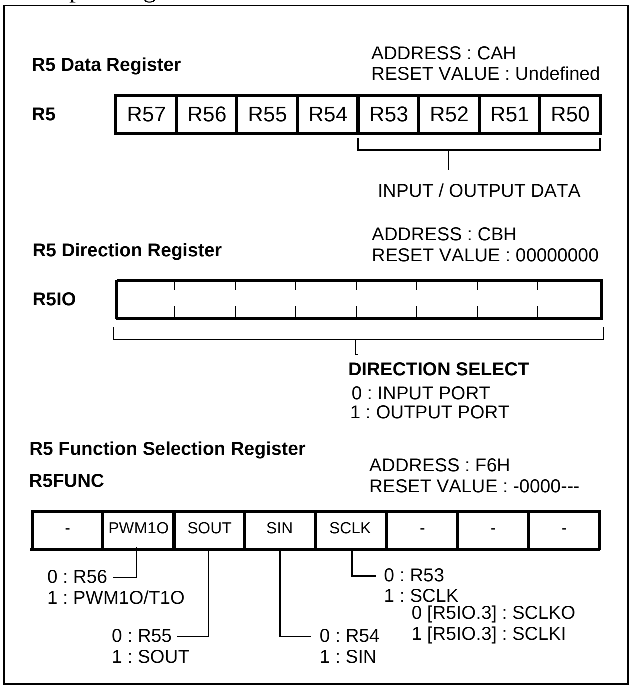


Figure 12-7 Registers of Port R5

PORT	R5FUNC [6:3]	Description
R56/ PWM1O/ T1O	0	R56 (Normal I/O Port)
	1	PWM1 Data Output / Timer 1 Data Output
R55/SOUT	0	R55 (Normal I/O Port)
	1	SPI Serial Data Output
R54/SIN	0	R54 (Normal I/O Port)
	1	SPI Serial Data Input
R53/SCLK	0	R53 (Normal I/O Port)
	0 [R5IO.3] SCLKO	SPI Synchronous Clock Output
	1 [R5IO.3] SCLKI	SPI Synchronous Clock Input

Table 12-1 Registers of Port R5FUNC

12.8 R6 and R6IO registers

R6 is an 8-bit bidirectional I/O port (address CC_H). Each port can be set individually as input and output through the R6IO register (address CD_H).

R67~R60 ports are multiplexed with Analog Input Port (AN7~AN0)..

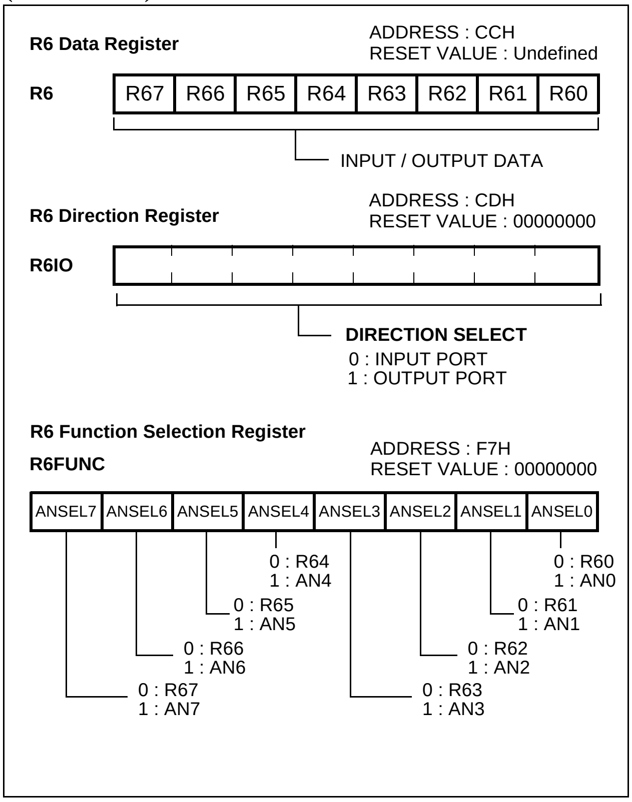


Figure 12-8 Registers of Port R6

The control register R6FUNC (address $F7_H$) 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)

PORT	R6FUNC [7:0]	Description
R67/AN7	0	R67 (Normal I/O Port)
	1	AN7 (ADS3~0=0111)
R66/AN6	0	R66 (Normal I/O Port)
	1	AN6 (ADS3~0=0110)
R65/AN5	0	R65 (Normal I/O Port)
	1	AN5 (ADS3~0=0101)
R64/AN4	0	R64 (Normal I/O Port)
	1	AN4 (ADS3~0=0100)
R63/AN3	0	R63 (Normal I/O Port)
	1	AN3 (ADS3~0=0011)
R62/AN2	0	R62 (Normal I/O Port)
	1	AN2 (ADS3~0=0010)
R61/AN1	0	R61 (Normal I/O Port)
	1	AN1 (ADS3~0=0001)
R60/AN0	0	R60 (Normal I/O Port)
	1	AN0 (ADS3~0=0000)

12.9 R7 and R7IO registers

R7 is an 4-bit bidirectional I/O port (address CE_H). Each port can be set individually as input and output through the R7IO register (address CF_H).

R73~R70 ports are multiplexed with Analog Input Port AN11~AN8)..

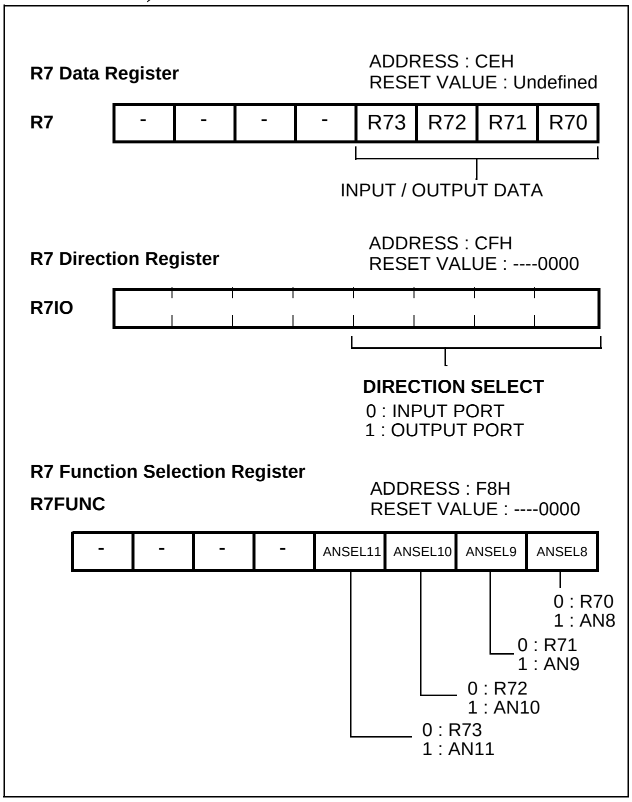


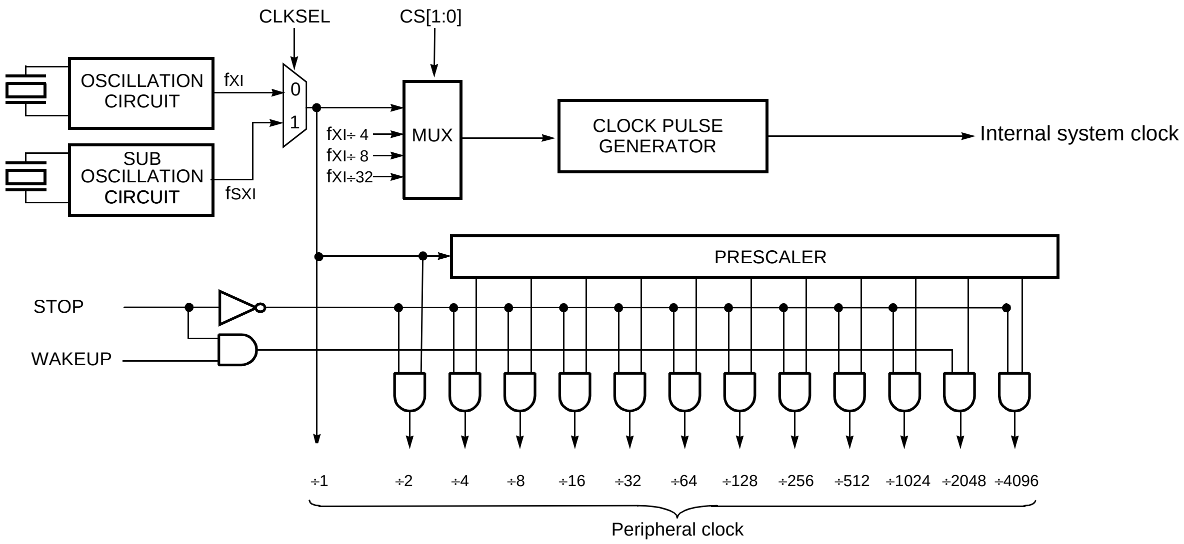
Figure 12-9 Registers of Port R6

The control register R7FUNC (address $F8_H$) 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.

PORT	R7FUNC [7:0]	Description
R73/AN11	0	R73 (Normal I/O Port)
	1	AN11 (ADS3~0=1011)
R72/AN10	0	R72 (Normal I/O Port)
	1	AN10 (ADS3~0=1010)
R71/AN9	0	R71 (Normal I/O Port)
	1	AN9 (ADS3~0=1001)
R70/AN8	0	R70 (Normal I/O Port)
	1	AN8 (ADS3~0=1000)

13. CLOCK GENERATOR

The clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and peripheral hardware. The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator connected to the XI and XO pins. External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the XI pin and open the XO pin.



System Clock Mode Register

SCMR	-	-	-	CS1	CS0	SUBOFF	CLKSEL	MAINOFF

ADDRESS : FAH
RESET VALUE : ---00000

CS[1:0] — Clock selection enable bits
00 : fXI ÷ 210 : fXI ÷16
01 : fXI ÷ 811 : 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

Figure 13-1 Block Diagram of Clock Pulse Generator

13.1 Oscillation Circuit

XI and XO are the input and output, respectively, a inverting amplifier which can be set for use as an on-chip oscillator, as shown in Figure 13-2 .

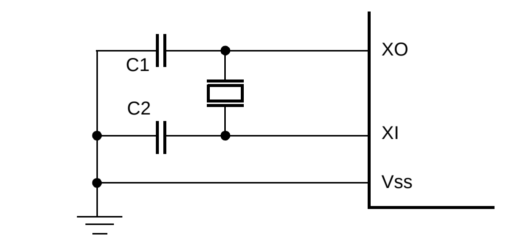


Figure 13-2 Oscillator Connections

SXI and SXO are the input and output, respectively, a inverting amplifier which can be set for use as an on-chip oscillator, as shown in Figure 13-2 .

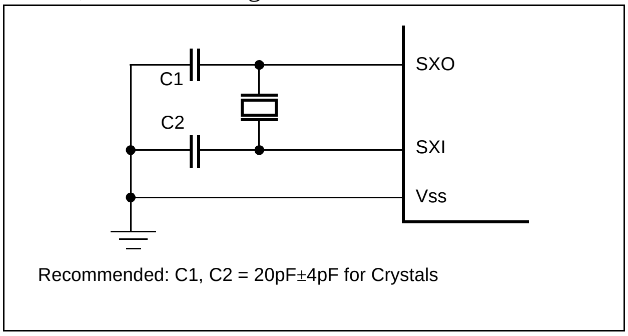
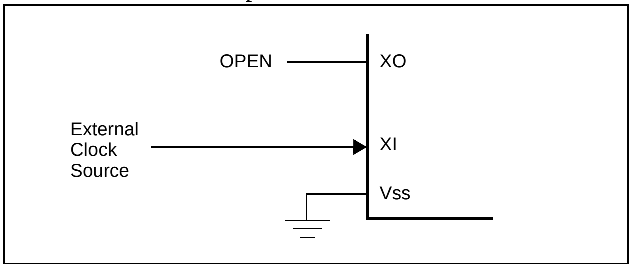


Figure 13-3 Sub Oscillator Connections

To drive the device from an external clock source, XO should be left unconnected while XI is driven as shown in Figure 13-4 . There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

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 external RC oscillator. Since External RC oscillator has their own characteristic, the user should figure out the appropriate value of external resister. (Please refer the DC Spec)

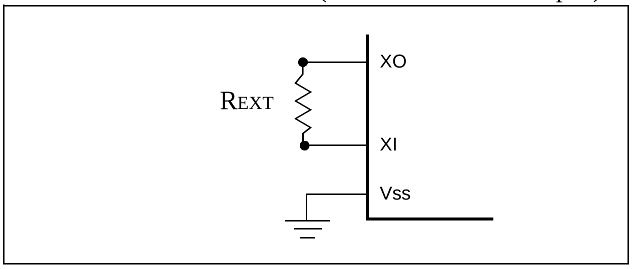


Figure 13-4 External R Connection

Note: *When using a system clock oscillator, carry out wiring in the broken line area in Figure 13-2 to prevent any effects from wiring capacities.*
- Minimize the wiring length.
- Do not allow wiring to intersect with other signal conductors.
- Do not allow wiring to come near changing high current.
- Set the potential of the grounding position of the oscillator capacitor to that of Vss. Do not ground to any ground pattern where high current is present.
- Do not fetch signals from the oscillator.

14. Basic Interval Timer

The GMS81C2020 and GMS81C2120 has one 8-bit Basic Interval Timer that is free-run, can not stop. Block diagram is shown in Figure 14-1 .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.

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 fXI÷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

Note: *All control bits of Basic interval timer are in CKCTLR register which is located at same address of BITR (address EC_H). Address EC_H is read as BITR, written to CKCTLR. Therefore, the CKCTLR can not be accessed by bit manipulation instruction.*

.

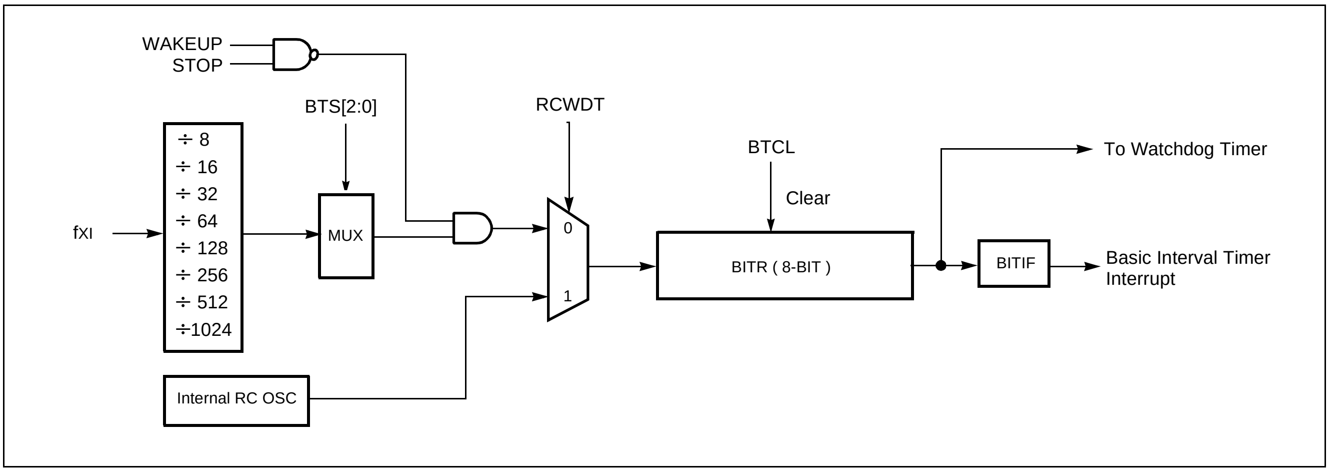


Figure 14-1 Block Diagram of Basic Interval Timer

Clock Control Register

CKCTLR	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0

ADDRESS : ECH
RESET VALUE : -0010111
Bit Manipulation Not Available

Basic Interval Timer Clock Selection

000 : fXI ÷ 8
001 : fXI ÷ 16
010 : fXI ÷ 32
011 : fXI ÷ 64
100 : fXI ÷ 128
101 : fXI ÷ 256
110 : fXI ÷ 512
111 : fXI ÷ 1024

Symbol	Function Description
WAKEUP	1: Enables Wake-up Timer 0: Disables Wake-up Timer
RCWDT	1: Enables Internal RC Watchdog Timer 0: Disables Internal RC Watchdog Timer
WDTON	1: Enables Watchdog Timer 0: Operates as a 7-bit Timer
BTCL	1: BITR is cleared and BTCL becomes "0" automatically after one machine cycle, and BITR continue to count-up

Figure 14-2 CKCTLR : Clock Control Register

15. TIMER / COUNTER

The GMS81C2020 and GMS81C2120 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 can be used either the two 8-bit Timer/Counter or one 16-bit Timer/Counter by combining 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 0-to-1 (rising & falling edge) transition at its corresponding external input pin, EC0(Timer 0).

In addition the "capture" function, the register is increased in response external interrupt same with timer function. When external interrupt 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 TMx as shown in Figure 15-1 and Table 12-1 .

Timer 0 Mode Register

TM0	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST

ADDRESS : D0H
RESET VALUE : --000000

CAP0 — Capture mode selection bit.
0 : Disables Capture
1 : Enables Capture

T0CK[2:0] — Input clock selection
000 : fXI ÷ 2100 : fXI ÷ 128
001 : fXI ÷ 4101 : fXI ÷ 512
010 : fXI ÷ 8110 : fXI ÷ 2048
011 : fXI ÷ 32111 : External Event (EC0)

T0CN — Continue control bit
0 : Stop counting
1 : Start counting continuously

T0ST — Start control bit
0 : Stop counting
1 : Counter register is cleared and start again

Timer 1 Mode Register

TM1	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST

ADDRESS : D2H
RESET VALUE : 00000000

POL — PWM Output Polarity
0 :Duty active low
1 : Duty active high

16BIT — 16-bit mode selection
0 : 8-bit mode
1 : 16-bit mode

PWM1E — PWM enable bit
0 : Disables PWM
1 : Enables PWM

CAP1 — Capture mode selection bit.
0 : Disables Capture
1 : Enables Capture

T1CK[2:0]] — Input clock selection
00 : fXI 10 : fXI ÷ 8
01 : fXI ÷ 211 : using the Timer 0 clock

T1CN — Continue control bit
0 : Stop counting
1 : Start counting continuously

T1ST — Start control bit
0 : Stop counting
1 : Counter register is cleared and start again

Figure 15-1 Timer Mode Register (TMx , x = 0~1)

16BIT	CAP0	CAP1	PWM1E	T0CK[2:0]	T1CK[1:0]	PWMO	TIMER 0	TIMER1
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	8-bit Compare output
0	X[1]	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	
1	0	0	0	XXX	11	1	16-bit Compare output	

Table 15-1 Operating Modes of Timer 0 and Timer 1

1. X : The value "0" or "1" corresponding your operation.

15.1 8-bit Timer/Counter Mode

The GMS81C2020 and GMS81C2120 has four 8-bit Timer/Counters, Timer 0, Timer 1 as shown in Figure 15-2 .

The "timer" or "counter" function is selected by mode registers TMx as shown in Figure 15-1 and Table 15-1 . 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 15-1).

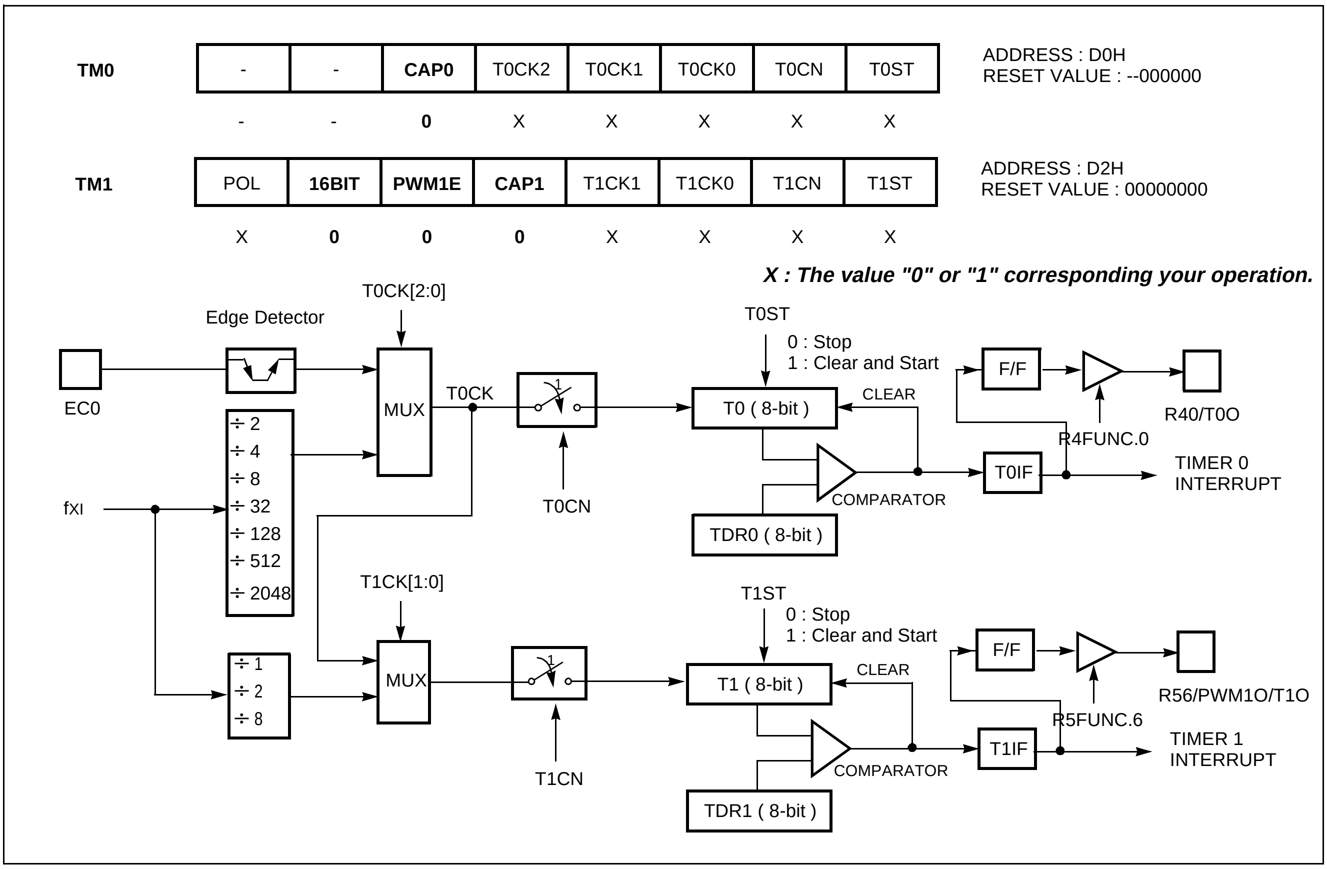


Figure 15-2 8-bit Timer / Counter Mode

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 T0CK2, T0CK1 and T0CK0 of register TM0) and 1, 2, 8 (selected by control bits T1CK1 and T1CK0 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.

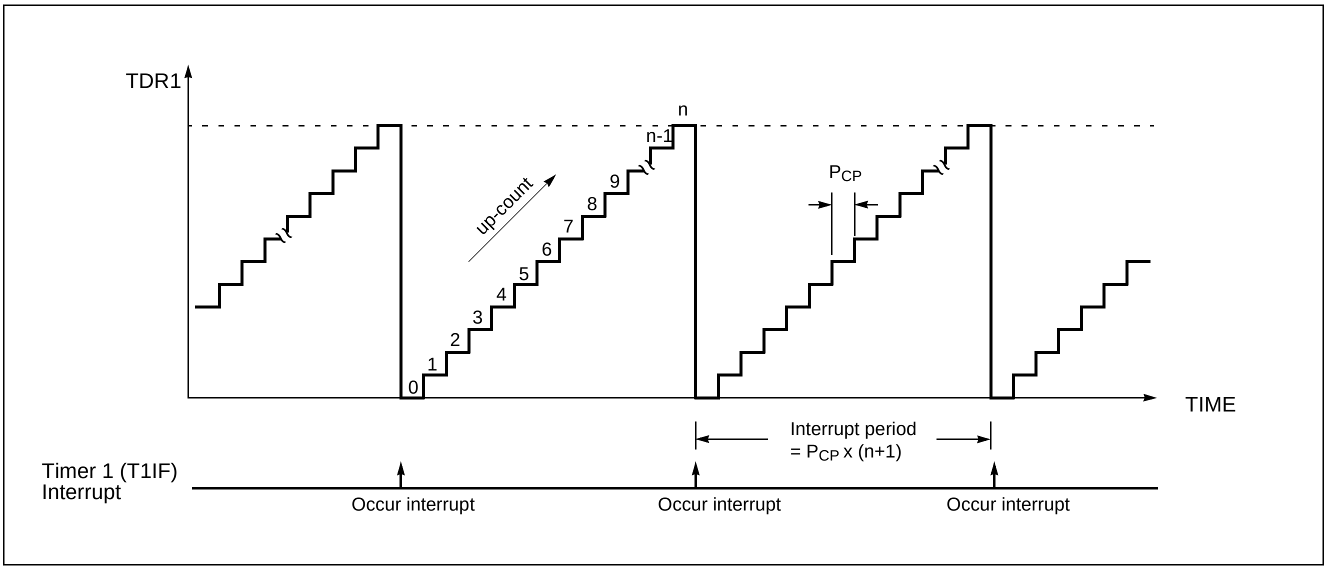


Figure 15-3 Counting Example of Timer Data Registers

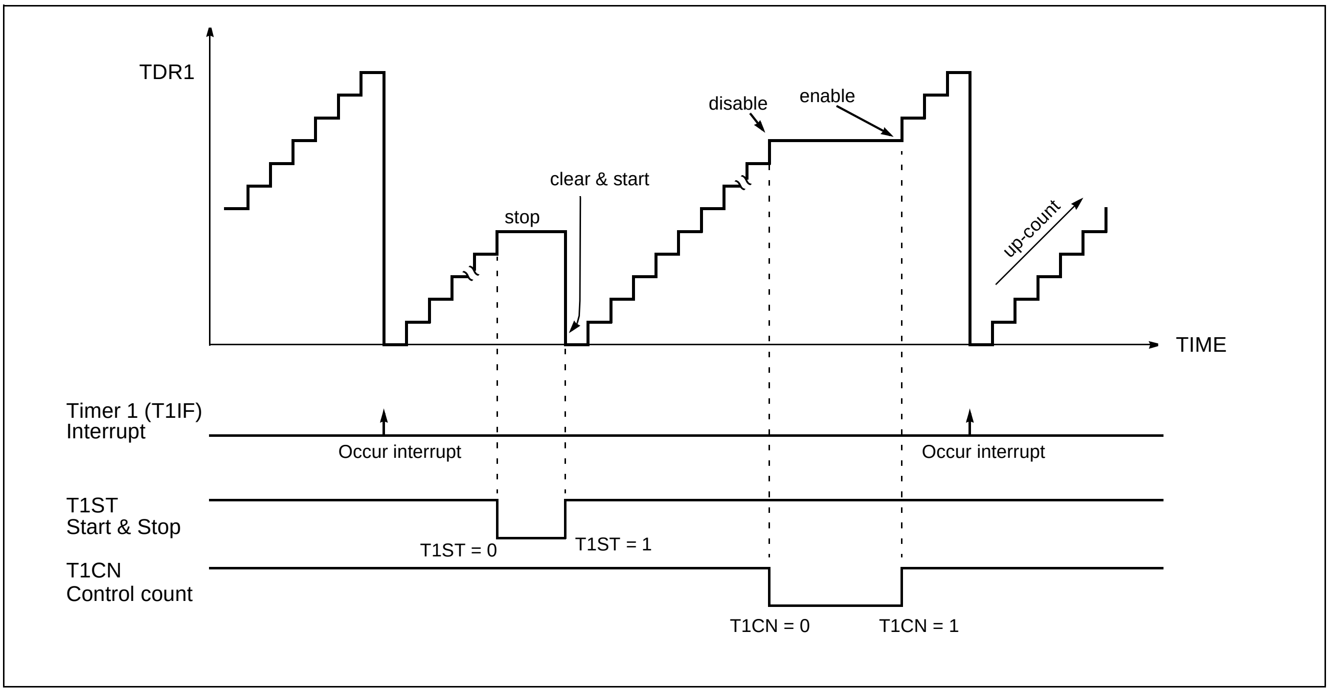


Figure 15-4 Timer Count Operation

15.2 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 T0CK2, T0CK1 and T0CK0.

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

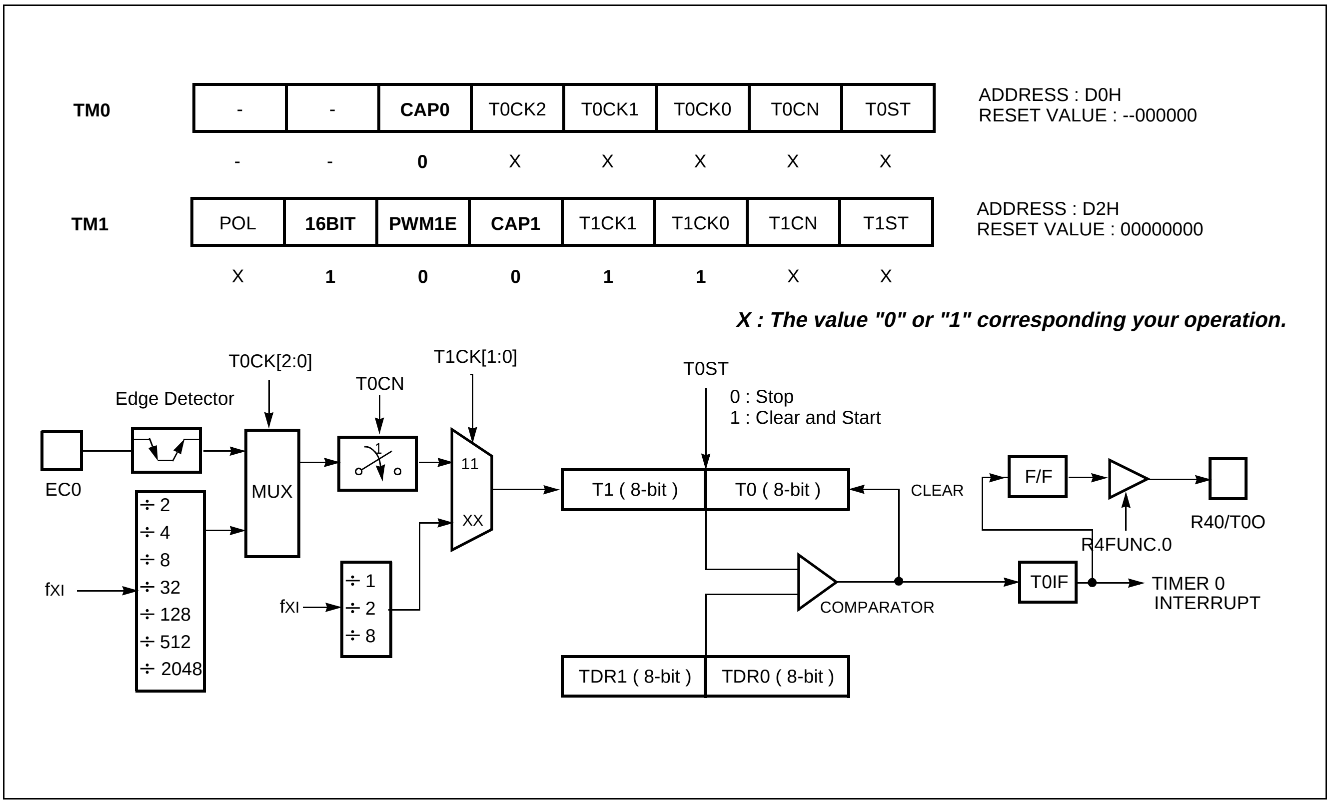


Figure 15-5 16-bit Timer / Counter Mode

15.3 8-bit Compare Output (16-bit)

The GMS81C2020 and GMS81C2120 has a function of Timer Compare Output. To pulse out, the timer match can goes to port pin(T0O, T1O) as shown in Figure 15-2 and Figure 15-5 . Thus, pulse out is generated by the timer match. These operation is implemented to pin, T0O, PWM1O/T1O.

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

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

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.

15.4 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 15-6 .

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 15-8 , the pulse width of captured signal is wider than the timer data value (FF_H) over 2 times. When external interrupt is occured, the captured value (13_H) is more little than wanted value. It can be obtained correct value by counting the number of timer overflow occurence.

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.

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.

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.*

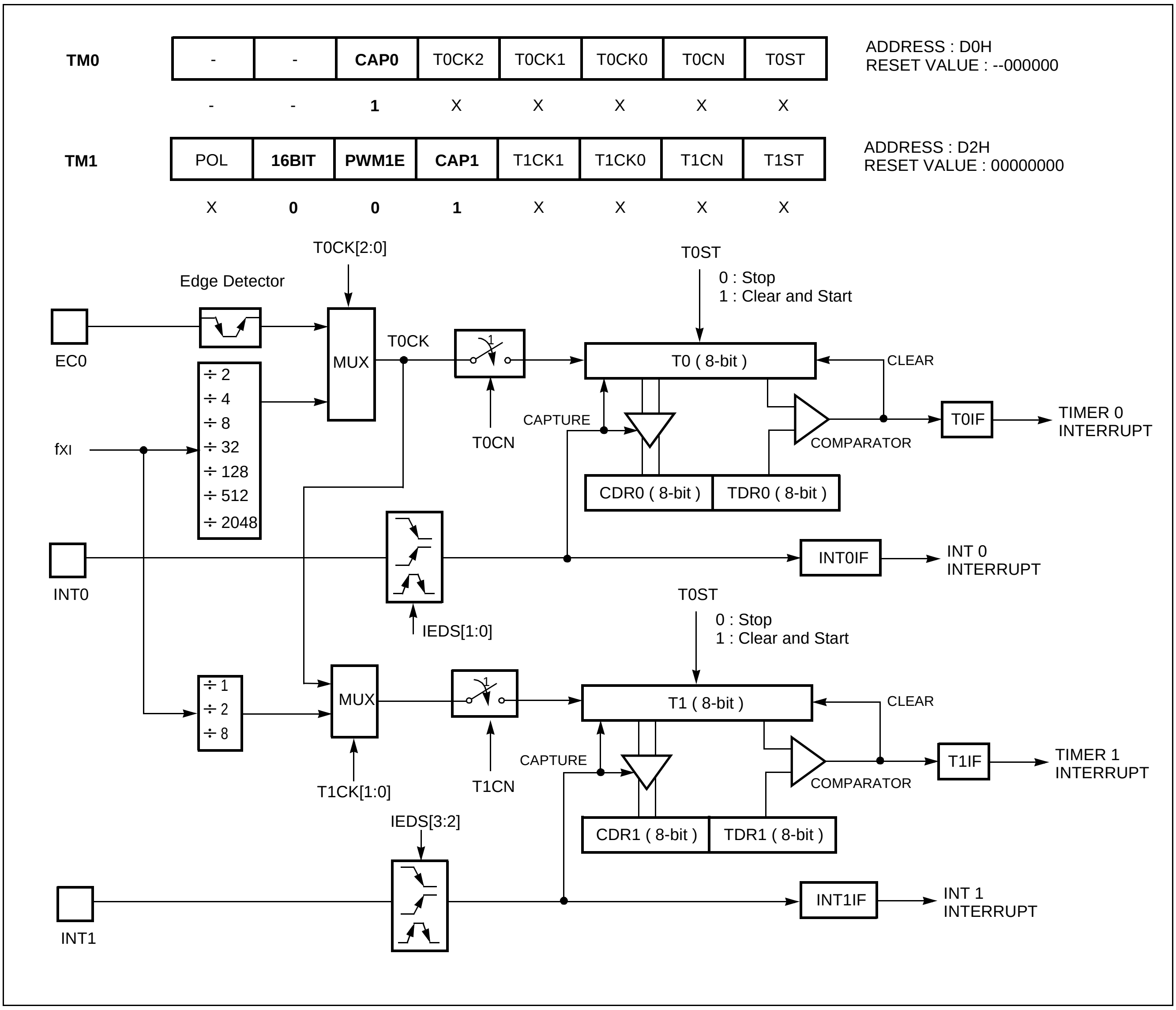


Figure 15-6 8-bit Capture Mode

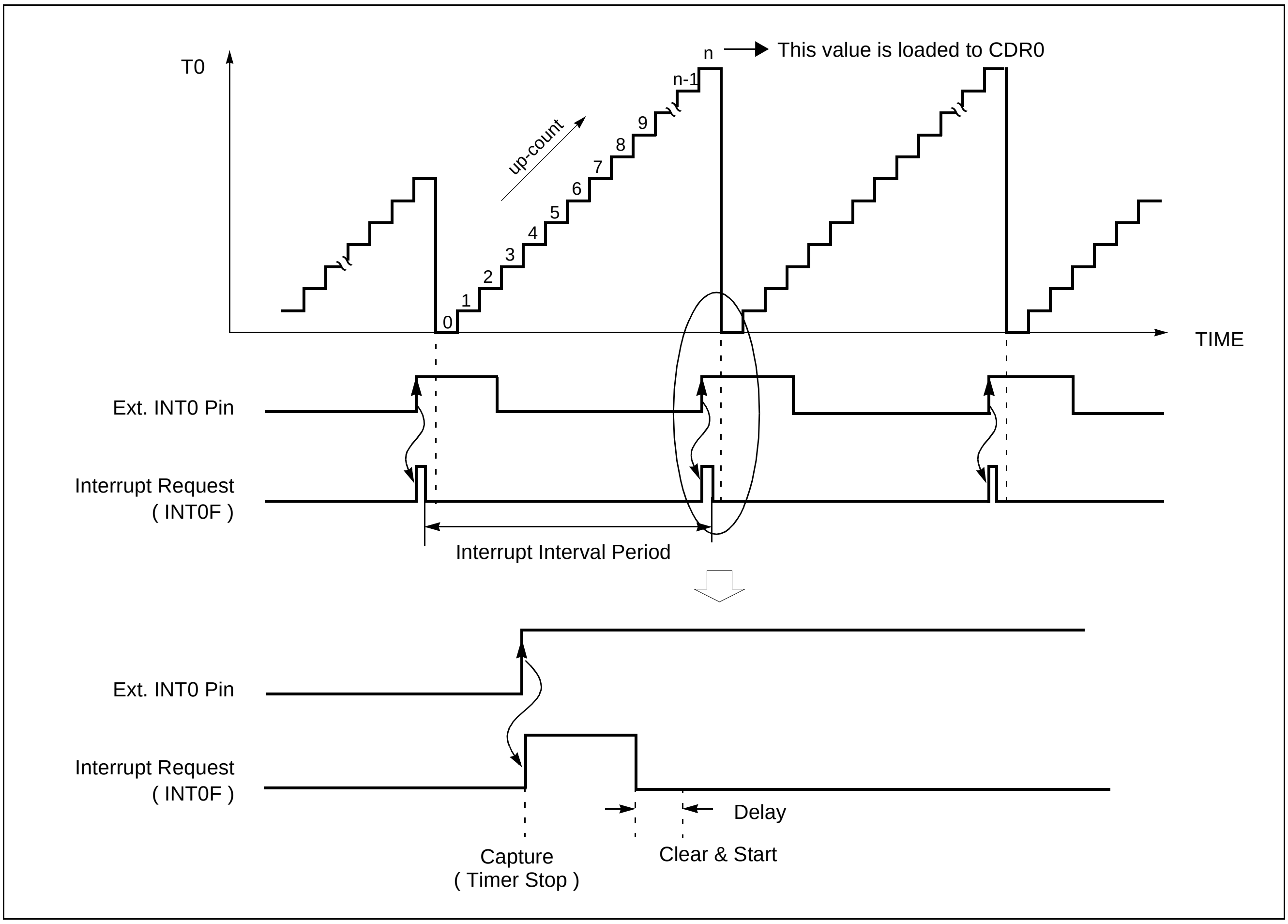


Figure 15-7 Input Capture Operation

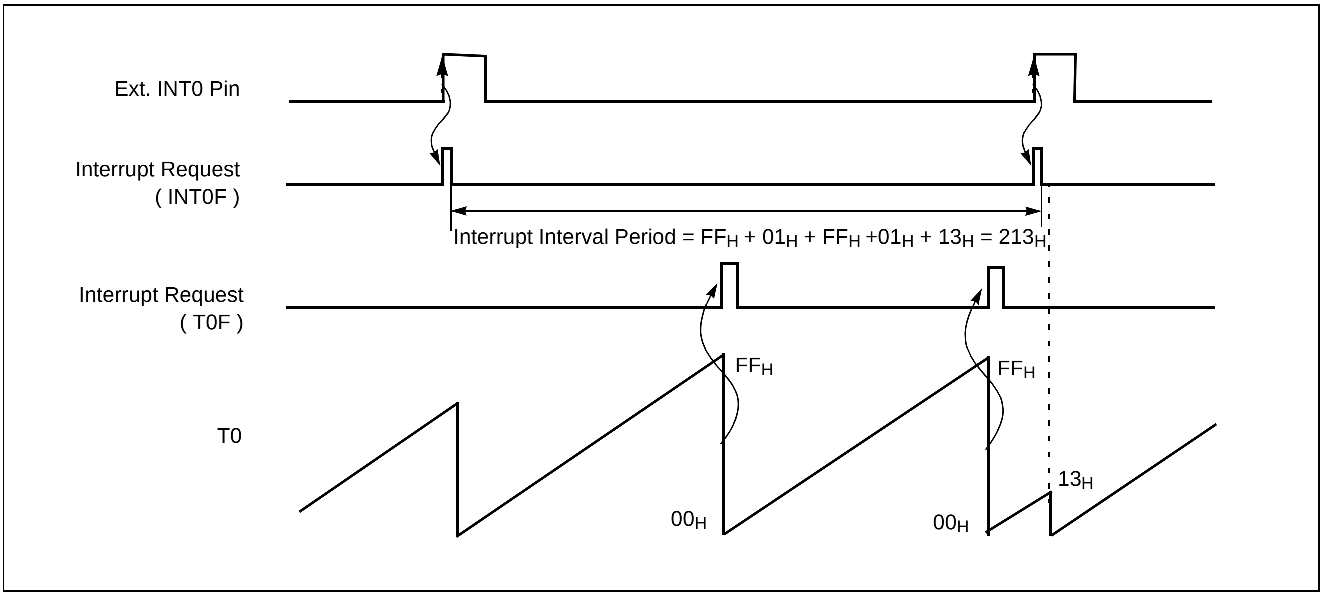


Figure 15-8 Excess Timer Overflow in Capture Mode

15.5 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.

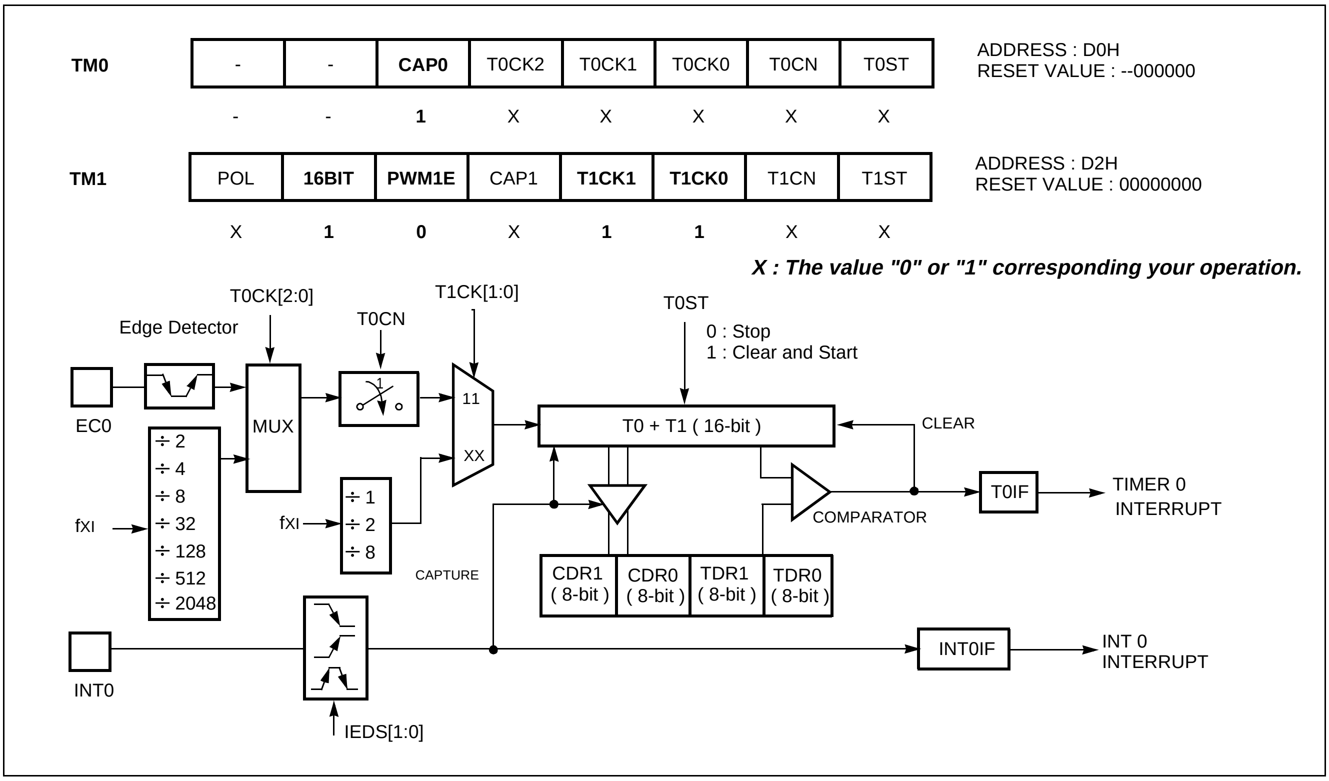


Figure 15-9 16-bit Capture Mode

15.6 PWM Mode

The GMS81C2020 and GMS81C2120 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 glitchless PWM output. In Figure 15-10 , the duty data is transfered 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 15-2 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 15-2 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 15-12 . As it were, the absolute duty time is not changed in varying frequency. But the changed period value must greater than the duty value.

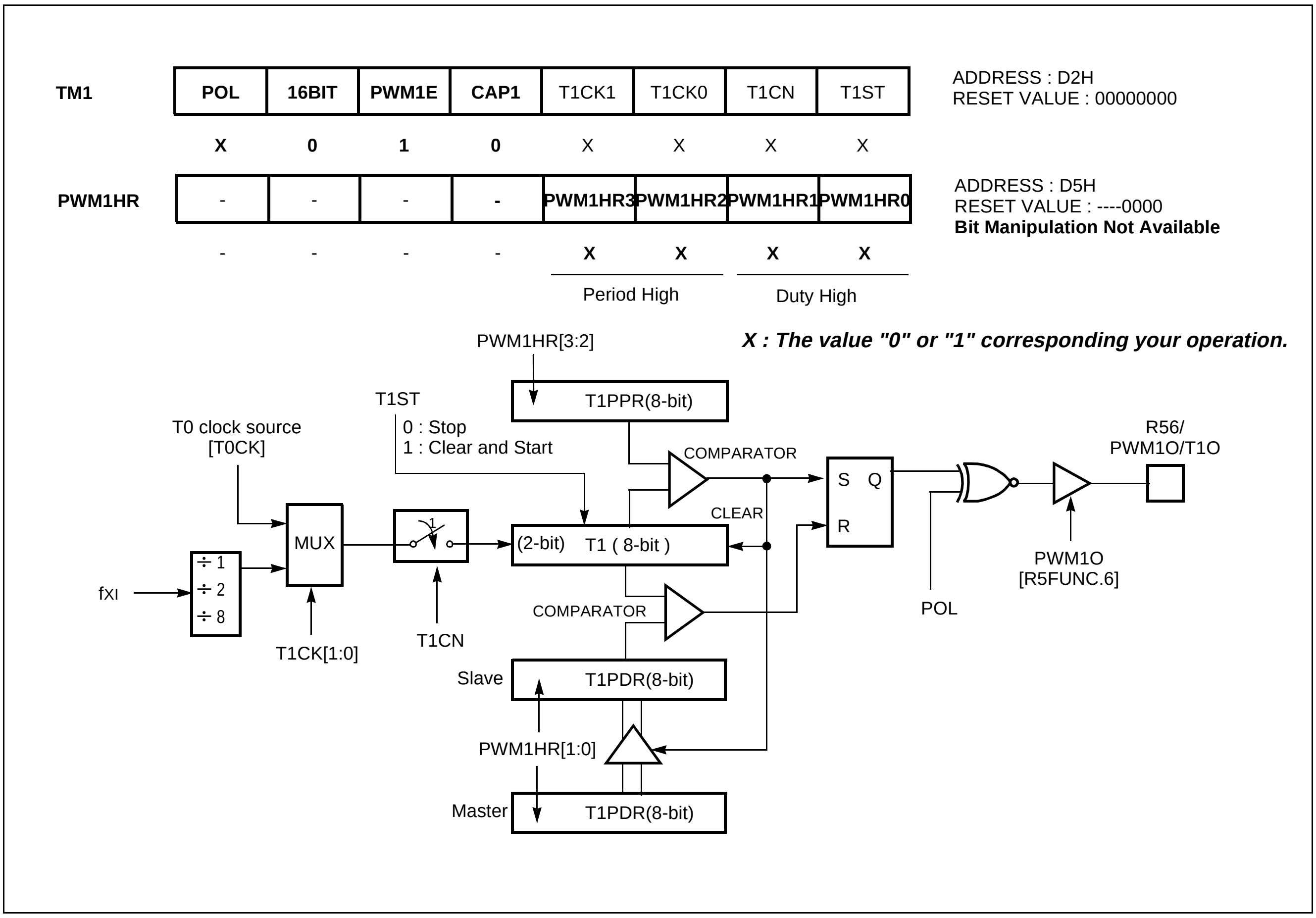


Figure 15-10 PWM Mode

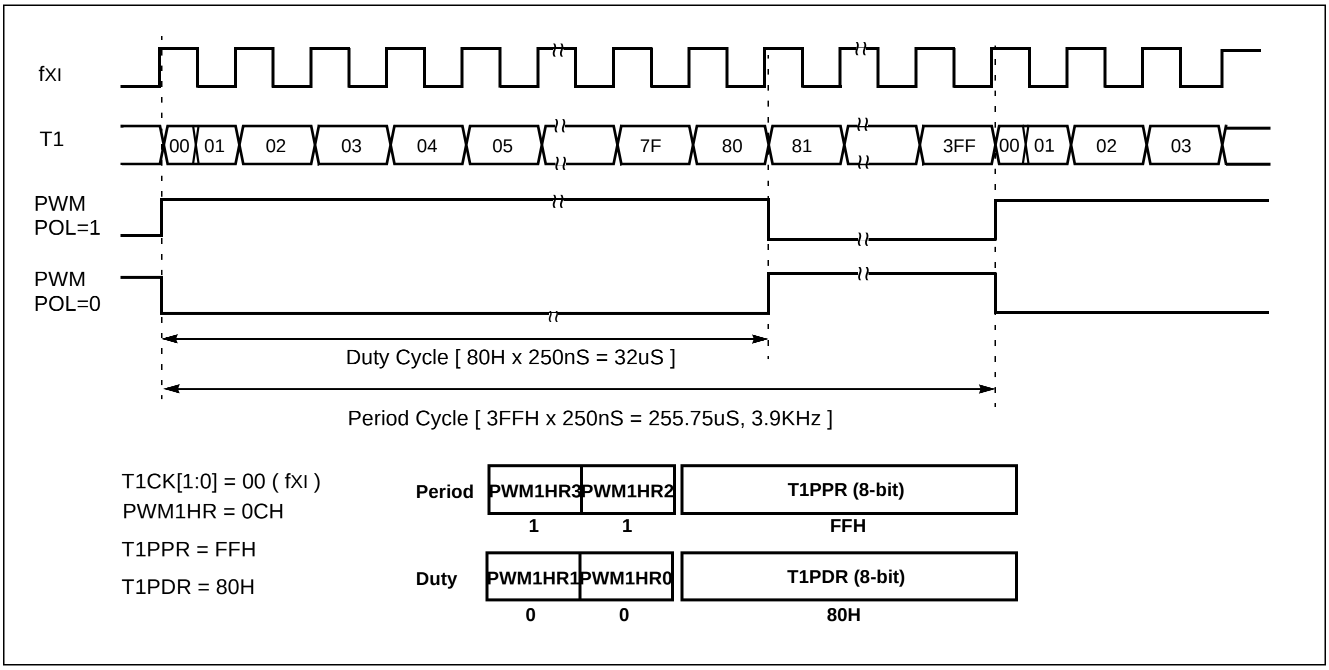


Figure 15-11 Example of PWM at 4MHz

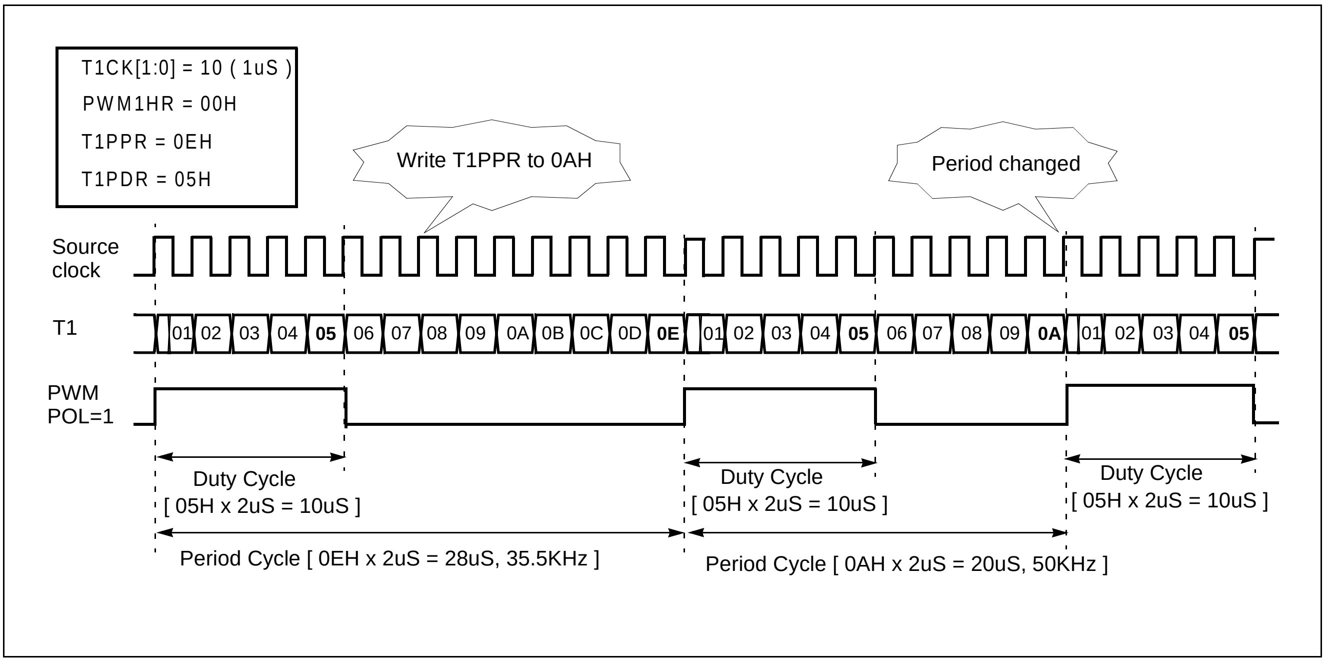


Figure 15-12 Example of Changing the Period in Absolute Duty Cycle (@4MHz)

16. 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.

SPI Mode Control Register

SIOM	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF

ADDRESS : E0H
RESET VALUE : 00000000

POL — Serial Clock Polarity Selection bit.
0 : Data Transmission at falling edge
(Received data latch at rising edge)
1 : Data Transmission at rising edge
(Received data latch at falling edge)

IOSW — Serial Input Pin Selection bit
0 : SIN(R54) Pin Selection
1 : SOUT(R55) Pin Selection

SM[1:0] — Serial Operation Mode Selection bits
00 : Normal Port (R55, R54, R53)
01 : Transmit Mode (SOUT,R54, SCLK)
10 : Receive Mode (R55, SIN, SCLK)
11 : Transmit & Receive Mode (SOUT, SIN, SCLK)

SCK[1:0] — Serial Clock Selection bits
00 : fXI ÷ 4
01 : fXI ÷ 16
10 : TMR0OV (Overflow of Timer 0)
11 : External Clock

SIOST — Serial Transmit Start bit
0 : Disable
1 : Start (After one SCLK, becomes “0”)

SIOSF — Serial Transmit Status bit
0 : During Transmission
1 : Finished

SPI Data Register

SIOR								

ADDRESS : E1H
RESET VALUE : Undefined

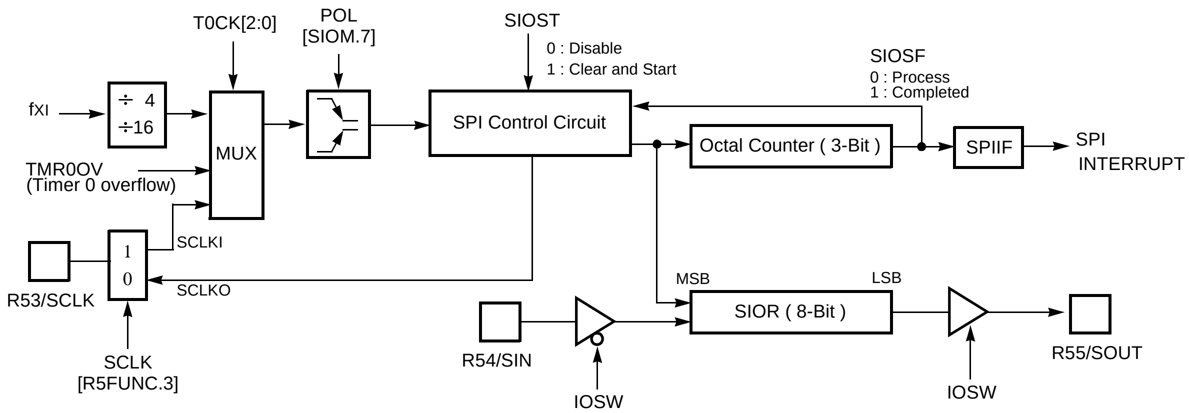


Figure 16-1 SPI Registers and Block Diagram

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

The serial data transfer operation mode is decided by setting the SM1 and SM0 of SPI Mode Control Register, and the transfer clock rate is decided by setting the SCK1 and SCK0 of SPI Mode Control Register as shown in Figure 16-1 . And the polarity of transfer clock is selected by setting the POL..

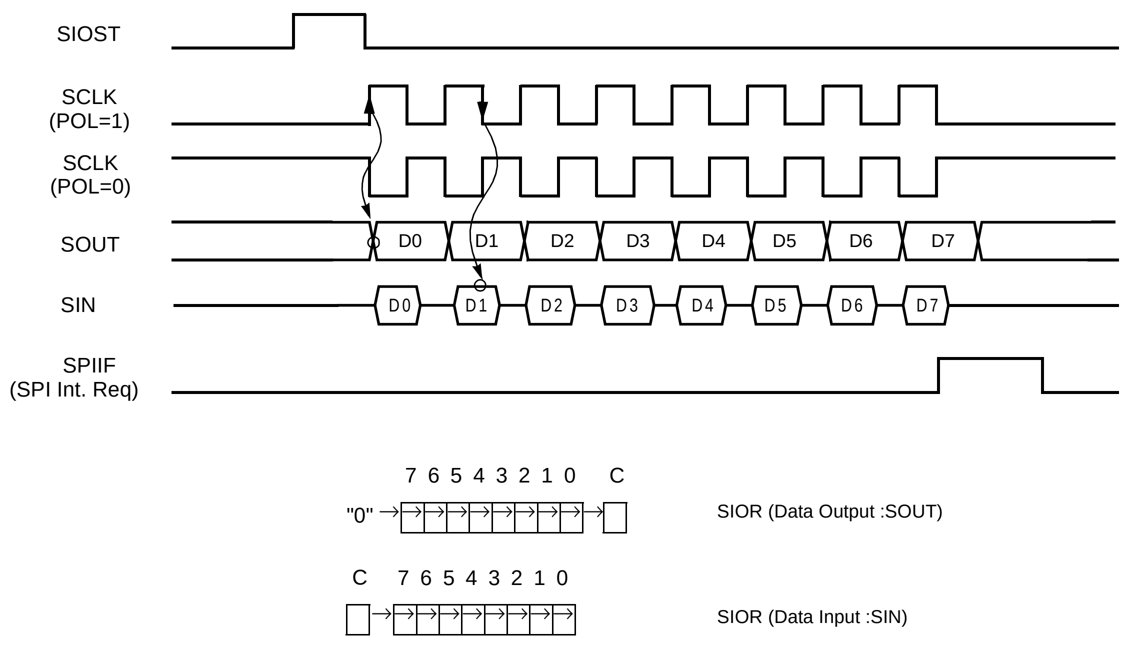


Figure 16-2 SPI Timing Diagram

17. Buzzer Output function

The buzzer driver consists of 6-bit binary counter, the buzzer register BUR and the clock selector. It generates square-wave which is very wide range frequency (480 Hz~250 KHz at fxin = 4 MHz) by user programmable counter.

Pin R03 is assigned for output port of Buzzer driver by setting the bit BUZO of R0FUNC to "1".

The 6-bit buzzer counter is cleared and start the counting by **writing signal** to the register BUR. It is increased from 00H until it matches 6-bit register BUR.

Also, it is cleared by **counter overflow** and count up to output the square wave pulse of duty 50%.

The bit 0 to 5 of BUR determines output frequency for buzzer driving. Frequency calculation is following as shown below.

$$f_{BUZ}(Hz) = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Ratio} \times (BUR + 1)}$$

The bits BUCK1, BUCK0 of BUR selects the source clock from prescaler output.

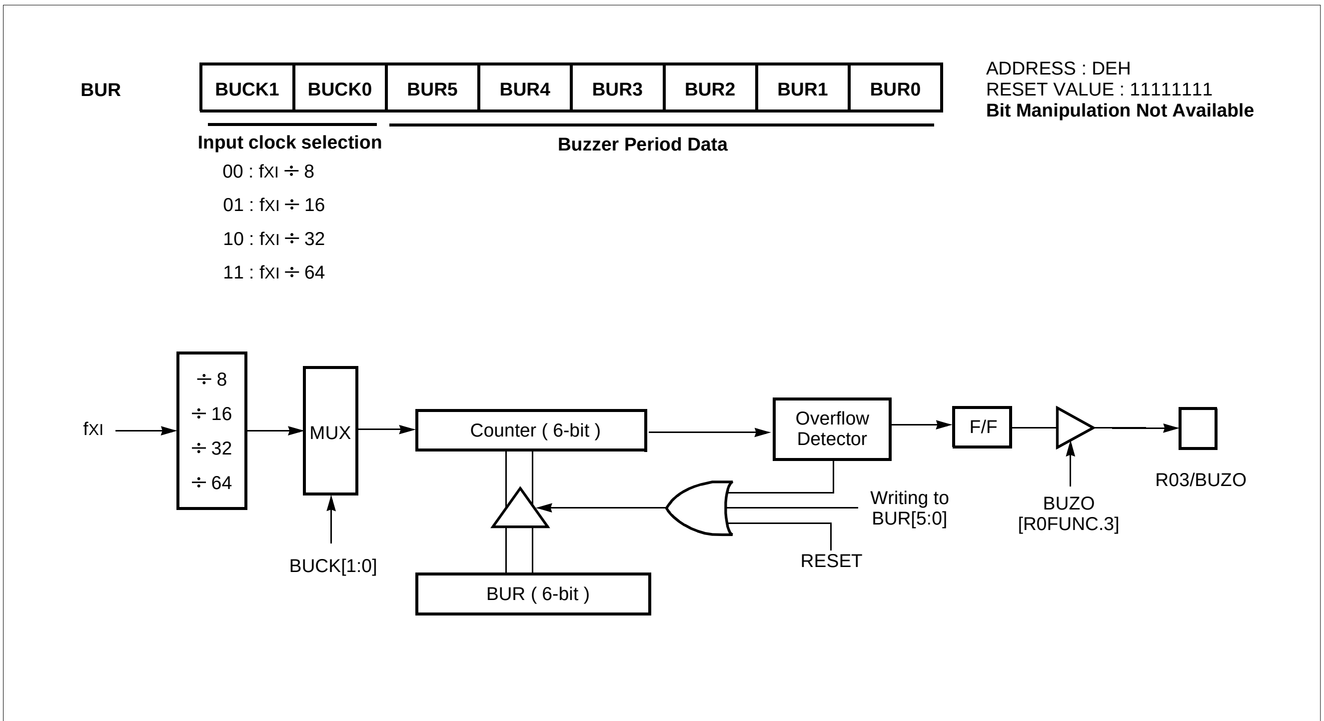


Figure 17-1 Buzzer Driver

18. ANALOG TO 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 twelve 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 A/D module has two registers which are the control register ADCM and A/D result register ADCR. The ADCM register, shown in Figure 18-2 , 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].

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 18-1 . The A/D status bit ADSF is set automatically when A/D conversion is completed, cleared when A/D conversion is in process. The conversion time takes maximum 20 uS (at fXI=4 MHz).

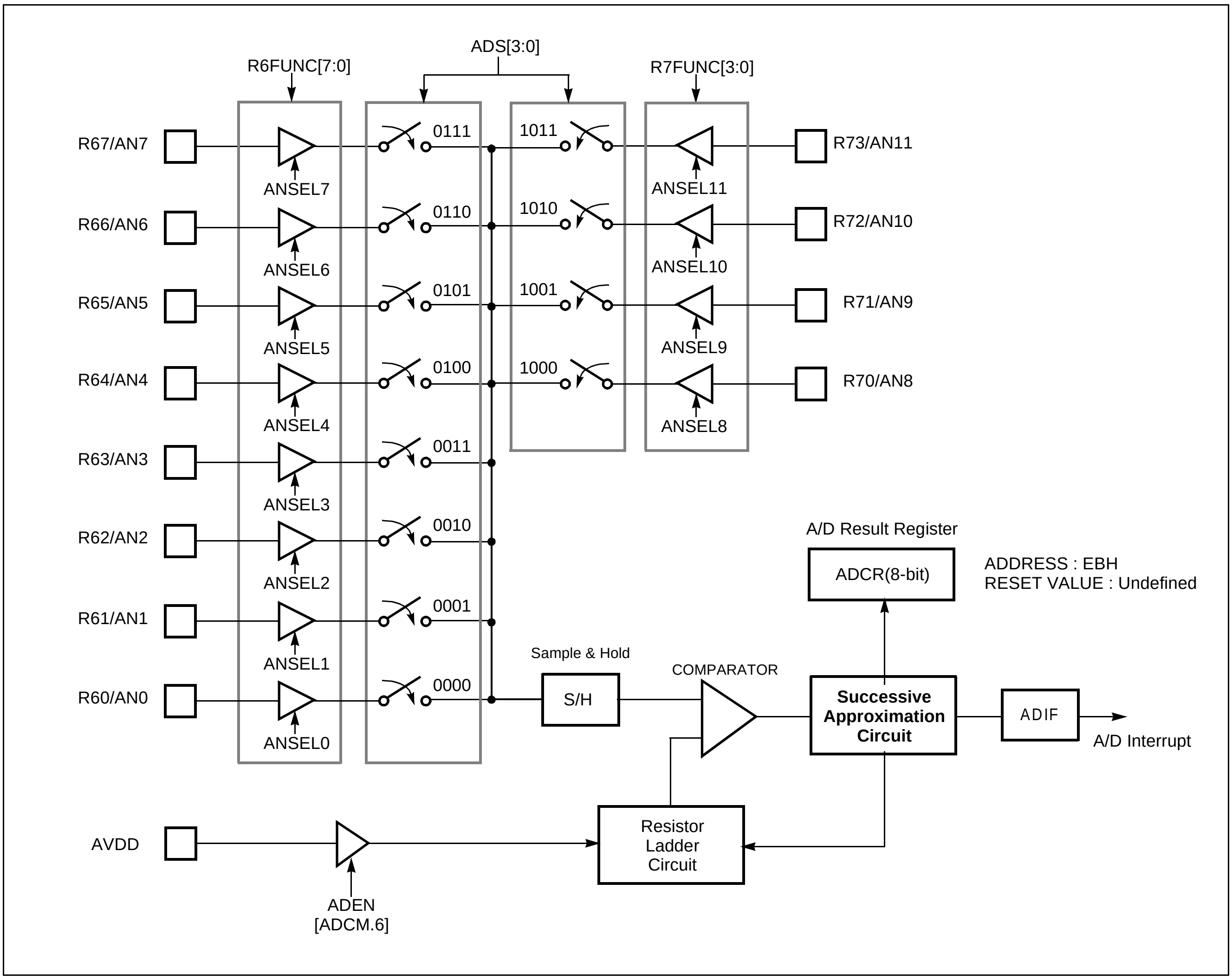


Figure 18-1 A/D Converter Block Diagram

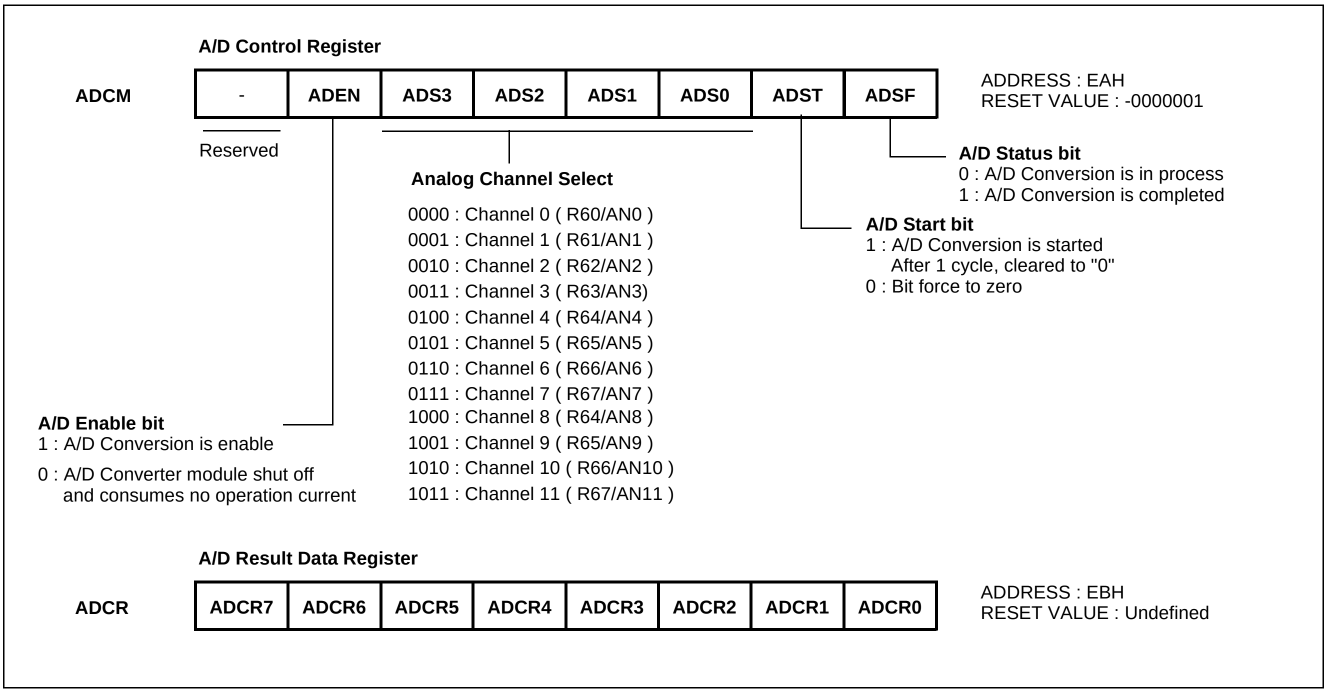


Figure 18-2 A/D Converter Registers

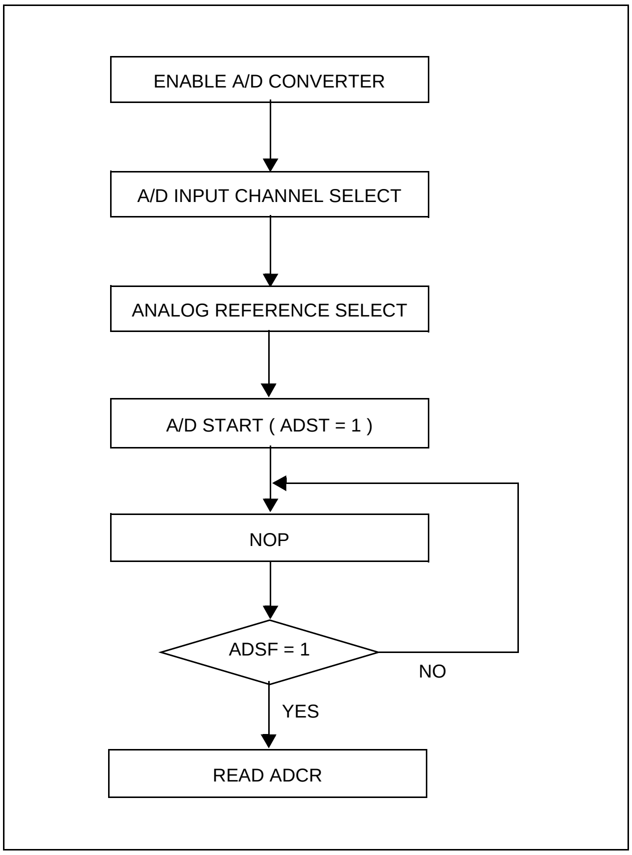


Figure 18-3 A/D Converter Operation Flow

A/D Converter Cautions

(1) Input range of AN11 to AN0

The input voltages 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 18-4 in order to reduce noise.

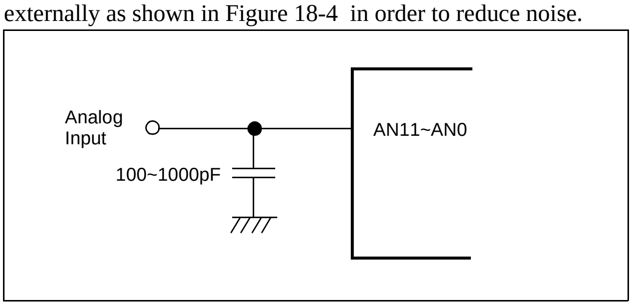


Figure 18-4 Analog Input Pin Connecting Capacitor

(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.

19. INTERRUPTS

The GMS81C2020 and GMS81C2120 interrupt circuits consist of Interrupt enable register (IENH, IENL), Interrupt request flags of IRQH, IRQL, Interrupt Edge Selection Register (IEDS), priority circuit and Master enable flag("I" flag of PSW). The configuration of interrupt circuit is shown in Figure and Interrupt priority is shown in Table 19-1 .

The External Interrupts INT0 and INT1 can each be transition-activated (1-to-0, 0-to-1 and both transiton).

The flags that actually generate these interrupts are bit INT0IF and INT1IF 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 and Timer 1 Interrupts are generated by T0IF and T1IF, which are set by a match in their respective timer/counter register. The AD converter Interrupt is generated by ADIF which is set by finishing the analog to digital conversion. The Watch dog timer Interrupt is generated by WDTIF which set by a match in Watch dog timer register (when the bit WDTON is set to "0"). The Basic Interval Timer Interrupt is generated by BITIF which is set by a overflowing of the Basic Interval Timer Register(BITR). The Serial Peripheral Interface (SPI) is generated by SPIIF which is set by communicating with other peripheral of microcontroller devices (by finishing the data transmission).

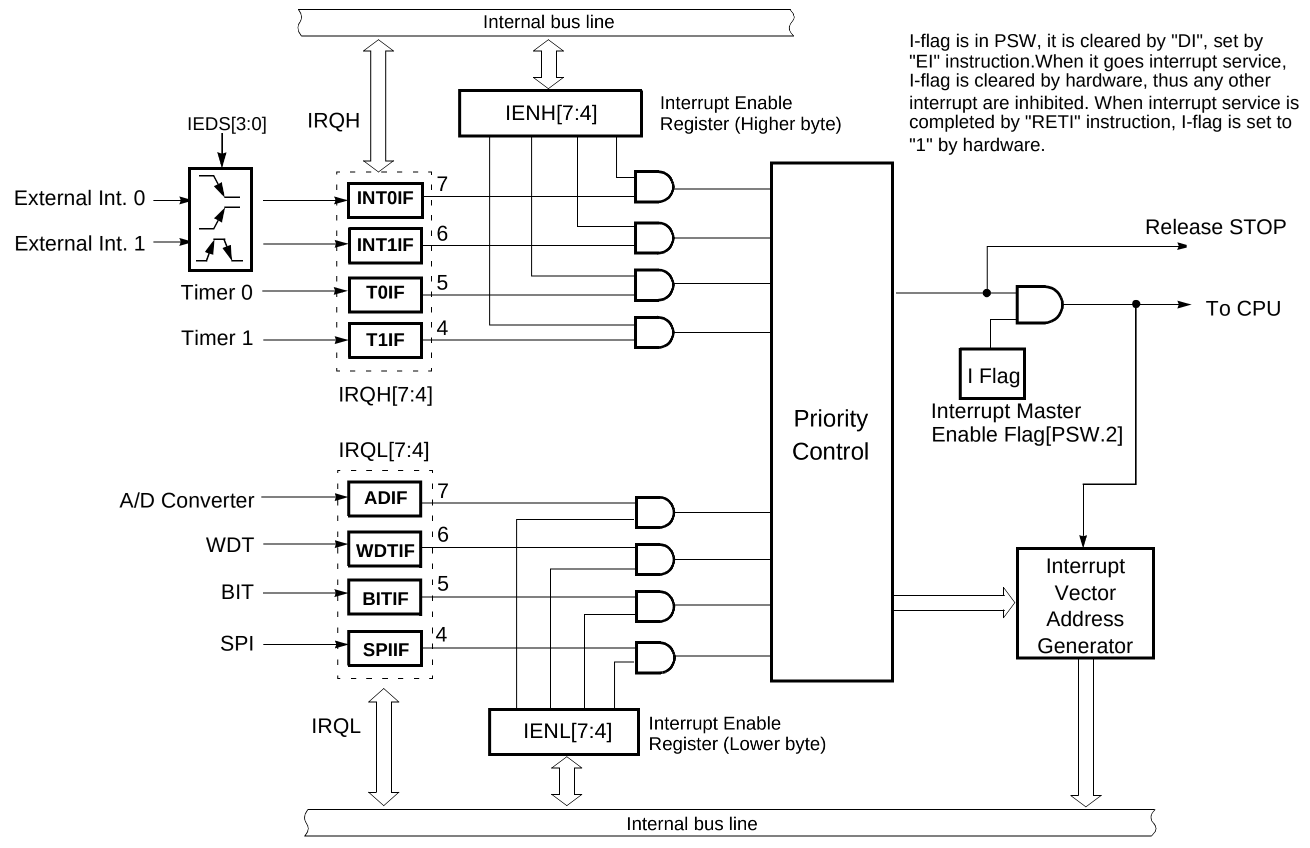


Figure 19-1 Block Diagram of Interrupt Function

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

Interrupt enable registers are shown in Figure 19-2 . These registers are composed of interrupt enable flags of each interrupt source, 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.

Reset/Interrupt	Symbol	Priority	Vector Addr.
Hardware Reset	RESET	-	$FFFE_H$
External Interrupt 0	INT0	1	$FFFA_H$
External Interrupt 1	INT1	2	$FFF8_H$
Timer 0	Timer 0	3	$FFF6_H$
Timer 1	Timer 1	4	$FFF4_H$
-	-	-	$FFF2_H$
-	-	-	$FFF0_H$
-	-	-	$FFEE_H$
-	-	-	$FFEC_H$
A/D Converter	A/D C	5	$FFEA_H$
Watch Dog Timer	WDT	6	$FFE8_H$
Basic Interval Timer	BIT	7	$FFE6_H$
Serial Interface	SPI	8	

Table 19-1 Interrupt Priority

Interrupt Enable Register High

IENH								
IENH	INT0E	INT1E	T0E	T1E	-	-	-	-

ADDRESS : E2H
RESET VALUE : 0000----

Interrupt Enable Register Low

IENL								
IENL	ADE	WDTE	BITE	SPIE	-	-	-	-

ADDRESS : E3H
RESET VALUE : 0000----

Enables or disables the interrupt individually
If flag is cleared, the interrupt is disabled.
0 : Disable
1 : Enable

Interrupt Request Register High

IRQH								
IRQH	INT0IF	INT1IF	T0IF	T1IF	-	-	-	-

ADDRESS : E4H
RESET VALUE : 0000----

Interrupt Request Register Low

IRQL								
IRQL	ADIF	WDTIF	BITIF	SPIIF	-	-	-	-

ADDRESS : E5H
RESET VALUE : 0000----

Shows the interrupt occurrence
0 : Not occurred
1 : Interrupt request is occurred

Figure 19-2 Interrupt Enable Registers and Interrupt Request Registers

When an interrupt is occured, the I-flag is cleared and disable any further interrupt, the return address and PSW are pushed into the stack and the PC is vectored to. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt request flag bits.

The interrupt request flag bit(s) must be cleared by software before re-enabling interrupts to avoid recursive interrupts. The Interrupt Request flags are able to be read and written.

19.1 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 8 f_{OSC} (2 μs at fXI=4MHz) 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.

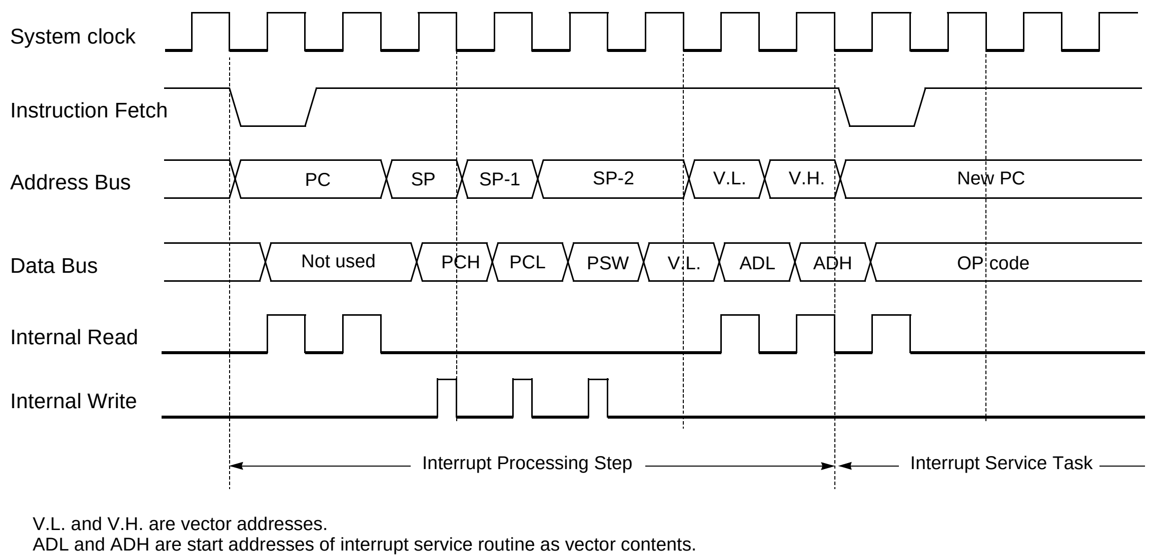
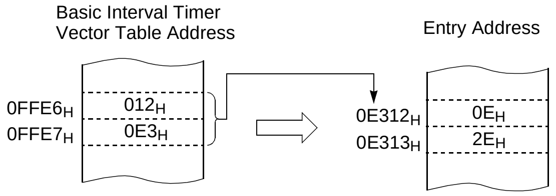


Figure 19-3 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



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.

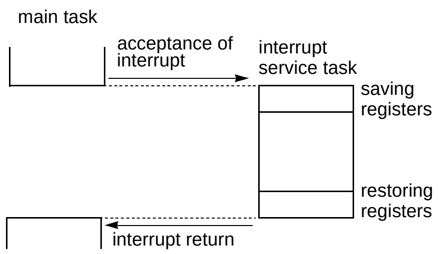
Example: Register save using push and pop instructions

```
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;



19.2 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 19-4 .

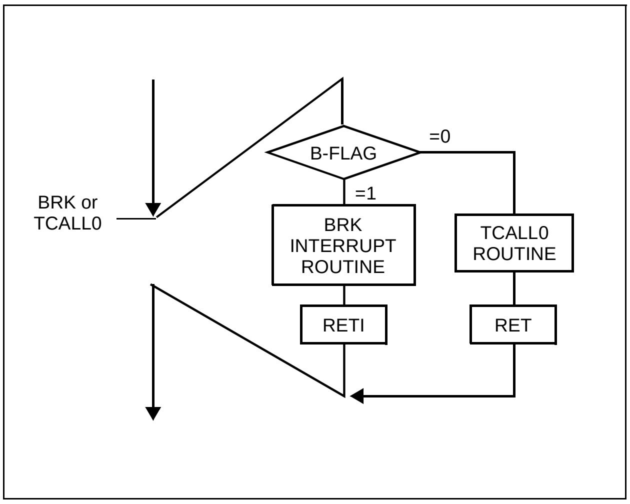
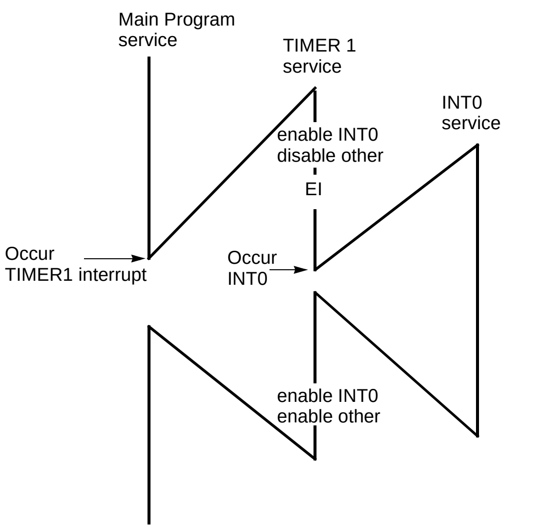


Figure 19-4 Execution of BRK/TCALL0

19.3 Multi Interrupt

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

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



In this example, the INT0 interrupt can be serviced without any pending, even TIMER1 is in progress.
Because of re-setting the interrupt enable registers IENH,IENL and master enable "EI" in the TIMER1 routine.

Figure 19-5 Execution of Multi Interrupt

Example: Even though 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,#0FFH   ;Enable all interrupts
        LDM    IENL,#0F0H
        POP    Y
        POP    X
        POP    A
        RETI
```

19.4 External Interrupt

The external interrupt on INT0 and INT1 pins are edge triggered depending on the edge selection register IEDS (address $0E6_H$) as shown in Figure 19-6 .

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

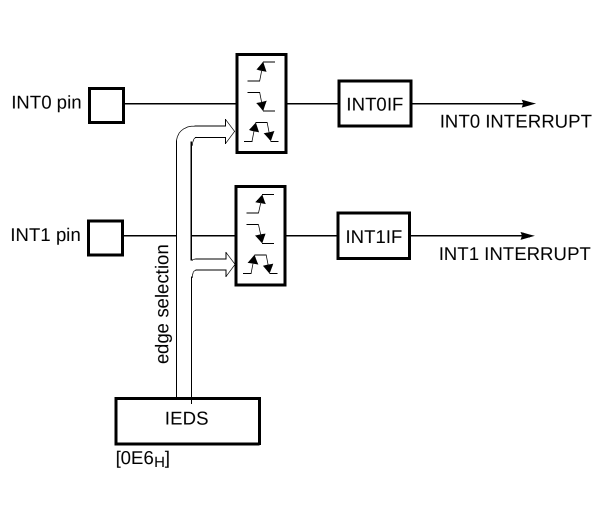
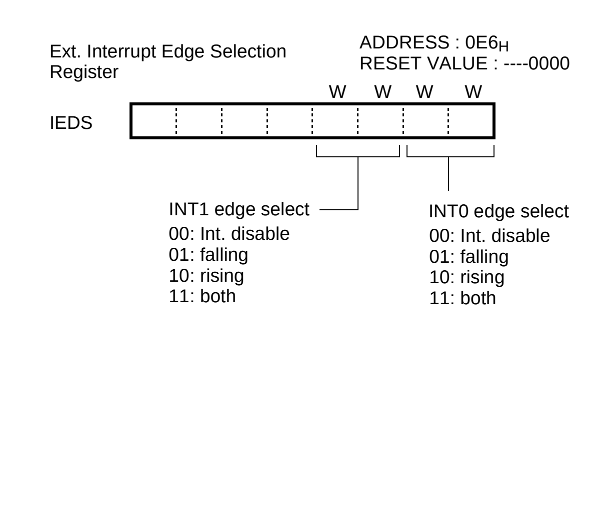


Figure 19-6 External Interrupt Block Diagram



Example: To use as an INT0, INT1

```
        :
        :
;**** Set port as an input port R00,R01
        LDM    R0IO,#1111_1100B
        ;
;**** Set port as an interrupt port
        LDM    R0FUNC,#03H
        ;
;**** Set Falling-edge Detection
        LDM    IEDS,#0000_0101B
        :
        :
        :
```

Response Time

The INT0 and INT1 edge are latched into INT0IF and INT3IF at every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The DIV itself takes twelve cycles. Thus, a minimum of twelve complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine.

shows interrupt response timings.

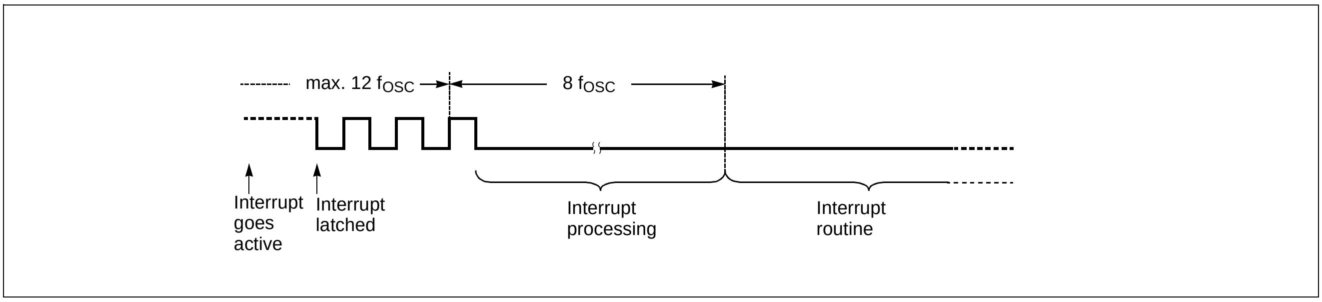


Figure 19-7 Interrupt Response Timing Diagram

20. WATCHDOG TIMER

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 maching 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.

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

Clock Control Register

CKCTLR	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0
	-	0	X	1	X	X	X	X

ADDRESS : ECH
RESET VALUE : -0010111
Bit Manipulation Not Available

Watchdog Timer Register

WDTR	WDTCL	7-bit Watchdog Counter Register

ADDRESS : EDH
RESET VALUE : 01111111
Bit Manipulation Not Available

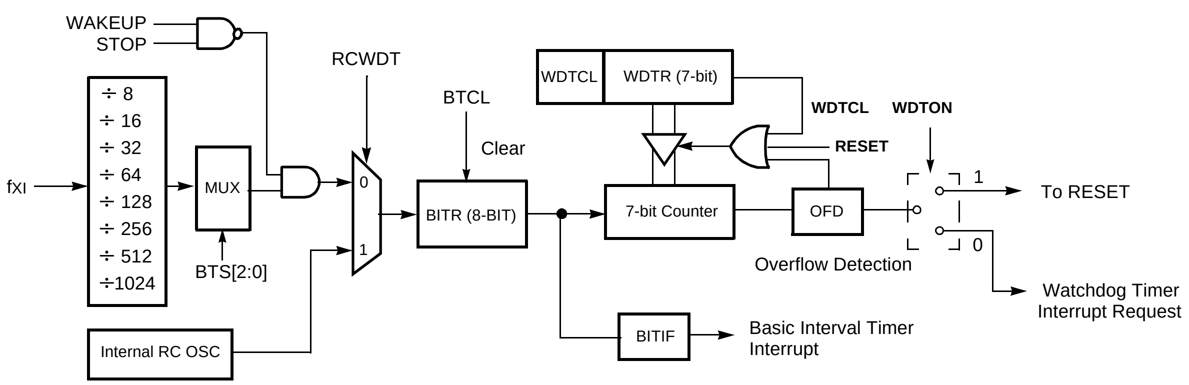


Figure 20-1 Block Diagram of Watchdog Timer

21. Power Saving Mode

For applications where power consumption is a critical factor, device provides four kinds of power saving functions, STOP mode, Subactive mode and Wake-up Timer mode(Standby mode, Watch mode).

Table 21-1 shows the status of each Power Saving Mode.

Peripheral	STOP Mode	Subactive Mode	Wake-up Timer Mode	
			Standby 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 21-1 Power Saving Mode

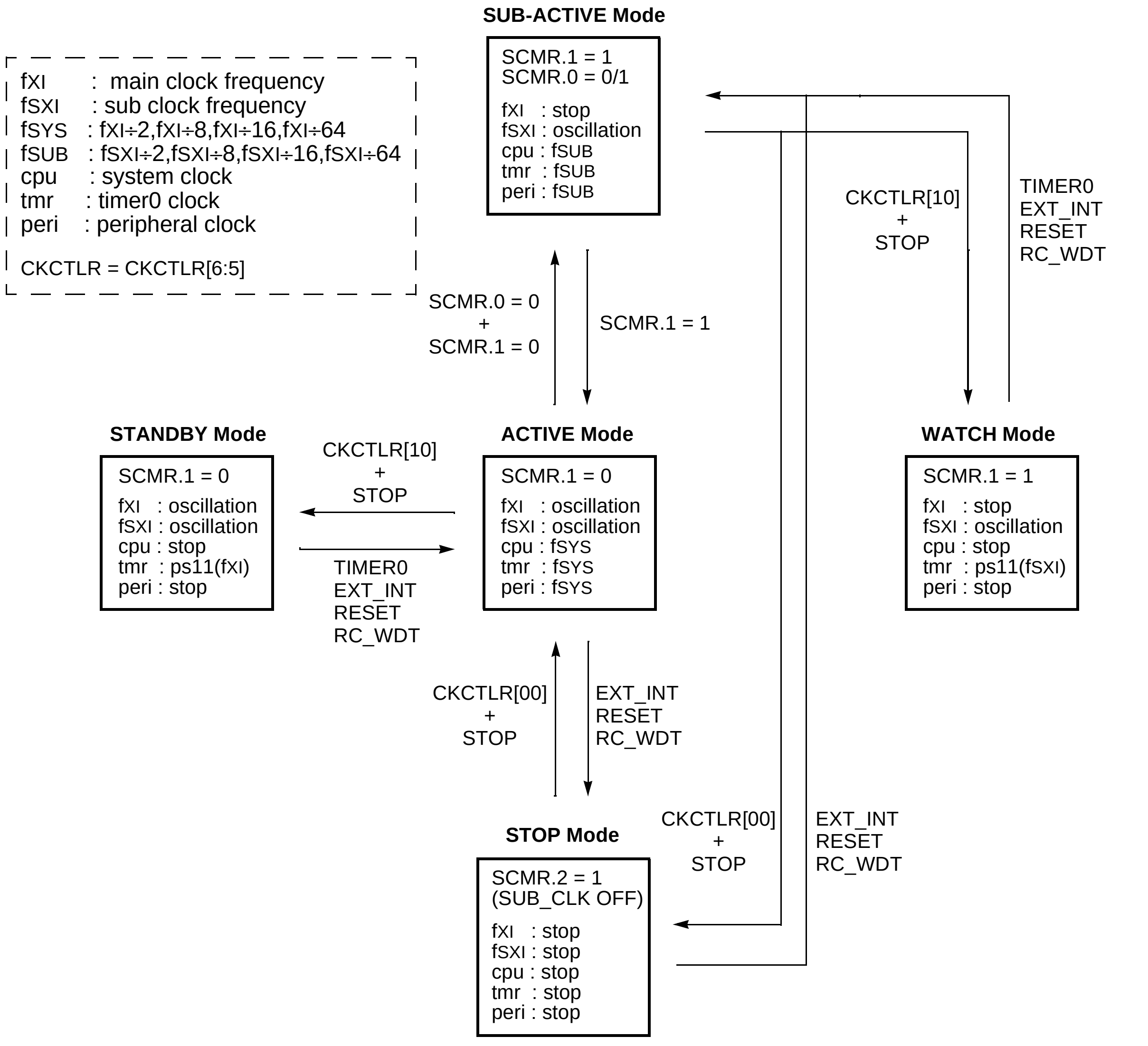
The power saving function is activated by execution of STOP instruction and by execution of STOP instruction after setting the corresponding status (WAKEUP) of CKCTLR.

we shows the release sources from each Power Saving Mode

Release Source	STOP Mode	Subactive Mode	Wake-up Timer Mode	
			Standby Mode	Watch Mode
RESET	O	O	O	O
RCWDT	O	O	O	O
EXT.INT	O	O	O	O
EXT.INT1				
Timer0	X	X	O	O

Table 21-2 Release Sources from Power Saving Mode

21.1 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 ÷ 210 : fXI ÷16
01 : fXI ÷ 811 : 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

21.2 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 opereation. If this register is set by bit manipulation instrunction, 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 21-1)

When exit from Stop mode by external interrupt, enough oscillation stabilization time is required to normal operation. Figure 21-4 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 21-5 .

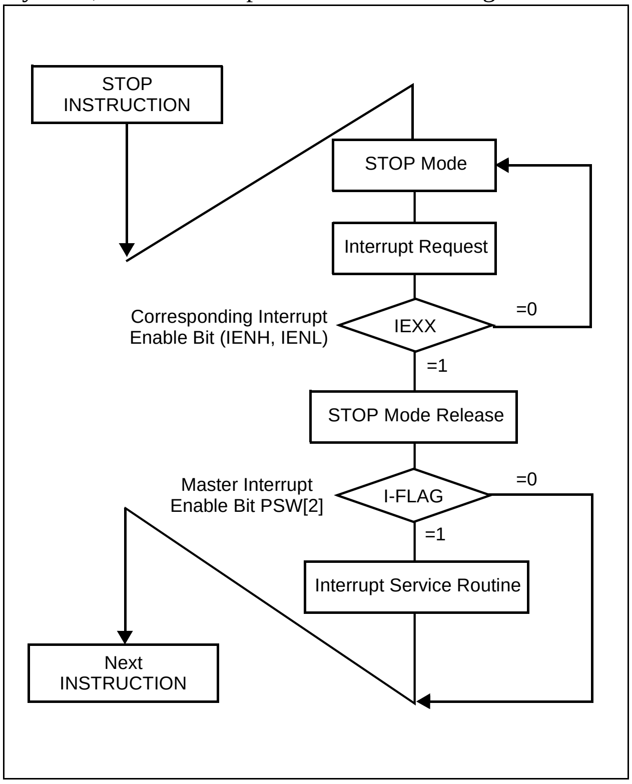


Figure 21-1 STOP Releasing Flow by Interrupts

Minimizing Current Consumption

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

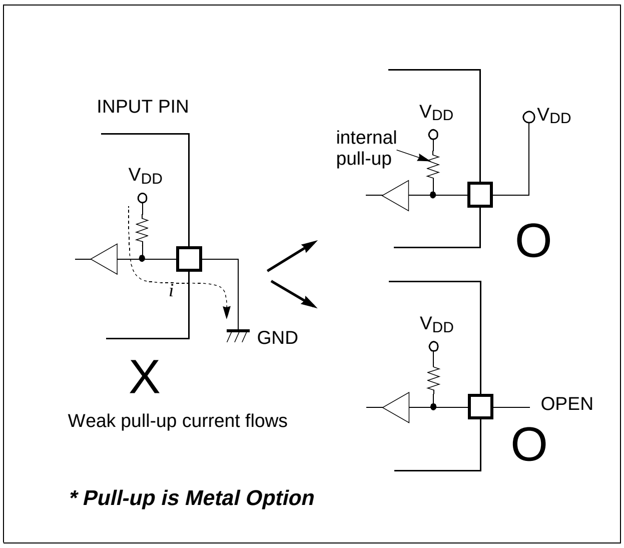
Note: *In the STOP operation, the power dissipation associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the STOP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

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

First 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.



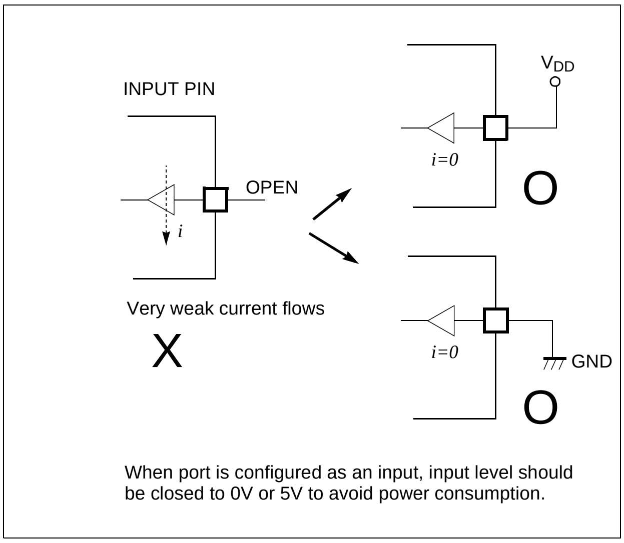
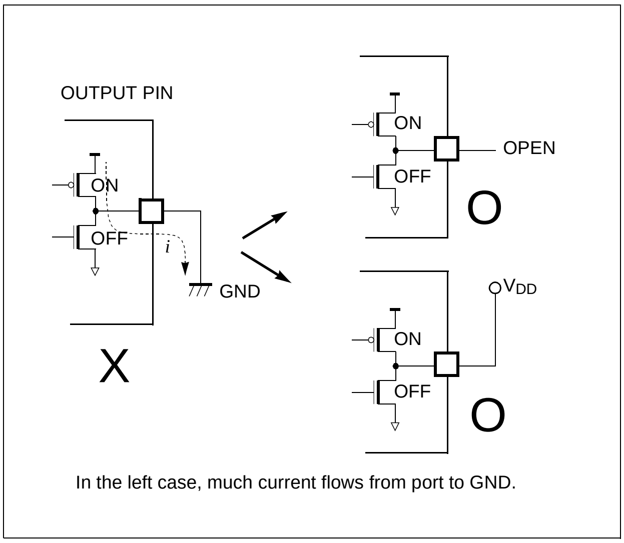


Figure 21-2 Application Example of Unused Input Port



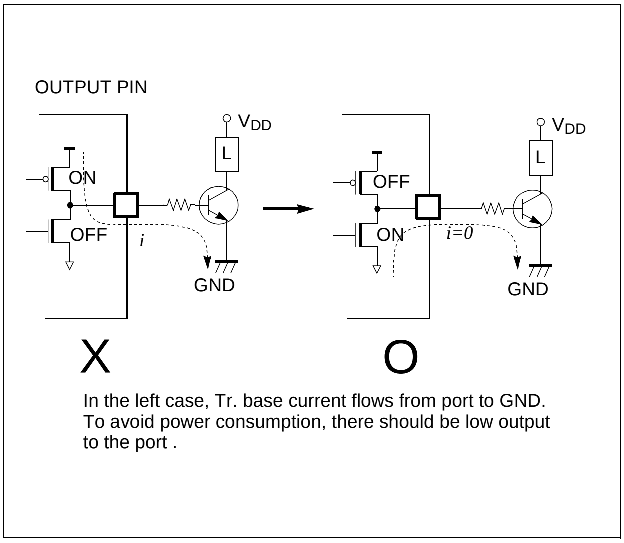


Figure 21-3 Application Example of Unused Input Port

Minimizing Current Consumption in Stop Mode

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. Weak pull-ups on port pins should be turned off, if possible. All inputs should be either as VSS or at VDD (or as close to rail as possible).

An intermediate voltage on an input pin causes the input buffer to draw a significant amount of current.

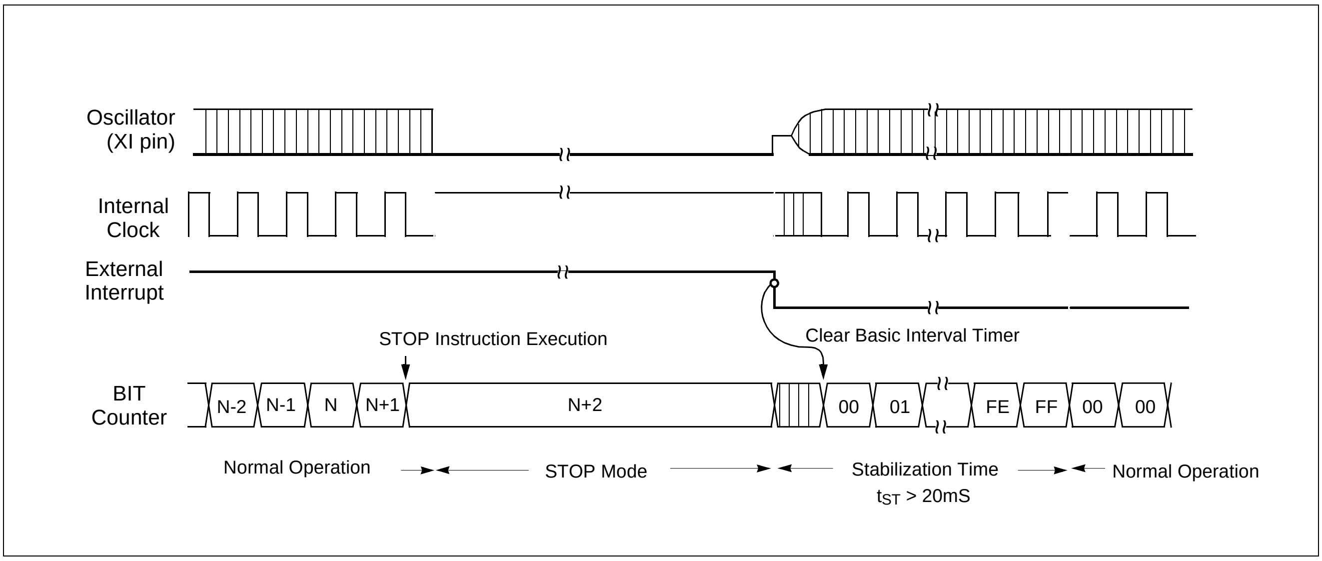


Figure 21-4 Timing of STOP Mode Release by External Interrupt

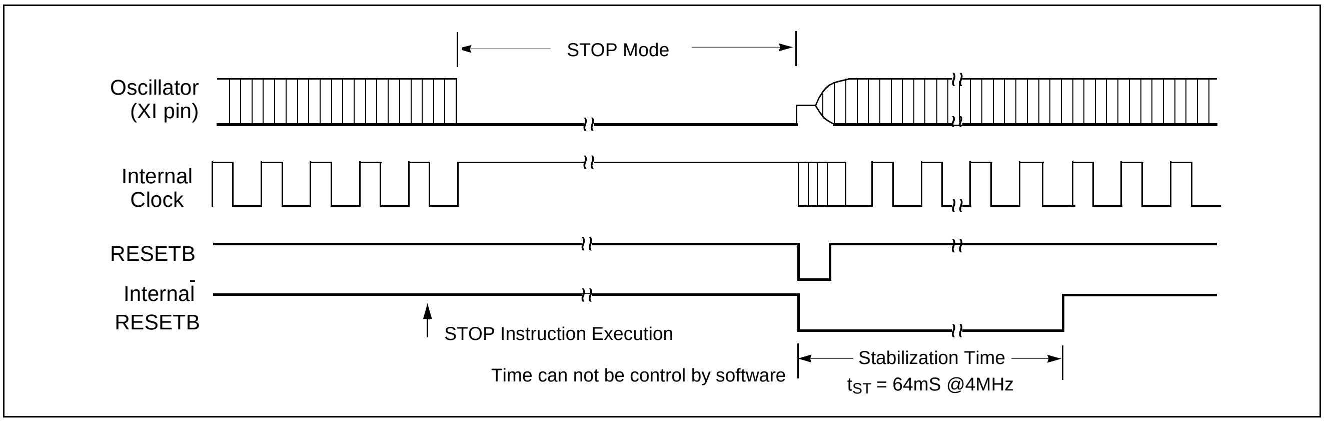


Figure 21-5 Timing of STOP Mode Release by RESET

21.3 Wake-up Timer Mode

In the Wake-up Timer mode, the on-chip oscillator is not stopped. Except the Prescaler(only 2048 devided 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 opereation. If this register is set by bit manipulation instrunction, 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 devided 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 21-1)

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 21-6 .

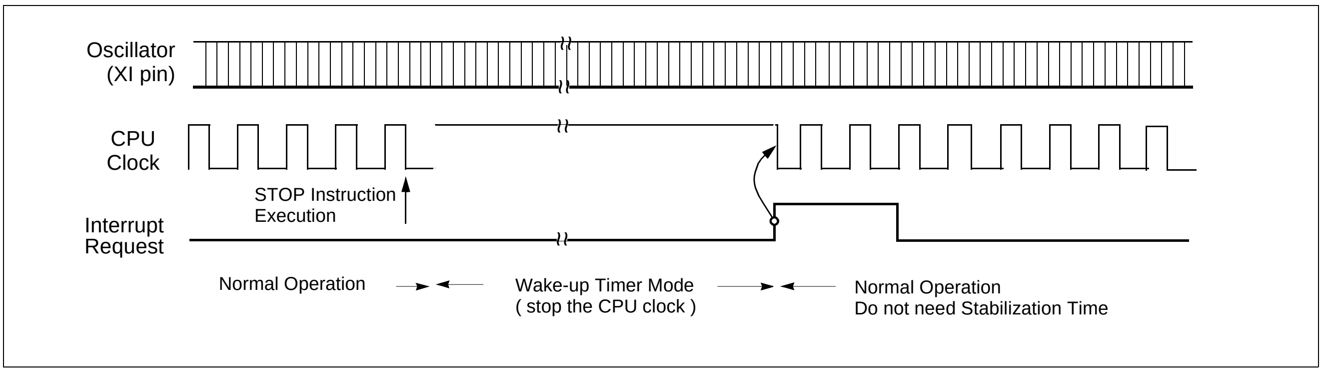


Figure 21-6 Wake-up Timer Mode Releasing by External Interrupt or Timer0 Interrupt

21.4 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 opereation. 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,#0**01**0_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 21-7) 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 21-8)

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 21-1)

When exit from Internal RC-Oscillated Watchdog Timer mode by external interrupt, the oscillation stabilization time is required to normal operation. Figure 21-7 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 21-8 .

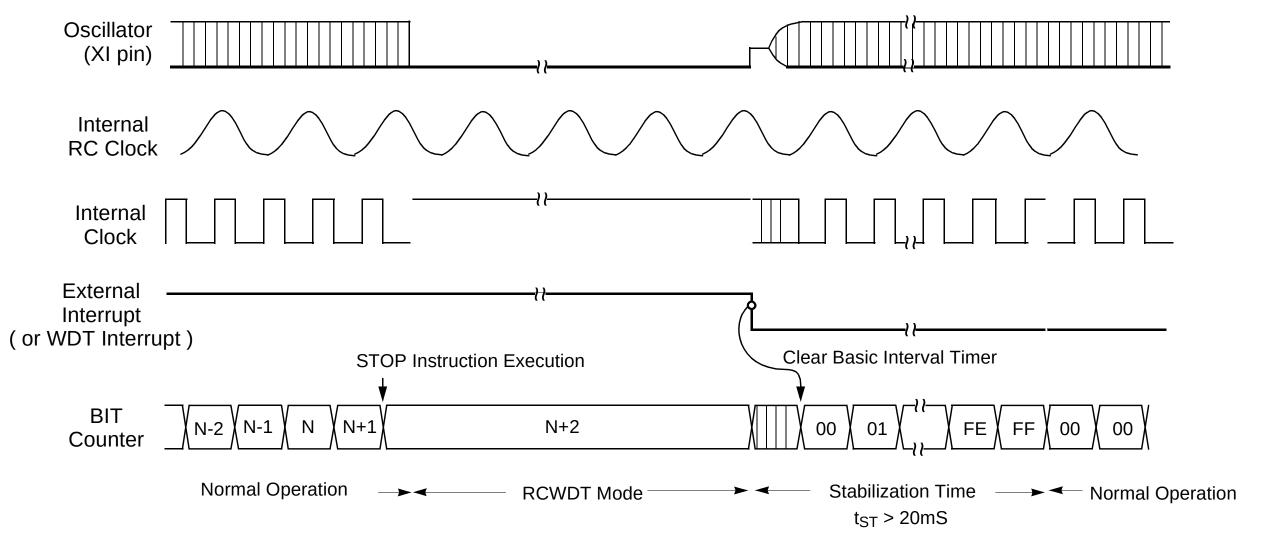


Figure 21-7 Internal RCWDT Mode Releasing by External Interrupt or WDT Interrupt

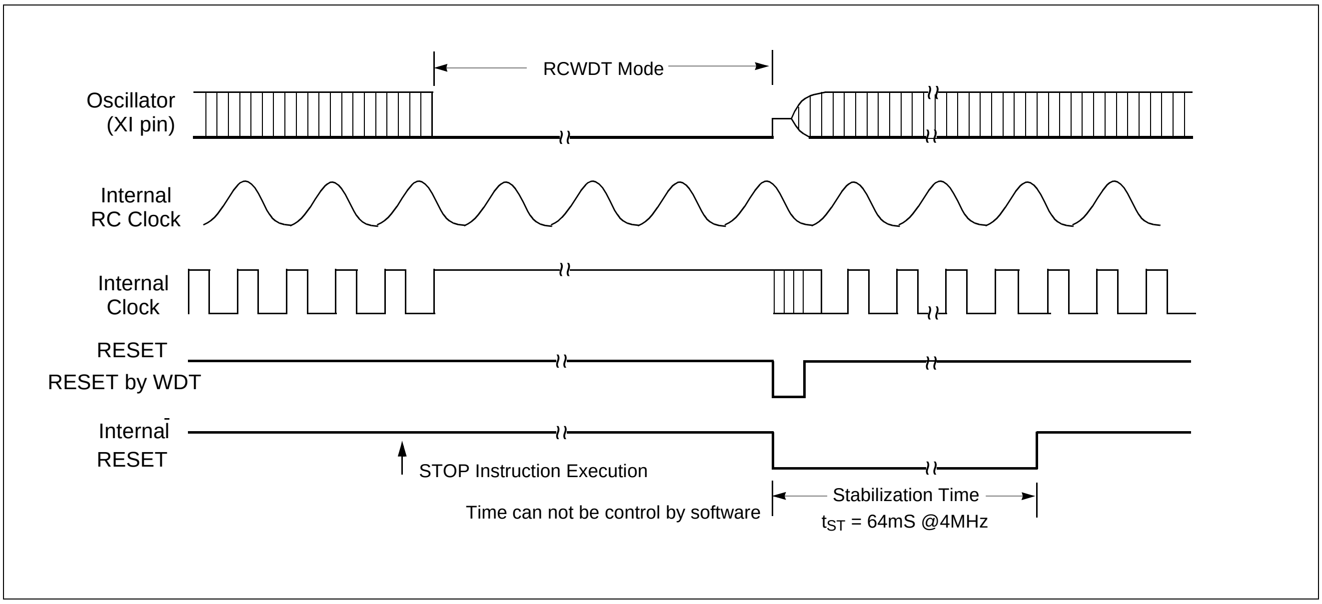


Figure 21-8 Internal RCWDT Mode Releasing by RESET

22. RESET

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, while the oscillator running. After reset, 64ms (at 4 MHz) add with 7 oscillator periods are required to start execution as shown in Figure 26-2 .

Internal RAM is not affected by reset. When VDD is turned on, the RAM content is indeterminate. Therefore, this RAM should be initialized before reading or testing it.

Initial state of each register is shown as Table 11-3 .

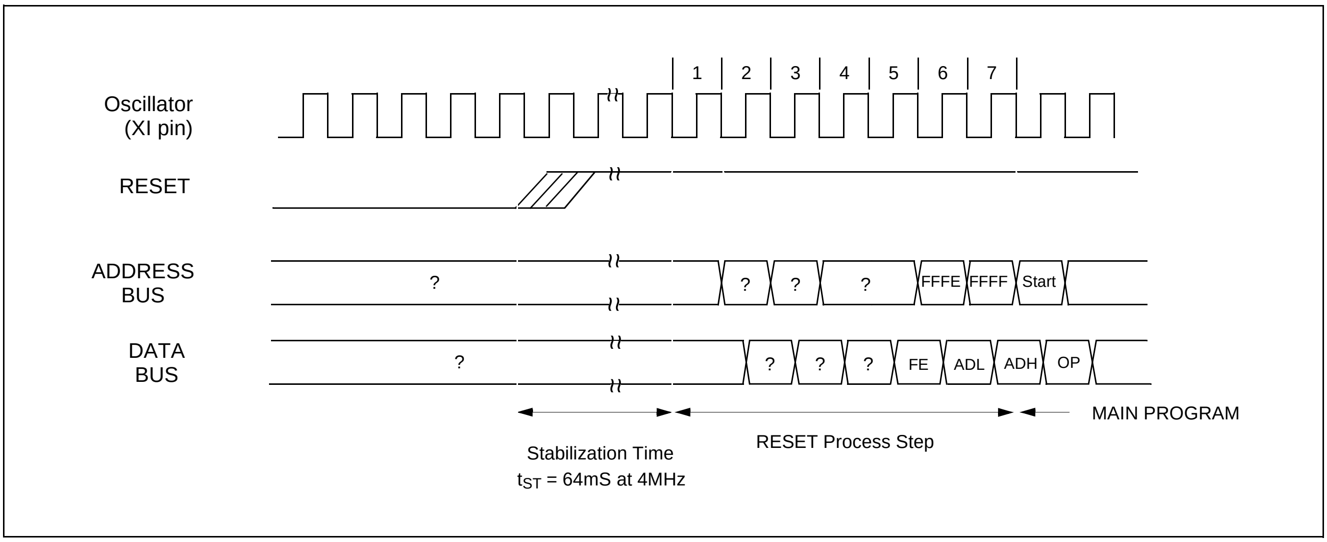


Figure 22-1 Timing Diagram after RESET

23. POWER FAIL PROCESSOR

The GMS81C2020 and GMS81C2120 has an on-chip power fail detection circuitry to immunize against power noise. A configuration register, PFDR, can enable (if clear/ programmed) or disable (if set) the Power-fail Detect circuitry. If VDD falls below 2.4~3.0V range for longer than 50 nS, the Power fail situation may reset MCU according to PFDM bit of PFDR.

As below PFDR register is not implemented on the in-circuit emulator, user can not experiment with it. Therefore, after final development of user program, this function may be experimented.

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

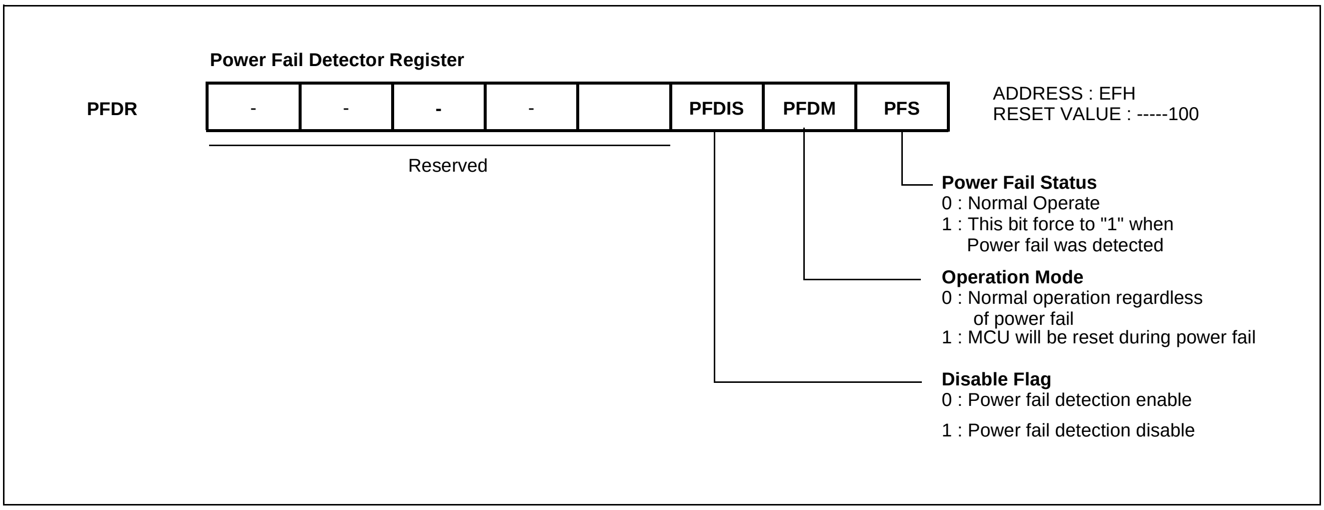


Figure 23-1 Power Fail Detector Register

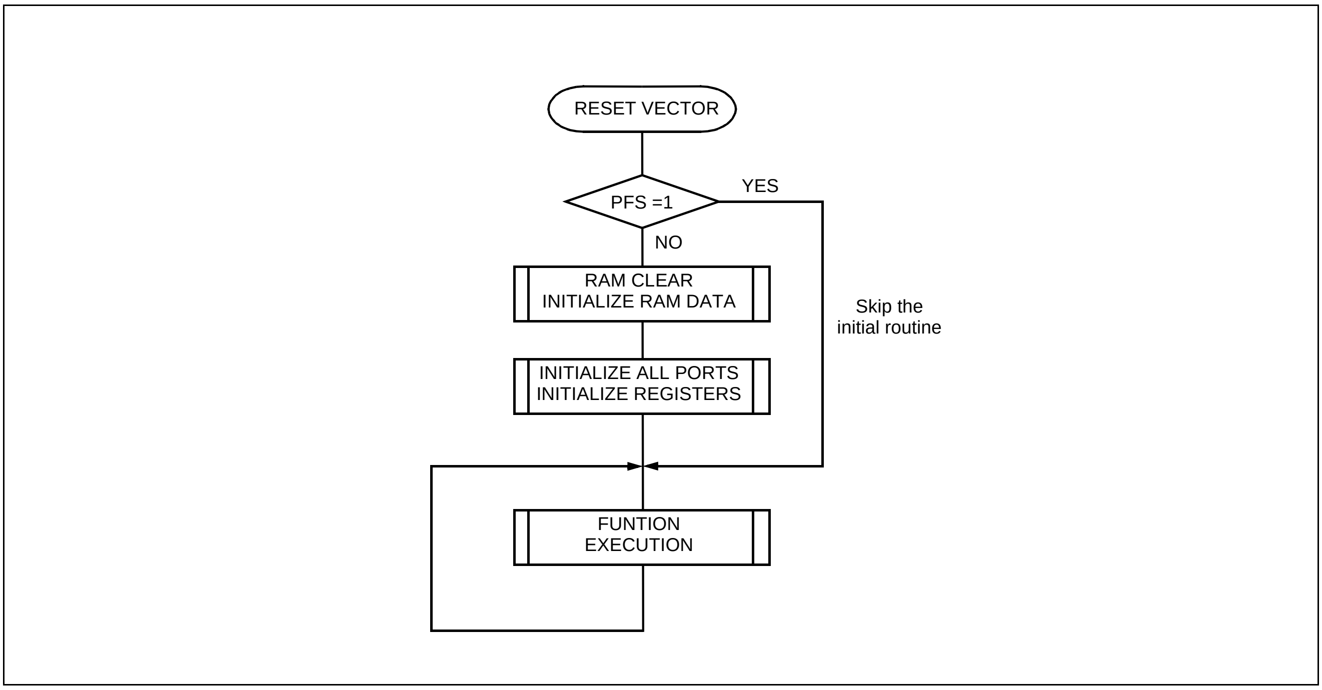


Figure 23-2 Example S/W of RESET by Power fail

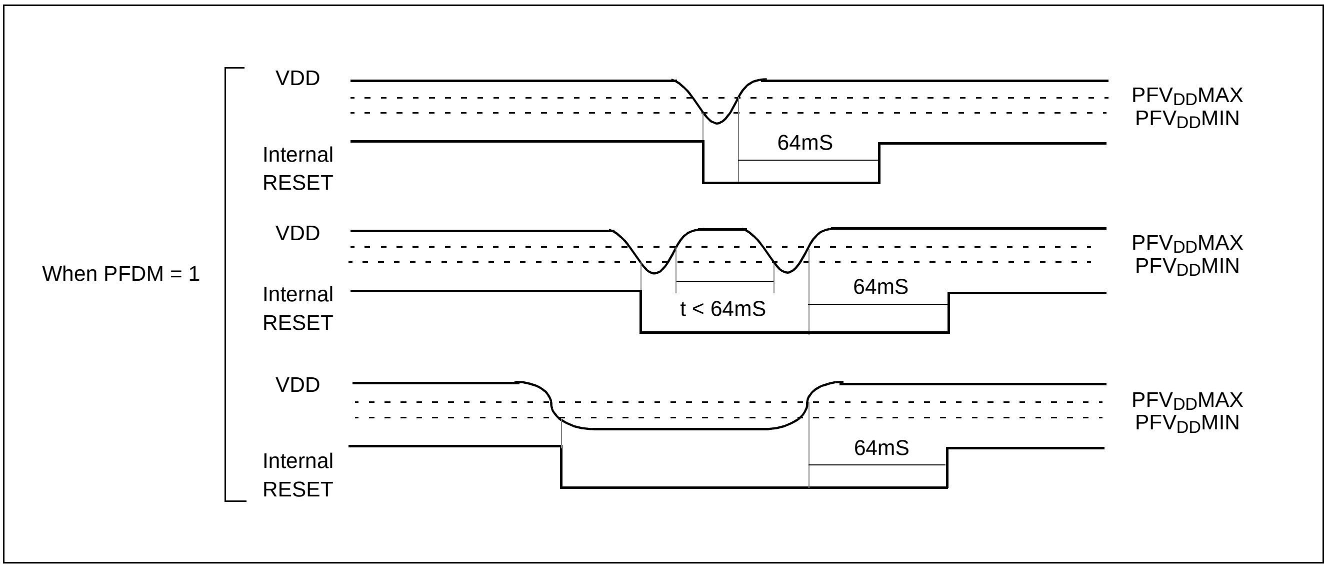


Figure 23-3 Power Fail Processor Situations

24. OTP PROGRAMMING

24.1 DEVICE CONFIGURATION AREA

The Device Configuration Area can be programmed or left unprogrammed to select device configuration such as security bit.
sixteen memory locations ($7030_H \sim 703F_H$) 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.

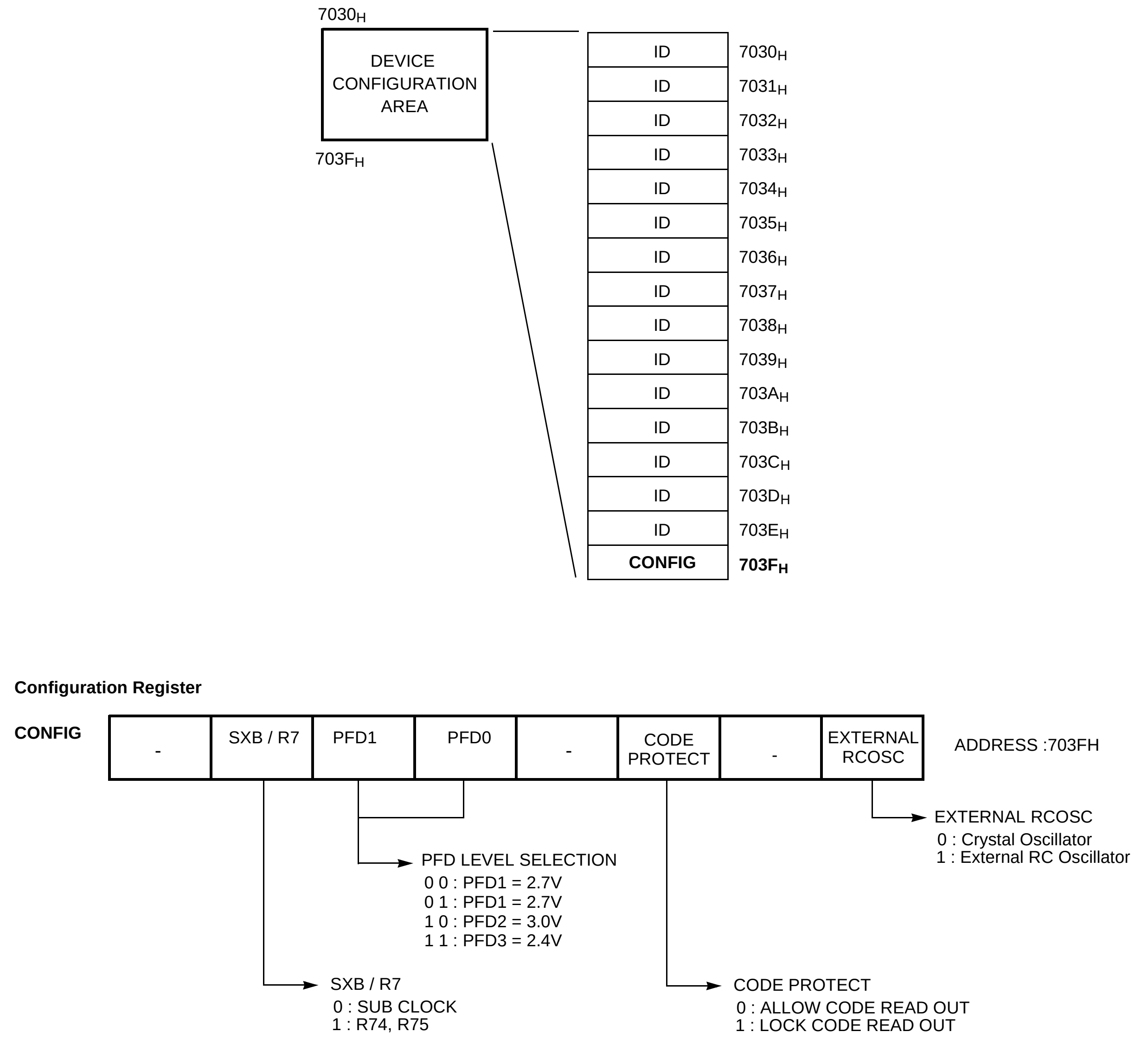


Figure 24-1 Device Configuration Area

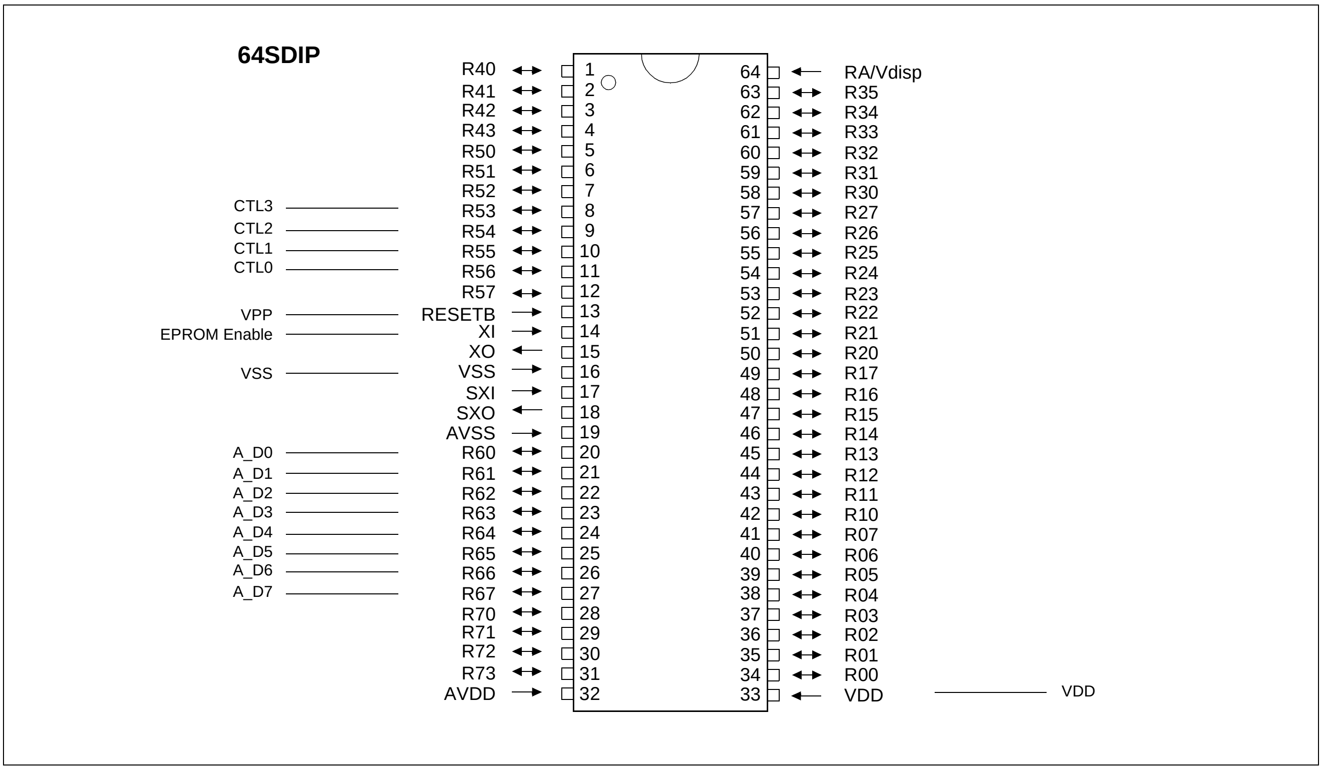


Figure 24-2 Pin Assignmen (64SDIP)t

Pin No.	User Mode Pin Name	EPROM MODE Pin Name	Description			
8	R53	CTL3	Read/Write Control			**P_Vb**
9	R54	CTL2	Address/Data Control			**D_Ab**
10	R55	CTL1	Write 8Bytes Control			**PGM8**
11	R56	CTL0	Write 4Bytes Control			**PGM4**
13	RESETB	VPP	Programming Power (0V, 12.75V)			
14	XI	EPROM Enable	High Active, Latch Address in falling edge			
15	XO	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 24-1 Pin Description in EPROM Mode (GMS81C2020)

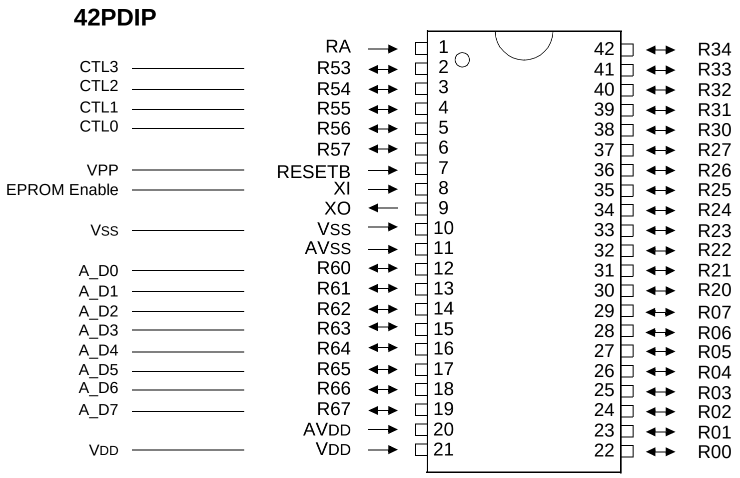


Figure 24-3 Pin Assignmen (42SDIP)t

Pin No.	User Mode	EPROM MODE				
	Pin Name	Pin Name	Description			
2	R53	CTL3	Read/Write Control			**P_Vb**
3	R54	CTL2	Address/Data Control			**D_Ab**
4	R55	CTL1	Write 8Bytes Control			**PGM8**
5	R56	CTL0	Write 4Bytes Control			**PGM4**
7	RESETB	VPP	Programming Power (0V, 12.75V)			
8	XI	EPROM Enable	High Active, Latch Address in falling edge			
9	XO	NC	No connection			
10	VSS	VSS	Connect to VSS (0V)			
12	R60	A_D0	Address Input Data Input/Output	A8	A0	D0
13	R61	A_D1		A9	A1	D1
14	R62	A_D2		A10	A2	D2
15	R63	A_D3		A11	A3	D3
16	R64	A_D4	Address Input Data Input/Output	A12	A4	D4
17	R65	A_D5		A13	A5	D5
18	R66	A_D6		A14	A6	D6
19	R67	A_D7		A15	A7	D7
21	VDD	VDD	Connect to VDD (6.0V)			

Table 24-2 Pin Description in EPROM Mode (GMS81C2120)

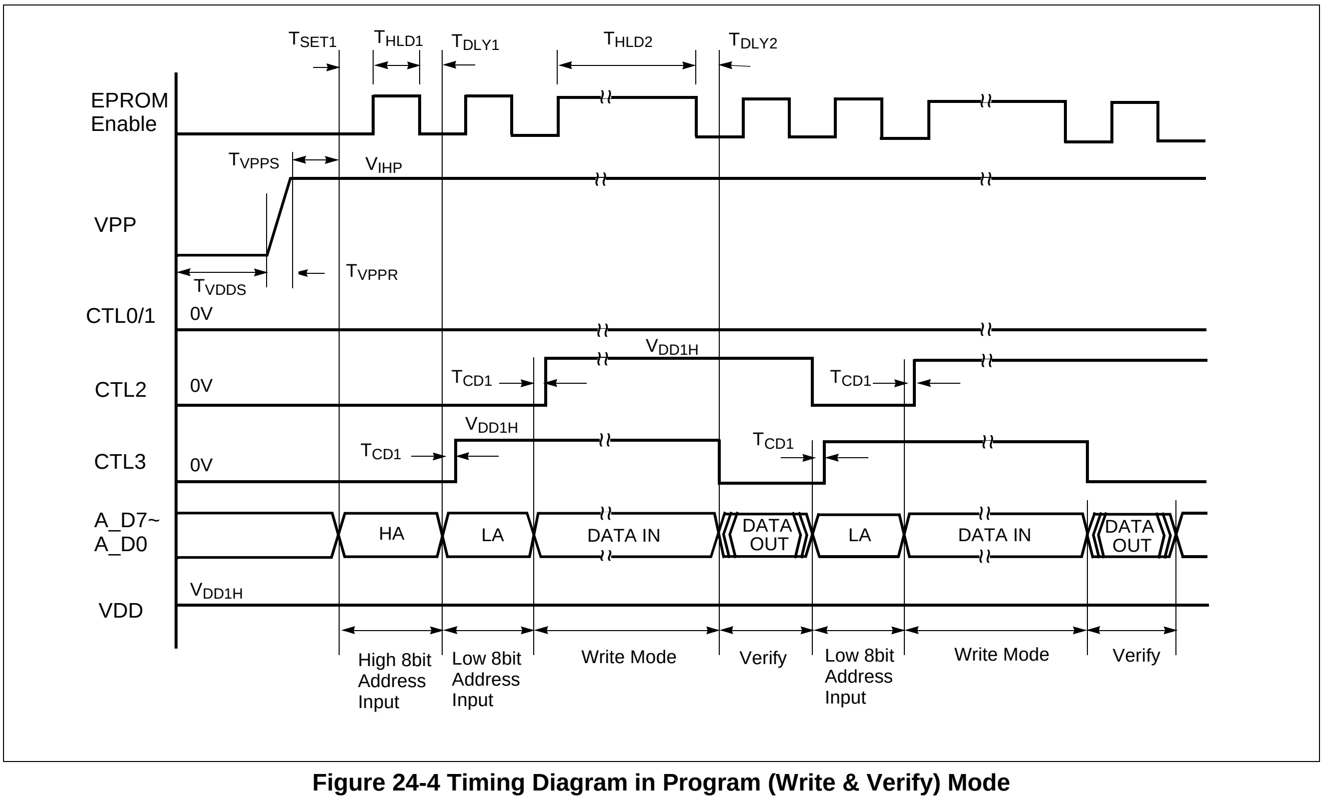


Figure 24-4 Timing Diagram in Program (Write & Verify) Mode

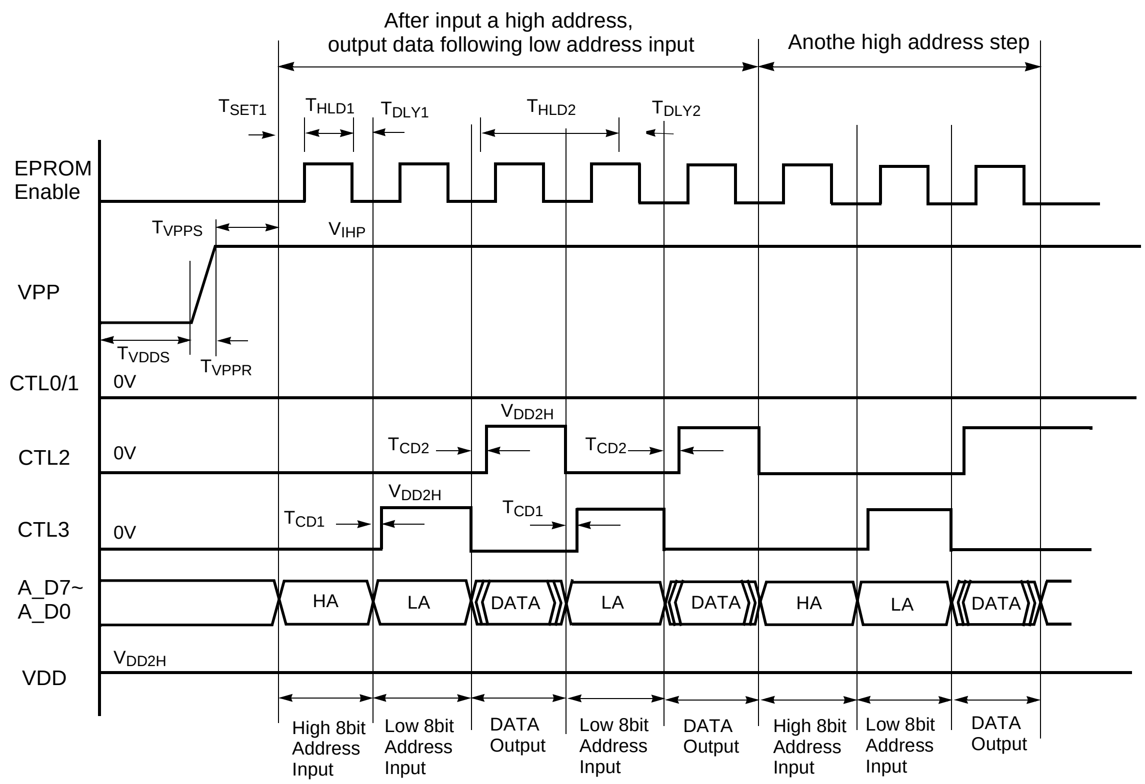


Figure 24-5 Timing Diagram in READ Mode

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 24-3 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 24-3 AC/DC Requirements for Program/Read Mode

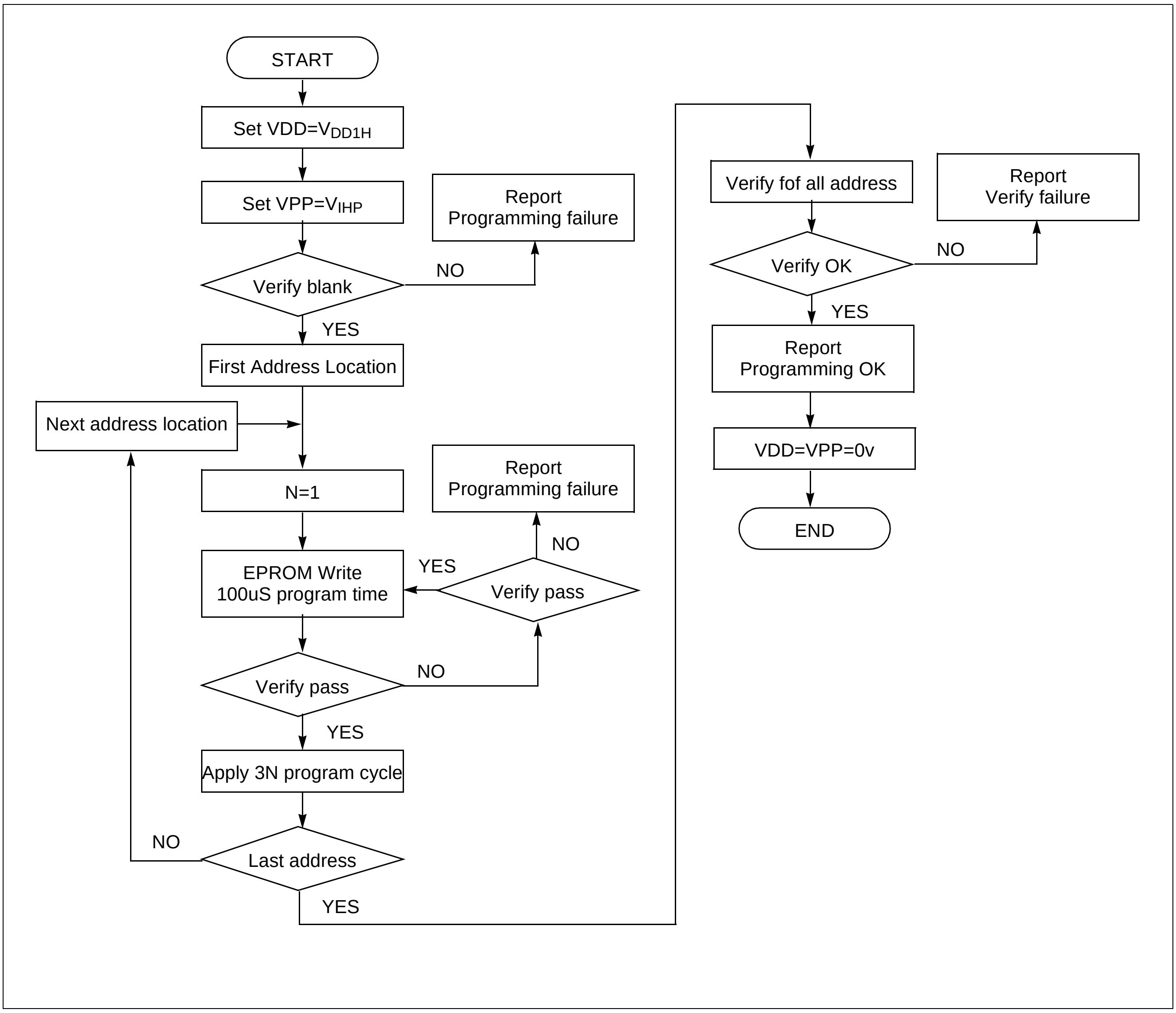
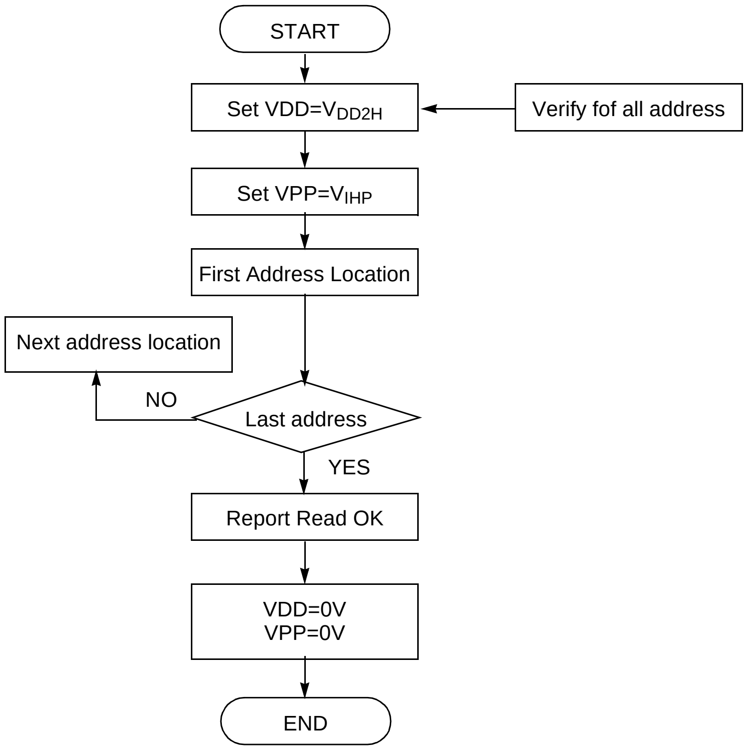


Figure 24-6 Programming Flow Chart


START
Set VDD=VDD2H
Verify fof all address
Set VPP=VIHP
First Address Location
Next address location
NO
Last address
YES
Report Read OK
VDD=0V
VPP=0V
END

GMS81C2 Series [GMS81C2020/12] Option List

Package

☐ 64SDIP	☐ 64LQFP
☐ 64MQFP	☐ 64TQFP

Date of Order	1999 / 2000. . .	
Customer		
Department		
Name		
ROM Code Name		
Check sum		
ROM Size	☐ 20KBytes	☐ 12KBytes

RA / Vdisp

☐ RA Without pull-down resistance
☐ Vdisp

*Note : In the I/O options list, *you must select Vdisp* even if only one pin is selected with pull-down resistance.

ROM Code Option List : 703FH

Bit7		Bit6		Bit5		Bit4		Bit3		Bit2		Bit1		Bit0	
-		SXB / R7		PFD1		PFD0		-		-		LOW VOLTAGE		RCOSC	
0 ✓	1	0	1	0	1	0	1	0 ✓	1	0 ✓	1	0	1	0	1

** Refer to Device Configuration Area*

I/O Option [VFD Driving Port]

Bit	I/O	I/O Option On	I/O Option Off
R00/INT0	I/O		
R01/INT1	I/O		
R02/EC0	I/O		
R03/BUZO	I/O		
R04	I/O		
R05	I/O		
R06	I/O		
R07	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R10	I/O		
R11	I/O		
R12	I/O		
R13	I/O		
R14	I/O		
R15	I/O		
R16	I/O		
R17	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R20	I/O		
R21	I/O		
R22	I/O		
R23	I/O		
R24	I/O		
R25	I/O		
R26	I/O		
R27	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R30	I/O		
R31	I/O		
R32	I/O		
R33	I/O		
R34	I/O		
R35	I/O		

* On : with pull-down resistance
* Off : without pull-down resistance

I/O Option [Normal Port]

Bit	I/O	I/O Option On	I/O Option Off
R40/T0O	I/O		
R41	I/O		
R42	I/O		
R43	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R50	I/O		
R51	I/O		
R52	I/O		
R53/SCLK	I/O		
R54/SIN	I/O		
R55/SOUT	I/O		
R56/PWM1O	I/O		
R57	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R60/AN0	I/O		
R61/AN1	I/O		
R62/AN2	I/O		
R63/AN3	I/O		
R64/AN4	I/O		
R65/AN5	I/O		
R66/AN6	I/O		
R67/AN7	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R70/AN8	I/O		
R71/AN9	I/O		
R72/AN10	I/O		
R73/AN11	I/O		
R74	I/O		
R75	I/O		

* On : with pull-up
* Off : without pull-up

GMS81C2 Series [GMS81C2120/12] Option List

Package

☐ 42SDIP	☐ 44MQFP
☐ 40PDIP	

Date of Order	1999 / 2000. . .
Customer	
Department	
Name	
ROM Code Name	
Check sum	
ROM Size	☐ 20KBytes ☐ 12KBytes

RA / Vdisp

☐ RA Without pull-down resistance
☐ Vdisp

*Note : In the I/O options list, *you must select Vdisp* even if only one pin is selected with pull-down resistance.

ROM Code Option List : 703FH

Bit7		Bit6		Bit5		Bit4		Bit3		Bit2		Bit1		Bit0	
-		-		PFD1		PFD0		-		-		LOW VOLTAGE		RCOSC	
0 ✓	1	0 ✓	1	0	1	0	1	0 ✓	1	0 ✓	1	0	1	0	1

** Refer to Device Configuration Area*

I/O Option [VFD Driving Port]

Bit	I/O	I/O Option On	I/O Option Off
R00/INT0	I/O		
R01/INT1	I/O		
R02/EC0	I/O		
R03/BUZO	I/O		
R04	I/O		
R05	I/O		
R06	I/O		
R07	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R20	I/O		
R21	I/O		
R22	I/O		
R23	I/O		
R24	I/O		
R25	I/O		
R26	I/O		
R27	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R30	I/O		
R31	I/O		
R32	I/O		
R33	I/O		
R34	I/O		

* On : with pull-down resistance
* Off : without pull-down resistance

I/O Option [Normal Port]

Bit	I/O	I/O Option On	I/O Option Off
R53/SCLK	I/O		
R54/SIN	I/O		
R55/SOUT	I/O		
R56/PWM1O	I/O		
R57	I/O		

Bit	I/O	I/O Option On	I/O Option Off
R60/AN0	I/O		
R61/AN1	I/O		
R62/AN2	I/O		
R63/AN3	I/O		
R64/AN4	I/O		
R65/AN5	I/O		
R66/AN6	I/O		
R67/AN7	I/O		

* On : with pull-up
* Off : without pull-up