HMS81C2232/48, HMS81C2332/48

CMOS Single-Chip 8-Bit Microcontroller
with VFD Controller & VFD Driver

1. OVERVIEW

1.1 Description

The HMS81C2232/48 is advanced CMOS 8-bit micro-controller with 32/48K bytes of ROM(EPROM). This is a powerful micro-controller which provides a highly flexible and cost effective solution to many VFD applications. This provides the following standard features : 32K/48K bytes of ROM(EPROM), 1K bytes of RAM, 8-bit timer/counter, 8-bit A/D converter, 6-bit Watch dog Timer, Programmable Buzzer Driving Port, Serial Peripheral Interface, 8-bit remote control timer, on-chip oscillator and clock circuitry. It also comes with high voltage I/O pins that can directly drive a VFD(Vacuum Fluorescent Display). In addition, the HMS8XC2232/48 support power saving modes to reduce power consumption.

Device Name	ROM Size	RAM Size	Display RAM	OTP	Package
HMS81C2232	32K bytes	896 bytes	112bytes	HMS87C2232	80TQFP 80MQFP
HMS81C2248	48K bytes			HMS87C2248	
HMS81C2332	32K bytes	896 bytes	112bytes	HMS87C2332	64SDIP,64LQFP 64MQFP
HMS81C2348	48K bytes			HMS87C2348	

1.2 HMS81C2232/48 Features

- **32K/48K bytes ROM(EPROM)**
- **896 Bytes of On-Chip Data RAM (Including STACK Area)**
- **112 Bytes of On-Chip Display RAM**
- **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**
- **Two 8-bit Serial Peripheral Interface**
- **Two External Interrupt Ports**
- **One Programmable 6-bit Buzzer Driving Port**
- **53 FIP Ports**
  - 16EA Output only pins
  - High-voltage pins Max. 40V
- **Operating Temperature**
  -40°C ~ 85°C
- **12 Interrupt Sources**
  - Two External Sources (INT0, INT1)
  - Two Timer/Counter Sources (Timer0, Timer1)
  - Three Remote Timer Sources(FE,RE,OVF)
  - Two SPI Sources(SIO1, SIO3)
  - Three Functional Sources (ADC,WDT,BIT)
- **7-Channel 8-bit On-Chip Analog to Digital Converter**
- **Oscillator:**
  - Crystal
  - Ceramic Resonator
- **Low Power Dissipation Modes**
  - STOP Mode
  - SLEEP Mode
- **Operating Voltage: 2.7V ~ 5.5V (@8MHz)**
  4.5V ~ 5.5V (@10MHz)
- **Operating Frequency: 1MHz ~ 10MHz**
- **Enhanced EMS Improvement**
  Power Fail Processor
  (Noise Immunity Circuit)

1.3 HMS2332/48 Features

- 32K/48K bytes ROM(EPROM)
- 896 Bytes of On-Chip Data RAM
  (Including STACK Area)
- 112 Bytes of On-Chip Display RAM
- 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
- Two 8-bit Serial Peripheral Interface
- Two External Interrupt Ports
- One Programmable 6-bit Buzzer Driving Port
- 41 FIP Ports
  - 12EA Output only pins
  - High-voltage pins Max. 40V
- Operating Temperature
  -40°C ~ 85°C
- 11 Interrupt Sources
  - Two External Sources (INT0, INT1)
  - Two Timer/Counter Sources (Timer0, Timer1)
  - Three Remote Timer Sources(FE,RE,OVF)
  - One SPI Sources(SIO1)
  - Three Functional Sources (ADC,WDT,BIT)
- 5-Channel 8-bit On-Chip Analog to Digital Converter
- Oscillator:
  - Crystal
  - Ceramic Resonator
- Low Power Dissipation Modes
  - STOP Mode
  - SLEEP Mode
- Operating Voltage: 2.7V ~ 5.5V (@8MHz)
  4.5V ~ 5.5V (@10MHz)
- Operating Frequency: 1MHz ~ 10MHz
- Enhanced EMS Improvement
  Power Fail Processor
  (Noise Immunity Circuit)

1.4 Development Tools

The HMS81C22xx/23xx are supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Dr.TM and OTP programmers. There are third different type programmers such as emulator add-on board type, single type, gang type. For mode detail, Refer to “25. OTP PROGRAMMING” on page 112. Macro assembler operates under the MS-Windows 95, 98, NT, 2000, XPTM.Please contact sales part of Hynix Semiconductor.

In Circuit Emulators	CHOICE-Dr.
Socket Adapter for OTP	OA87C23XX-64SD (64SDIP) OA87C23XX-64QF (64MQFP) OA87C23XX-64QT (64LQFP) OA87C22XX-80QF (80MQFP) OA87C22XX-80QT (64TQFP)
POD	CHPOD81C22D-64SD (64SDIP)
Assembler	HYNIX Macro Assembler

1.5 Ordering Information

HMS81C22xx	Device name	ROM Size	RAM size	Package
Mask version	HMS81C2232 Q HMS81C2232 TQ HMS81C2248 Q HMS81C2248 TQ	32K bytes 32K bytes 48K bytes 48K bytes	896 bytes	80MQFP 80TQFP 80MQFP 80TQFP
OTP version	HMS81C2232 Q HMS81C2232 TQ HMS81C2248 Q HMS81C2248 TQ	32K bytes OTP 32K bytes OTP 48K bytes OTP 48K bytes OTP	896 bytes	80MQFP 80TQFP 80MQFP 80TQFP

HMS81C23xx	Device name	ROM Size	RAM size	Package
Mask version	HMS81C2332 K HMS81C2332 Q HMS81C2332 LQ HMS81C2348 K HMS81C2348 Q HMS81C2348 LQ	32K bytes 32K bytes 32K bytes 48K bytes 48K bytes 48K bytes	896 bytes	64SDIP 64MQFP 64LQFP 64SDIP 64MQFP 64LQFP
OTP version	HMS87C2332 K HMS87C2332 Q HMS87C2332 LQ HMS87C2348 K HMS87C2348 Q HMS87C2348 LQ	32K bytes OTP 32K bytes OTP 32K bytes OTP 48K bytes OTP 48K bytes OTP 48K bytes OTP	896 bytes	64SDIP 64MQFP 64LQFP 64SDIP 64MQFP 64LQFP

2. BLOCK DIAGRAM

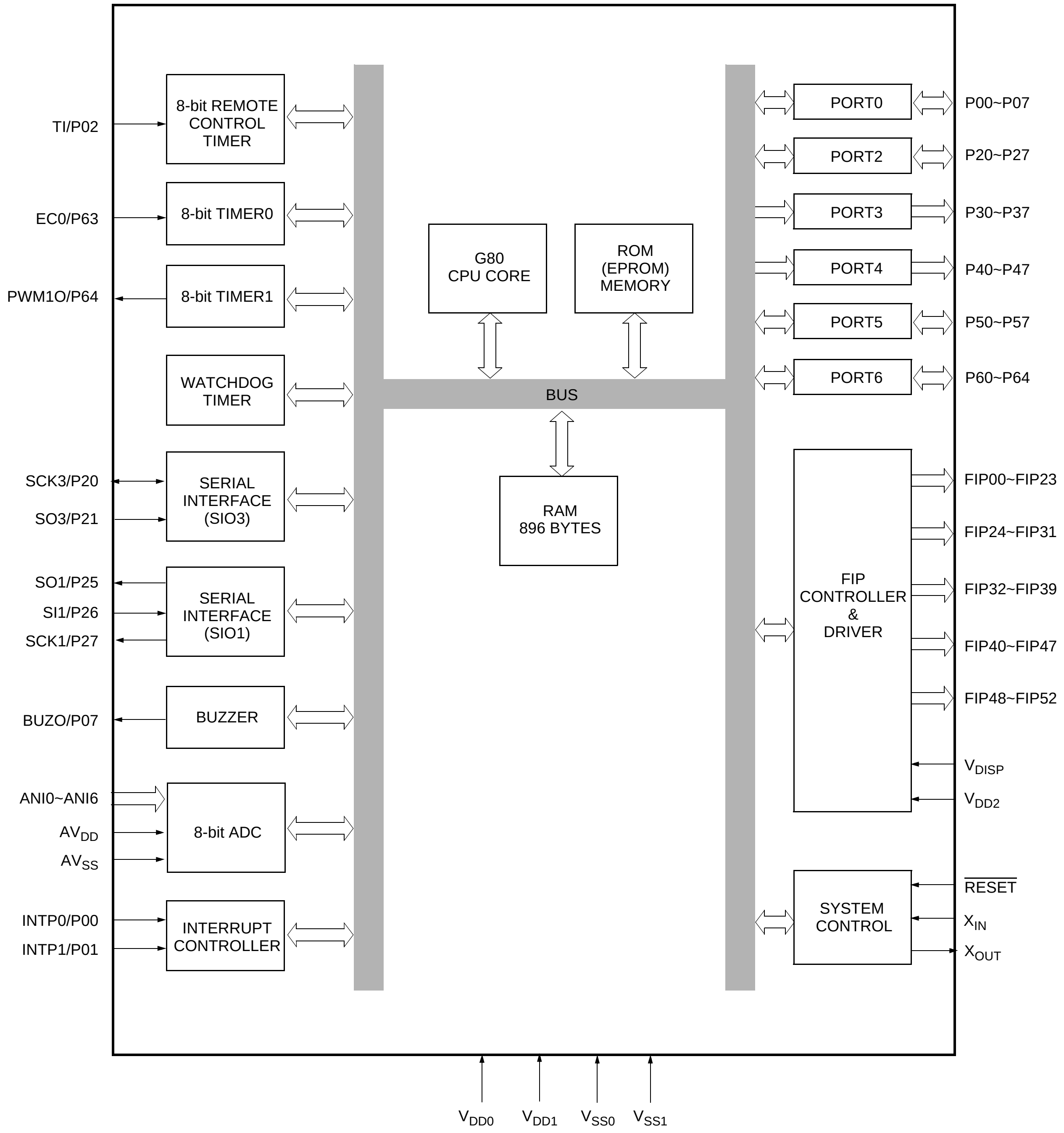


Figure 2-1 HMS81C2232/48 Block Diagram

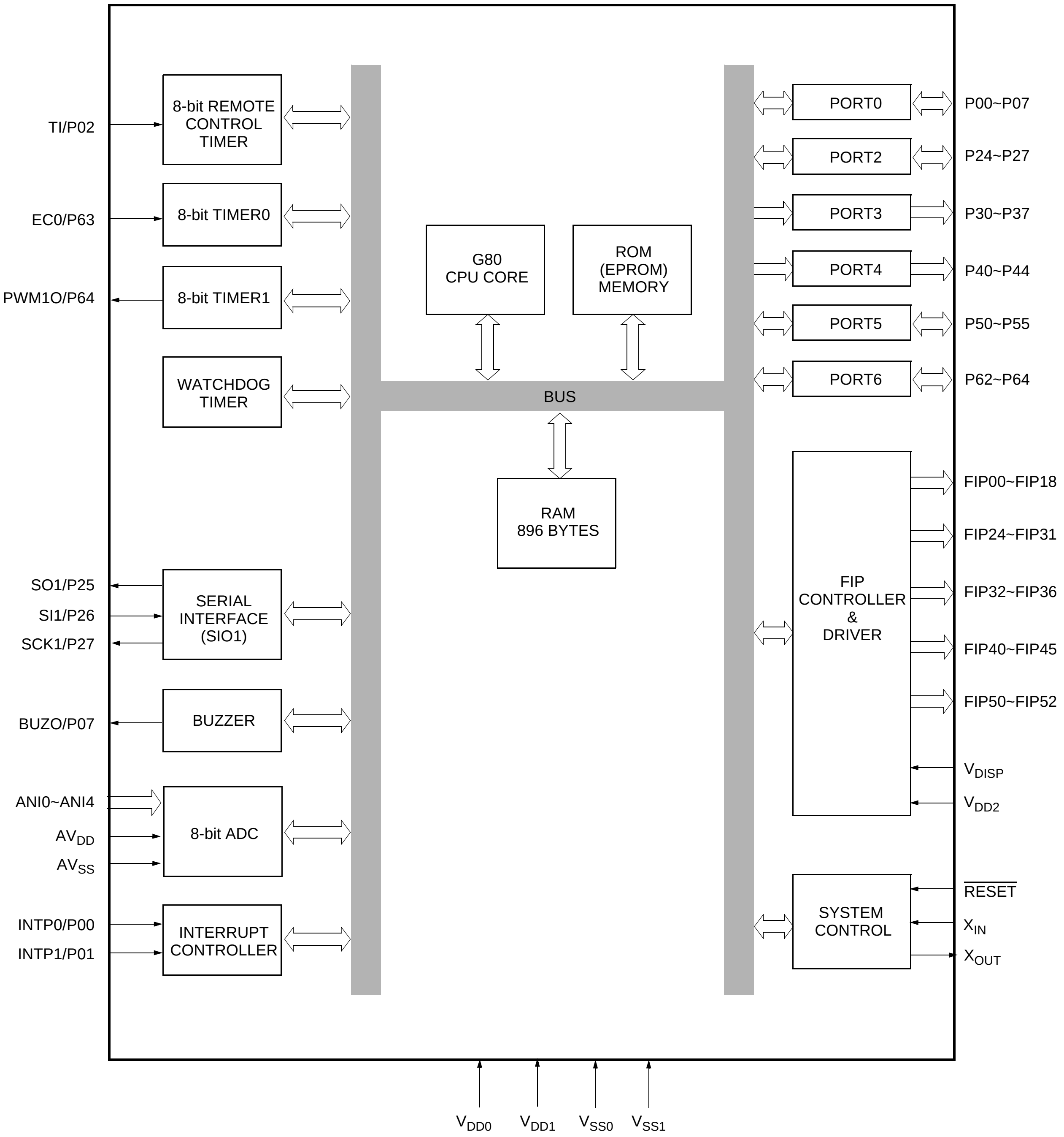
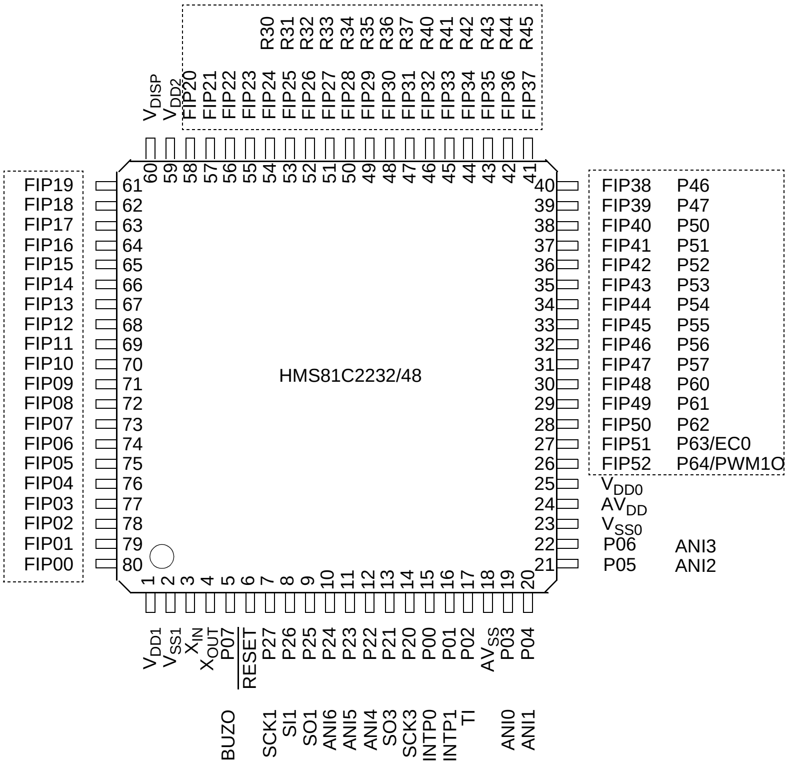


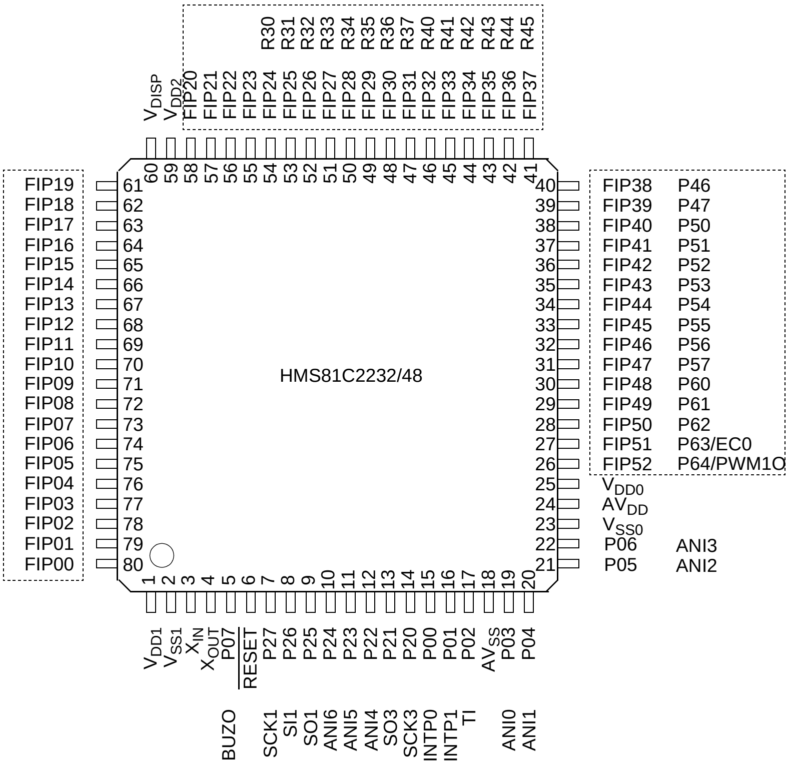
Figure 2-2 HMS81C2332/48 Block Diagram

3. PIN ASSIGNMENT

80TQFP



80MQFP



High Voltage Port

Figure 3-1 HMS81C2232/48 Pin Assignment

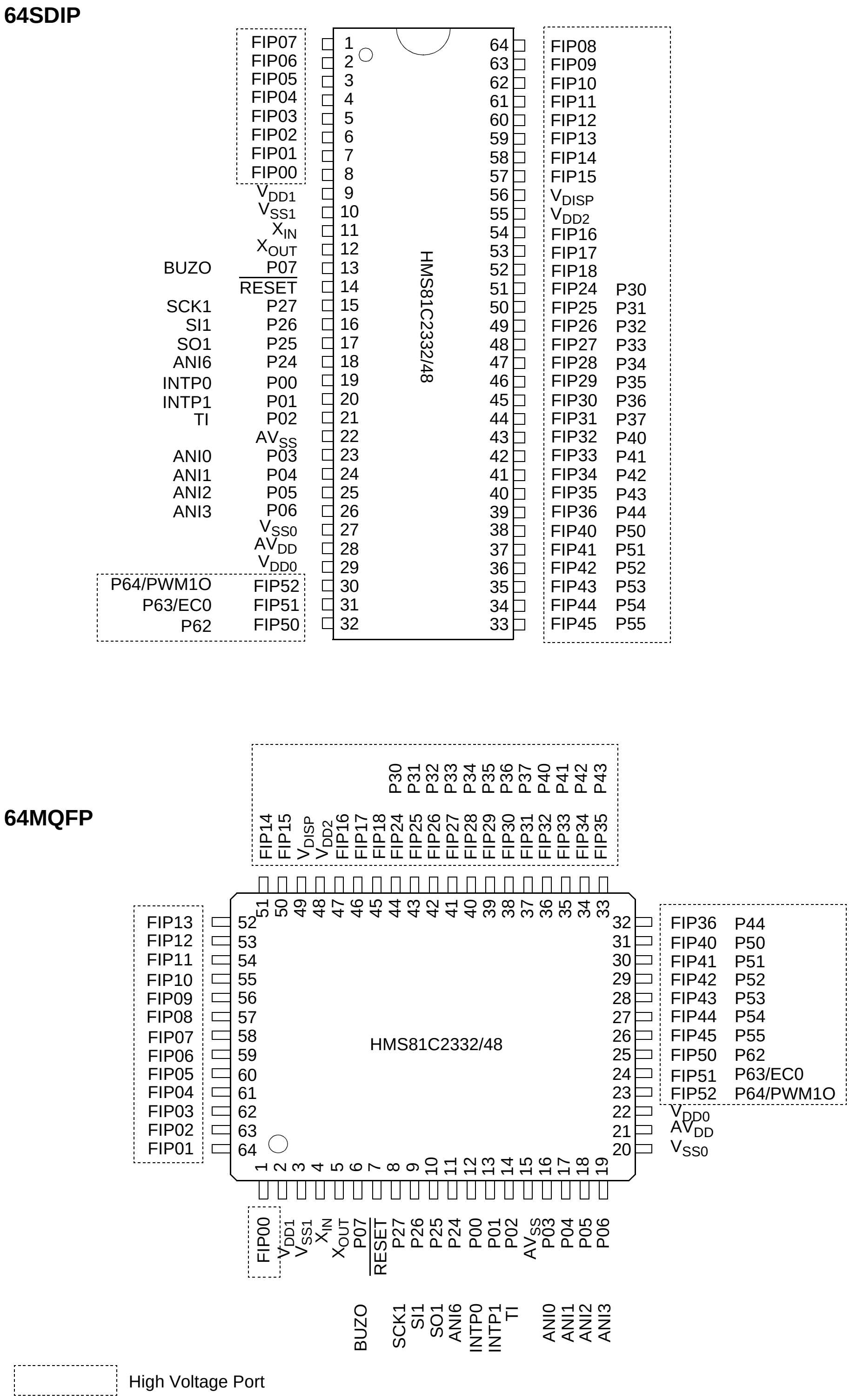


Figure 3-2 HMS81C2332/48 64SDIP & 64MQFP Pin Assignment

64LQFP

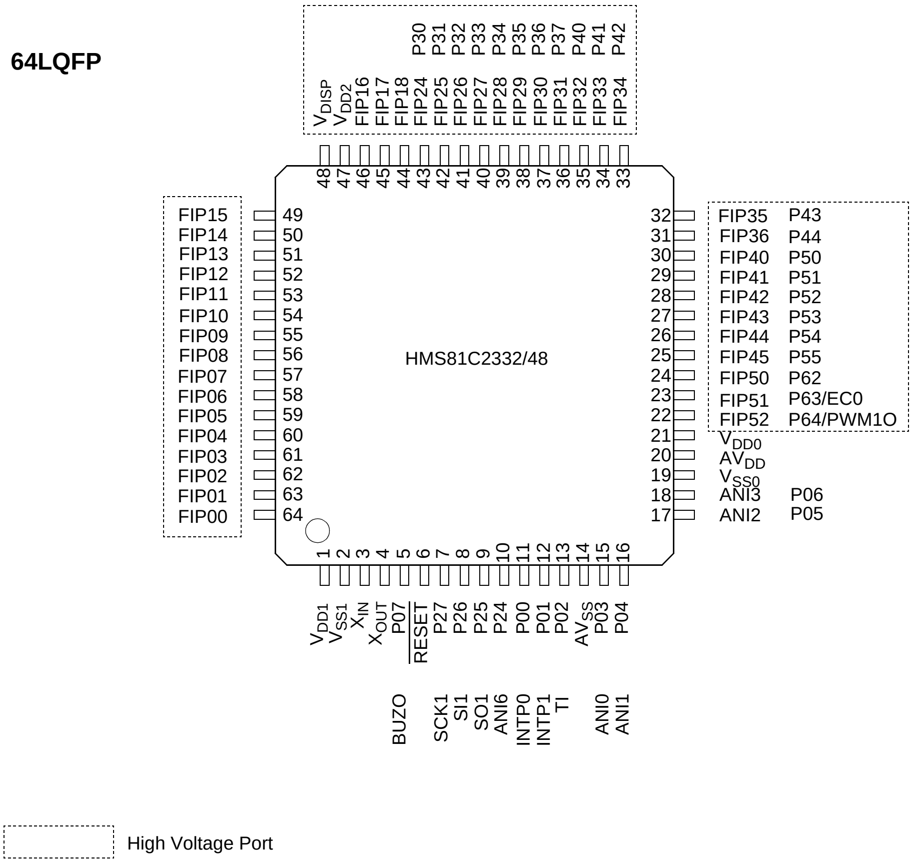
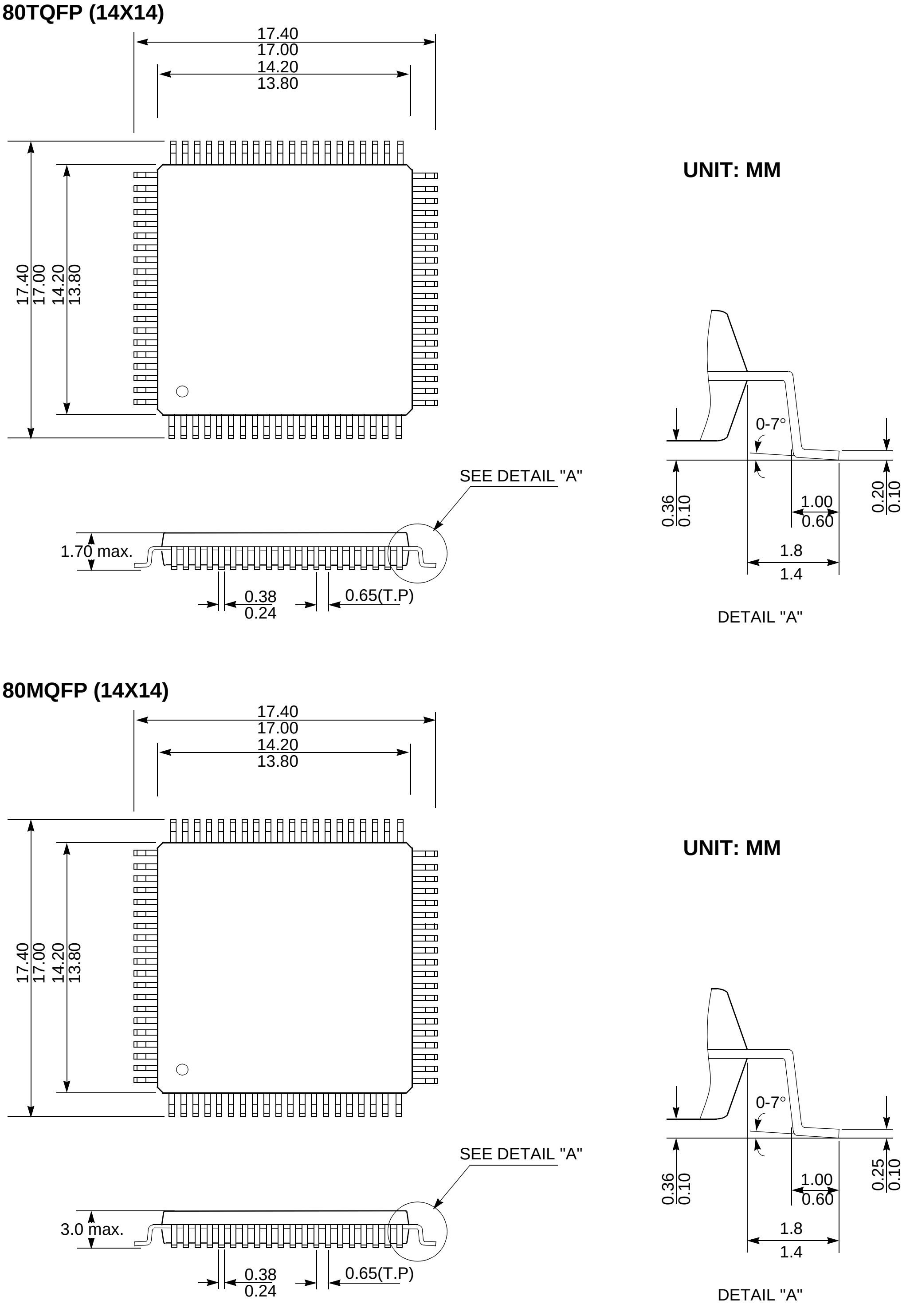


Figure 3-3 HMS81C2332/48 64LQFP Pin Assignment

4. PACKAGE DIAGRAM



80TQFP (14X14)

80MQFP (14X14)

Figure 4-1 HMS81C2232/48 Package Diagram

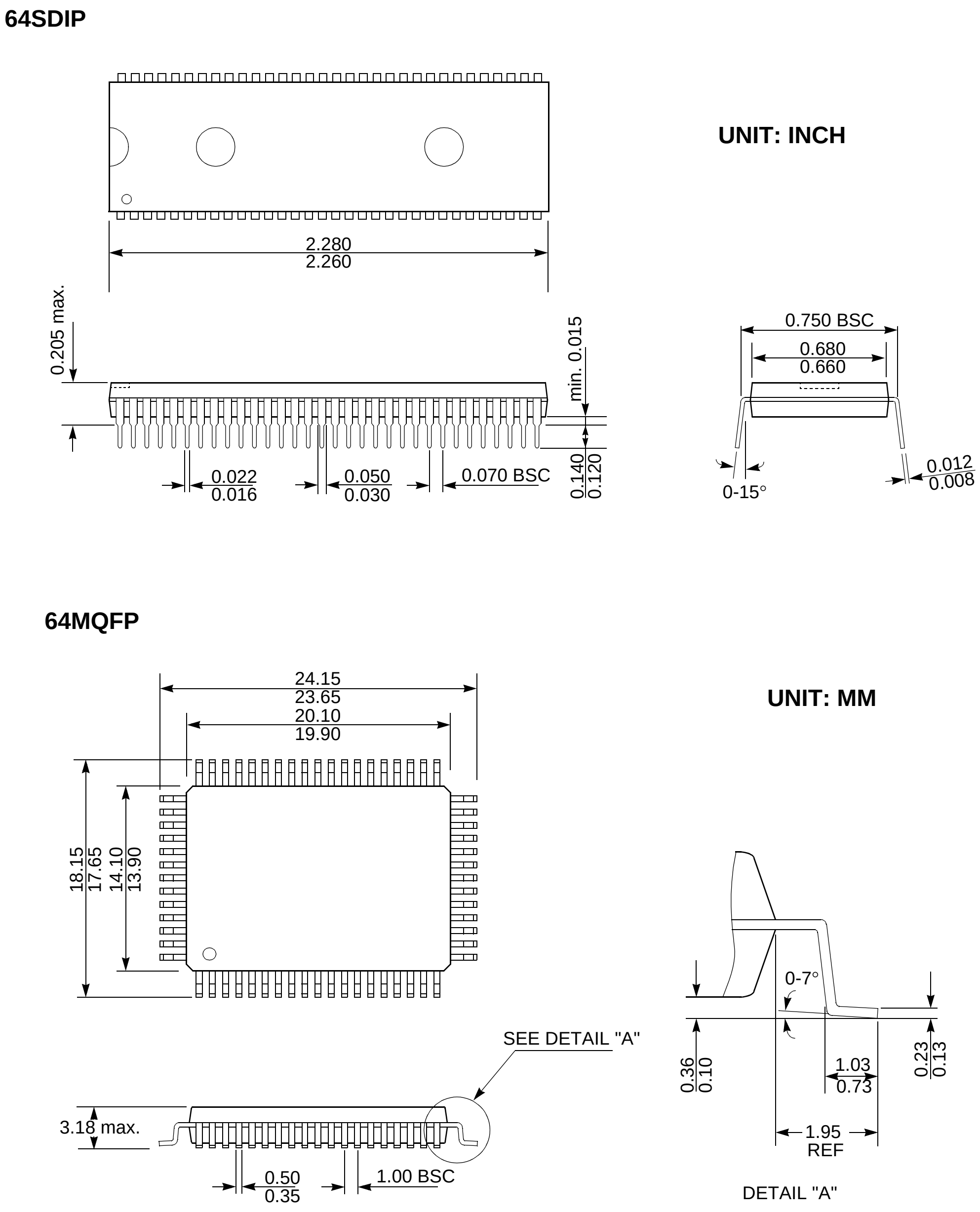


Figure 4-2 HMS81C2332/48 64SDIP & 64MQFP Package Diagram

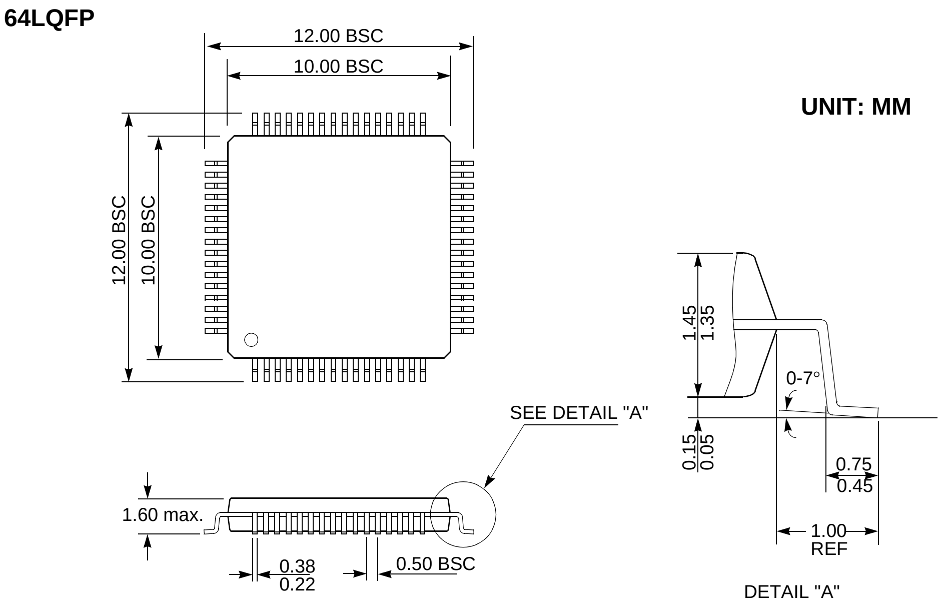


Figure 4-3 HMS81C2332/48 64LQFP Package Diagram

5. PIN FUNCTION

5.1 P00 ~ P07 (Port 0)

P00 through P07 are used as a 8-bit I/O port. These pins also have external interrupt request input and timer input functions in addition to the I/O port function. Port 0 can be set in the following operation modes in 1-bit units.

(1) Port mode

P00 through P07 function as a 8-bit I/O port in this mode. This 2-bit port can be set in the input or output mode in 1-bit units by the port 0 direction register (P0IO). When used as an input port, the internal pull-up resistor can be connected by using the pull-up resistor option register0 (PU0).

(2) Alternate mode

P00 through P01 functions as external interrupt request input and P02 functions as timer input pin. P03 through P06 functions as ADC input pin and P07 functions as buzzer dirver output pin.

(a) INTP0, INTP1

INTP0 and INTP1 input external interrupt requests whose valid edge can be specified (to be the rising edge, falling edge, or both the rising and falling edges).

(b) TI

TI input timer of the 8-bit remote control timer.

(c) ANI0, ANI1, ANDI2, ANI3

These are input pins of the A/D converter.

(d) BUZO

This is a output pin of buzzer dirver output.

Port pin	Alternate function
P00	INTP0 (External interrupt 0)
P01	INTP1 (External interrupt 1)
P02	TI (Timer input of remote control timer)
P03	ANI0 (Analog Input 0)
P04	ANI1 (Analog Input 1)
P05	ANI2 (Analog Input 2)
P06	ANI3 (Analog Input 3)
P07	BUZO (Buzzer driver output)

5.2 P20 ~ P27 (Port 2)

P20 through P27 constitute an 8-bit I/O port, port 2. These pins also have functions to input/output data of the serial interface, clock, and automatic transmit/receive busy input. The following operation modes can be specified in 1-bit units.

(1) Port mode

In this mode, P20 through P27 function as an 8-bit I/O port. This port can be set in the input or output mode in 1-bit units by using the port 2 direction register (P2IO). When the port is used as an input port, the internal pullup resistor can be used if so specified by the pull-up resistor option register 2 (PU2).

(2) Alternate mode

In this mode, P20 through P21 are used to output serial interface data, clock. P22 through P24 functions as ADC input pin. P25 through P27 are used to input/output serial interface data, clock.

(a) SI1, SO1, SO3

These are I/O pins of the serial data of the serial interface.

(b) SCK1, SCK3

These are I/O pins of the serial clock of the serial interface.

(c) ANI4, ANI5, ANI6

These are input pins of the A/D converter.

Port pin	Alternate function
P20	SCK3 (Serial3 clock input/output)
P21	SO3 (Serial3 data output)
P22	ANI4 (Analog Input 4)
P23	ANI5 (Analog Input 5)
P24	ANI6 (Analog Input 6)
P25	SO1 (Serial1 data output)
P26	SI1 (Serial1data input)
P27	SCK1 (Serial1 clock input/output)

5.3 P30 ~ P37 (Port 3)

P30 through P37 constitute an 8-bit output port. These pins are also used as FIP controller/driver output pins. The following operation modes can be specified in 1-bit units.

(1) Port mode

P30 through P37 function as an 8-bit output port in this mode. These pins are P-ch open-drain pins. Pull-down resistors can be connected to these pins of the mask ROMmodels by mask option. The HMS87C2232/48 does not have pull-down resistors.

(2) Alternate mode

In this mode, P30 through P37 function as the output pins of the FIP controller/driver (FIP24 through FIP31).

Port pin	Alternate function
P30~P37	FIP24-FIP31

5.4 P40 ~ P47 (Port 4)

P40 through P47 constitute an 8-bit output port. These pins are also used as FIP controller/driver output pins. The following operation modes can be specified in 1-bit units.

(1) Port mode

P40 through P47 function as an 8-bit output port in this mode. These pins are P-ch open-drain pins. Pull-down resistors can be connected to these pins of the mask ROM models by mask option. The HMS87C2232/48 does not have pull-down resistors.

(2) Alternate mode

In this mode, P40 through P47 function as the output pins of the FIP controller/driver (FIP32 through FIP39).

Port pin	Alternate function
P40~P47	FIP32-FIP39

5.5 P50 ~ P57 (Port 5)

P50 through P57 constitute an 8-bit I/O port. These pins are also used as FIP controller/driver output pins. The following operation modes can be specified in 1-bit units.

(1) Port mode

P50 through P57 function as an 8-bit I/O port in this mode. These pins are P-ch open-drain pins. Pull-down resistors can be connected to these pins of the mask ROM models by mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 does not have pull-down resistors.

(2) Alternate mode

In this mode, P50 through P57 function as the output pins of the FIP controller/driver (FIP40 through FIP47).

Port pin	Alternate function
P50~P57	FIP40-FIP47

5.6 P60 ~ P64 (Port 6)

P60 through P64 constitute a 5-bit I/O port. These pins are also used as FIP controller/driver output pins. The following operation modes can be specified in 1-bit units.

(1) Port mode

P60 through P64 function as a 5-bit input/output port in this mode. These pins are P-ch open-drain pins. Pull-down resistors can be connected to these pins of the mask ROM models by mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 does not have pull-down resistors.

(2) Alternate mode

In this mode, P60 through P64 function as the output pins of the FIP controller/driver (FIP48 through FIP52).

(a) EC0

This is input pin of the Event Counter0.

(b) PWM1O

This is outinput pin of the PWM.

Port pin	Alternate function
P60 P61 P62 P63 P64	FIP48 FIP49 FIP50 FIP51 / EC0 FIP52 / PWM1O

5.7 FIP0 ~ FIP23

These are the output pins of the FIP controller/driver.

5.8 V_{DISP}

This pin connects a pull-down resistor to the FIP controller/driver.

5.9 AV_{DD}

This pin supply an analog voltage to the A/D converter. Always keep this pin at the same potential as the V_{DD1} pin even when the A/D converter is not used.

5.10 AV_{SS}

This is the ground pin of the A/D converter. Always keep this pin at the same potential as the V_{SS1} pin even when the A/D converter is not used.

5.11 /RESET

This pin inputs an active-low system reset signal.

5.12 X_{IN} and X_{OUT}

These pins connect a crystal resonator for main system clock oscillation. To supply an external clock, input it to X_{IN}, and input a signal reverse to that input to X_{IN}, to X_{OUT}.

5.13 ANI0 through ANI6

These are the input pins of the A/D converter.

5.14 V_{DD0} ~ V_{DD2}

V_{DD0} supplies a positive voltage to the ports. V_{DD1} supplies a positive voltage to the internal function blocks other than the ports, analog block, and FIP controller/driver. V_{DD2} supplies a positive voltage to the FIP controller/driver.

5.15 V_{SS0} and V_{SS1}

V_{SS0} is the ground pin for the ports. V_{SS1} is the ground pin for the internal function blocks other than the ports and analog block.

PIN NAME	In/Out	Function	
		Basic	Alternate
P00 (INTP0)	I/O (I)	Port0 8-bit I/O ports Can be set in input or output mode in 1-bit units. Internal pull-up resistor can be used via software when this port is used as input port	External interrupt 0 input
P01 (INTP1)	I/O (I)		External interrupt 1 input
P02 (TI)	I/O (I)		Timer input of 8-bit remote control timer
P03 (ANI0)	I/O (I)		Analog input channel 0 for A/D converter
P04 (ANI1)	I/O (I)		Analog input channel 1 for A/D converter
P05 (ANI2)	I/O (I)		Analog input channel 2 for A/D converter
P06 (ANI3)	I/O (I)		Analog input channel 3 for A/D converter
P07 (BUZO)	I/O (O)		Buzzer driving output
P20 (SCK3)	I/O (I/O)	Port2 8-bit I/O ports Can be set in input or output mode in 1-bit units. Internal pull-up resistor can be used via software when this port is used as input port	Serial3 clock input/output
P21 (SO3)	I/O (O)		Serial3 data output
P22 (ANI4)	I/O (I)		Analog input channel 4 for A/D converter
P23 (ANI5)	I/O (I)		Analog input channel 5 for A/D converter
P24 (ANI6)	I/O (I)		Analog input channel 6 for A/D converter
P25 (SO1)	I/O (O)		Serial1 data output
P26 (SI1)	I/O (I)		Serial1data input
P27 (SCK1)	I/O (I/O)		Serial1clock input/output
FIP0~FIP23	O	High voltage high-current of FIP controller/driver	-
P30~P37 (FIP24-FIP31)	O	Port3 P-ch open-drain 8-bit high-voltage output port. Pull-down resistor for V_{DISP} or V_{SS0} can be used by mask option in 1-bit units(mask ROM models only) OTP models do not have pull-down resistor	FIP24-FIP31
P40~P47 (FIP32-FIP39)	O	Port4 P-ch open-drain 8-bit high-voltage output port. Pull-down resistor for V_{DISP} or V_{SS0} can be used by mask option in 1-bit units(mask ROM models only) OTP models do not have pull-down resistor	FIP32-FIP39

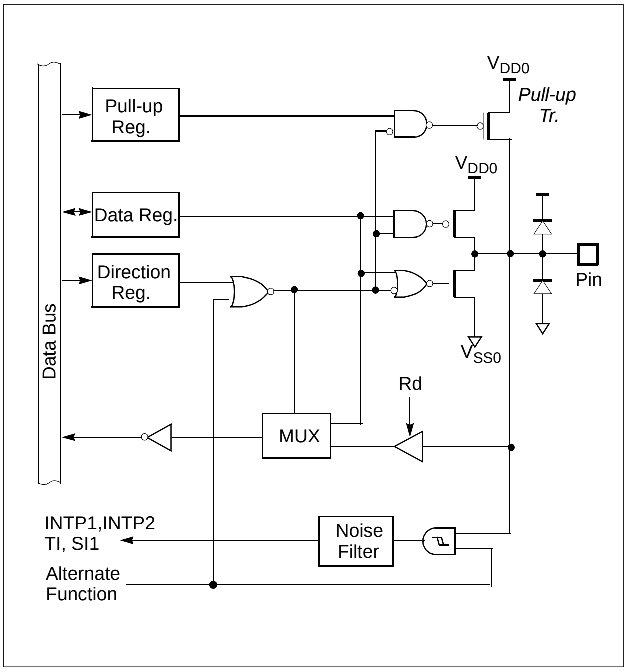
Table 5-1 HMS81C2232/48 Port Function Description

PIN NAME	In/Out	Function	
		Basic	Alternate
P50~P57 (FIP40-FIP47)	I/O (O)	Port5 P-ch open-drain 8-bit high-voltage output port. Can be set in input or output mode in 1-bit units. When it is used as input port, set the output latch to “0”, and read port level read(P50 through P57). Pull-down resistor for V_{DISP} or V_{SS0} can be used by mask option in 1-bit units(mask ROM models only). OTP models do not have pull-down resistor.	FIP40-FIP47
P60	I/O (O)	Port6 P-ch open-drain 8-bit high-voltage output port. Can be set in input or output mode in 1-bit units. When it is used as input port, set the output latch to “0”, and read port level read(P50 through P57). Pull-down resistor for V_{DISP} or V_{SS0} can be used by mask option in 1-bit units(mask ROM models only). OTP models do not have pull-down resistor.	FIP48
P61	I/O (O)		FIP49
P62	I/O (O)		FIP50
P63	I/O (I/O)		FIP51/EC0
P64	I/O (O)		FIP52/Timer1 PWM 1 pulse output
AVDD	-	Analog power/reference voltage input to A/D converter Set the same potential as V_{DD}	
AVSS	-	Ground potential for A/D converter. Set the same potential as V_{DD}	
VDD0	-	Positive power supply to ports	
VSS0	-	Ground potential to ports.	
VDD1	-	Positive power supply to internal function block	
VSS1	-	Ground potential(except ports, analog block)	
VDD2	-	Positive power supply to FIP controller/driver.	
V_{disp}	-	Pull-down resistor connection of FIP controller/driver	
$\overline{\text{RESET}}$	I	System reset signal input	
XIN	I	Main system clock oscillation input	
XOUT	O	Main system clock oscillation output	

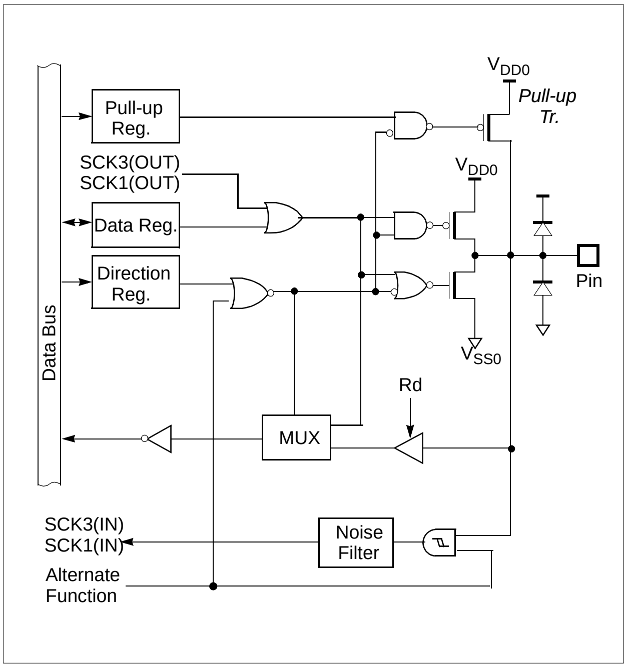
Table 5-1 HMS81C2232/48 Port Function Description

6. PORT STRUCTURES

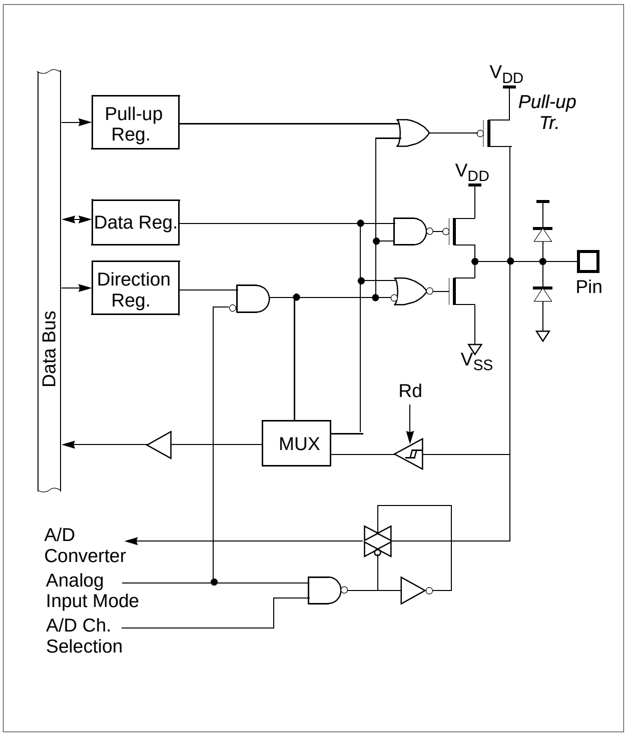
P00~P01/INTP0~INTP1, P02/TI, P26/SI1



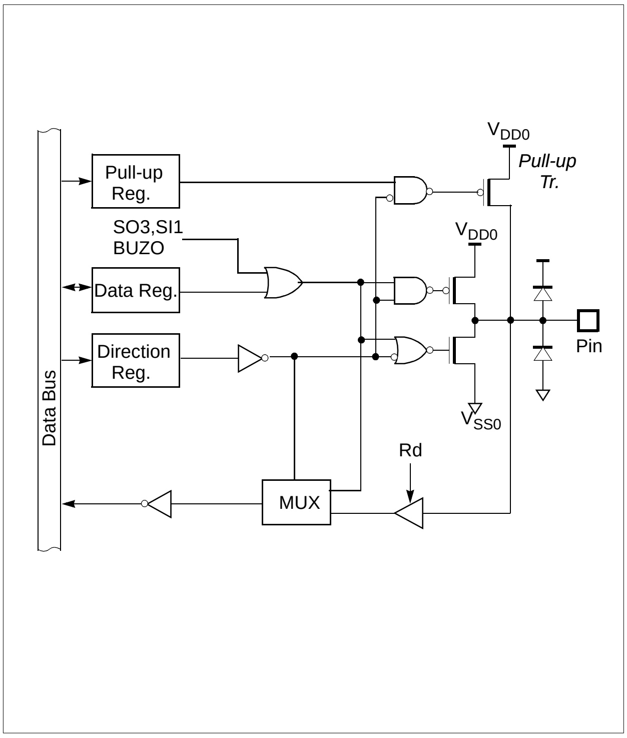
P20/SCK3, P27/SCK1



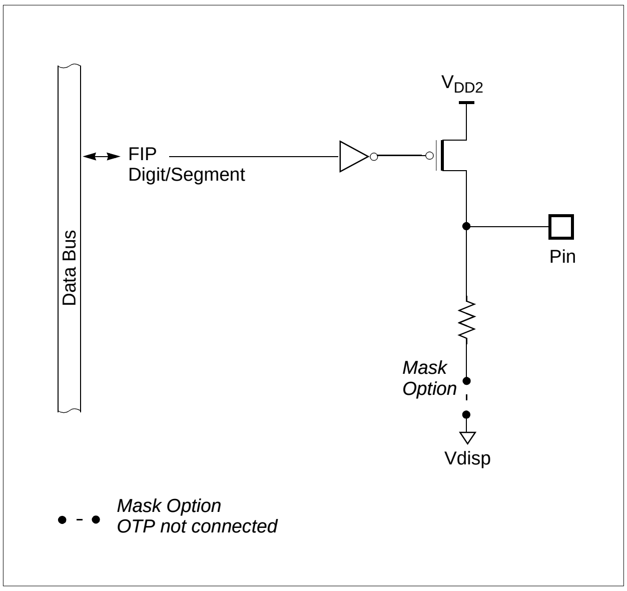
P03~P06/AN0~AN3, P22~P24/AN4~AN6



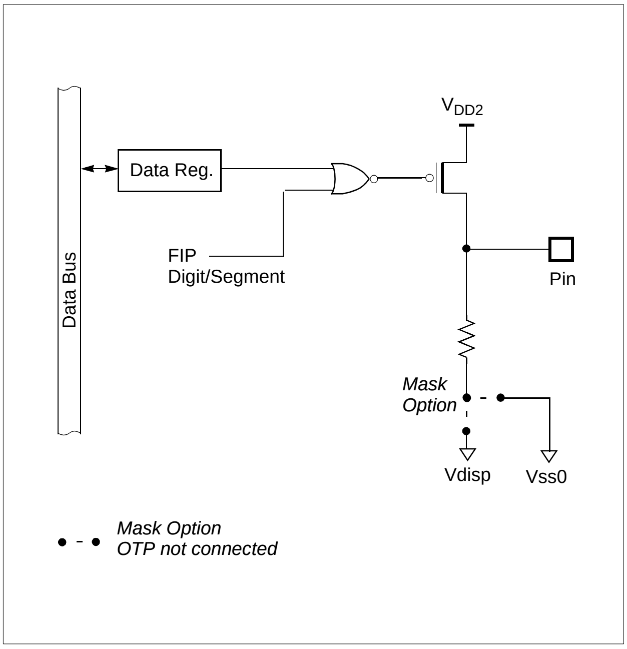
P21/SO3, P25/SO1, P07/BUZO



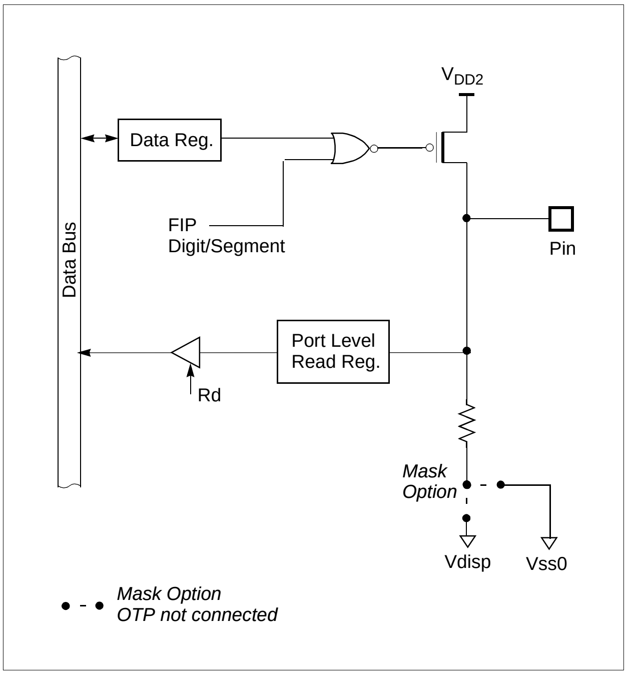
FIP00~FIP23



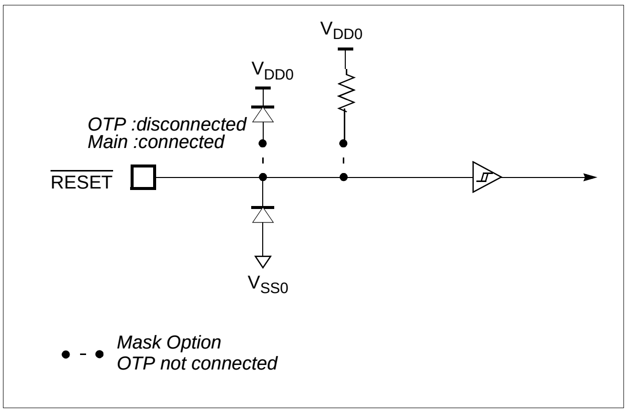
P30~37/FIP24~FIP31, P40~47/FIP32~FIP39



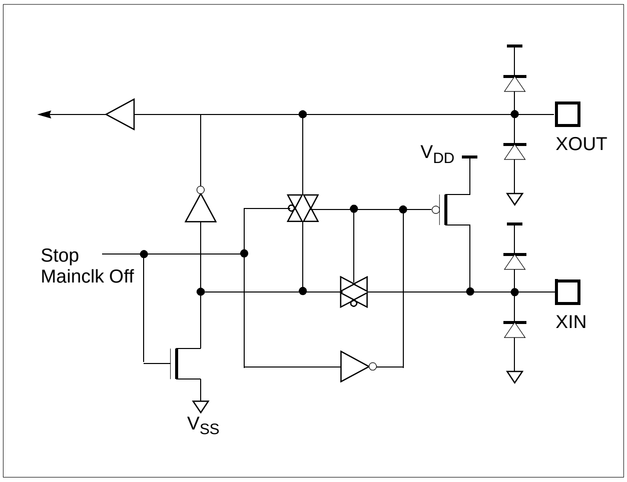
P50~P57/FIP40~FIP47, P60~P64/FIP48~FIP52



$\overline{\text{RESET}}$



XIN, XOUT



7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings

Parameter	Symbol	Specifications		Unit
		Min.	Max.	
Supply Voltage	V_{DD}	-0.3	6.0	V
	AV_{DD}	V_{DD}-0.3	V_{DD}+0.3	V
	AV_{SS}	-0.3	0.3	V
	V_{DISP}	V_{DD}-45	V_{DD}+0.3	V
Normal Voltage Pin	V_{I1}	-0.3	V_{DD}+0.3	V
	V_{O1}	-0.3	V_{DD}+0.3	V
	I_{OH1}	-8		mA
	ΣI_{OH1}	-30		mA
	I_{OL1}	15		mA
	ΣI_{OL1}	50		mA
Hige Voltage Pin	V_{I2}	V_{DD}-45	V_{DD}+0.3	V
	V_{O2}	V_{DD}-45	V_{DD}+0.3	V
	I_{OH2}	-30		mA
	ΣI_{OH2}	-120		mA
Total Power Dissipation	PT	700		mW
Storage Temperature	TSTG	-40	125	°C

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

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Specifications		Unit
			Min.	Max.	
Supply Voltage	V_{DD}	f_{XI} = 5 MHz	2.7	5.5	V
Operating Frequency	f_{XIN}	V_{DD} = 2.7V~5.5V	1	10	MHz
Operating Temperature	T_{OPR}	V_{DD} = 2.7V~5.5V	-40	85	°C

7.3 A/D Converter Characteristics

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

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.[1]	Max.	
Analog Power Supply Input Voltage Range	AV_{DD}		AV_{SS}	-	AV_{DD}	V
Analog Input Voltage Range	V_{AN}		AV_{SS}		AV_{DD}	V
Current Following Between AV_{DD} and AV_{SS}	I_{AVDD}		-	-	200	uA
Overall Accuracy	CA_{IN}		-	-	±2	LSB
Non-Linearity Error	N_{NLE}		-	-	±2	LSB
Differential Non-Linearity Error	N_{DNLE}		-	-	±2	LSB
Zero Offset Error	N_{ZOE}		-	-	±2	LSB
Full Scale Error	N_{FSE}		-	-	±2	LSB
Gain Error	N_{NLE}		-	-	±2	LSB
Conversion Time	T_{CONV}	f_{XIN}=4MHz	-	-	30	uS

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

7.4 DC Electrical Characteristics

(V_{DD} = 5.0V ± 10%, V_{SS} = 0V, T_A = -40 ~ 85°C, f_{XIN} = 5 MHz, Vdisp = V_{DD}-40V to V_{DD}),

Parameter	Symbol	Pin	Test Condition	Specification			Unit
				Min	Typ.[1]	Max	
Input High Voltage	V_{IH1}	XIN	External Clock	$0.9V_{DD}$		$V_{DD}+0.3$	V
	V_{IH2}	$\overline{RESET}$,SI1,INTP0,INTP1,TI, SCK1,EC0		$0.8V_{DD}$		$V_{DD}+0.3$	
	V_{IH3}	P00~P07,P20~P27,P50~P57 P60~P64		$0.7V_{DD}$		$V_{DD}+0.3$	
Input Low Voltage	V_{IL1}	XIN	External Clock	-0.3		$0.1V_{DD}$	V
	V_{IL2}	$\overline{RESET}$,SI1,INTP0,INTP1,TI, SCK1,EC0,SCK3		-0.3		$0.2V_{DD}$	
	V_{IL3}	P00~P07,P20~P27		-0.3		$0.3V_{DD}$	
	V_{IL4}	P50~P57,P60~P64				$0.3V_{DD}$	
Output High Voltage	V_{OH1}	P00~P07,P20~P27	I_{OH} = -1.0mA	V_{DD}-1.0		V_{DD}	V
		P00~P07,P20~P27	I_{OH} = -100uA	V_{DD}-0.5		V_{DD}	
	V_{OH2}	XOUT	I_{OH} = -50uA	V_{DD}-2.0		V_{DD}	
Output Low Voltage	V_{OL1}	P00~P07,P20~P27	I_{OL} = 400uA	0		0.5	V
	V_{OL2}	XOUT	I_{OL} = 50uA	0		2	
Input High Leakage Current	I_{IH1}	P00~P07,P20~P27,P50~P57 P60~P64,$\overline{RESET}$	$V_{IN} = V_{DD}$			1	uA
Input Low Leakage Current	I_{IL1}	P00~P07,P20~P27,P50~P57 P60~P64,$\overline{RESET}$	V_{IN} = 0			-1	uA
	I_{IL2}	P50~P57,P60~P64	$V_{IN} = V_{DD}$-40V			-10	
Input Pull-up Resistor(*Option)	R_{PU}	P00~P07,P20~P27	V_{DD}=5V	10	60	100	KΩ
OSC Feed Back Resistor	R_{FB}	XIN, XOUT	V_{DD}=5V	0.25		2.5	MΩ
VFD Output Current	I_{OD1}	FIP00~FIP19	$V_{OD}=V_{DD}$-2V			-15	mA
	I_{OD2}	FIP20~FIP52	$V_{OD}=V_{DD}$-2V			-5	mA
On-Chip Mask Option Pull-down Resistance	R_{D1}	P50~P57,P60~P64	V_{SS0} Connection	15	35	90	KΩ
	R_{D2}	FIP00~FIP52	V_{DISP} Connection $V_{DD}-V_{DISP}$=40V	30	60	135	KΩ
Power Fail Detect Voltage	V_{PFD}	V_{DD}			2.7		V
Current Dissipation in Active Mode	I_{DD}	V_{DD}	fXIN=5MHz		5	10	mA
Current Dissipation in SLEEP Mode	I_{SLEEP}	V_{DD}	fXIN=5MHz		2	3	mA

Current Dissipation in STOPO Mode	I_{STOP}	V_{DD}	f_{XIN}=Off		1	10	uA
Internal RC WDT Frequency	T_{RCWDT}	XOUT		8		30	KHz

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

7.5 AC Characteristics

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

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Operating Frequency	f_{CP}	XIN	1	-	10	MHz
System Clock Cycle Time	t_{SYS}	-	200		2000	nS
Oscillation Stabilizing Time	t_{ST}	XIN, XOUT	-	-	20	mS
External Clock Pulse Width	t_{CPW}	XIN	40	-	-	nS
External Clock Transition Time	t_{RCP},t_{FCP}	XIN	-	-	10	nS
Interrupt Input Pulse Width	t_{EPW}	INTP0, INTP1	2	-	-	t_{SYS}
Event Counter Input Pulse Width	t_{ECW}	EC0	2			t_{SYS}
Event Counter Transition Time	t_{REP},t_{FEP}	EC0	-	-	20	nS
$\overline{\text{RESET}}$ Input Width	t_{RST}	$\overline{\text{RESET}}$	8	-	-	t_{SYS}

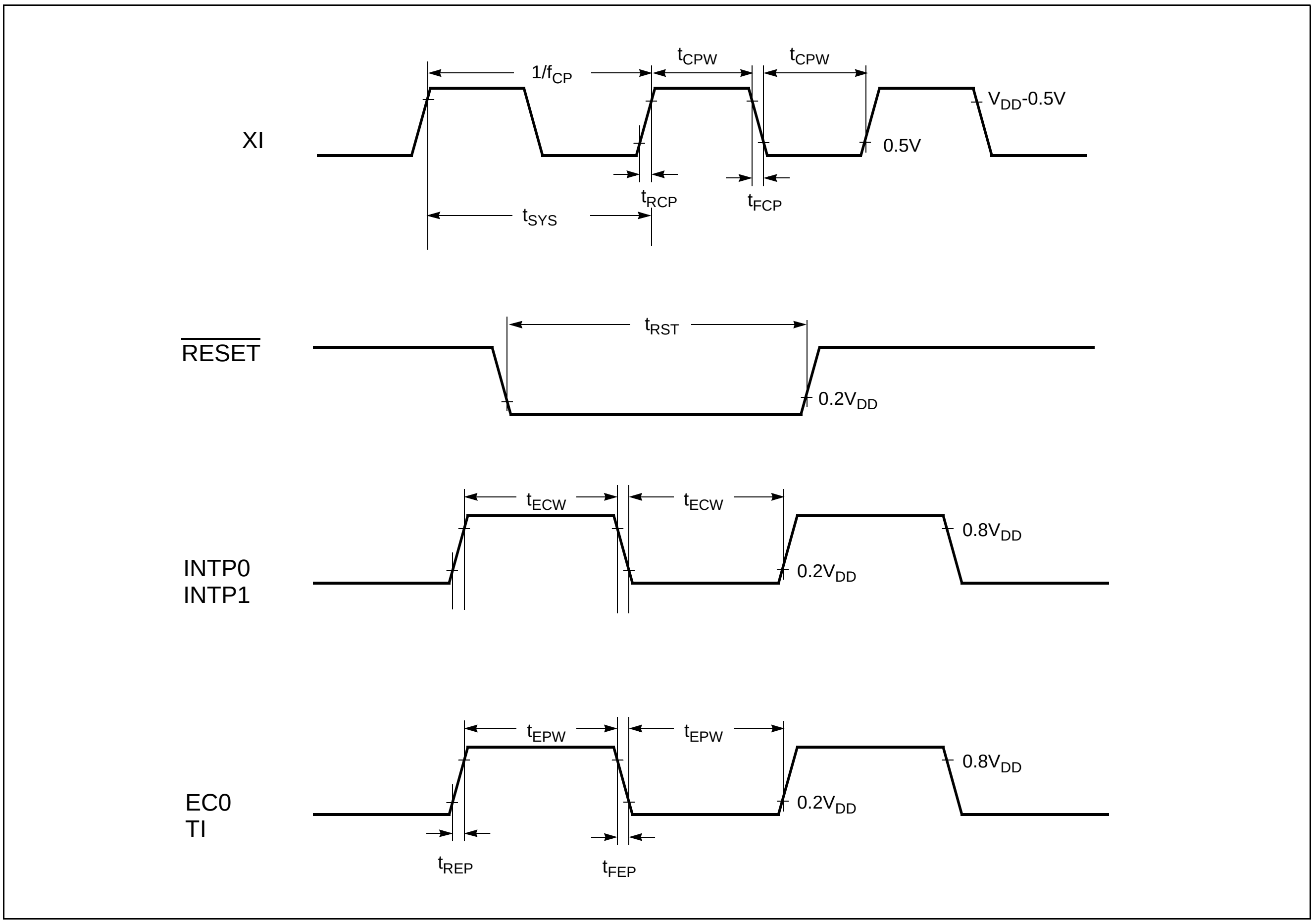


Figure 7-1 Timing Chart

7.6 AC Characteristics

(T_A=-40~+85°C, V_{DD}=5V±10%, V_{SS}=0V, f_{XIN}=4MHz)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Serial Input Clock Pulse	t_{SCYC}	SCK1, SCK3	$2t_{SYS}$+200	-	-	ns
Serial Input Clock Pulse Width	t_{SCKW}	SCK1, SCK3	t_{SYS}+70	-	-	ns
Serial Input Clock Pulse Transition Time	t_{FSCK} t_{RSCK}	SCK1, SCK3	-	-	30	ns
Serial Output Clock Cycle Time	t_{SCYC}	SCK1, SCK3	$4t_{SYS}$	-	$16t_{SYS}$	ns
Serial Output Clock Pulse Width	t_{SCKW}	SCK1, SCK3	$2t_{SYS}$-30			ns
Serial Output Clock Pulse Transition Time	t_{FSCK} t_{RSCK}	SCK1, SCK3			30	ns
Serial Output Delay Time	t_{DS}	SO			100	ns
SI Input Pulse Transition Time	t_{FSIN} t_{RSIN}	SI1	-	-	30	ns
SI Input Setup Time (External SCK)	t_{SUS}	SI1	100	-	-	ns
SI Input Setup Time (Internal SCK)	t_{SUS}	SI1	200	-		ns
SI Input Hold Time	t_{HS}	SI1	t_{SYS}+100	-		ns

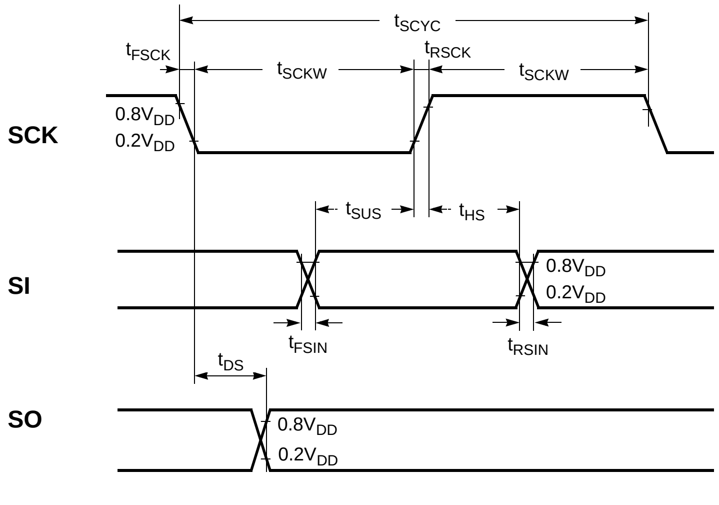


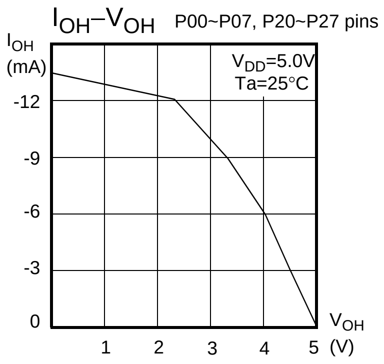
Figure 7-2 Serial I/O Timing Chart

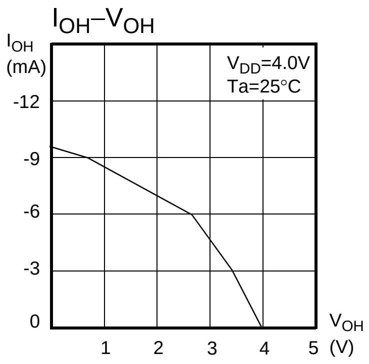
7.7 Typical Characteristics

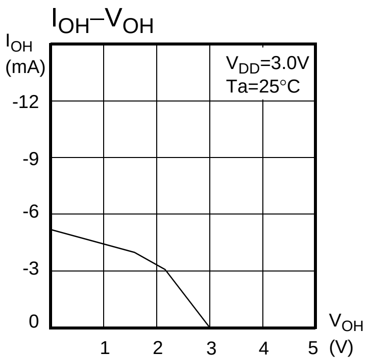
This graphs and tables provided in this section are for design guidance only and are not tested or guaranteed.

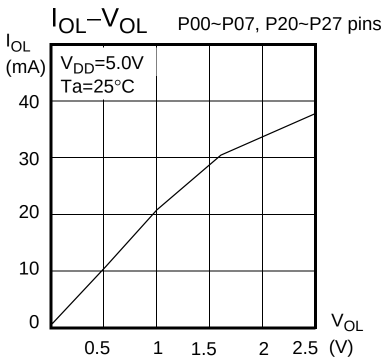
In some graphs or tables the data presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for information only and devices are guaranteed to operate properly only within the specified range.

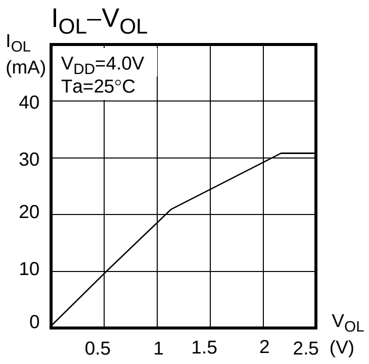
The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean – 3σ) respectively where σ is standard deviation

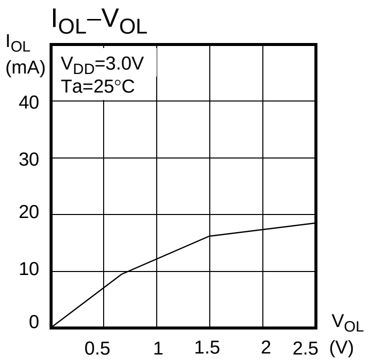


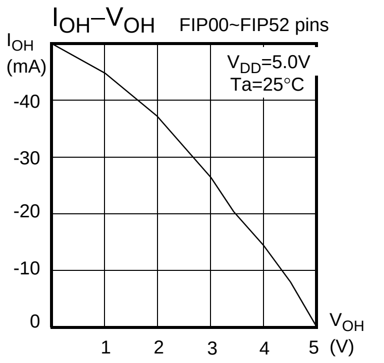


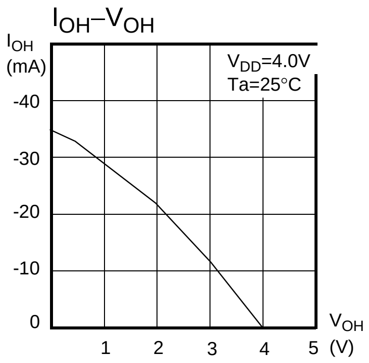


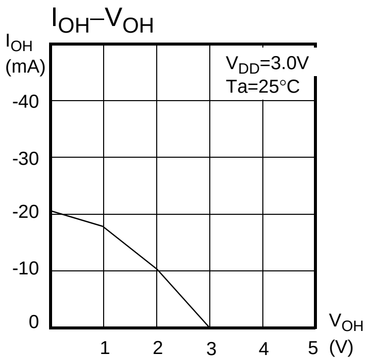




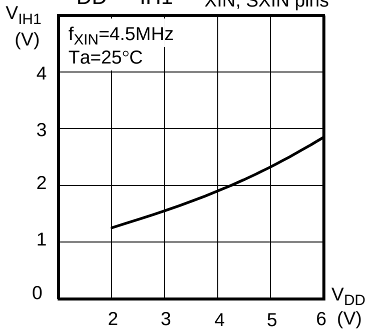




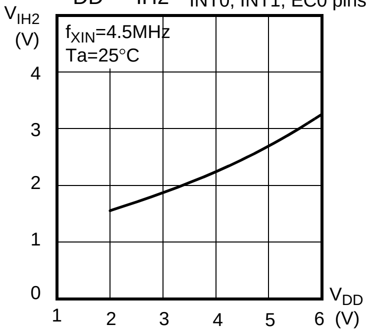




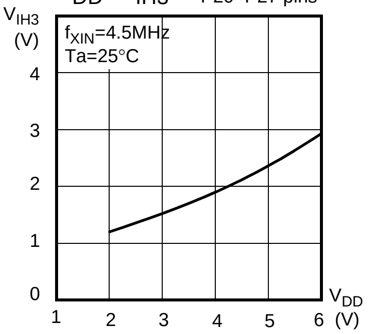
V_{DD}–V_{IH1} XIN, SXIN pins



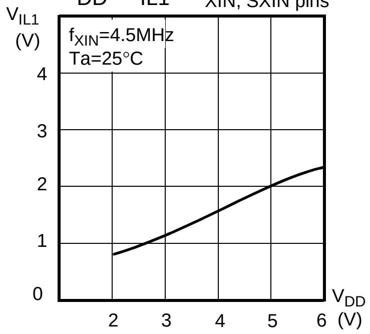
$\overline{\text{RESET}}$, INT0, INT1, EC0 pins V_{DD}–V_{IH2}



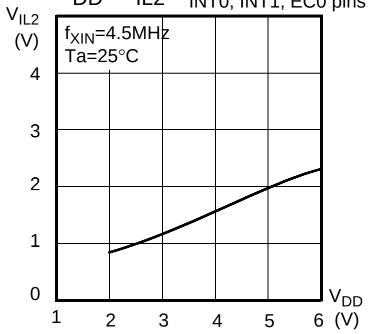
R03~P07, P20~P27 pins V_{DD}–V_{IH3}



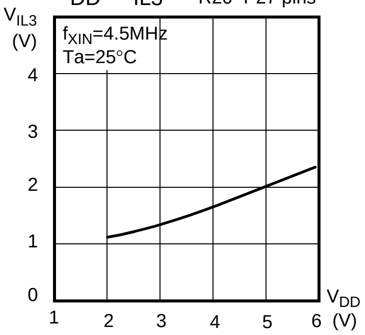
V_{DD}–V_{IL1} XIN, SXIN pins



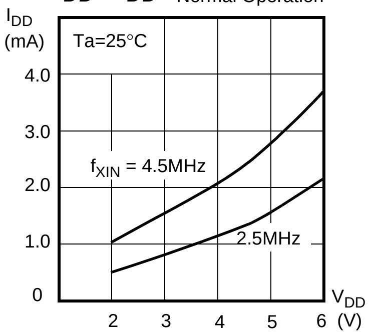
$\overline{\text{RESET}}$, INT0, INT1, EC0 pins V_{DD}–V_{IL2}



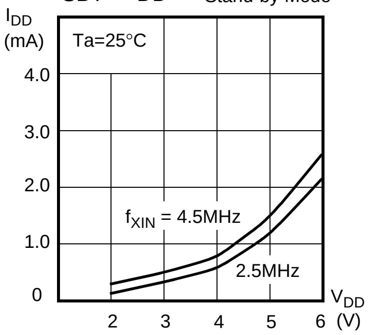
R03~P07, R20~P27 pins V_{DD}–V_{IL3}



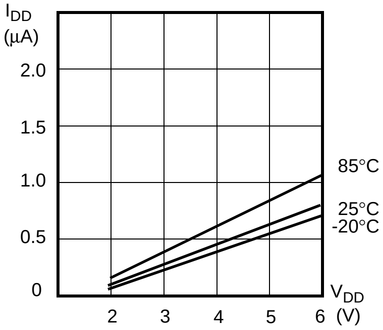
I_{DD}–V_{DD} Normal Operation



I_{SBY}–V_{DD} Stand-by Mode



I_{STOP}–V_{DD} Stop Mode



8. MEMORY ORGANIZATION

The HMS81C2232/48 have separate address spaces for Program memory and Data Memory. Program memory can only be read, not written to. It can be up to 32K/48K bytes of Program memory. Data memory can be read and written to up to 448 bytes including the stack area.

8.1 Registers

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

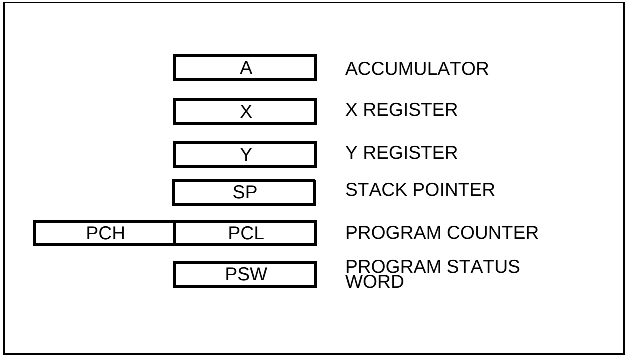


Figure 8-1 Configuration of Registers

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

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

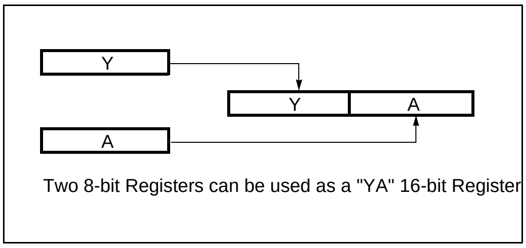


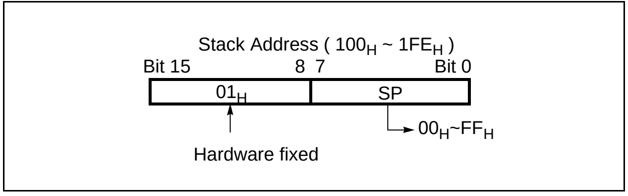
Figure 8-2 Configuration of YA 16-bit Register

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

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

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

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



Note: *The Stack Pointer must be initialized by software because its value is undefined after RESET.*
Example: To initialize the SP

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

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

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

[Carry flag C]

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

[Zero flag Z]

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

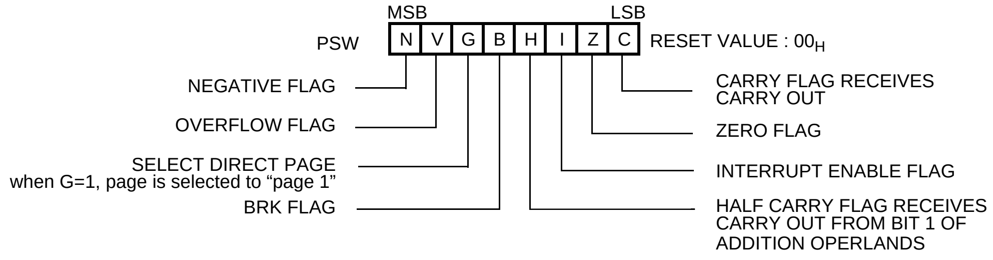

MSB
LSB
PSW
N
V
G
B
H
I
Z
C
RESET VALUE : 00H
NEGATIVE FLAG
OVERFLOW FLAG
SELECT DIRECT PAGE
when G=1, page is selected to “page 1”
BRK FLAG
CARRY FLAG RECEIVES CARRY OUT
ZERO FLAG
INTERRUPT ENABLE FLAG
HALF CARRY FLAG RECEIVES CARRY OUT FROM BIT 1 OF ADDITION OPERLANDS

Figure 8-3 PSW (Program Status Word) Register

[Interrupt disable flag I]

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

[Half carry flag H]

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

[Break flag B]

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

[Direct page flag G]

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

[Overflow flag V]

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

[Negative flag N]

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

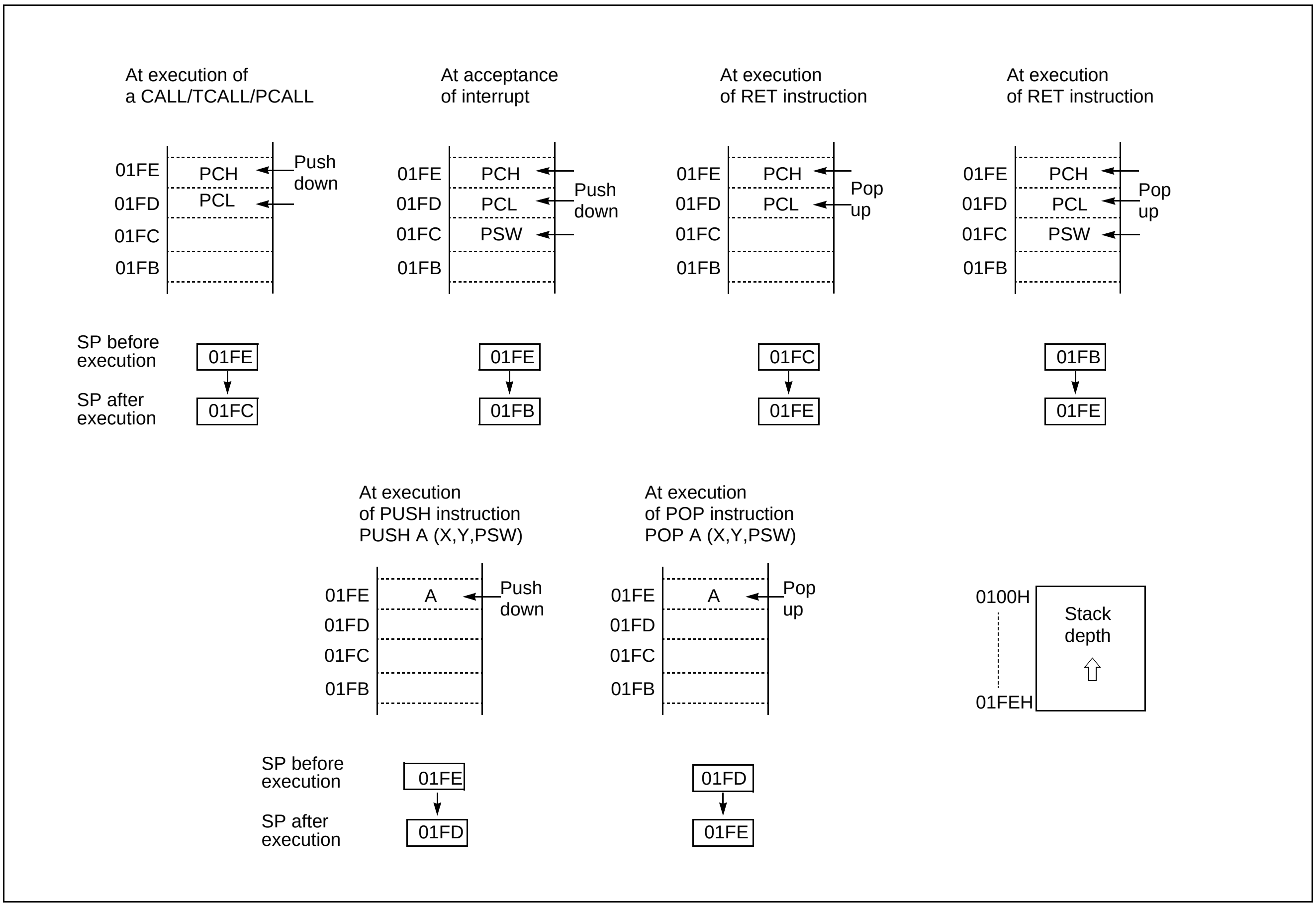


Figure 8-4 Stack Operation

8.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 32/48K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

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

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

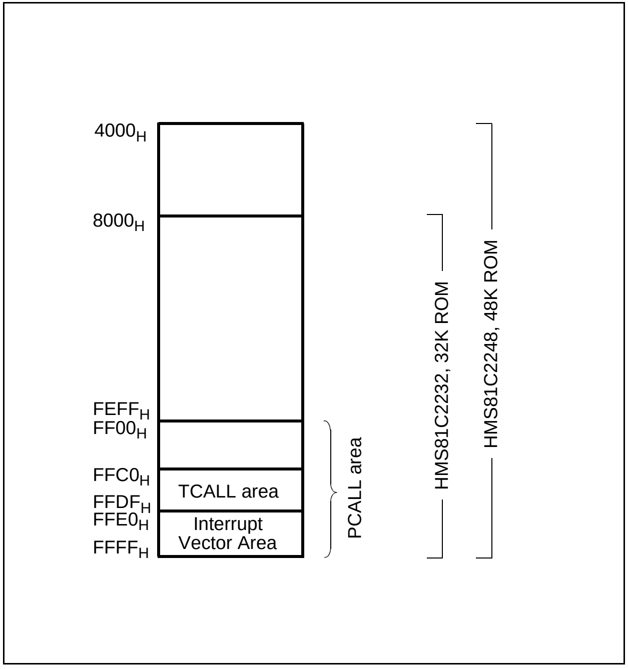


Figure 8-5 Program Memory Map

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

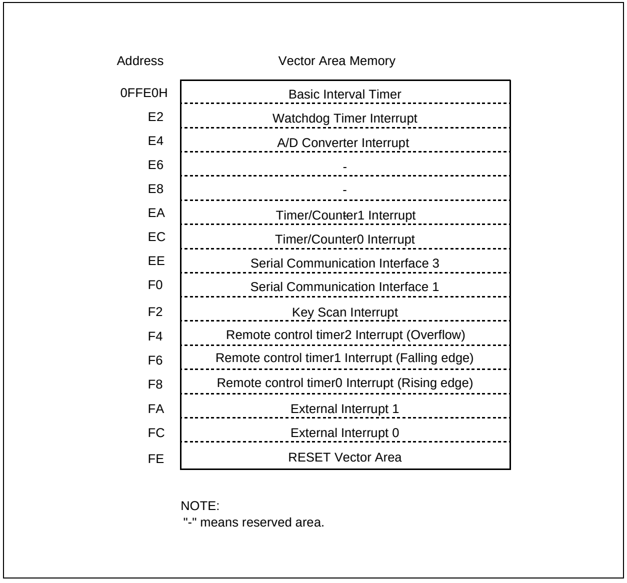
Table Call (TCALL) causes the CPU to jump to each TCALL address, where it commences the execution of the service routine. The Table Call service area spaces 2-byte for every TCALL: $0FFC0_H$ for TCALL15, $0FFC2_H$ for TCALL14, etc., as shown in Figure 8-7.

Example: Usage of TCALL

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

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

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



Address	Vector Area Memory
0FFE0H	Basic Interval Timer
E2	Watchdog Timer Interrupt
E4	A/D Converter Interrupt
E6	-
E8	-
EA	Timer/Counter1 Interrupt
EC	Timer/Counter0 Interrupt
EE	Serial Communication Interface 3
F0	Serial Communication Interface 1
F2	Key Scan Interrupt
F4	Remote control timer2 Interrupt (Overflow)
F6	Remote control timer1 Interrupt (Falling edge)
F8	Remote control timer0 Interrupt (Rising edge)
FA	External Interrupt 1
FC	External Interrupt 0
FE	RESET Vector Area

NOTE:
"-" means reserved area.

Figure 8-6 Interrupt Vector Area

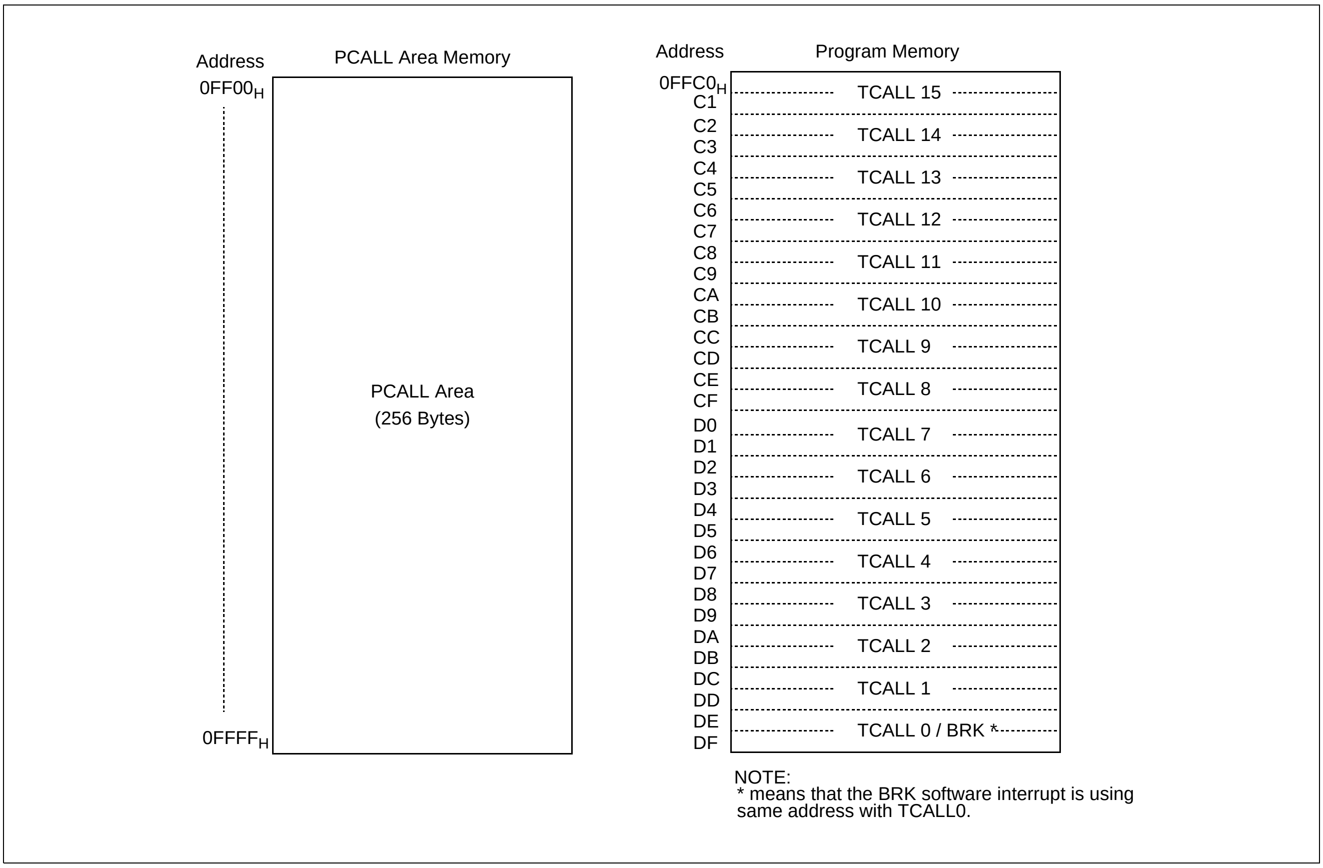


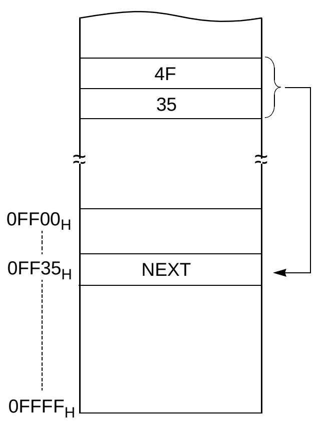
Figure 8-7 PCALL and TCALL Memory Area

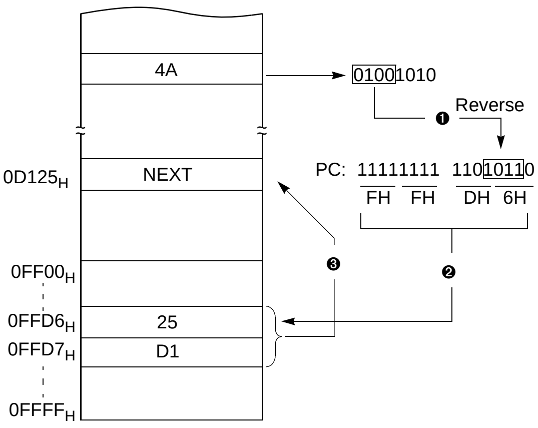
PCALL→ rel

```
4F35     PCALL  35H
```

TCALL→ n

```
4A       TCALL  4
```





Example: The usage software example of Vector address for HMS81C2232/48.

```
;************************************************************************;
;                   Interrupt Vector Table                               ;
;************************************************************************;
                    ORG       0FFE0H
                    DW        Not_Used            ; Basic Interval Timer
                    DW        Not_Used            ; Watch Dog Timer
                    DW        Not_Used            ; A/D Converter
                    DW        Not_Used            ; Not Usedr
                    DW        Not_Used            ; Not Used
                    DW        Timer1_ISR          ; Timer1
                    DW        Not_Used            ; Timer0
                    DW        Not_Used            ; SIO3
                    DW        SPI_ISR             ; SIO1
                    DW        Not_Used            ; Key Scan Interrupt
                    DW        Not_Used            ; Remote Timer2(Overflow)
                    DW        Not_Used            ; Remote Falling Edge
                    DW        Not_Used            ; Remote Rising Edge
                    DW        Not_Used            ; Ext.INT1
                    DW        INT0_ISR            ; Ext.INT0
                    DW        Reset               ; Reset
;************************************************************************;
;                   Program Initial Part                                 ;
;************************************************************************;
                    ORG04000h             ;HMS81C2248/2348 Program Start
Reset:
                    DI                     ;Disable All Interrupt
                    ;=====================;
                    ;  RAM Clear Routine  ;
                    ;=====================;
                    LDX    #0
                    LDY    #0
RAM_Clear0:
                    LDA    #0             ;Page0 RAM Clear(0000h ~ 00BFh)
                    STA    {X}+
                    CMPX   #0C0h
                    BNE    RAM_Clear0

                    INC    Y
                    STY    !RPR           ;Page Select
                    SETG

                    LDX    #0
RAM_Clear1:
                    LDA    #0
                    STA    {X}+
                    CMPX   #00h
                    BNE    RAM_Clear1

                    INC    Y
                    CMPY   #5             ;Page1~4 RAM Clear(0100h ~ 03FFh)
                    BCS    RAM_Clear_Bye

                    STY    !RPR
                    SETG                          ;
                    BRA    RAM_Clear1
RAM_Clear_Bye:
                    CLRG                          ;Page0 Select
                    LDX    #0FFh                 ;Initial Stack Pointer
                    TXSP

                    CALL      Initial_IO          ;I/O Port Initial
                    CALL      Initial_Reg         ;Register Initial
                    EI                            ;Enable Interrupt
;******************************************************************************;
;                   Main Program Part                                          ;
;******************************************************************************;
Main:
                    BRA    Main
```

8.3 Data Memory

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

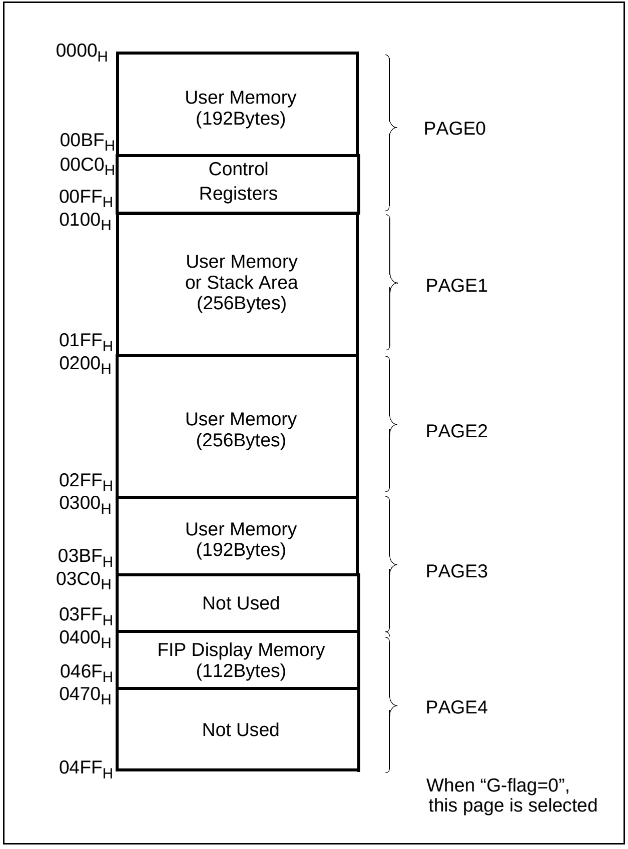


Figure 8-8 Data Memory Map

User Memory

The HMS81C2232/48 has 896 × 8 bits for the user memory (RAM). RAM pages are seleted by RPR.

Note: *After setting RPR(RAM Page Slect Register), be sure to execute SETG instruction. When executing CLRG instruction, be selected PAGE0 regardless of RPR.*

Control Registers

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

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

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

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

Example; To write at CKCTLR

```
        LDM    CLCTLR,#09H ;Divide ratio(÷16)
```

Stack Area

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

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

The save/restore locations in the stack are determined by the stack pointed (SP). The SP is automatically decreased after the saving, and increased before the restoring. This means the value of the SP indicates the stack location number for the next save. Refer to Figure 8-4 on page 28.

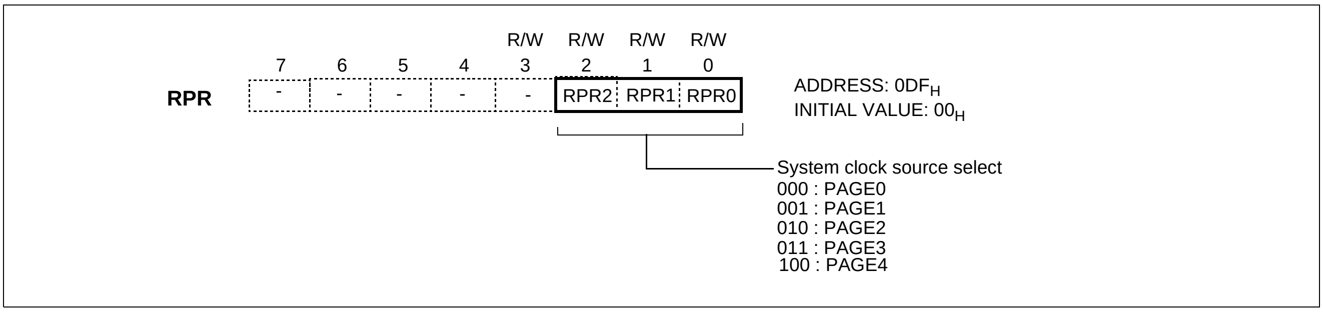


Figure 8-9 RPR(RAM Page Select Register)

8.4 Control Registers

Address	Symbol	Register Name	R/W	Reset Value								Addressing Mode
				7	6	5	4	3	2	1	0	
00C0	P0	Port0 data register	R/W	Undefined								Byte,Bit
00C1	P0IO	Port0 I/O direction register	W	0	0	0	0	0	0	0	0	Byte
00C2	-	Reserved	-	-								-
00C3	-	Reserved	-	-								-
00C4	P2	Port2 data register	R/W	Undefined								Byte,Bit
00C5	P2IO	Port2 I/O direction register	W	0	0	0	0	0	0	0	0	Byte
00C6	P3	Port3 data register	R/W	Undefined								Byte,Bit
00C7	-	Reserved	-	-								-
00C8	P4	Port4 data register	R/W	Undefined								Byte,Bit
00C9	-	Reserved	-	-								-
00CA	P5	Port5 data register	R/W	Undefined								Byte,Bit
00CB	-	Reserved	-	-								-
00CC	P6	Port6 data register	R/W	Undefined								Byte,Bit
00CD	-	Reserved	-	-								-
00CE	-	Reserved	-	-								-
00CF	-	Reserved	-	-								-
00D0	T0CR	Timer0 mode control register	R/W	-	-	0	0	0	0	0	0	Byte,Bit
00D1	T0	Timer0 register	R	0	0	0	0	0	0	0	0	Byte
	T0DR	Timer0 data register	W	1	1	1	1	1	1	1	1	Byte
	CDR0	Timer0 capture data register	R	0	0	0	0	0	0	0	0	Byte
00D2	T1CR	Timer1 mode control register	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00D3	T1DR	Timer1 data register	W	1	1	1	1	1	1	1	1	Byte
	T1PPR	Timer1 PWM period register	W	1	1	1	1	1	1	1	1	Byte
00D4	T1	Timer1 register	R	0	0	0	0	0	0	0	0	Byte
	T1PDR	Timer1 PWM duty register	R/W	0	0	0	0	0	0	0	0	Byte,Bit
	CDR1	Timer1 capture data register	R	0	0	0	0	0	0	0	0	Byte
00D5	PWM1HR	Timer1 PWM high register	W	-	-	-	-	0	0	0	0	Byte
00D6	-	Reserved	-	-								-
00D7	-	Reserved	-	-								-
00D8	-	Reserved	-	-								-
00D9	-	Reserved	-	-								-
00DA	-	Reserved	-	-								-
00DB	-	Reserved	-	-								-
00DC	SIO3M	SIO3 mode control register	R/W	0	-	-	0	0	0	0	0	Byte,Bit
00DD	SIO3R	SIO3 Data shift register	R/W	Undefined								Byte,Bit

Table 8-1 Control Registers

Address	Symbol	Register Name	R/W	Reset Value								Addressing Mode
				7	6	5	4	3	2	1	0	
00DE	BUZ	Buzzer driver register	W	1	1	1	1	1	1	1	1	Byte
00DF	RPR	RAM page selection register	R/W	-	-	-	-	-	0	0	0	Byte,Bit
00E0	SIO1M	SIO1 mode control register	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E1	SIO1R	SIO1 Data shift register	R/W	Undefined								Byte,Bit
00E2	IENH	Interrupt enable register high	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E3	IENL	Interrupt enable register low	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E4	IRQH	Interrupt request flag register high	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E5	IRQL	Interrupt request flag register low	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E6	IEDS	External interrupt edge selection register	W	0	0	0	0	0	0	0	0	Byte,Bit
00E7	RTCR	Remote Timer control register	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00E8	RT	Remote Timer register	R	0	0	0	0	0	0	0	0	Byte
	RTDR	Remote Timer data register	W	1	1	1	1	1	1	1	1	Byte
	RTCP0	Remote Timer capture register0	R	0	0	0	0	0	0	0	0	Byte
00E9	RTCP1	Remote Timer capture register1	R	0	0	0	0	0	0	0	0	Byte
00EA	ADCM	A/D converter mode register	R/W	-	0	0	0	0	0	0	1	Byte
00EB	ADR	A/D converter data register	R	Undefined								Byte
00EC	BITR	Basic interval timer register	R	Undefined								Byte
	CKCTR	Clock control register	W	-	0	0	1	0	1	1	1	Byte
00ED	WDTR	Watchdog Timer Register	R	0	0	0	0	0	0	0	0	Byte
			W	0	1	1	1	1	1	1	1	Byte
00EE	-	Reserved	-	-								-
00EF	PFDR	Power fail detection register	R/W	-	-	-	-	-	1	0	0	Byte
00F0	DSPM0	Display mode register0	R/W	0	0	0	1	0	0	0	0	Byte,Bit
00F1	DSPM1	Display mode register1	R/W	0	0	0	0	0	0	0	1	Byte,Bit
00F2	DSPM2	Display mode register2	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00F3	-	Reserved	-	-								-
00F4	PSR	Port selection register	W	-	-	-	0	0	0	0	0	Byte
00F5	-	Reserved	-	-								-
00F6	-	Reserved	-	-								-
00F7	-	Reserved	-	-								-
00F8	-	Reserved	-	-								-
00F9	-	Reserved	-	-								-
00FA	SCMR	System clock mode register	R/W	-	-	-	0	0	0	0	0	Byte,Bit
00FB	SMR	Sleep mode register	W	-	-	-	-	-	-	-	0	Byte
00FC	PU0	Pull-up resistor option register0	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00FD	-	Reserved	-	-								-

Table 8-1 Control Registers

Address	Symbol	Register Name	R/W	Reset Value								Addressing Mode
				7	6	5	4	3	2	1	0	
00FE	PU2	Pull-up resistor option register2	R/W	0	0	0	0	0	0	0	0	Byte,Bit
00FF	STPC	Stop control register	W	0	0	0	0	0	0	0	0	Byte

Table 8-1 Control Registers

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

- : this bit location is reserved.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0H	P0	Port0 Data Register (Bit[7:0])							
C1H	P0IO	Port0 Direction Register (Bit[7:0])							
C2H	-	Reserved							
C3H	-	Reserved							
C4H	P2	Port2 Data Register (Bit[7:0])							
C5H	P2IO	Port2 Direction Register (Bit[7:0])							
C6H	P3	Port3 Data Register (Bit[7:0])							
C7H	-	Reserved							
C8H	P4	Port4 Data Register (Bit[7:0])							
C9H	-	Reserved							
CAH	P5	Port5 Data Register (Bit[7:0])							
CBH	-	Reserved							
CCH	P6	Port6 Data Register (Bit[7:0])							
CDH	-	Reserved							
CEH	-	Reserved							
CFH	-	Reserved							
D0H	T0CR	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
D1H	T0/T0DR/ CDR0	Timer0 Register/Timer0 Data Register Capture0 Data Register							
D2H	T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
D3H	T1DR 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])**							
D6H	-	Reserved							
D7H	-	Reserved							
D8H	-	Reserved							
D9H	-	Reserved							
DAH	-	Reserved							
DBH	-	Reserved							
DCH	SIO3M	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF
DDH	SIO3R	SPI3 DATA REGISTER							
DEH	BUR	BUCK1	BUCK0	BUR5	BUR4	BUR3	BUR2	BUR1	BUR0
DFH	RPR	RAM Page Selection Register							
E0H	SIO1M	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF

Table 8-2 Control Registers of HMS81C2248

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E1H	SIO1R	SPI1 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
E7H	RTCR	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF
E8H	RT/RTDR RTCP0	Remotae Timer Register/Remote Timer Data Registor Remote Timer Capture Register0							
E9H	RTCP0	Remote Timer Capture Register1							
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						
EEH	-	Reserved							
EFH	PFDR[2]	-	-	-	-	-	PFDIS	PFDM	PFDS
F0H	DSPM0					IED1H	IED1L	IED0H	IED0L
F1H	DSPM1	POL	IOSW	SM1	SM0	SCK1	SCK0	SIOST	SIOSF
F2H	DSPM2					IED1H	IED1L	IED0H	IED0L
F3H	-	Reserved							
F4H	PSR	-	-	-	-	BUZO	EC0	INT1	INT0
F5H	-	Reserved							
F6H	-	Reserved							
F7H	-	Reserved							
F8H	-	Reserved							
F9H	-	Reserved							
FAH	SCMR	-	-	-	-	-	-	CS1	CS0
FBH	-	Reserved							
FCH	PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00
FDH	-	Reserved							
FEH	PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20
FFH	STPC	Stop Control Register							

Table 8-2 Control Registers of HMS81C2248

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

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

8.5 Addressing Mode

The GMS800 series MCU uses six addressing modes;

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

(1) Register Addressing

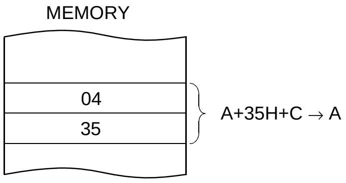
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

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

Example:

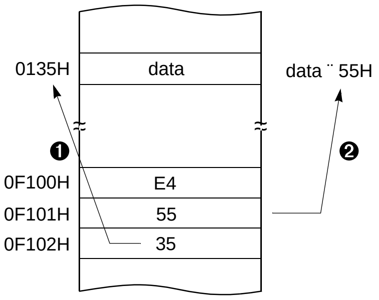
```
0435     ADC    #35H
```



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

Example: G=1

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

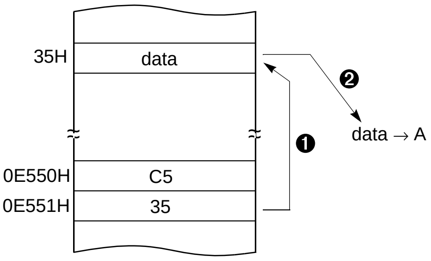


(3) Direct Page Addressing → dp

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

Example; G=0

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



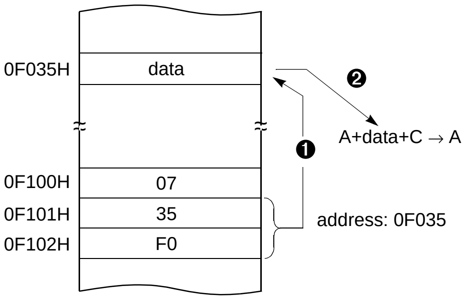
(4) Absolute Addressing → !abs

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

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

Example;

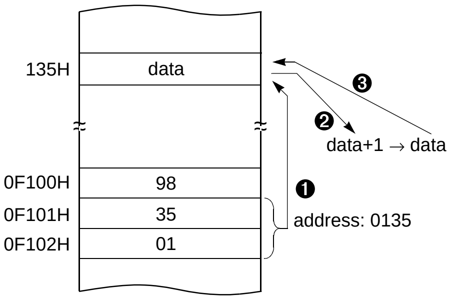
```
0735F0   ADC    !0F035H      ;A ←ROM[0F035H]
```



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

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

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



(5) Indexed Addressing

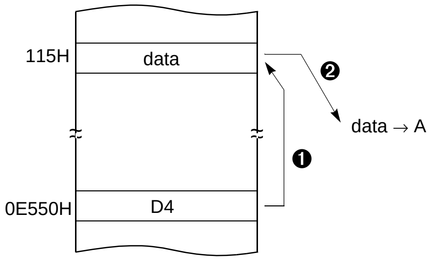
X indexed direct page (no offset) → {X}

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

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

Example; X=15_H, G=1

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



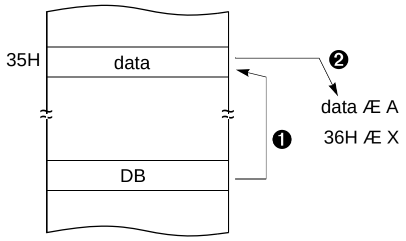
X indexed direct page, auto increment→ {X}+

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

LDA, STA

Example; G=0, X=35_H

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



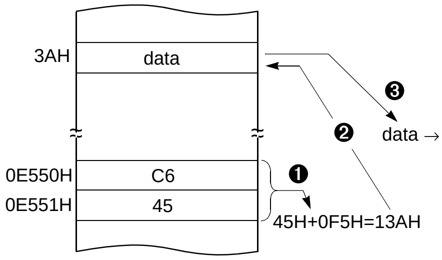
X indexed direct page (8 bit offset) → dp+X

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

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

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

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



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

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

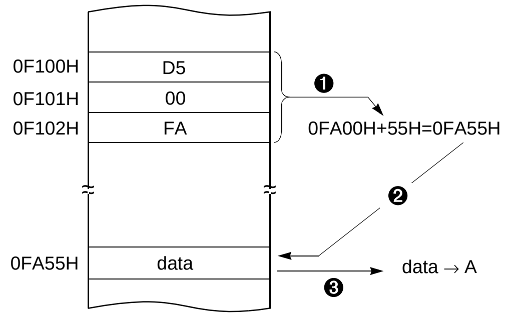
This is same with above (2). Use Y register instead of X.

Y indexed absolute → !abs+Y

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

Example; Y=55_H

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



(6) Indirect Addressing

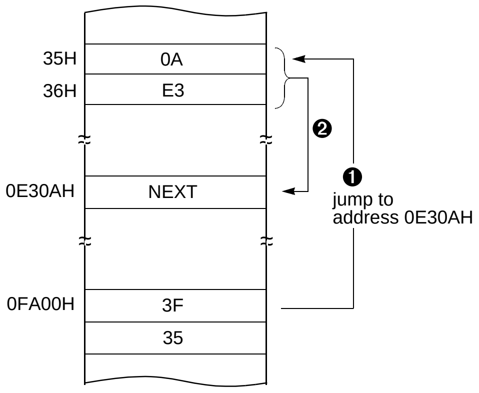
Direct page indirect → [dp]

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

JMP, CALL

Example; G=0

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



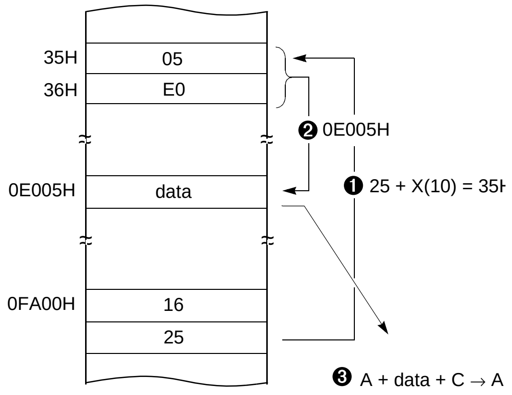
X indexed indirect → [dp+X]

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

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

Example; G=0, X=10_H

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



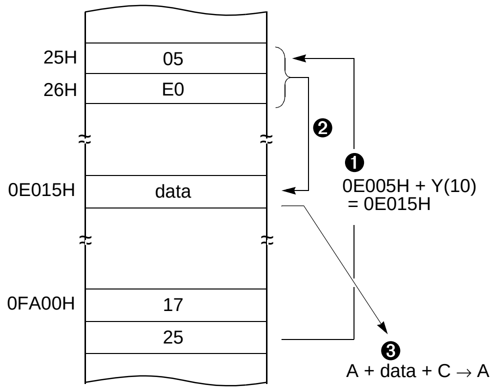
Y indexed indirect → [dp]+Y

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

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

Example; G=0, Y=10_H

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



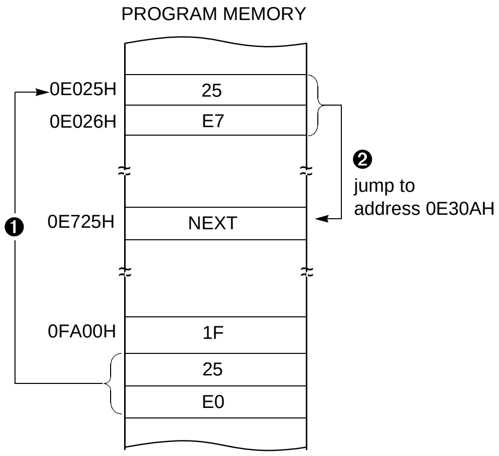
Absolute indirect → [!abs]

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

JMP

Example; G=0

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



9. I/O PORTS

The HMS81C2232/48 has six ports (P0, P2, P3, P4, P5, and P6). These ports pins may be multiplexed with an alternate function for the peripheral features on the device.

The HMS81C2232/48 incorporates 16 output ports and 24 input/output ports. Every port is capable of 1-bit and 8-bit manipulations and can carry out considerably varied control operations. Besides port functions, the ports can also serve as built-in hardware input/output pins.

P0 and P2 port have data direction registers which can define these ports as output or input. A “1” in the port direction register configure the corresponding port pin as output. Conversely, write “0” to the corresponding bit to specify it as input pin. For example, to use the even numbered bit of P0 as output ports and the odd numbered bits as input ports, write “55$_H$” to address 0C1$_H$ (P0 port direction register) during initial setting as shown in Figure 9-1.

All the port direction registers in the HMS81C2232/48 have 0 written to them by reset function. On the other hand, its initial status is input.

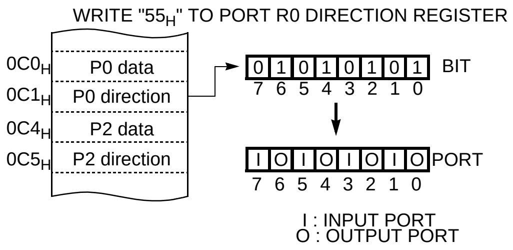


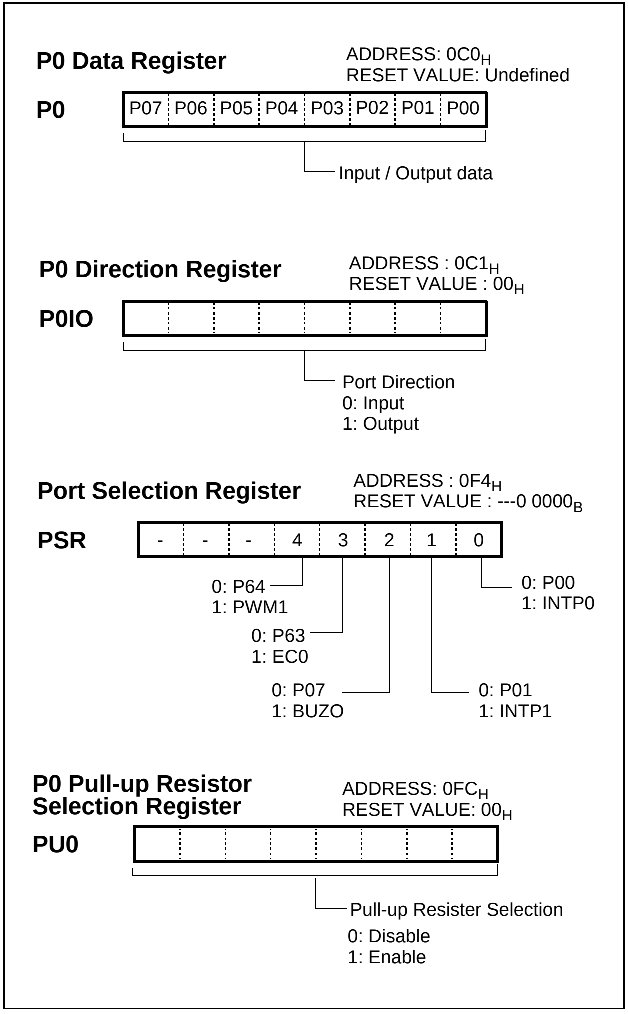
Figure 9-1 Example of Port I/O Assignment

9.1 P0 and P0IO register:

P0 is an 8-bit high-voltage CMOS bidirectional I/O port (address 0C0$_H$). Each port can be set individually as input and output through the P0IO register (address 0C1$_H$).

Port pin	Alternate function
P00	INTP0 (External interrupt 0)
P01	INTP1 (External interrupt 1)
P02	TI (Timer input of remote control timer)
P03	AN0 (Analog Input 0)
P04	AN1 (Analog Input 1)
P05	AN2 (Analog Input 2)
P06	AN3 (Analog Input 3)
P07	BUZO (Buzzer driver output)

When P00 through P07 pins are used as input ports, an on-chip pull-up resistor can be connected to them in 1-bit units with a pull-up resistor option register 0 (PU0). Alternate functions include external interrupt request input and timer input. RESET input sets port 0 to input mode.



P00~P01 ports are multiplexed with External Interrupt Input Port(INTP1, INTP0), and P02 port is multiplexed with Event Counter Input Port (EC0). P03~P06 ports are multiplexed with Analog Input Port and P07 port is multiplexed with Buzzer Output Port(BUZO).

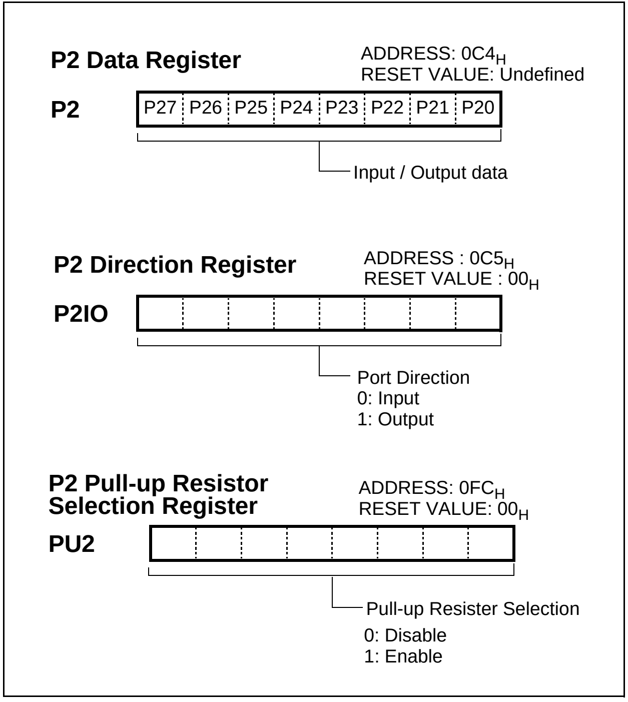
.The control register PSR (address F4$_H$) controls to select alternate function. After reset, this value is "0", port may be used as general I/O ports. To select alternate function such as Buzzer Output, External Event Counter Input and External Interrupt Input, write "1" to the corresponding bit of PSR. Regardless of the direction register P0IO, PSR is selected to use as alternate functions, port pin can be used as a corresponding alternate features (BUZO, EC0, INT1, INT0)

9.2 P2 and P2IO register:

P2 is an 8-bit high-voltage CMOS bidirectional I/O port (address $0C4_H$). Each port can be set individually as input and output through the P2IO register (address $0C5_H$).

Port pin	Alternate function
P20	SCK3 (Serial3 clock input/output)
P21	SO3 (Serial3 data output)
P22	AN4 (Analog Input 4)
P23	AN5 (Analog Input 5)
P24	AN6 (Analog Input 6)
P25	SO1 (Serial1 data output)
P26	SI1 (Serial1data input)
P27	SCK1 (Serial1 clock input/output)

When P20 to P27 pins are used as input ports, an on-chip pull-up resistor can be connected to them in 1-bit units with a pull-up resistor option register 2 (PU2). Alternate functions include serial interface data input/output and clock input/output. RESET input sets port 2 to input mode.



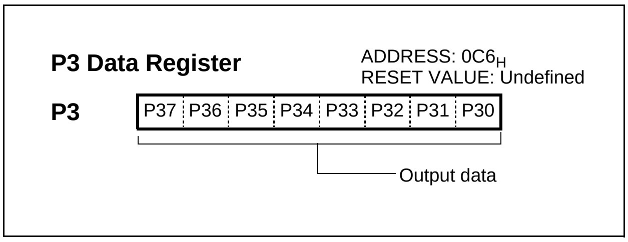
P20 to P21 port is multiplexed with serial interface3 data input/output(SO3), clock input/output(SCK3). P22~P24 ports are multiplexed with Analog Input Port (ANI4~ANI6). P25~P27 port is multiplexed with serial interface1 data input(SI1)/output(SO1), clock input/output(SCK1).

9.3 P3 register:

P3 is an 8-bit output only port(address $0C6_H$). On-chip pull-down resistors can be connected in 1-bit units with the mask option in case of mask ROM model. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver segment/digit output is provided as an alternate function.

Port pin	Alternate function
P30~P37	FIP24-FIP31

On-chip pull-down resistors can be connected in 1-bit units with the mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver output is provided as an alternate function.

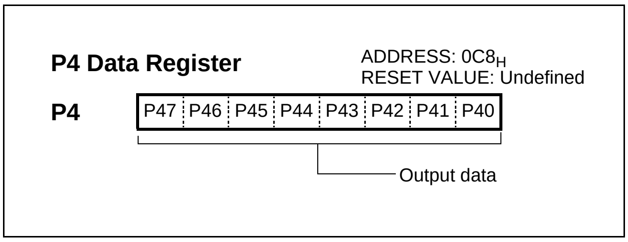


9.4 P4 register:

P4 is an 8-bit output only port(address $0C8_H$). On-chip pull-down resistors can be connected in 1-bit units with the mask option in case of mask ROM model. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver segment/digit output is provided as an alternate function.

Port Pin	Alternate Function
P40~P47	FIP32-FIP39

On-chip pull-down resistors can be connected in 1-bit units with the mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver output is provided as an alternate function.

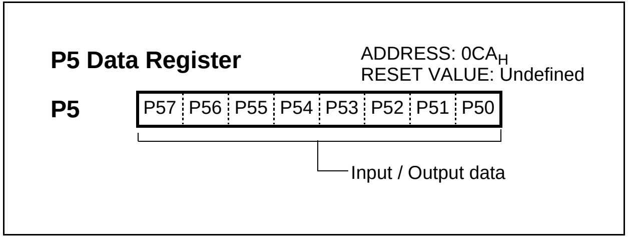


9.5 P5 register:

P5 is an 8-bit bidirectional I/O port (address $0CA_H$). Port 5 is an 8-bit input/output port with output latch. When using this port as an output port, the value assigned to the output latch (P50 through P57) is output. When it is used as an input port, set the output latch (P50 through P57) to “0”, and read the port level read (P50 through P57).

Port Pin	Alternate Function
P50~P57	FIP40-FIP47

On-chip pull-down resistors can be connected in 1-bit units with the mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver output is provided as an alternate function. RESET input sets port 5 to input mode.



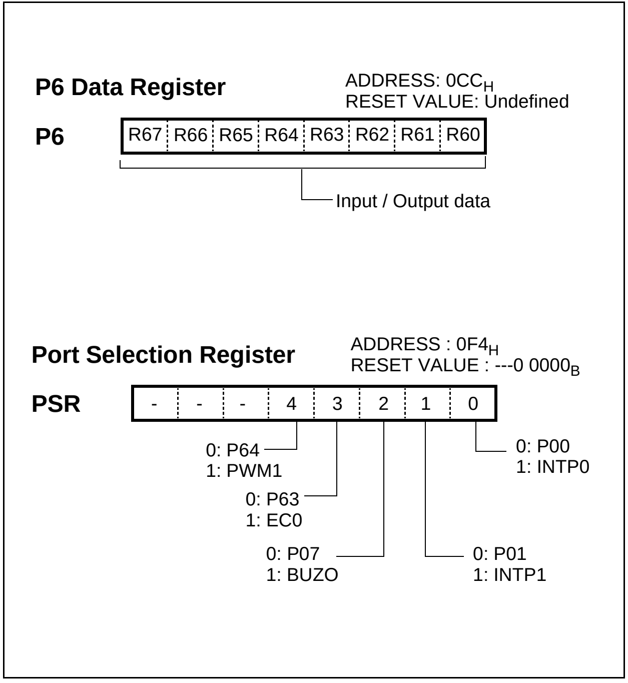
9.6 P6 register:

P6 is an 5-bit bidirectional I/O port (address $0CC_H$). Port 64 is multiplexed with Pulse Width Modulator (PWM).

Port 6 is an 5-bit input/output port with output latch. When using this port as an output port, the value assigned to the output latch (P60 through P64) is output. When it is used as an input port, set the output latch (P60 through P64) to “0”, and read the port level read (P60 through P64).

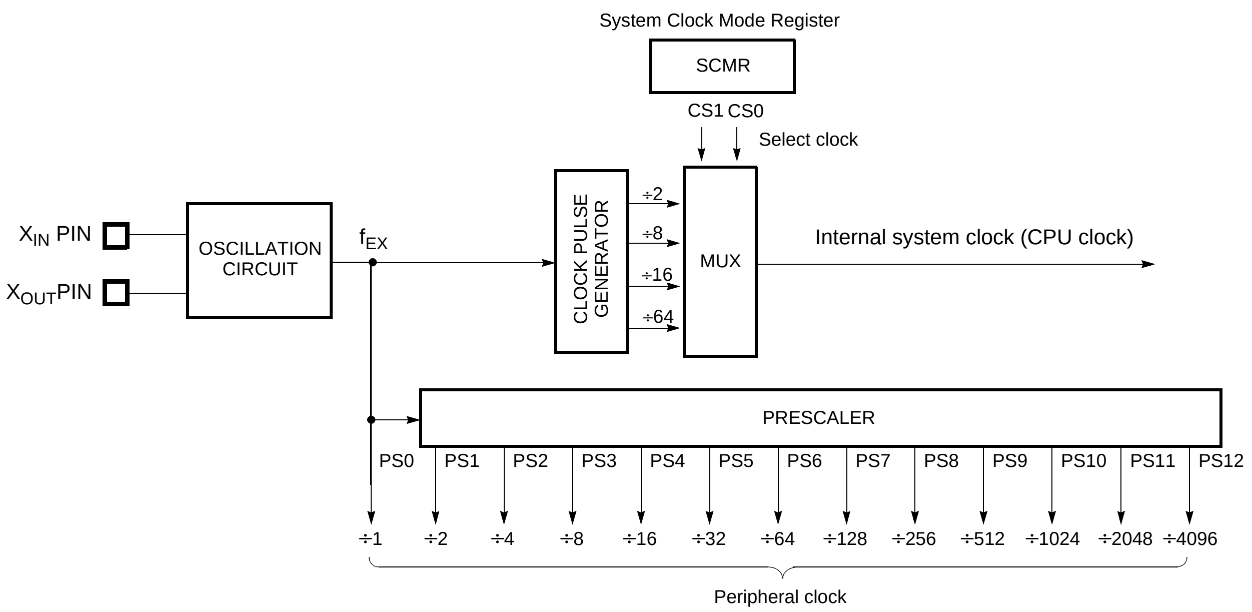
Port pin	Alternate function
P60 P61 P62 P63 P64	FIP48 FIP49 FIP50 FIP51 / EC0 FIP52 / PWM1O

On-chip pull-down resistors can be connected in 1-bit units with the mask option. Pull-down resistor to V_{DISP} or V_{SS0} can be selected in 1-bit units. The HMS87C2232/48 has no pull-down resistor. In addition, FIP controller/driver output is provided as an alternate function. RESET input sets port 6 to input mode.



10. CLOCK GENERATOR

As shown in Figure 10-1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. 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 Xin and Xout pins. External clocks can be input to the main system clock oscillator. In this case, input a clock signal to the Xin pin and open the Xout pin The system clock can also be obtained from the external oscillator. .



f_{EX}(MHz)		PS0	PS1	PS2	PS3	PS4	PS5	PS6	PS7	PS8	PS9	PS10	PS11	PS12
4	Frequency	4M	2M	1M	500K	250K	125K	62.5K	31.25K	15.63K	7.183K	3.906K	1.953K	976.6
	period	250n	500n	1u	2u	4u	8u	16u	32u	64u	128u	256u	512u	1024u

Figure 10-1 Block Diagram of Clock Generator

The clock generator produces the system clocks forming clock pulse, which are supplied to the CPU and the peripheral hardware. The internal system clock can be selected by bit1, and bit0 of the System Clock Mode Register(SCMR). The register is shown in Figure 10-2.

To the peripheral block, the clock among the not-divided original clocks, divided by 2, 4,..., up to 4096 can be provided. Peripheral clock is enabled or disabled by STOP instruction. On the initial reset, internal system clock is PS1 which is the fastest and other clock can be provided by bit1 and bit0 of SCMR.

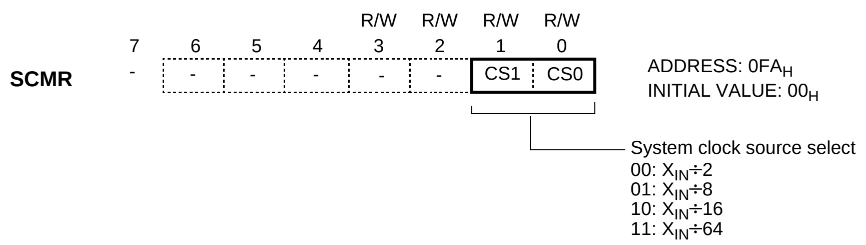


Figure 10-2 System Clock Control Register

11. BASIC INTERVAL TIMER

The HMS81C2232/48 has one 8-bit Basic Interval Timer that is free-run, can not stop. Block diagram is shown in Figure 11-1. In addition, the Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt (BITIF).

The 8-bit Basic interval timer register (BITR) is increased every internal count pulse which is divided by prescaler. Since prescaler has divided ratio by 8 to 1024, the count rate is 1/8 to 1/1024 of the oscillator frequency. As the count overflows from FF_H to 00_H, this overflow causes to generate the Basic interval timer interrupt. The BITIF is interrupt request flag of Basic interval timer. The Basic Interval Timer is controlled by the clock control register (CKCTLR) shown in Figure 11-2.

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

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

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

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

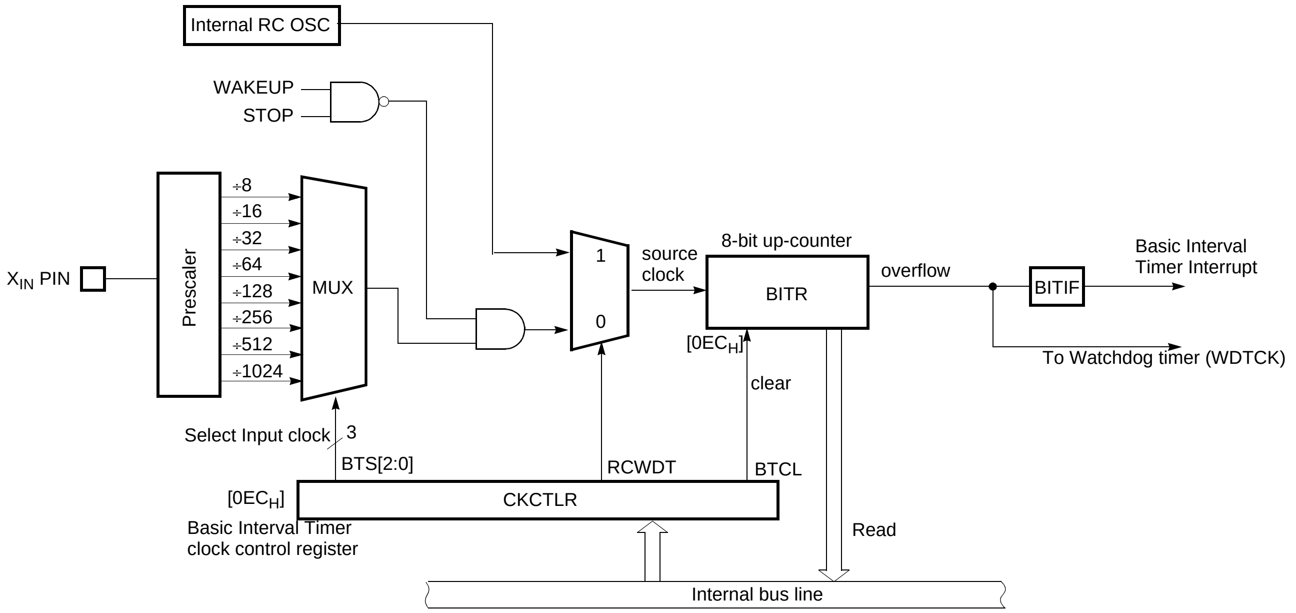
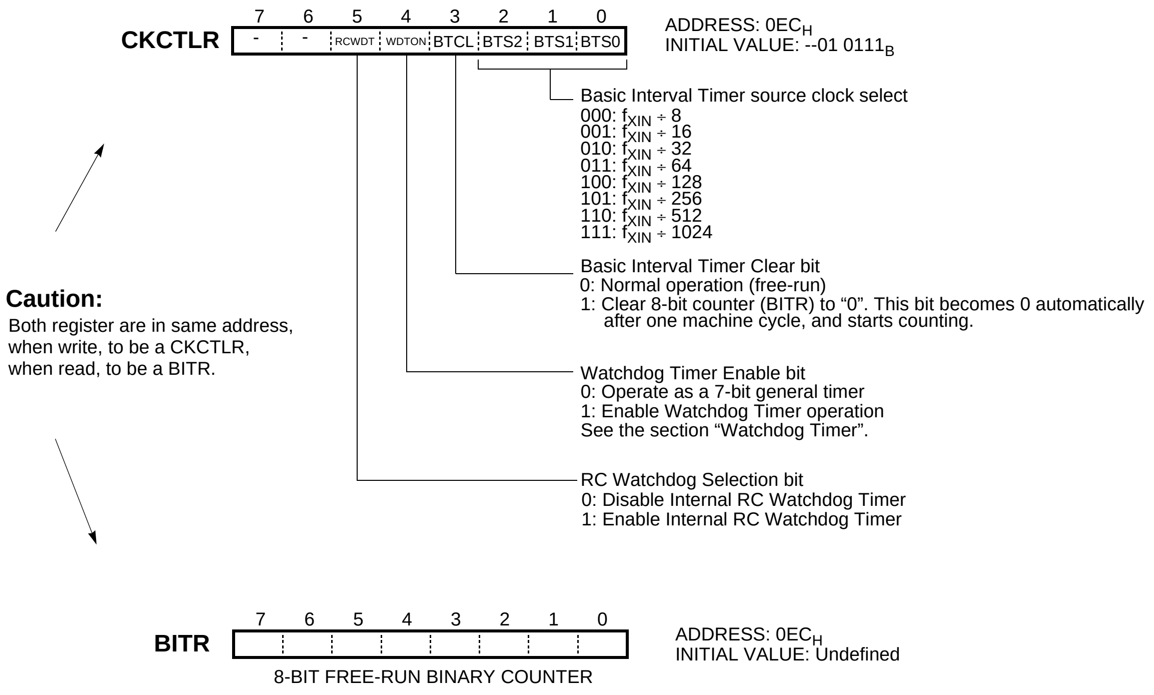


Figure 11-1 Block Diagram of Basic Interval Timer

CKCTLR [2:0]	Source clock	Interrupt (overflow) Period (ms) @ f_{XIN} = 4MHz
000	PS3(f_{XIN}÷8)	0.512
001	PS4(f_{XIN}÷16)	1.024
010	PS5(f_{XIN}÷32)	2.048
011	PS6(f_{XIN}÷64)	4.096
100	PS7(f_{XIN}÷128)	8.192
101	PS8(f_{XIN}÷256)	16.384
110	PS9(f_{XIN}÷512)	32.768
111	PS10(f_{XIN}÷1024)	65.536

Table 11-1 Basic Interval Timer Interrupt Time


CKCTLR
7 6 5 4 3 2 1 0
- - RCWDT WDTON BTCL BTS2 BTS1 BTS0
BITR
7 6 5 4 3 2 1 0
8-BIT FREE-RUN BINARY COUNTER

ADDRESS: $0EC_H$
INITIAL VALUE: --01 0111_B

Basic Interval Timer source clock select
000: f_{XIN} ÷ 8
001: f_{XIN} ÷ 16
010: f_{XIN} ÷ 32
011: f_{XIN} ÷ 64
100: f_{XIN} ÷ 128
101: f_{XIN} ÷ 256
110: f_{XIN} ÷ 512
111: f_{XIN} ÷ 1024

Basic Interval Timer Clear bit
0: Normal operation (free-run)
1: Clear 8-bit counter (BITR) to “0”. This bit becomes 0 automatically after one machine cycle, and starts counting.

Watchdog Timer Enable bit
0: Operate as a 7-bit general timer
1: Enable Watchdog Timer operation
See the section “Watchdog Timer”.

RC Watchdog Selection bit
0: Disable Internal RC Watchdog Timer
1: Enable Internal RC Watchdog Timer

Caution:
Both register are in same address, when write, to be a CKCTLR, when read, to be a BITR.

ADDRESS: $0EC_H$
INITIAL VALUE: Undefined

Figure 11-2 BITR: Basic Interval Timer Mode Register

Example 1:

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

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

Example 2:

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

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

12. WATCHDOG TIMER

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

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

The watchdog timer has two types of clock source.

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

The other type is a prescaled system clock.

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

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

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

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

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

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

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

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.

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

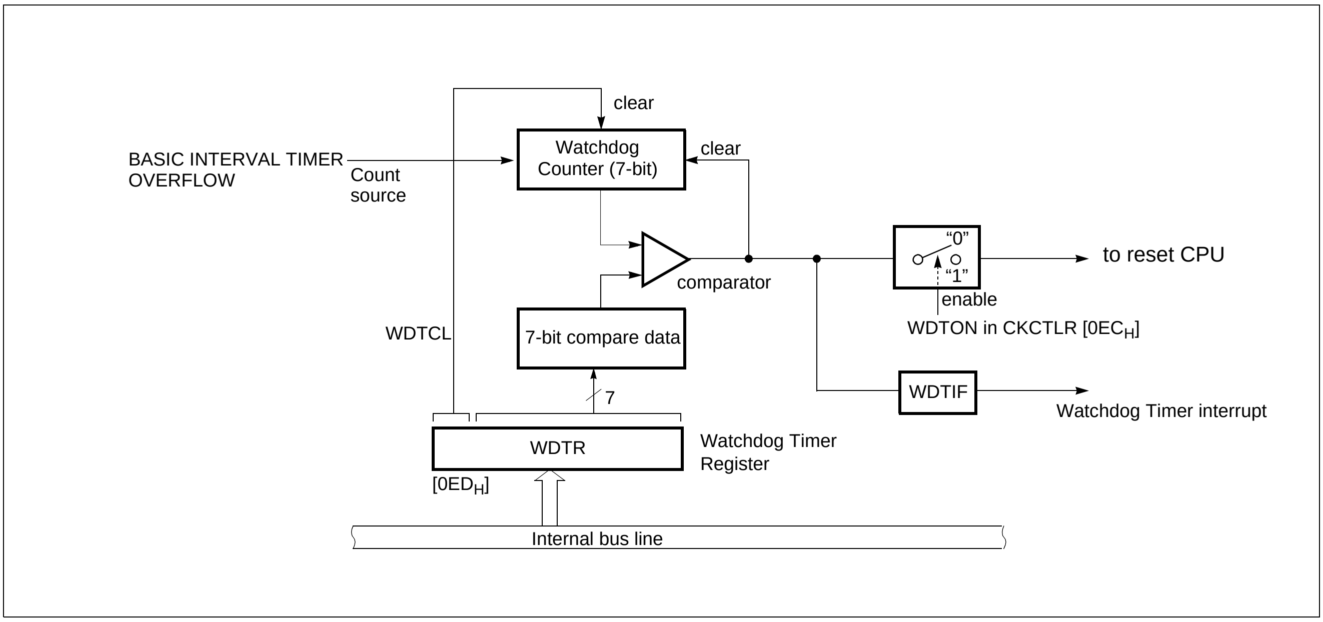


Figure 12-1 Block Diagram of Watchdog Timer

Watchdog Timer Control

Figure 12-2 shows the watchdog timer control register. The watchdog timer is automatically disabled after reset.

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

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

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

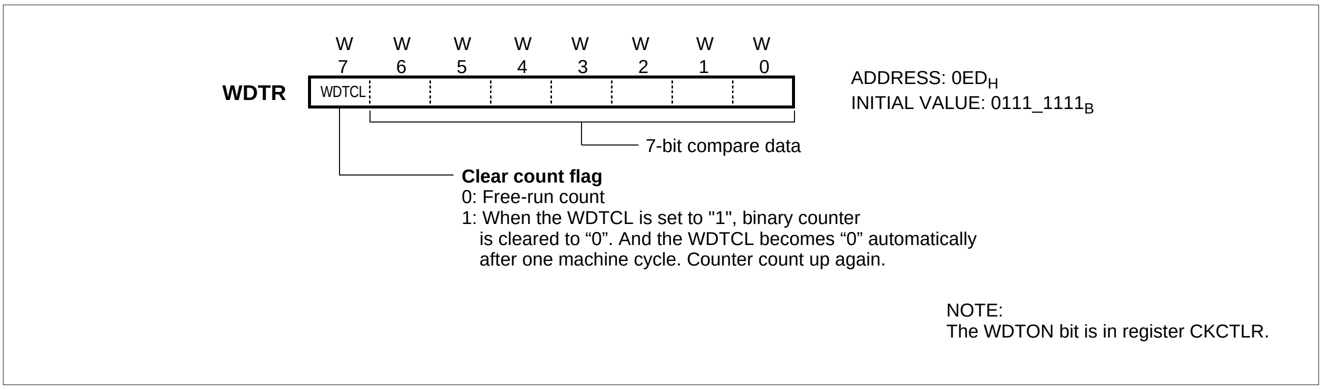

WDTR
W 7
W 6
W 5
W 4
W 3
W 2
W 1
W 0
WDTCL
ADDRESS: $0ED_H$
INITIAL VALUE: 0111_1111_B
7-bit compare data
Clear count flag
0: Free-run count
1: When the WDTCL is set to "1", binary counter is cleared to “0”. And the WDTCL becomes “0” automatically after one machine cycle. Counter count up again.
NOTE:
The WDTON bit is in register CKCTLR.

Figure 12-2 WDTR: Watchdog Timer Data Register

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

```
                LDM     CKCTLR,#3FH       ;Select 1/2048 clock source,  WDTON ← 1, Clear Counter
                LDM     WDTR,#04FH

              ┌─LDM     WDTR,#04FH        ;Clear counter
Within WDT    │  :
detection time│  :
              │  :
              └─LDM     WDTR,#04FH        ;Clear counter
              ┌─ :
Within WDT    │  :
detection time│  :
              └─LDM     WDTR,#04FH        ;Clear counter
```

Enable and Disable Watchdog

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

Example: Enables watchdog timer for Reset

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

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

Watchdog Timer Interrupt

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

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

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

Example: 7-bit timer interrupt set up.

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

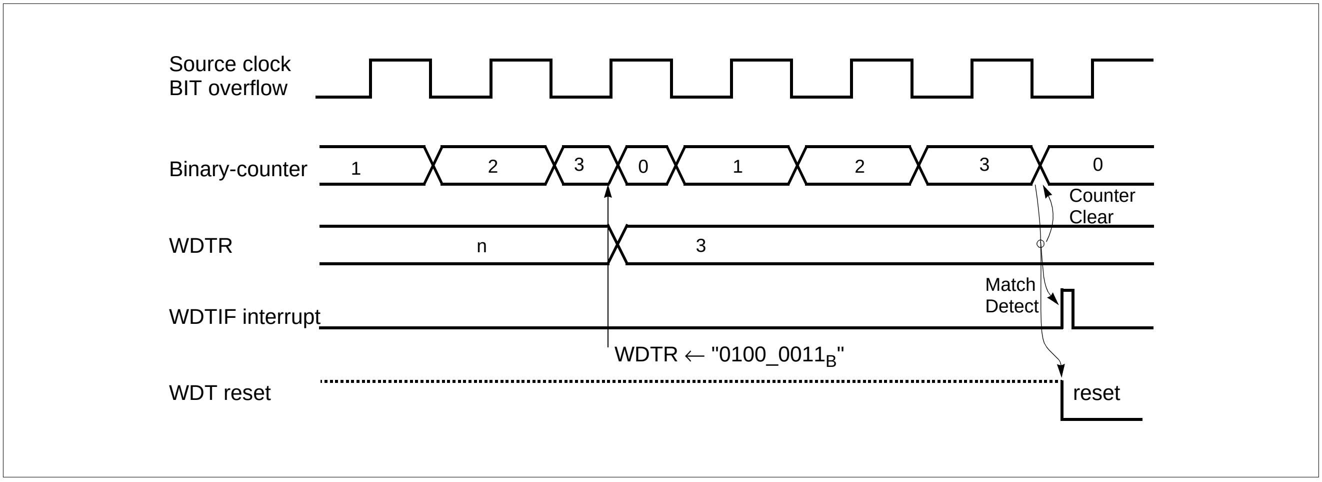


Figure 12-3 Watchdog timer Timing

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

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

13. TIMER/EVENT COUNTER

The HMS81C2232/48 has two Timer/Counter registers. Each module can generate an interrupt to indicate that an event has occurred (i.e. timer match).

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

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

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

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

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

It has seven operating modes: "8-bit timer/counter", "16-bit timer/counter", "8-bit capture", "16-bit capture", "8-bit compare output", "16-bit compare output" and "10-bit PWM" which are selected by bit in Timer mode register TM0 and TM1 as shown in Figure 13-1 and Table 13-1.

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

Table 13-1 Operating Modes of Timer0 and Timer1

T0CR

		R/W	R/W	R/W	R/W	R/W	R/W
		5	4	3	2	1	0
-	-	CAP0	T0Ck2	T0CK1	T0Ck0	T0CN	T0ST

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

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

T1CR

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

ADDRESS: $0D2_H$
INITIAL VALUE: 00_H

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

T0DR

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

ADDRESS: $0D1_H$
INITIAL VALUE: Undefined

T1DR

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

ADDRESS: $0D3_H$
INITIAL VALUE: Undefined

Read: Count value read
Write: Compare data write

Figure 13-1 TM0, TM1 Registers

13.1 8-bit Timer / Counter Mode

The HMS81C2232/48 has two 8-bit Timer/Counters, Timer 0, Timer 1 as shown in Figure 13-2.

The "timer" or "counter" function is selected by mode registers TMx as shown in Figure 13-1 and Table 13-1. To use as an 8-bit timer/counter mode, bit CAP0 of T0CR is cleared to "0" and bits 16BIT of T1CR should be cleared to “0”(Table 13-1).

T0CR

	7	6	5	4	3	2	1	0
T0CR	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
	-	-	0	X	X	X	X	X

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

X means don’t care

T1CR

	7	6	5	4	3	2	1	0
T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	0	0	0	X	X	X	X

ADDRESS: $0D2_H$
INITIAL VALUE: 00_H

X means don’t care

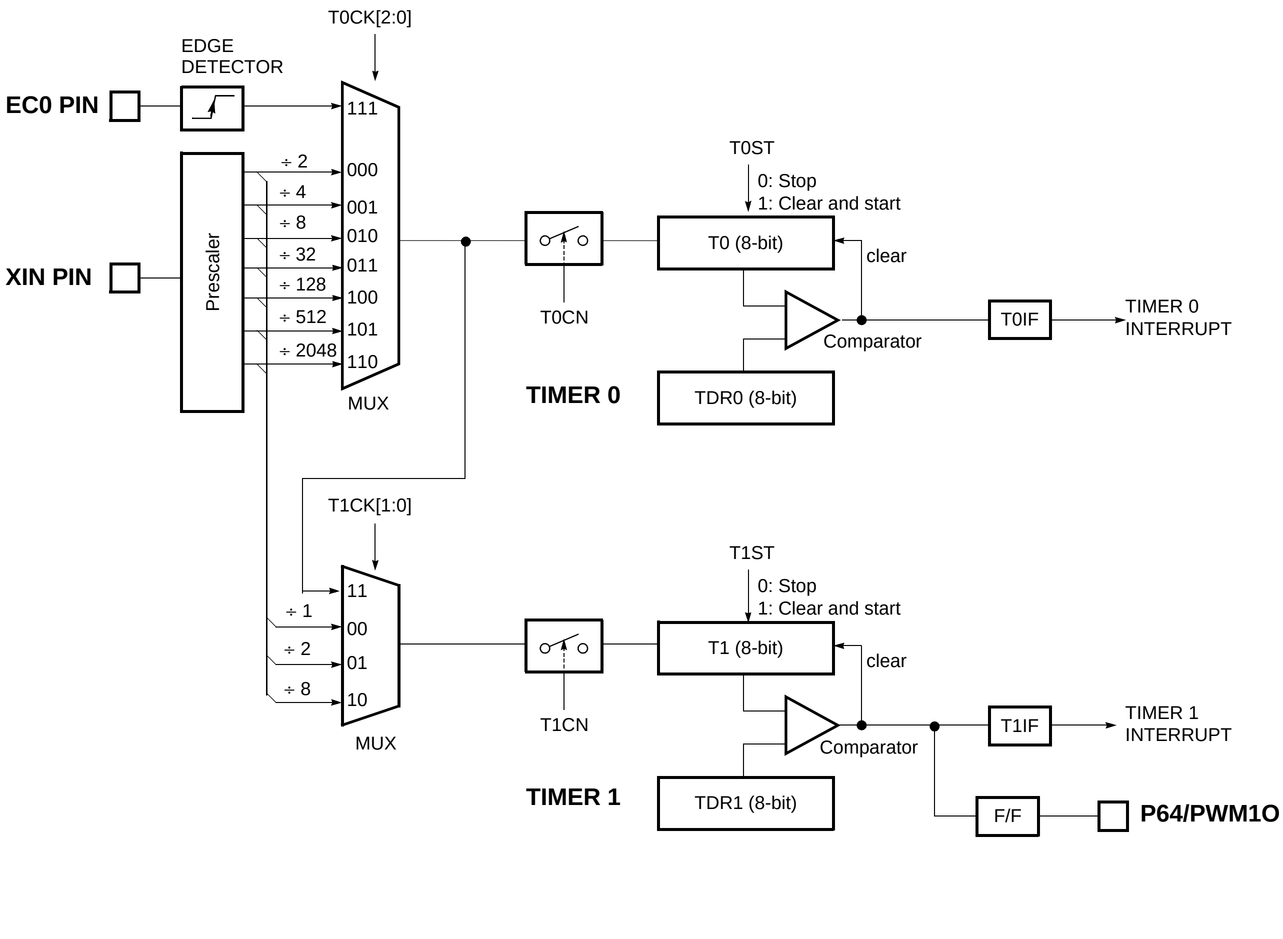


Figure 13-2 8-bit Timer/Counter 0, 1

Example 1:

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

```
        LDM    TDR0,#249
        LDM    TDR1,#249
        LDM    T0CR,#0000_1111B
        LDM    T1CR,#0000_1011B
        SET1   T0E
        SET1   T1E
        EI
```

Example 2:

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

```
        LDM    TDR0,#249
        LDM    TDR1,#249
        LDM    T0CR,#0001_1111B
        LDM    T1CR,#0000_1011B
        SET1   T0E
        SET1   T1E
        EI
```

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

These timers have each 8-bit count register and data register. The count register is increased by every internal or external clock input. The internal clock has a prescaler divide ratio option of 2, 4, 8, 32,128, 512, 2048 selected by control bits T0CK[2:0] of register (T0CR) and 1, 2, 8 selected by control bits T1CK[1:0] of register (T1CR). In the Timer 0, timer register T0 increases from 00_H until it matches T0DR 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 Port Selection Register(PSR.3) is set to "1". The Timer 0 can be used as a counter by pin EC0 input, but Timer 1 can not.

8-bit Timer Mode

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

As the value of TDR*n* is changeable by software, time interval is set as you want

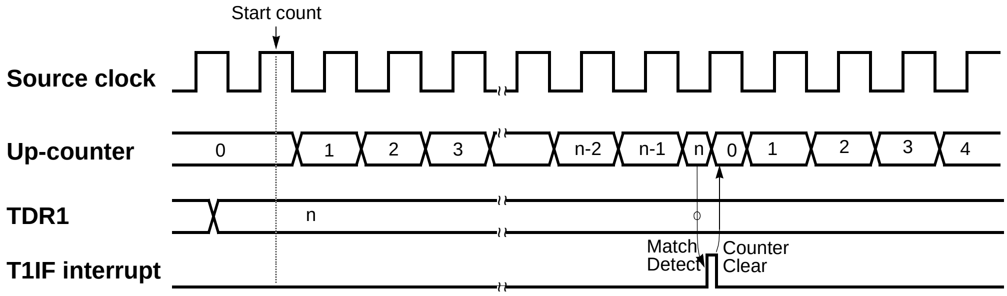


Figure 13-3 Timer Mode Timing Chart

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

```
LDM    T0CR,#0FH   ; divide by 32
LDM    TDR0,#124   ; 8us x (124+1)= 1ms
SET1   T0E         ; Enable Timer 0 Interrupt
EI                 ; Enable Master Interrupt
```

When TM0 = 0000 1111_B (8-bit Timer mode, Prescaler divide ratio = 32)
TDR0 = 124_D = $7C_H$
f_{XIN} = 4 MHz

INTERRUPT PERIOD = $\frac{1}{4 \times 10^6 \text{ Hz}} \times 32 \times (124+1) = 1$ ms

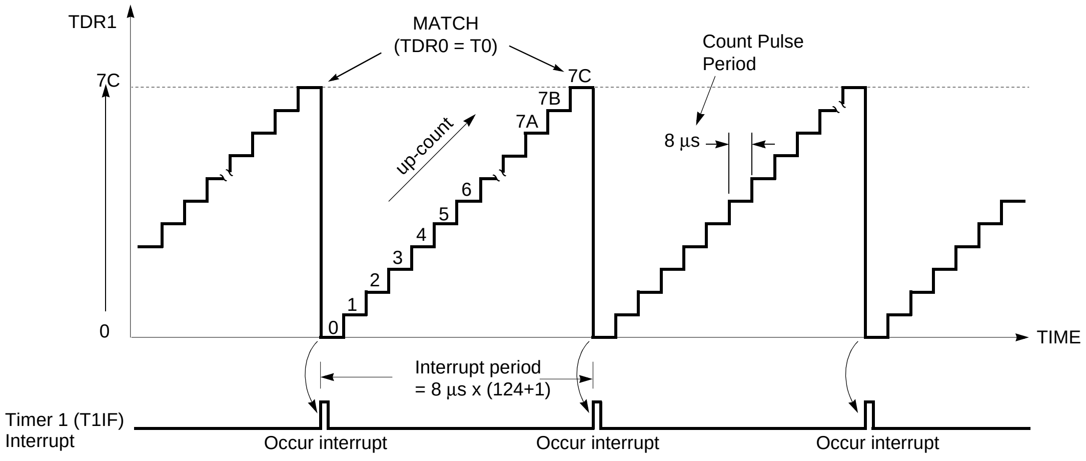


Figure 13-4 Timer Count Example

8-bit Event Counter Mode

In this mode, counting up is started by an external trigger. This trigger means falling edge or rising edge of the EC0 pin input. Source clock is used as an internal clock selected with timer mode register T0CR. The contents of timer data register T0DR is compared with the contents of the up-counter T0. If a match is found, an timer interrupt request flag T0IF is generated, and the counter is cleared to “0”. The counter is restart and count up continuously by every falling edge or rising edge of the EC0 pin input.

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

In order to use event counter function, the bit 3 of thePort Selection Register(PSR.3) is required to be set to “1”.

After reset, the value of timer data register T0DR is undefined, it should be initialized to between 0_H~FE_H, not to "0"The interval period of Timer is calculated as below equation.

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

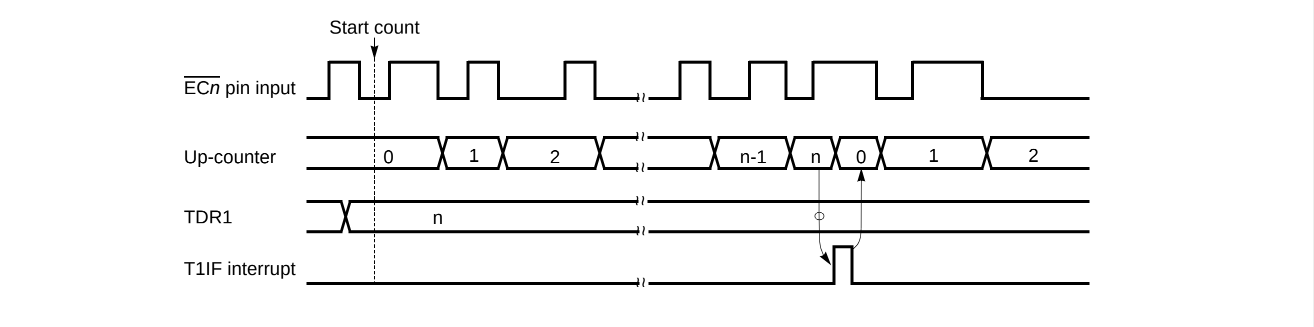


Figure 13-5 Event Counter Mode Timing Chart

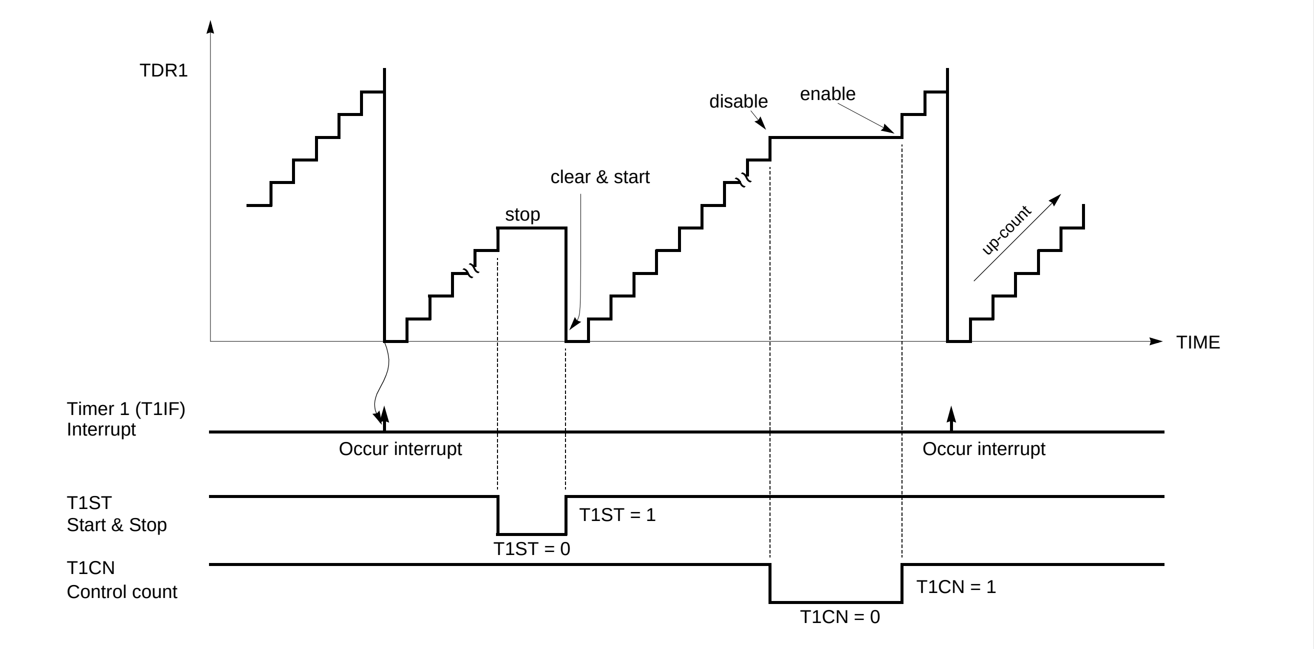


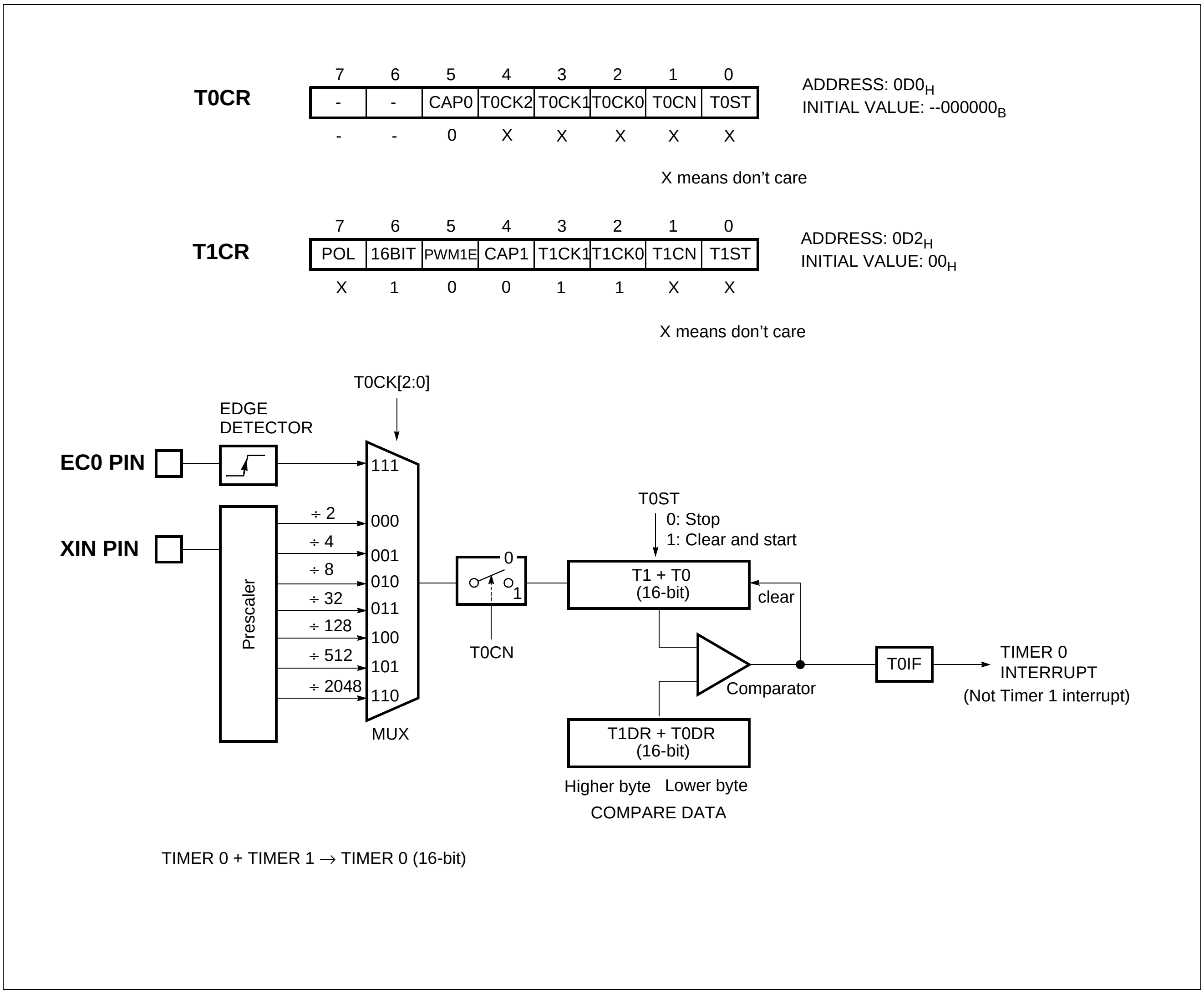
Figure 13-6 Count Operation of Timer / Event counter

13.2 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 T0DR, T1DR and then resets to 0000_H. The match output generates Timer 0 interrupt not Timer 1 interrupt.

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

In 16-bit mode, the bits T1CK[1:0] and 16BIT of T1CR should be set to "1" respectively.



T0CR

	7	6	5	4	3	2	1	0
T0CR	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
	-	-	0	X	X	X	X	X

ADDRESS: $0D0_H$
INITIAL VALUE: $\text{--}000000_B$

X means don’t care

T1CR

	7	6	5	4	3	2	1	0
T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	1	0	0	1	1	X	X

ADDRESS: $0D2_H$
INITIAL VALUE: 00_H

X means don’t care

Figure 13-7 16-bit Timer/Counter

13.3 8-bit Capture Mode

The Timer 0 capture mode is set by bit CAP0 of timer mode register TCR1 (bit CAP1 of timer mode register T1CE for Timer 1) as shown in Figure 13-8.

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

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

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

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

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

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

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

T0CR

7	6	5	4	3	2	1	0
-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
-	-	1	X	X	X	X	X

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

X means don’t care

T1CR

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

ADDRESS: $0D2_H$
INITIAL VALUE: 00_H

X means don’t care

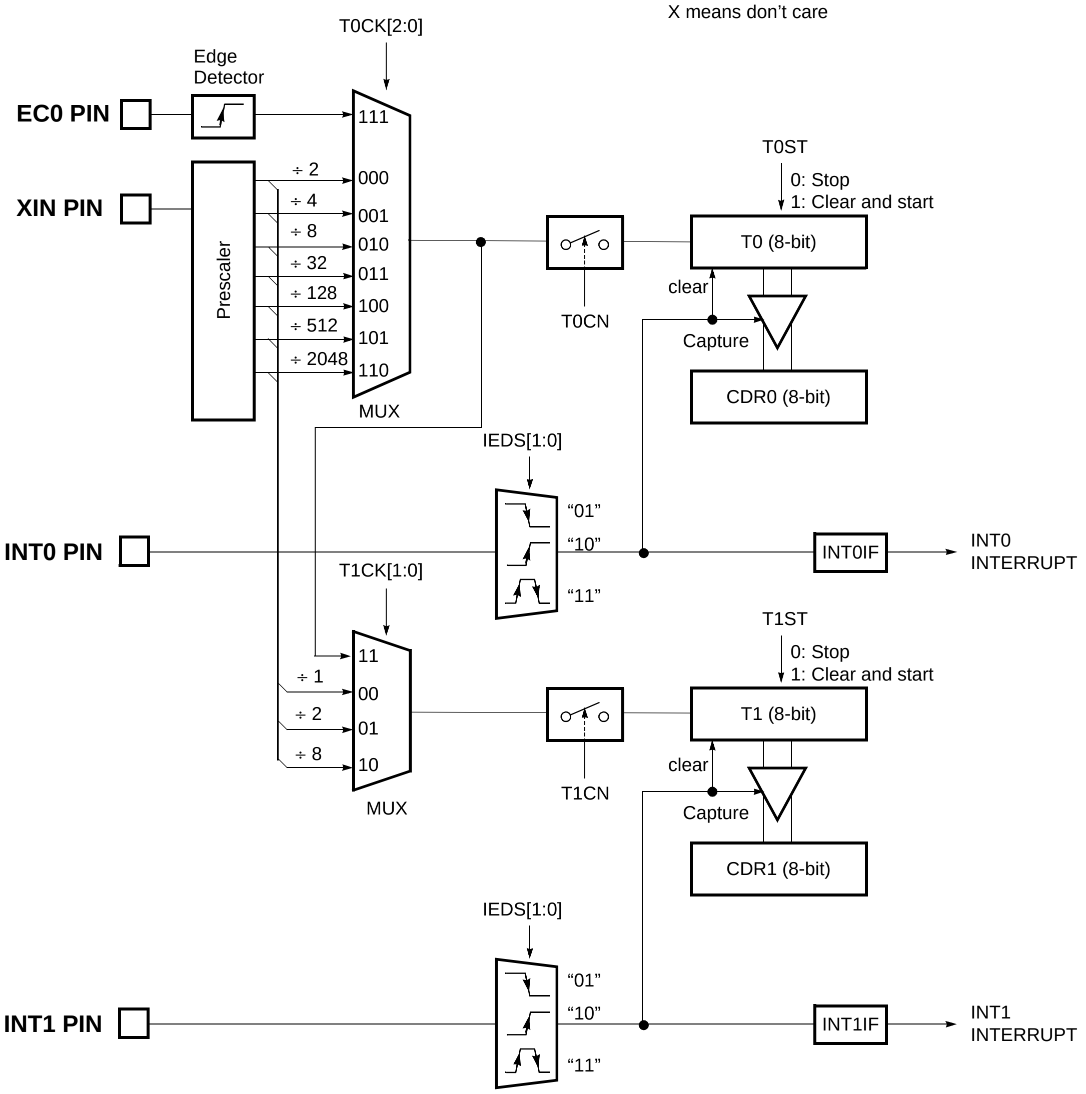


Figure 13-8 8-bit Capture Mode

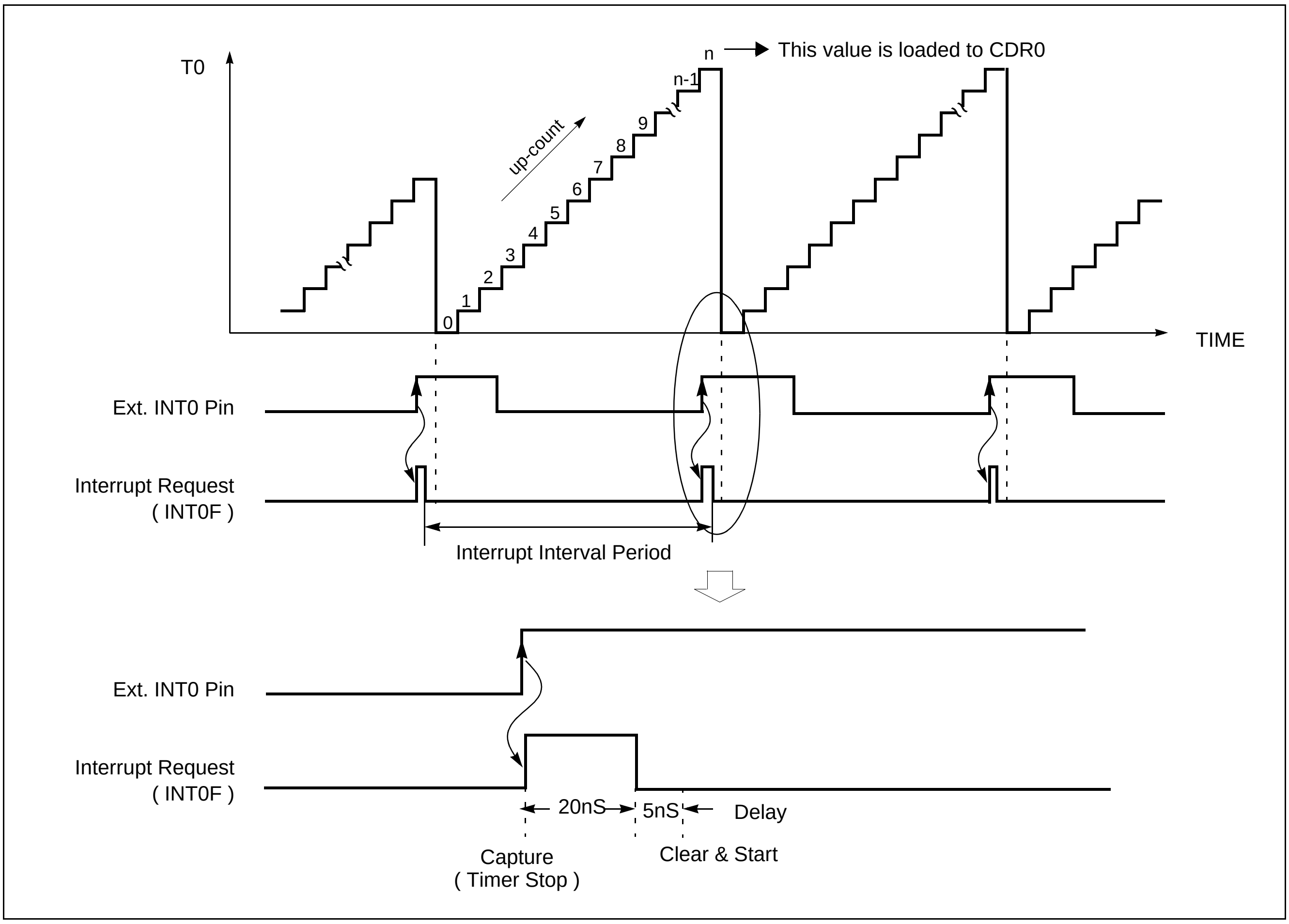


Figure 13-9 Input Capture Operation

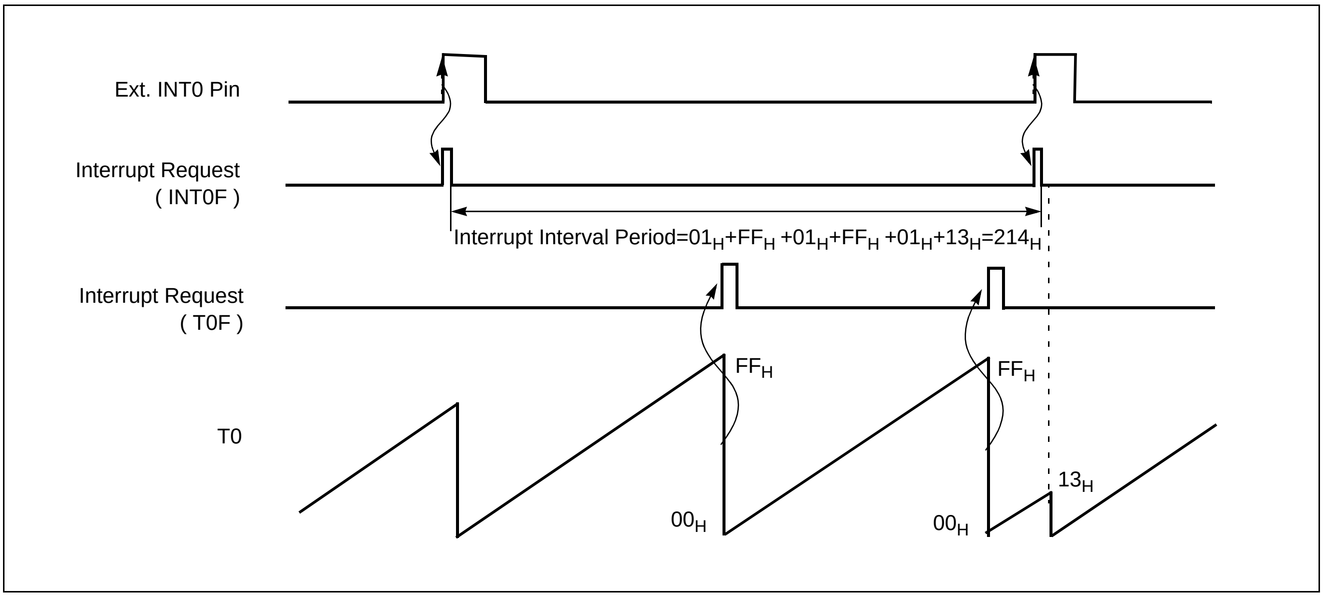


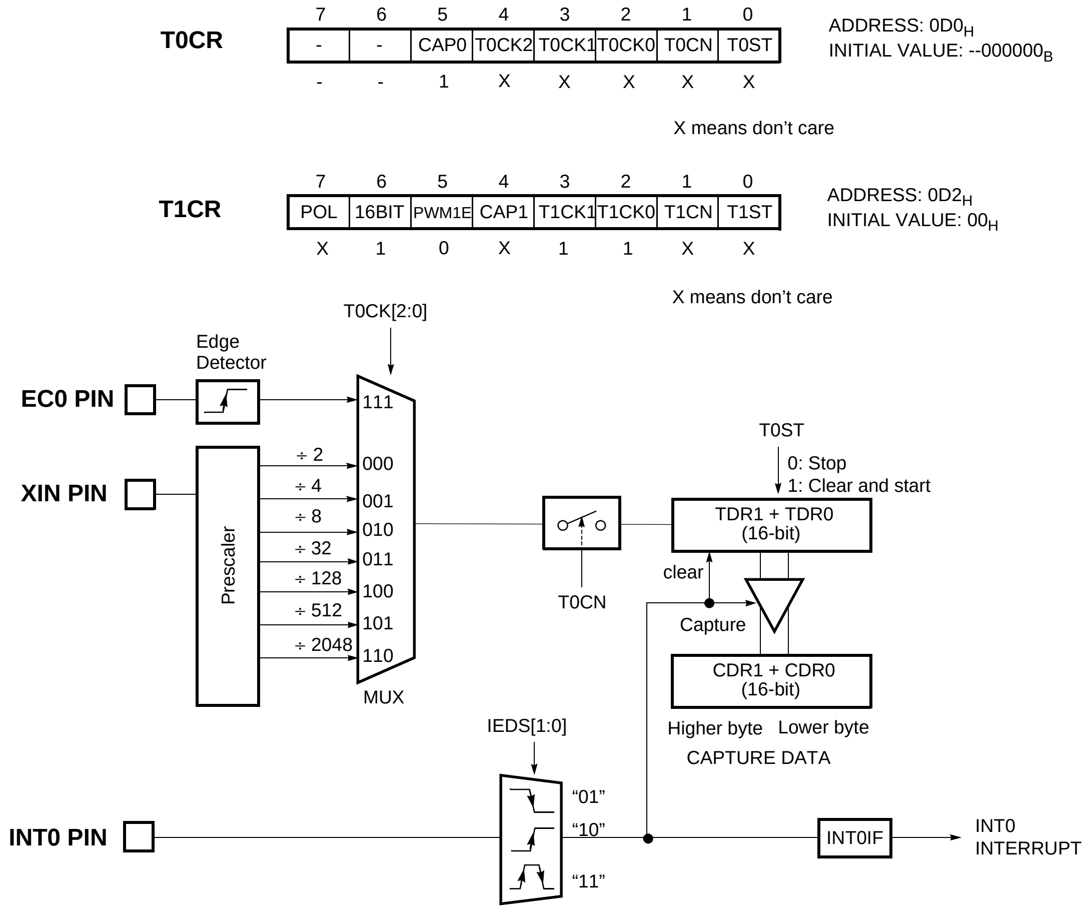
Figure 13-10 Excess Timer Overflow in Capture Mode

13.4 16-bit Capture Mode

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

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 T1CR should be set to "1" respectively.



T0CR — ADDRESS: $0D0_H$, INITIAL VALUE: $\text{--}000000_B$

7	6	5	4	3	2	1	0
-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
-	-	1	X	X	X	X	X

X means don’t care

T1CR — ADDRESS: $0D2_H$, INITIAL VALUE: 00_H

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

X means don’t care

Figure 13-11 16-bit Capture Mode

Example 1:

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

```
LDM    T0CR,#0000_1111B;8uS
LDM    T1CR,#0100_1100B;16bit Mode
LDM    TDR0,#<62499    ;8uS X 62500
LDM    TDR1,#>62499    ;=0.5s
SET1   T0E
EI
:
:
```

Example 2:

Timer0 = 16-bit event counter mode

```
LDM    PSR,#0000_1000B;EC0 Set
LDM    T0CR,#0001_1111B;CounterMode
LDM    T1CR,#0100_1100B;16bit Mode
LDM    TDR0,#<0FFH     ;
LDM    TDR1,#>0FFH     ;
SET1   T0E
EI
:
:
```

Example 3:

Timer0 = 16-bit capture mode

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

13.5 PWM Mode

The HMS81C2232/48 has a high speed PWM (Pulse Width Modulation) functions which shared with Timer1.

In PWM mode, pin R64/PWM1O outputs up to a 10-bit resolution PWM output. This pin should be configured as a PWM output by setting "1" bit PWM1O in PSR.4 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 13-12, the duty data is transferred from the master to the slave when the period data matched to the counted value. (i.e. at the beginning of next duty cycle)

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

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

The relation of frequency and resolution is in inverse proportion. Table 13-2 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 13-2 PWM Frequency vs. Resolution at 4MHz

The bit POL of T1CR 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 13-14. As it were, the absolute duty time is not changed in varying frequency. But the changed period value must greater than the duty value.

Note: *If changing the Timer1 to PWM function, it should be stop the timer clock firstly, and then set period and duty register value. If user writes register vlaues while timer is in operation, these register could be set with certain values.*

Ex) Sample Program @4MHz 2uS

```
LDM    T1CR,#1010_1010b; Set Clock & PWM1E
LDM    T1PPR,#199          ; Period :400uS=2uSX(199+1)
LDM    T1PDR,#99           ; Duty:200uS=2uSX(99+1)
LDM    PWM1HR,00H
LDM    T1CR,#1010_1011b ; Start timer1
```

13.6 8-bit Compare Output (16-bit)

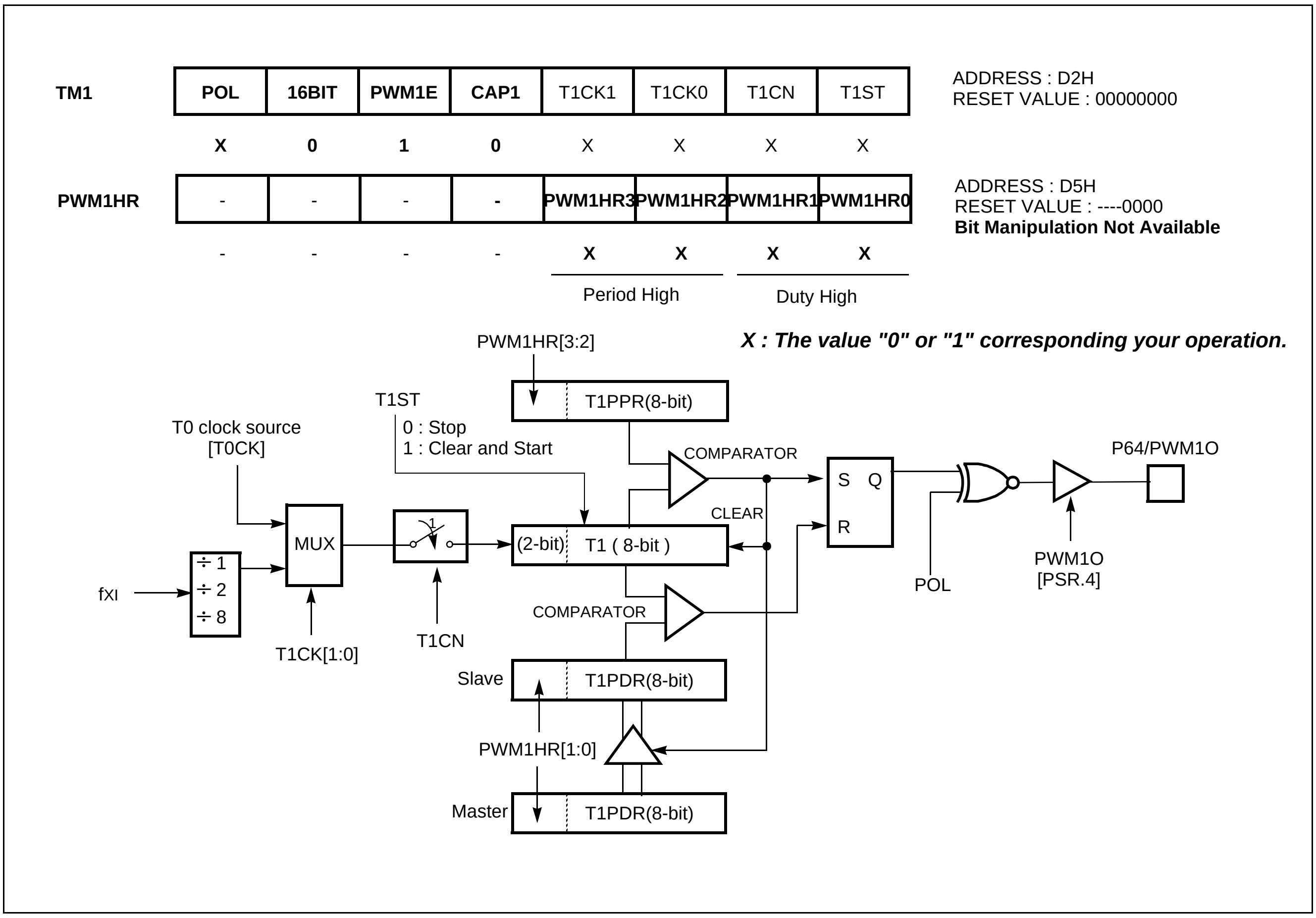
The HMS81C2232/48 has a function of Timer Compare Output. To pulse out, the timer match can goes to port pin(P64/PWM1O) as shown in Figure 13-2 and Figure 13-7. Thus, pulse out is generated by the timer match. These operation is implemented to pin, P64/PWM1O.

In this mode, the bit P64/PWM1O of Port Selection register (PSR.4) should be set to "1", and the bit PWM1E of timer1 mode register (T1CR) should be set to "0". In addition, 16-bit Compare output mode is available, also.

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

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

.



TM1	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	0	1	0	X	X	X	X

ADDRESS : D2H
RESET VALUE : 00000000

PWM1HR	-	-	-	-	PWM1HR3	PWM1HR2	PWM1HR1	PWM1HR0
	-	-	-	-	X	X	X	X
					Period High		Duty High	

ADDRESS : D5H
RESET VALUE : ----0000
Bit Manipulation Not Available

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

Figure 13-12 PWM Mode

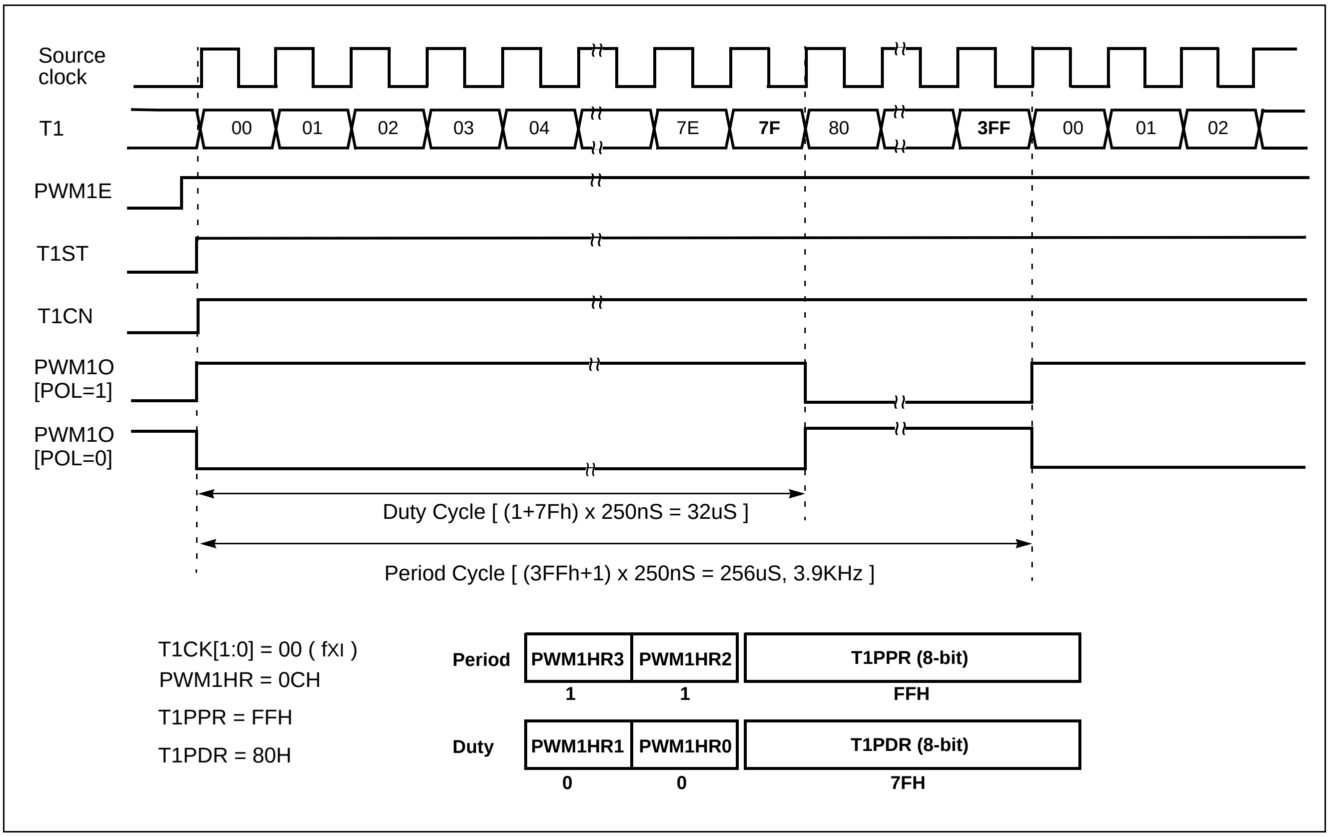


Figure 13-13 Example of PWM at 4MHz

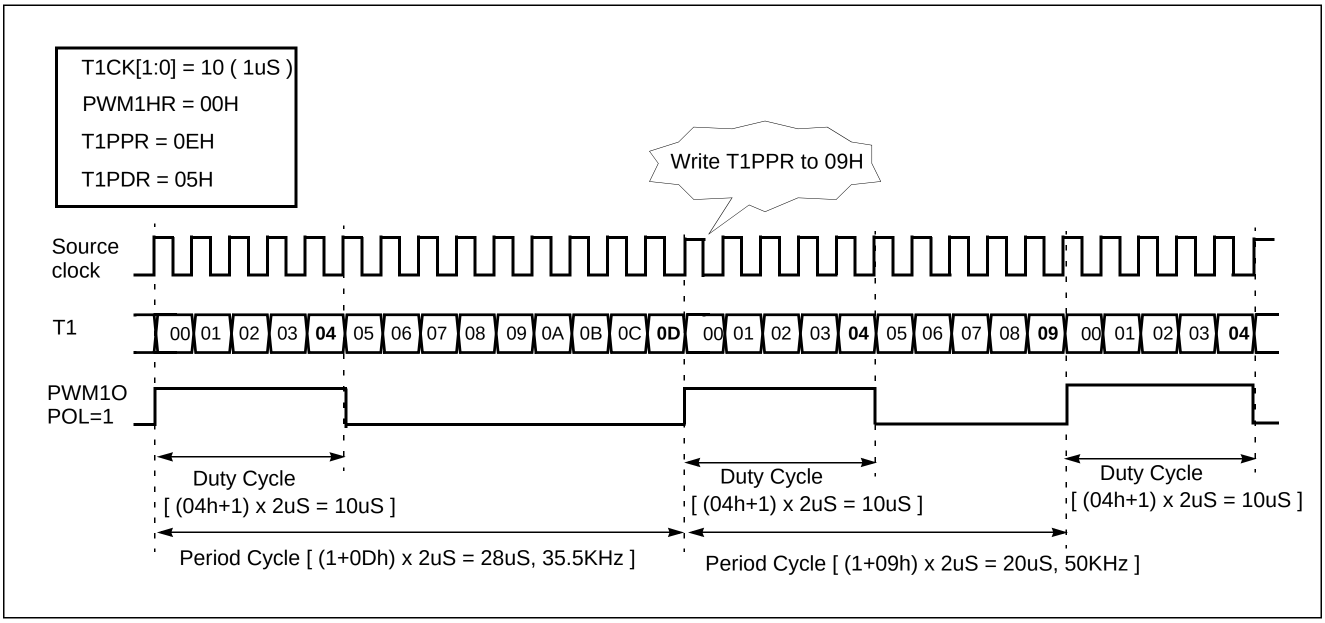

T1CK[1:0] = 10 (1uS)
PWM1HR = 00H
T1PPR = 0EH
T1PDR = 05H
Write T1PPR to 09H
Source clock
T1
00
01
02
03
04
05
06
07
08
09
0A
0B
0C
0D
00
01
02
03
04
05
06
07
08
09
00
01
02
03
04
PWM1O POL=1
Duty Cycle [(04h+1) x 2uS = 10uS]
Duty Cycle [(04h+1) x 2uS = 10uS]
Duty Cycle [(04h+1) x 2uS = 10uS]
Period Cycle [(1+0Dh) x 2uS = 28uS, 35.5KHz]
Period Cycle [(1+09h) x 2uS = 20uS, 50KHz]

Figure 13-14 Example of Changing the Period in Absolute Duty Cycle (@4MHz)

14. REMOTE CONTROL TIMER

The 8-bit remote control timer has a pulse width measurement function with a resolution of 8 bits. Pulse width is measured from a difference in count value when the valid edge has been detected while the timer operates in the free-running mode.

6.3 Registers Controlling 8-Bit Remote Control Timer

The following three types of registers control the 8-bit remote control timer.

- Remote control timer control register (RTCR)
- Remote control timer capture registers (RTCP0 and RTCP1)
- 8-bit timer register (RT)

(1) Remote control timer control register(RTCR)

This register enables or disables the operation of the 8-bit timer (RT), and sets the count clock. TMC9 is set by using a 1-bit or 8-bit memory manipulation instruction. This register is initialized to 00_H by RESET input.

(2) Remote control timer capture registers (RTCP0 and RTCP1)

These 8-bit registers capture the contents of the 8-bit timer (RT). The capture operation is performed in synchronization with the valid edge input to the TI pin (capture trigger). The contents of RTCP0 are retained until the next rising edge of the TI pin is detected. The contents of RTCP1 are retained until the next falling edge of the TI pin is detected.

RTCP0 and RTCP1 can be read by using an 8-bit memory manipulation instruction. The values of these registers are initialized to 00_H by RESET input.

(3) 8-bit timer register (TM9)

This 8-bit register counts the count pulse. It can be read by using an 8-bit memory manipulation instruction. The value of this register is initialized to 00H by RESET input or by clearing the RTST bit.

6.4 Operation of 8-Bit Remote Control Timer

The 8-bit remote control timer operates as a pulse width measuring circuit. The width of a high-level or low-level external pulse input to the TI pin is measured by operating the 8-bit timer (TM9) in the free-running mode. Detection of the valid edge is sampled every 2 cycles of the count clock selected by TCL0, TCL1 and TCL2, and the capture operation is not performed until the valid level has been detected two times. Therefore, the pulse width input to the TI pin must be 5 or more of the count clock set by TCL0, TCL1 and TCL2, regardless of whether the level is high or low. If the pulse width is less than 5 clocks, it cannot be detected, and the capture operation is not performed. The value of timer register 9 (TM9) is loaded to and retained in the capture registers (CP90 and CP91) in synchronization with the valid edge of the pulse input to the TI pin, as shown in Figure 6-3. Figure 6-3 shows the timing of pulse width measurement.

s shown in Figure 13-1 and Table 13-1.

RTCR

		R/W 5	R/W 4	R/W 3	R/W 2	R/W 1	R/W 0
-	-	RCAP	RTCk2	RTCK1	RTCK0	RTCN	RTST

ADDRESS: $0E7_H$
INITIAL VALUE: $--00\ 0000_B$

Bit Name	Bit Position	Description
RCAP	RTCR.5	0: Timer/Counter mode 1: Capture mode selection flag
RTCK2 RTCK1 RTCK0	RTCR.4 RTCR.3 RTCR.2	000: 8-bit Timer, Clock source is $f_{XIN} \div 2$ 001: 8-bit Timer, Clock source is $f_{XIN} \div 4$ 010: 8-bit Timer, Clock source is $f_{XIN} \div 8$ 011: 8-bit Timer, Clock source is $f_{XIN} \div 32$ 100: 8-bit Timer, Clock source is $f_{XIN} \div 128$ 101: 8-bit Timer, Clock source is $f_{XIN} \div 512$ 110: 8-bit Timer, Clock source is $f_{XIN} \div 2048$ 111: $\overline{\text{EC0}}$ (External clock)
RTCN	RTCR.1	0: Stop the timer 1: A logic 1 starts the timer.
RTST	RTCR.0	0: When cleared, stop the counting. 1: When set, Timer 0 Count Register is cleared and start again.

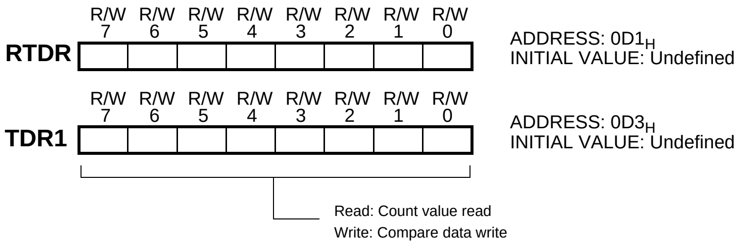
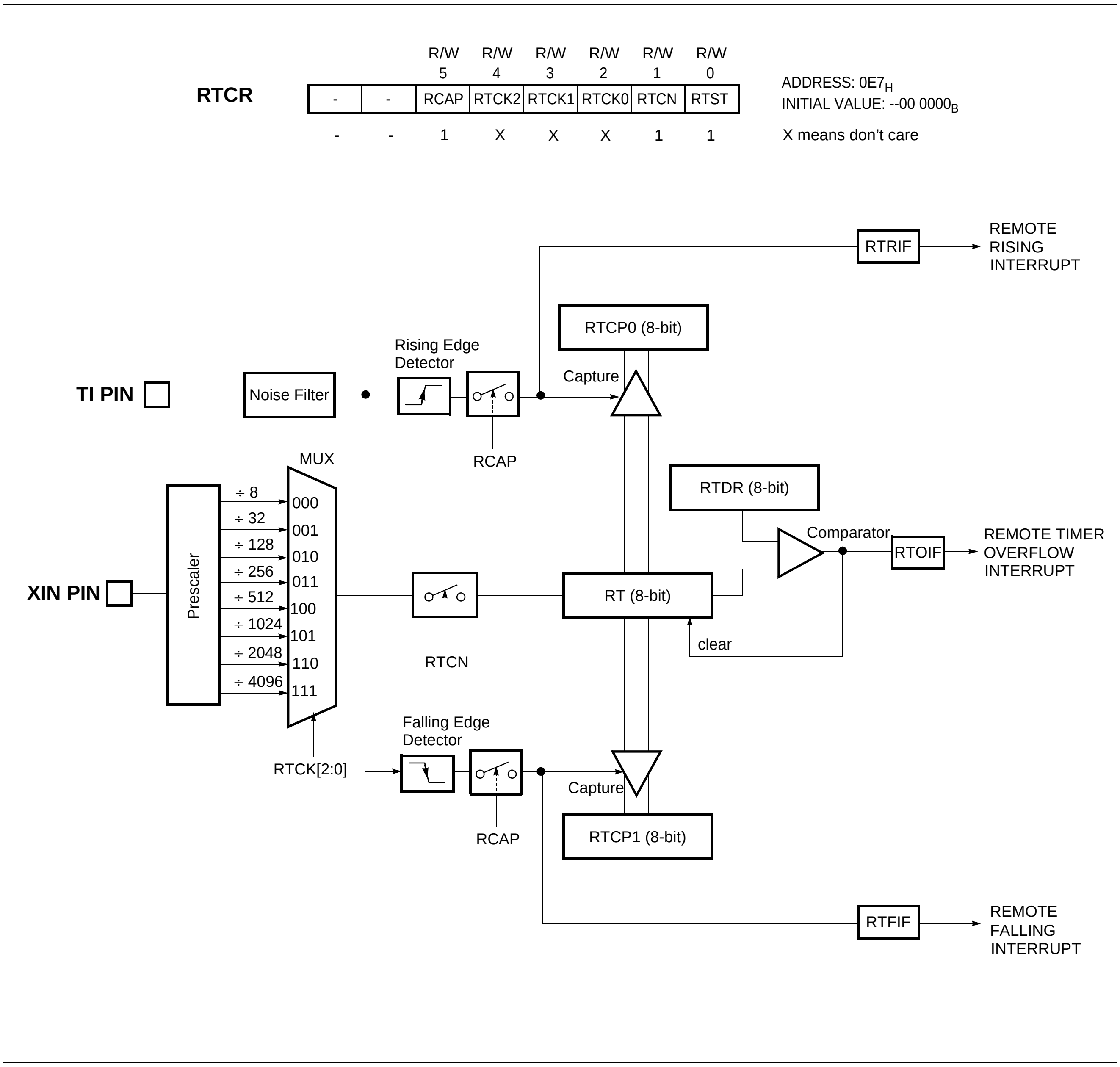


Figure 14-1 TM0, TM1 Registers

Note: *The RTCP0, RTDR and RT are in same address. In the capture mode, reading operation is read the RTCP0, not RT because path is opened to the RTCP0, and RTDR is only for writing operation.*

.



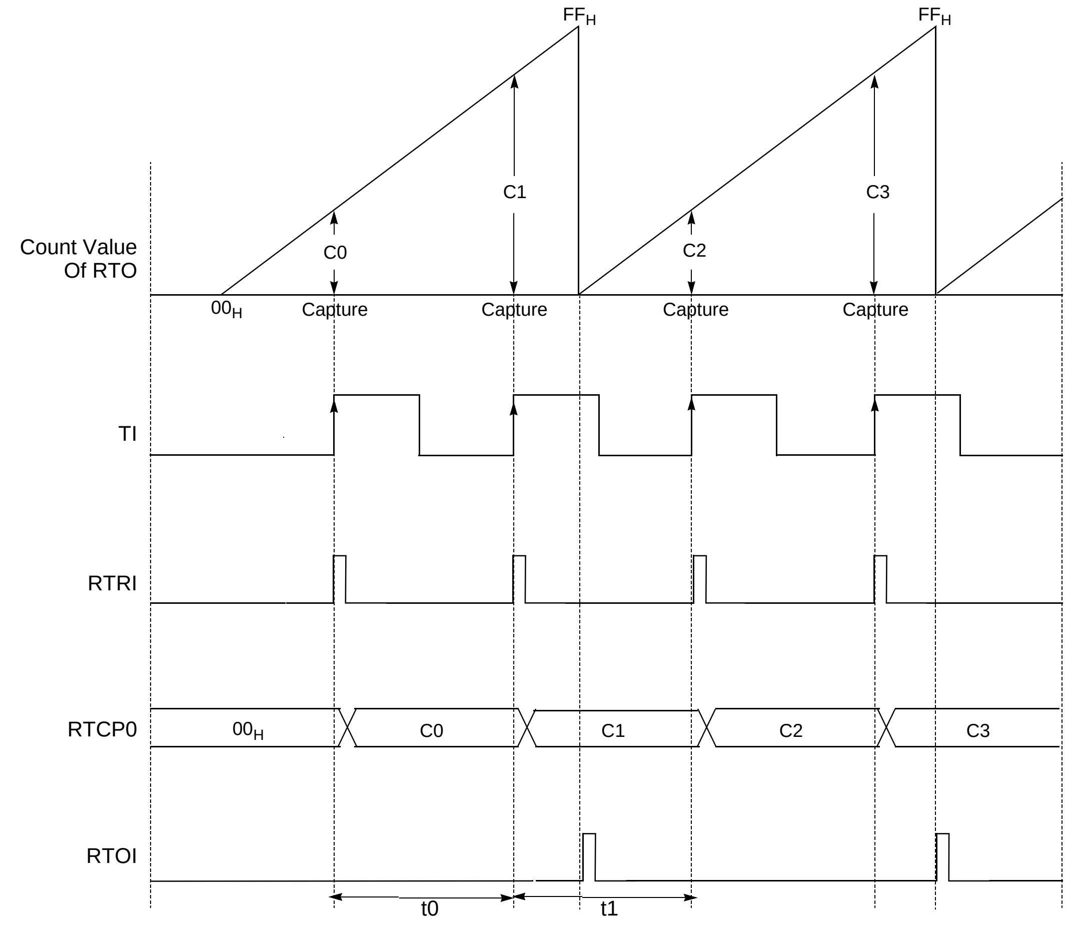
RTCR

ADDRESS: 0E7$_H$
INITIAL VALUE: --00 0000$_B$

		R/W 5	R/W 4	R/W 3	R/W 2	R/W 1	R/W 0
-	-	RCAP	RTCK2	RTCK1	RTCK0	RTCN	RTST
-	-	1	X	X	X	1	1

X means don’t care

Figure 14-2 Block Diagram of Remote Control Timer

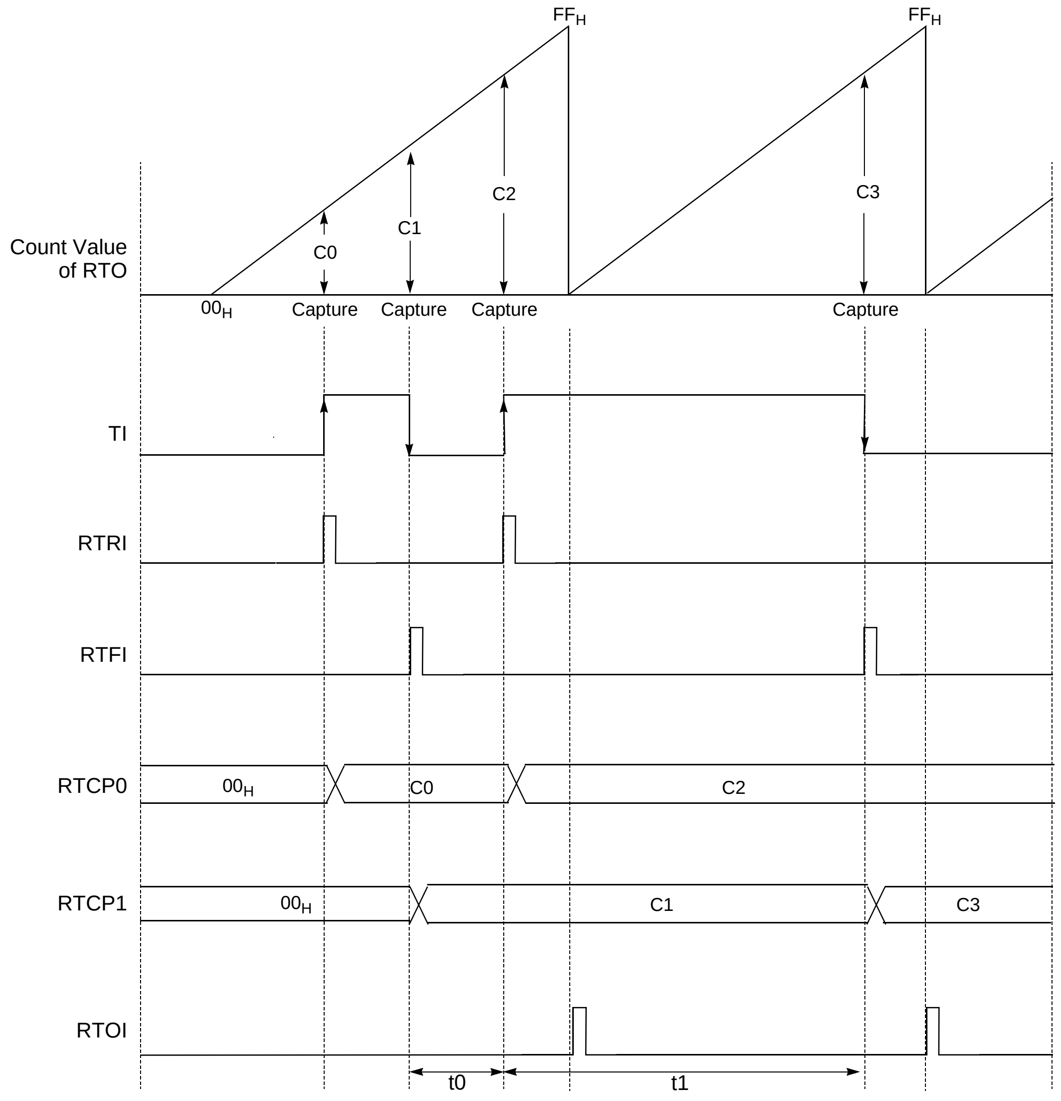


Remarks : **1.** t0 = (C1-C0) X 1/fCOUNT

2. t1 = (100H - C1 + C2) X 1/fCOUNT

3. fCOUNT : Count clock frequency set by TCL0~ TCL2

Figure 14-3 To measure pulse width in synchronization with rising edge



Remarks : **1.** t0 = (C2-C1) X 1/f$_{COUNT}$

2. t1 = (100H - C2 + C3) X 1/f$_{COUNT}$

3. f$_{COUNT}$: Count clock frequency set by TCL0~ TCL2

Figure 14-4 To measure pulse width in synchronization with both rising and falling edge

15. ANALOG DIGITAL CONVERTER

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

The A/D module has two registers which are the control register ADCM and A/D result register ADR. The register ADCM, shown in Figure 15-1, controls the operation of the A/D converter module. The port pins can be configured as analog inputs or digital I/O.

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

How to Use A/D Converter

The processing of conversion is start when the start bit ADST is set to "1". After one cycle, it is cleared by hardware. The register ADCR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCR, the A/D conversion status bit ADSF is set to "1", and the A/D interrupt flag ADIF is set. The block diagram of the A/D module is shown in Figure 15-2. The A/D status bit ADSF is set automatically when A/D conversion is completed, cleared when A/D conversion is in process. The conversion time takes maximum 20 uS (at fXI=4 MHz)

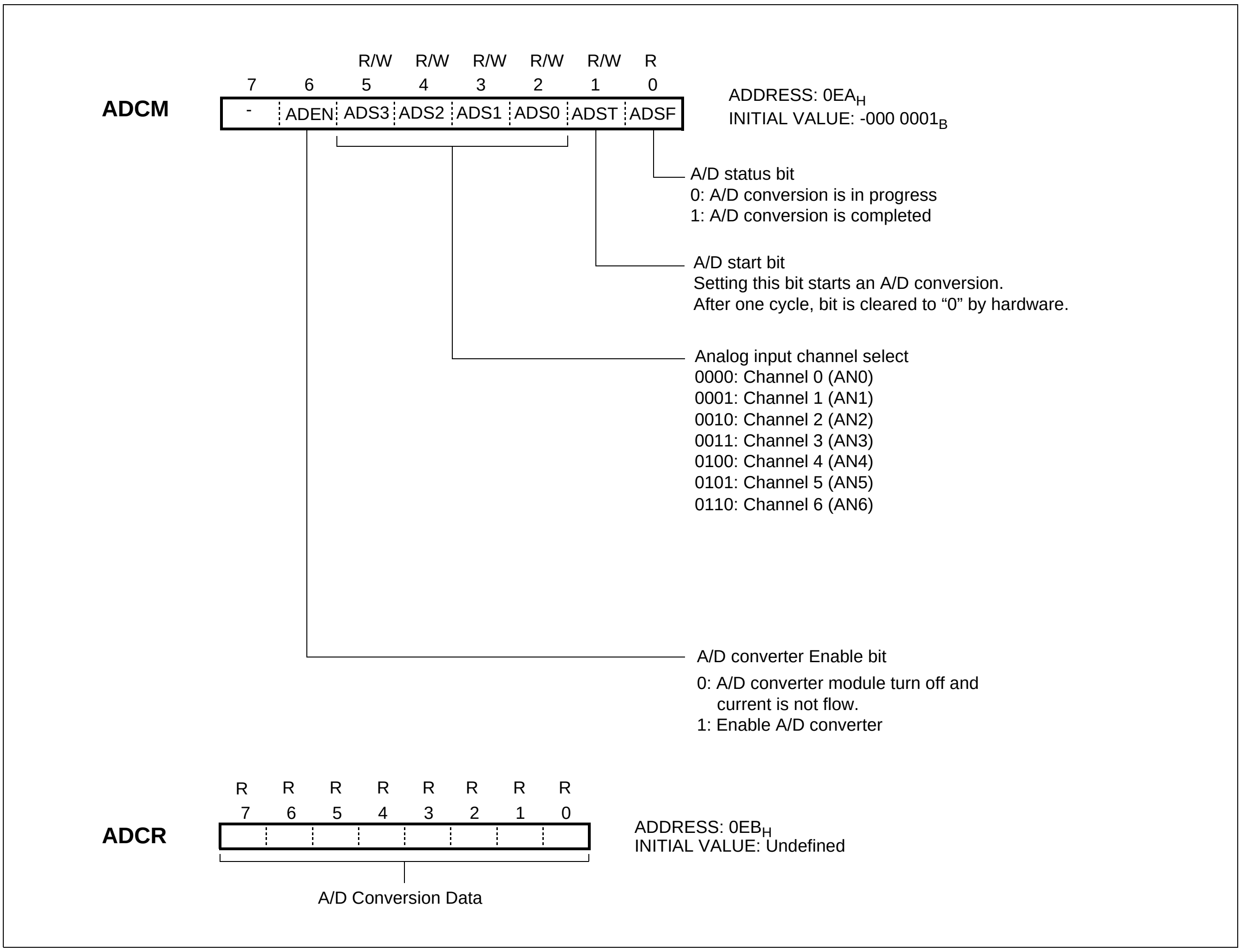


Figure 15-1 A/D Converter Control Register

.

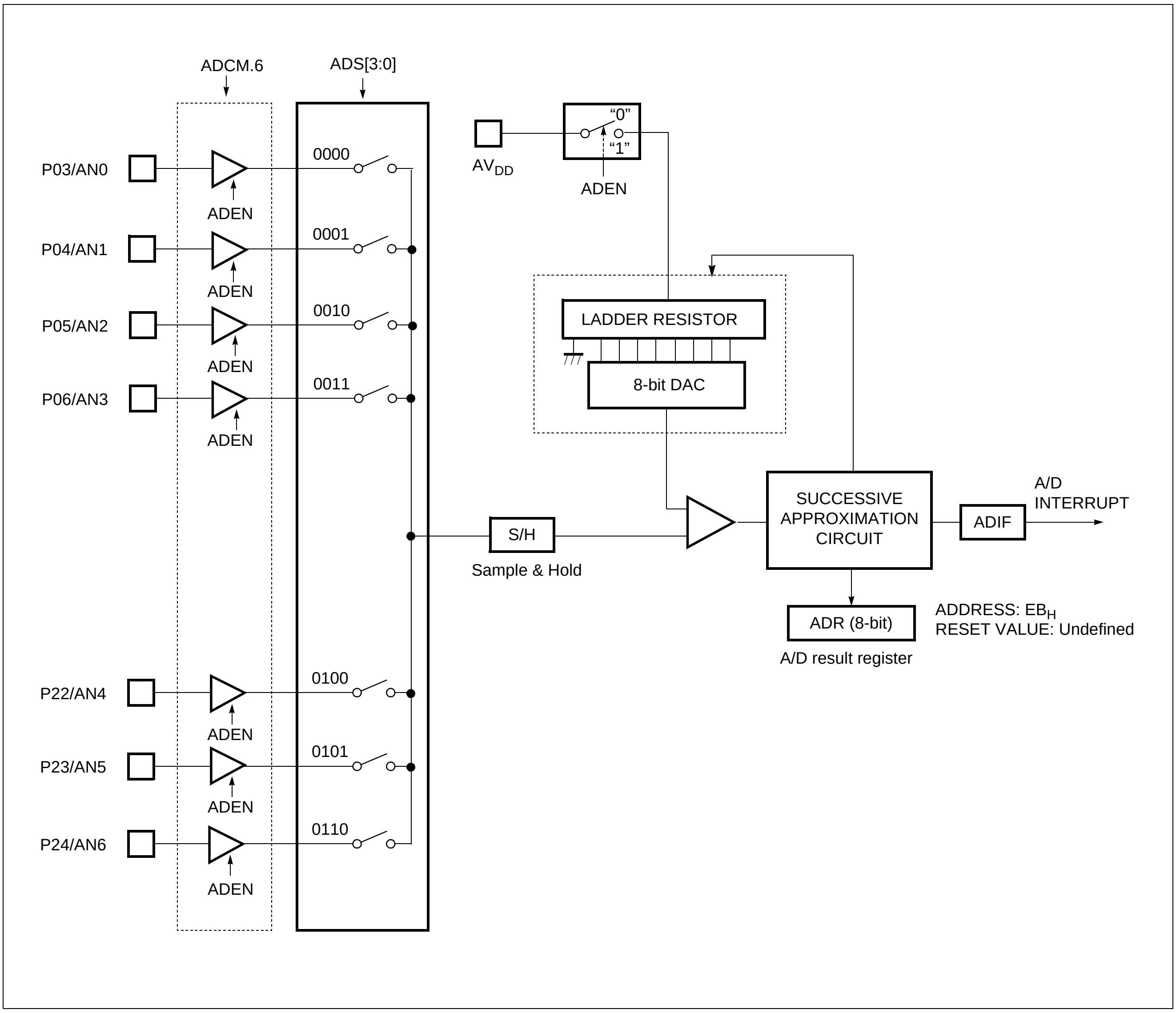


Figure 15-2 A/D Block Diagram

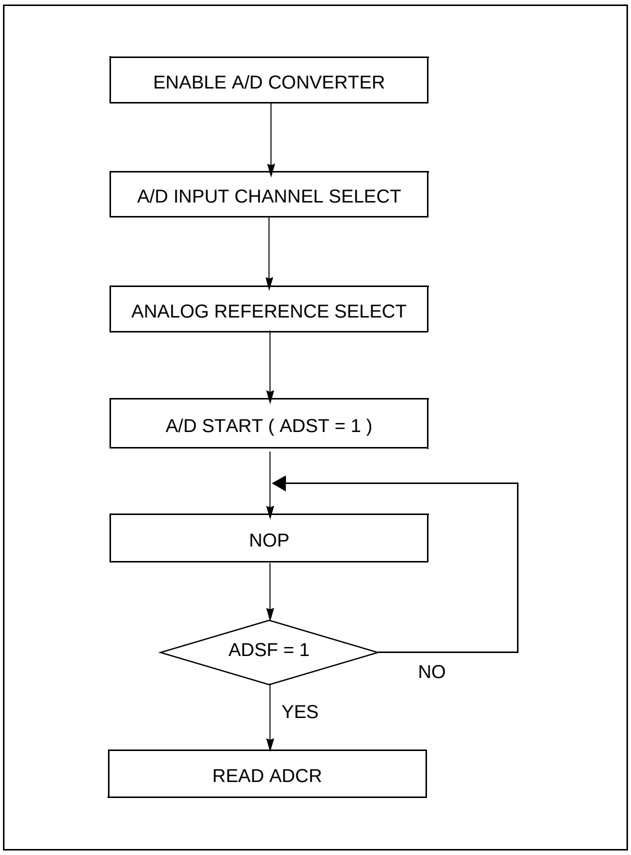


Figure 15-3 A/D Converter Operation Flow

A/D Converter Cautions

(1) Input range of AN0 to AN6

The input voltage of AN6 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 AV_{DD} 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 15-4 in order to reduce noise.

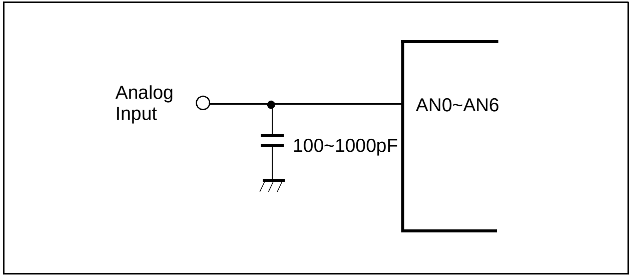


Figure 15-4 Analog Input Pin Connecting Capacitor

(3) Pins AN0/P00 to AN3/P03 and AN4/P22 to AN6/P24

The analog input pins AN6 to AN0 also function as input/output port (PORT P7 and P2) pins. When A/D conversion is performed with any of pins AN6 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) AV_{DD} pin input impedance

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

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

16. SERIAL PERIPHERAL INTERFACE (SPI1)

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

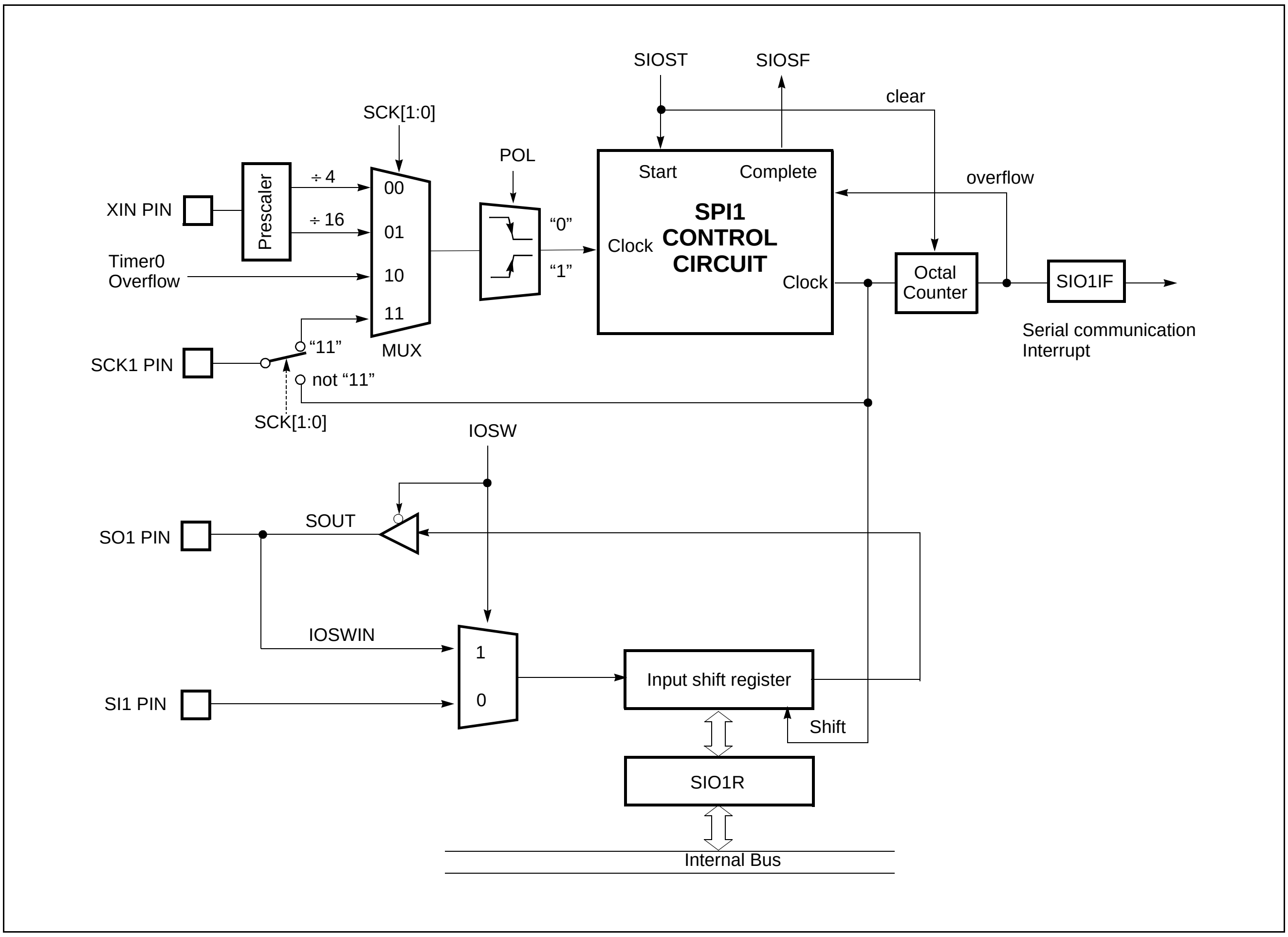


Figure 16-1 SPI1 Block Diagram

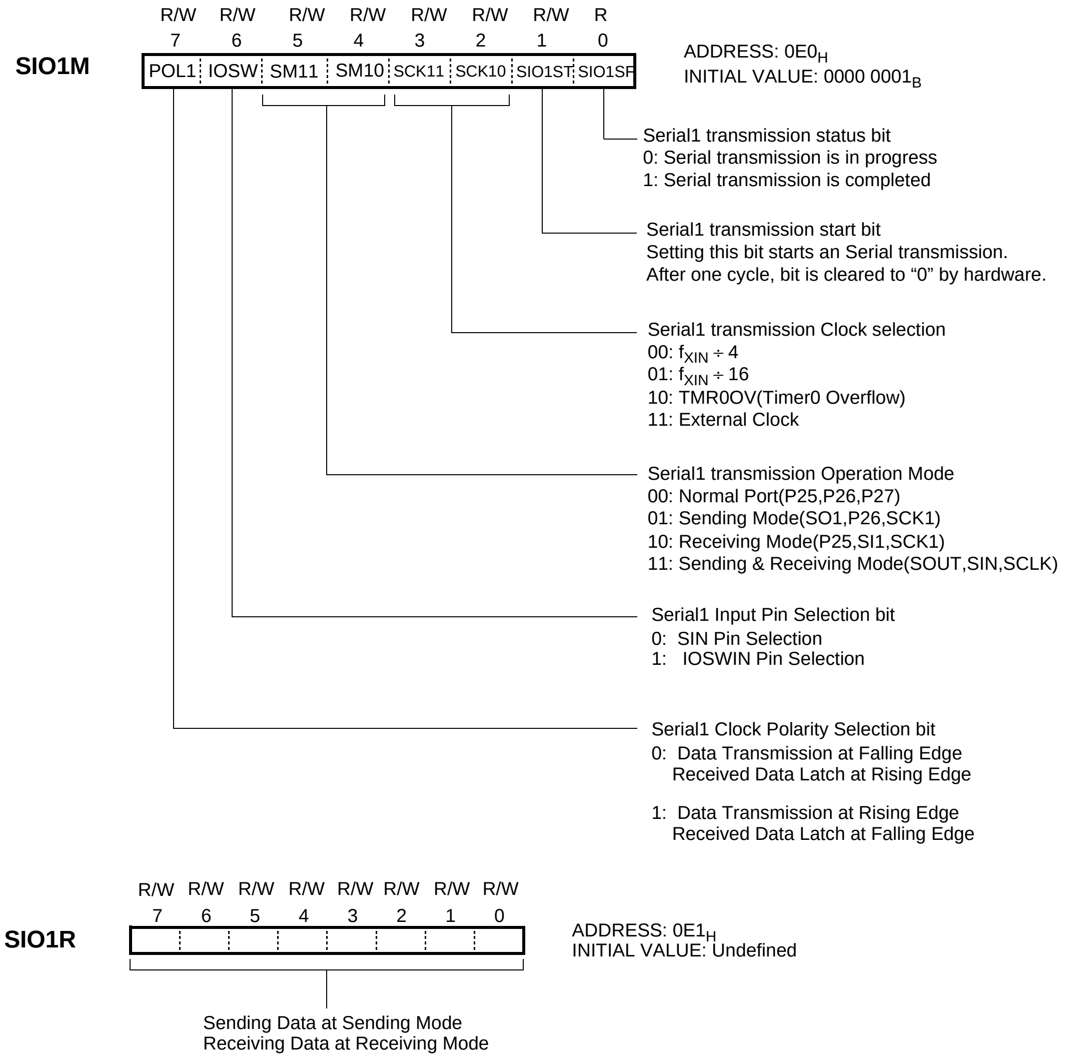
Serial I/O 1 Mode Register(SIO1M) controls serial I/O function. According to SCK1 and SCK0, the internal clock or external clock can be selected. The serial transmission operation mode is decided by setting the SM1 and SM0, and the polarity of transfer clock is selected by setting the POL.

Serial I/O Data Register(SIO1R) is a 8-bit shift register. First LSB is send or is received. When receiving mode, serial input pin is selected by IOSW. The SPI allows 8-bits of data to be synchronously transmitted and received.

To accomplish communication, typically three pins are used:

- Serial Data In	P26/SI1
- Serial Data Out	P25/SO1
- Serial Clock	P27/SCK1

.


SIO1M
R/W R/W R/W R/W R/W R/W R/W R
7 6 5 4 3 2 1 0
POL1 IOSW SM11 SM10 SCK11 SCK10 SIO1ST SIO1SF
SIO1R
R/W R/W R/W R/W R/W R/W R/W R/W
7 6 5 4 3 2 1 0

ADDRESS: 0E0$_H$
INITIAL VALUE: 0000 0001$_B$

Serial1 transmission status bit
0: Serial transmission is in progress
1: Serial transmission is completed

Serial1 transmission start bit
Setting this bit starts an Serial transmission.
After one cycle, bit is cleared to “0” by hardware.

Serial1 transmission Clock selection
00: $f_{XIN} \div 4$
01: $f_{XIN} \div 16$
10: TMR0OV(Timer0 Overflow)
11: External Clock

Serial1 transmission Operation Mode
00: Normal Port(P25,P26,P27)
01: Sending Mode(SO1,P26,SCK1)
10: Receiving Mode(P25,SI1,SCK1)
11: Sending & Receiving Mode(SOUT,SIN,SCLK)

Serial1 Input Pin Selection bit
0: SIN Pin Selection
1: IOSWIN Pin Selection

Serial1 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

ADDRESS: 0E1$_H$
INITIAL VALUE: Undefined

Sending Data at Sending Mode
Receiving Data at Receiving Mode

Figure 16-2 SPI1 Control Register

16.1 Transmission/Receiving Timing

The serial transmission is started by setting SIO1ST(bit1 of SIO1M) to “1”. After one cycle of SCK1, SIO1ST is cleared automatically to “0”. The serial output data from 8-bit shift register is output at falling edge of SCK1. And input data is latched at rising edge of SCK1 pin. When transmission clock is counted 8 times, serial I/O counter is cleared as ‘0”. Transmission clock is halted in “H” state and serial I/O interrupt(IFSIO1) occurred.

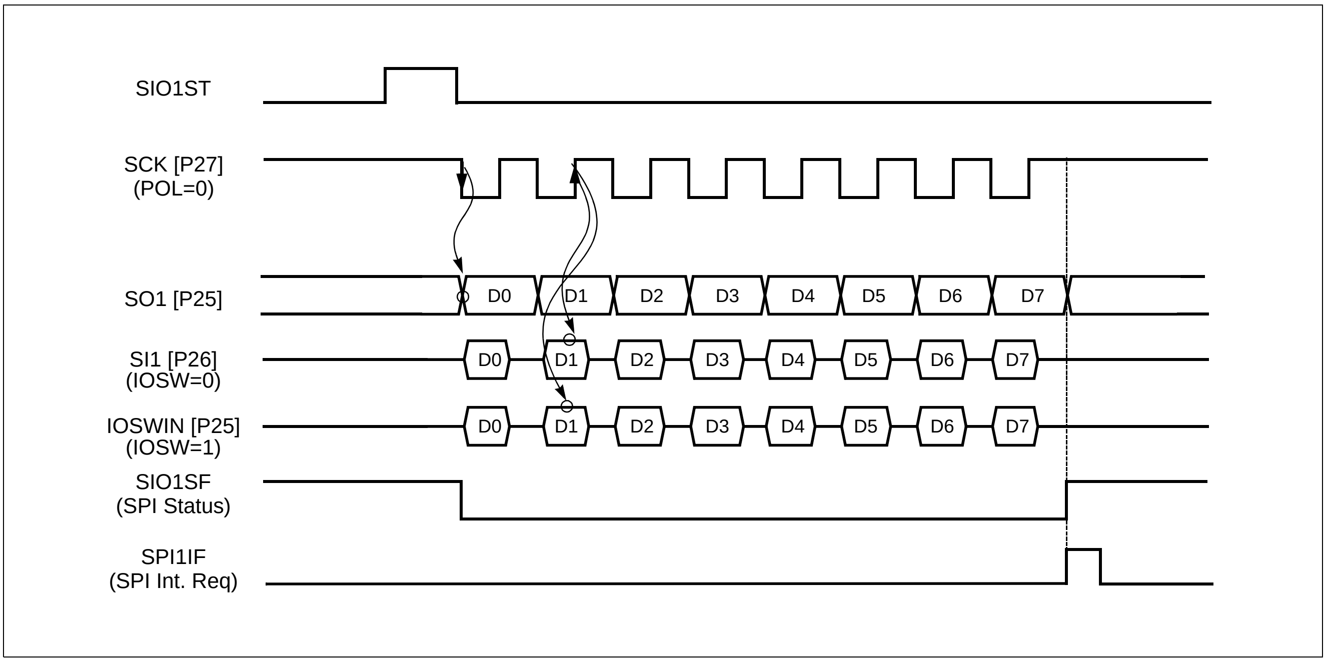


Figure 16-3 SPI1 Timing Diagram at POL=0

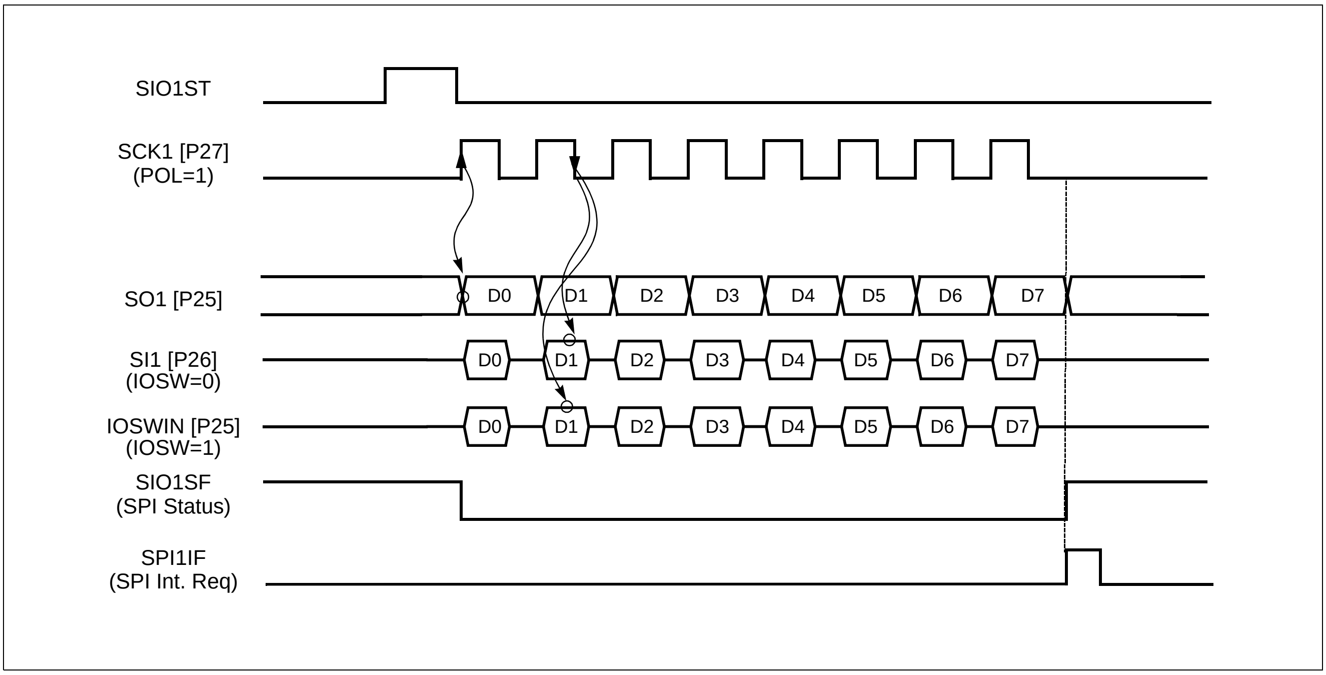


Figure 16-4 SPI Timing Diagram at POL=1

16.2 The method of Serial I/O

① Select transmission/receiving mode

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

② In case of sending mode, write data to be send to SIO1R.

③ Set SIO1ST to “1” to start serial transmission.

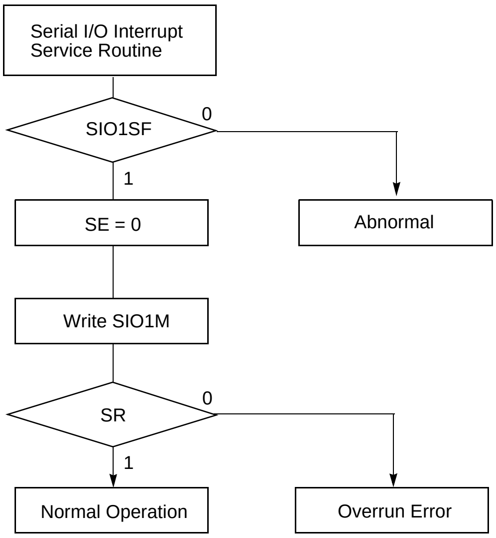
Note: *If both transmission mode is selected and transmission is performed simultaneously it would be made error.*

④ The SIO1 interrupt is generated at the completion of SIO1 and SIO1SF is set to “1”. In SIO1 interrupt service routine, correct transmission should be tested.

⑤ In case of receiving mode, the received data is acquired by reading the SIO1R.

```
LDM      SIO1R,#0AAh      ;SIO1R Initial
                          ;Value
LDM      SIO1M,#0011_1100b;SIO1M Select
         NOP
         NOP
         SIO1M,#0011_1110b;SIO1 Start
```

16.3 The Method to Test Correct Transmission



- SE : Interrupt Enable Register Low IENL(Bit3)

- SR : Interrupt Request Flag Register Low IRQL(Bit3)

Figure 16-5 Serial1 Method to Test Transmission

17. SERIAL PERIPHERAL INTERFACE (SPI3)

The Serial Peripheral Interface (SPI3) module is a serial interface useful for communicating with other peripheral of microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The Serial Peripheral Interface(SPI3) is 8-bit clock synchronous type and consists of serial I/O register, serial I/O mode register, clock selection circuit octal counter and control circuit. The SOUT pin is designed to input and output. So Serial Peripheral Interface(SPI) can be operated with minimum two pin

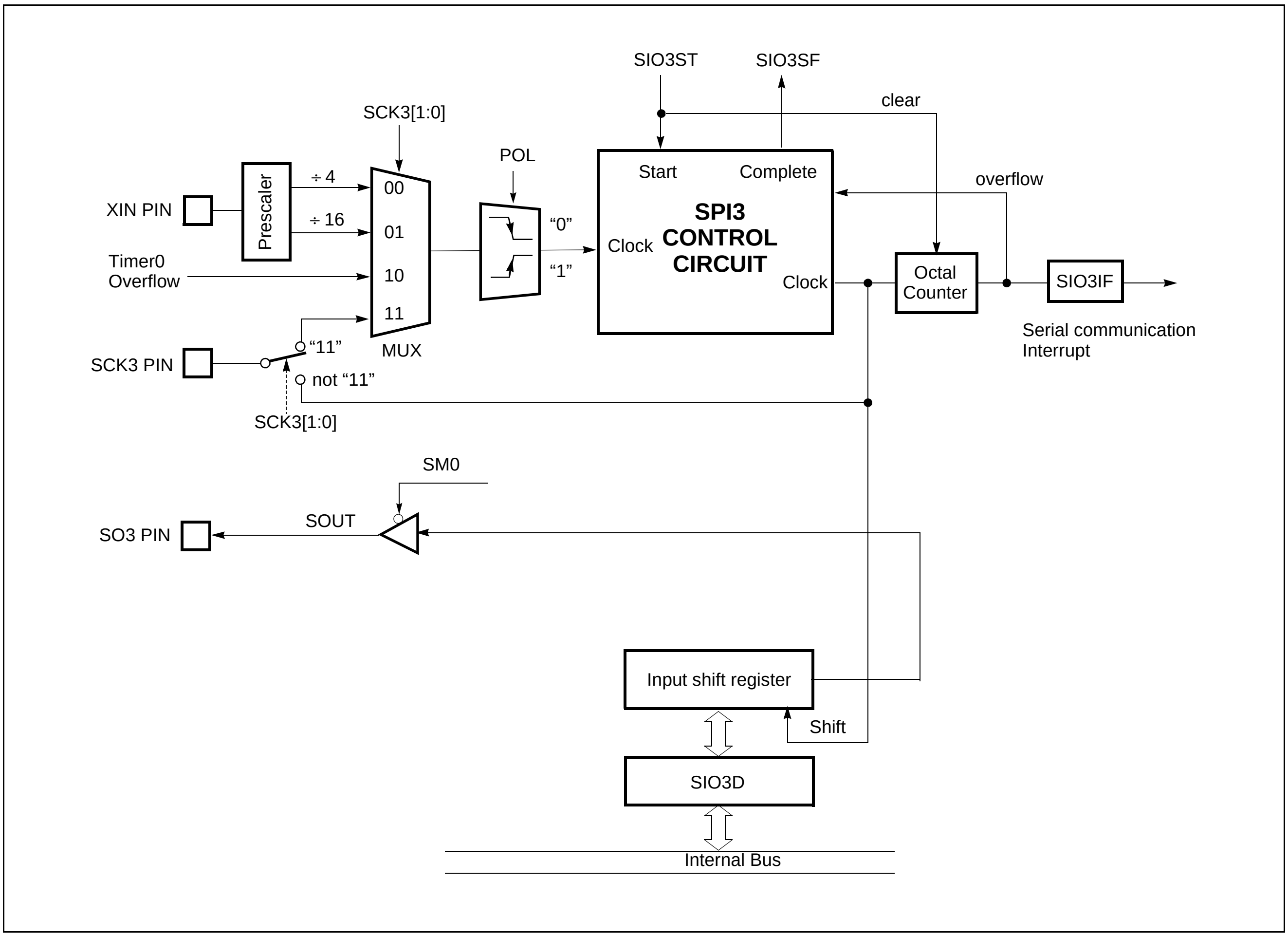


Figure 17-1 SPI3 Block Diagram

Serial I/O Mode Register(SIO3M) controls serial I/O function. According to SCK31 and SCK30, the internal clock or external clock can be selected. The serial transmission operation mode is decided by setting the SM30, and the polarity of transfer clock is selected by setting the POL3.

Serial I/O Data Register(SIO3R) is a 8-bit shift register. First LSB is send or is received. The SPI3 allows 8-bits of data to be synchronously transmitted and received.

To accomplish communication, typically two pins are used:

- Serial3 Data Out	P21/SO3
- Serial3 Clock	P20/SCK3

.

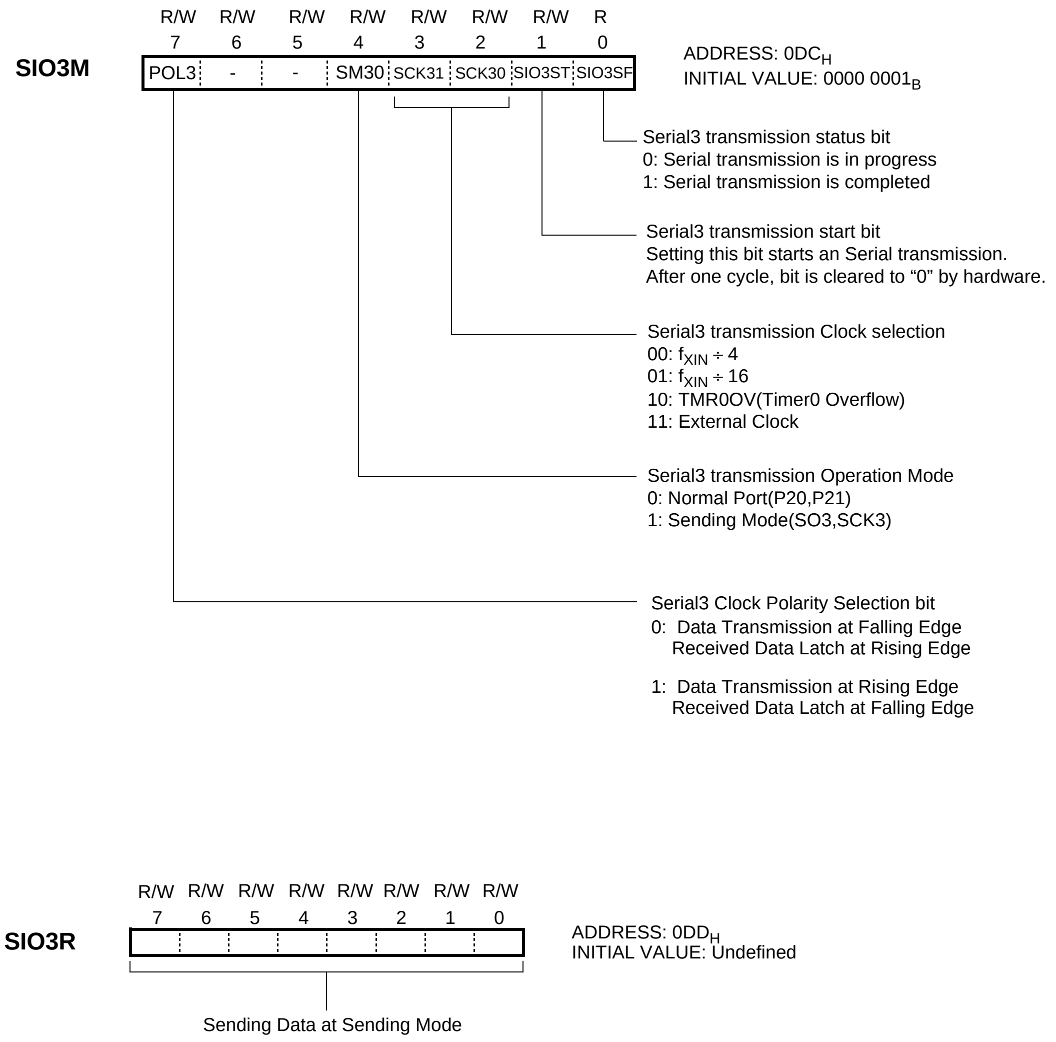

SIO3M
R/W R/W R/W R/W R/W R/W R/W R
7 6 5 4 3 2 1 0
POL3 - - SM30 SCK31 SCK30 SIO3ST SIO3SF
ADDRESS: 0DC$_H$
INITIAL VALUE: 0000 0001$_B$
Serial3 transmission status bit
0: Serial transmission is in progress
1: Serial transmission is completed
Serial3 transmission start bit
Setting this bit starts an Serial transmission.
After one cycle, bit is cleared to "0" by hardware.
Serial3 transmission Clock selection
00: $f_{XIN} \div 4$
01: $f_{XIN} \div 16$
10: TMR0OV(Timer0 Overflow)
11: External Clock
Serial3 transmission Operation Mode
0: Normal Port(P20,P21)
1: Sending Mode(SO3,SCK3)
Serial3 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
SIO3R
R/W R/W R/W R/W R/W R/W R/W R/W
7 6 5 4 3 2 1 0
ADDRESS: 0DD$_H$
INITIAL VALUE: Undefined
Sending Data at Sending Mode

Figure 17-2 SPI3 Control Register

17.1 Transmission/Receiving Timing

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

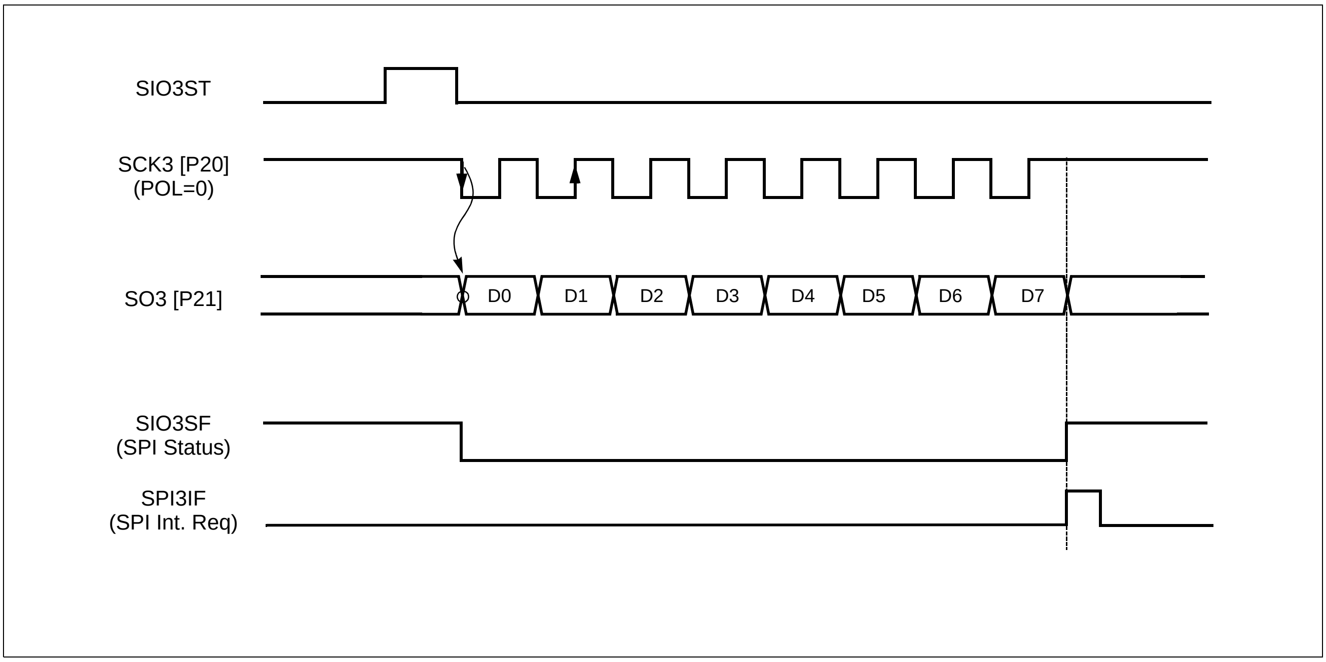


Figure 17-3 SPI3 Timing Diagram at POL=0

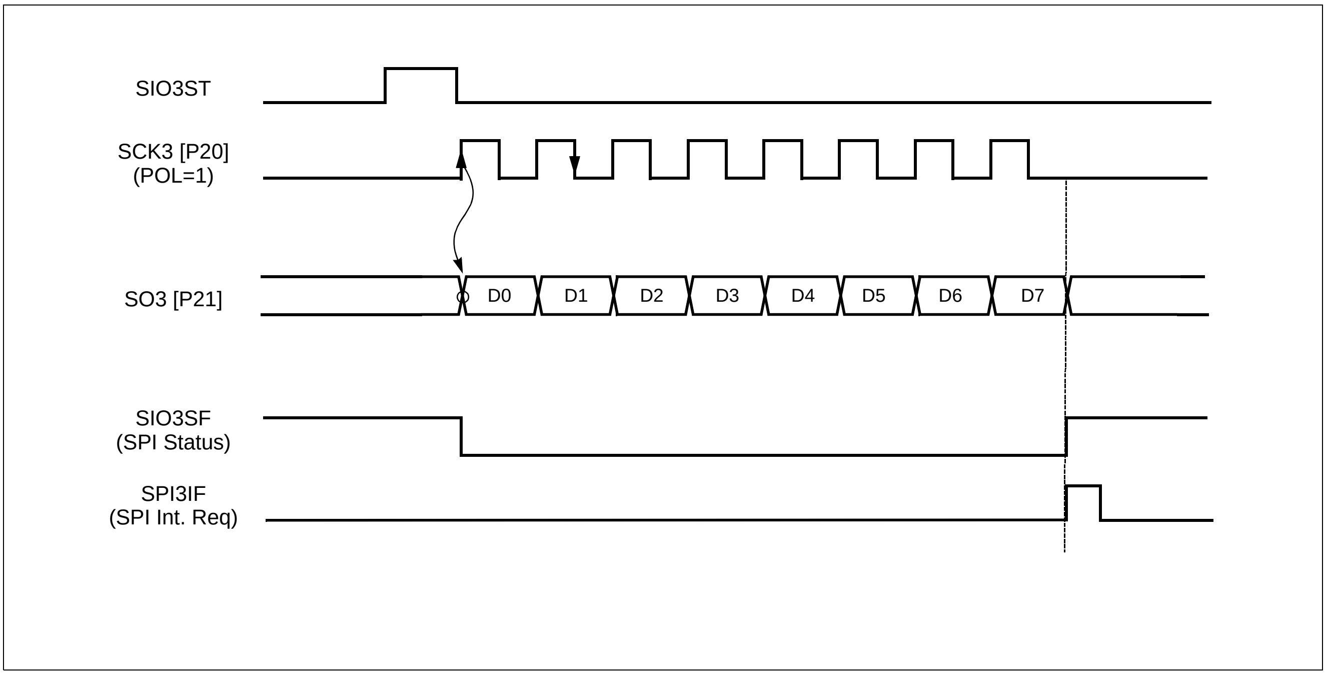


Figure 17-4 SPI3 Timing Diagram at POL=1

17.2 The method of Serial I/O

① Select transmission/receiving mode

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

② In case of sending mode, write data to be send to SIO3R.

③ Set SIO3ST to “1” to start serial transmission.

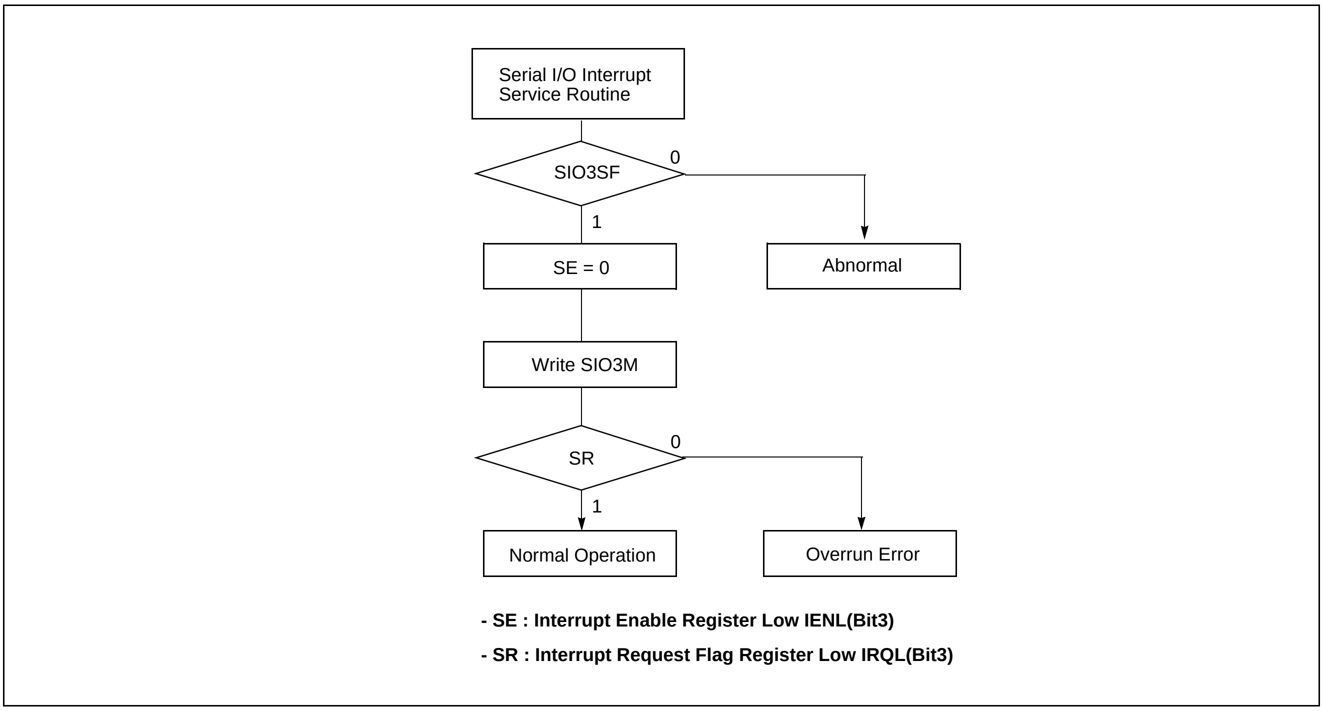
Note: *If both transmission mode is selected and transmission is performed simultaneously it would be made error.*

④ The SIO3 interrupt is generated at the completion of SIO3 and SIO3SF is set to “1”. In SIO3 interrupt service routine, correct transmission should be tested.

⑤ In case of receiving mode, the received data is acquired by reading the SIO3R.

```
LDM      SIO3R,#0AAh      ;SIO1R Initial
                          ;Value
LDM      SIO3M,#0001_1100b;SIO3M Select
         NOP
         NOP
         SIO3M,#0001_1110b;SIO3 Start
```

17.3 The Method to Test Correct Transmission



- SE : Interrupt Enable Register Low IENL(Bit3)

- SR : Interrupt Request Flag Register Low IRQL(Bit3)

Figure 17-5 Serial3 Method to Test Transmission

18. BUZZER FUNCTION

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

A 50% duty pulse can be output to R03/BUZO pin to use for piezo-electric buzzer drive. Pin R03 is assigned for output port of Buzzer driver by setting the bit 3 of R0FUNC(address 0F4$_H$) to “1”. At this time, the pin R03 must be defined as output mode (the bit 3 of R0IO=1).

Example: 5kHz output at 4MHz.

```
LDM    P0IO,#XXXX_X1XXB
LDM    BUR,#0011_0010B

LDM    PSR,#XXXX_X1XXB
```

X means don’t care

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

Equation of frequency calculation is shown below.

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

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

The frequency of output signal is controlled by the buzzer control register BUR.The bit 0 to bit 5 of BUR determine output frequency for buzzer driving.

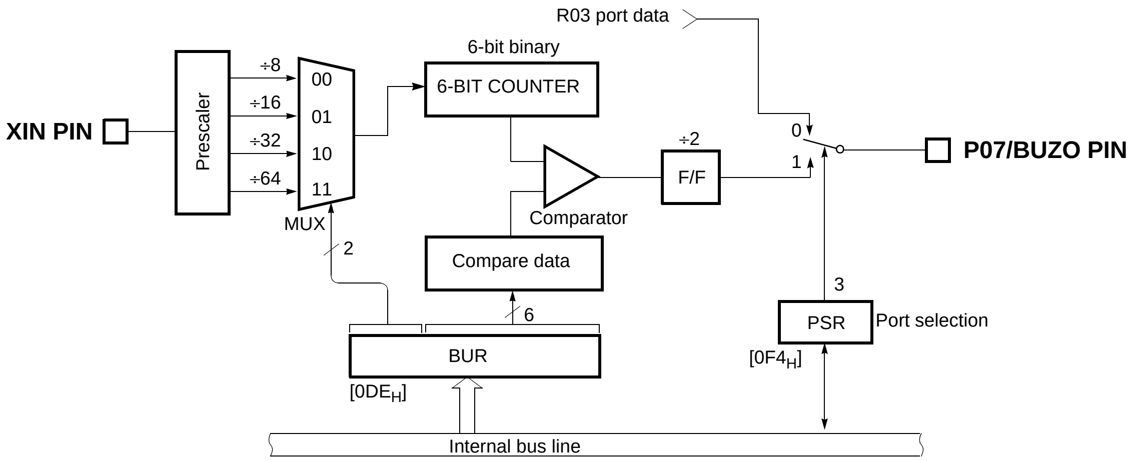


Figure 18-1 Block Diagram of Buzzer Driver

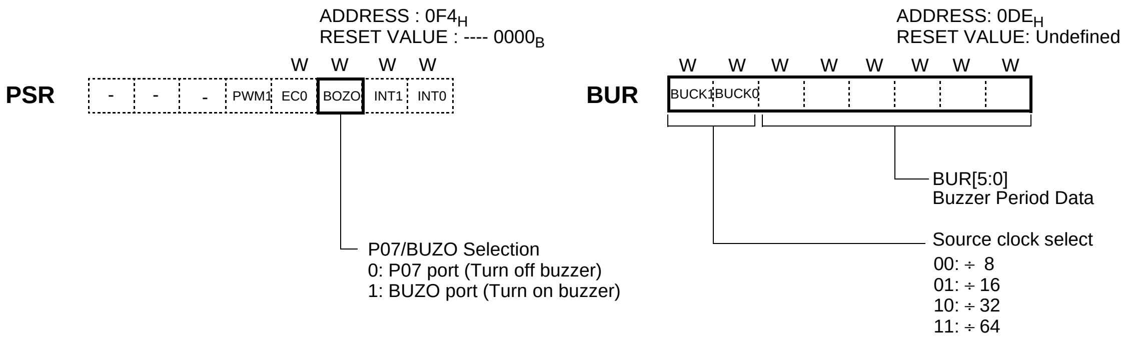


Figure 18-2 PSR and Buzzer Register

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

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

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

BUR [5:0]	**BUR[7:6]**			
	00	**01**	**10**	**11**
00	250.000	125.000	62.500	31.250
01	125.000	62.500	31.250	15.625
02	83.333	41.667	20.833	10.417
03	62.500	31.250	15.625	7.813
04	50.000	25.000	12.500	6.250
05	41.667	20.833	10.417	5.208
06	35.714	17.857	8.929	4.464
07	31.250	15.625	7.813	3.906
08	27.778	13.889	6.944	3.472
09	25.000	12.500	6.250	3.125
0A	22.727	11.364	5.682	2.841
0B	20.833	10.417	5.208	2.604
0C	19.231	9.615	4.808	2.404
0D	17.857	8.929	4.464	2.232
0E	16.667	8.333	4.167	2.083
0F	15.625	7.813	3.906	1.953
10	14.706	7.353	3.676	1.838
11	13.889	6.944	3.472	1.736
12	13.158	6.579	3.289	1.645
13	12.500	6.250	3.125	1.563
14	11.905	5.952	2.976	1.488
15	11.364	5.682	2.841	1.420
16	10.870	5.435	2.717	1.359
17	10.417	5.208	2.604	1.302
18	10.000	5.000	2.500	1.250
19	9.615	4.808	2.404	1.202
1A	9.259	4.630	2.315	1.157
1B	8.929	4.464	2.232	1.116
1C	8.621	4.310	2.155	1.078
1D	8.333	4.167	2.083	1.042
1E	8.065	4.032	2.016	1.008
1F	7.813	3.906	1.953	0.977

BUR [5:0]	**BUR[7:6]**			
	00	**01**	**10**	**11**
20	7.576	3.788	1.894	0.947
21	7.353	3.676	1.838	0.919
22	7.143	3.571	1.786	0.893
23	6.944	3.472	1.736	0.868
24	6.757	3.378	1.689	0.845
25	6.579	3.289	1.645	0.822
26	6.410	3.205	1.603	0.801
27	6.250	3.125	1.563	0.781
28	6.098	3.049	1.524	0.762
29	5.952	2.976	1.488	0.744
2A	5.814	2.907	1.453	0.727
2B	5.682	2.841	1.420	0.710
2C	5.556	2.778	1.389	0.694
2D	5.435	2.717	1.359	0.679
2E	5.319	2.660	1.330	0.665
2F	5.208	2.604	1.302	0.651
30	5.102	2.551	1.276	0.638
31	5.000	2.500	1.250	0.625
32	4.902	2.451	1.225	0.613
33	4.808	2.404	1.202	0.601
34	4.717	2.358	1.179	0.590
35	4.630	2.315	1.157	0.579
36	4.545	2.273	1.136	0.568
37	4.464	2.232	1.116	0.558
38	4.386	2.193	1.096	0.548
39	4.310	2.155	1.078	0.539
3A	4.237	2.119	1.059	0.530
3B	4.167	2.083	1.042	0.521
3C	4.098	2.049	1.025	0.512
3D	4.032	2.016	1.008	0.504
3E	3.968	1.984	0.992	0.496
3F	3.907	1.953	0.977	0.488

19. FIP CONTROLLER/DRIVER

19.1 Function of FIP Controller/Driver

The FIP controller/driver of the HMS81C2232/48 has the following functions.

(1) Can output display signals (DMA operation) by automatically reading display data.

(2) The pins not used for FIP display can be used as I/O port or output port pins (FIP24 through FIP52 pins only).

(3) Luminance can be adjusted in 8 steps by display mode register 1 (DSPM1).

(4) Hardware for key scan application

• Generates an interrupt signal (INTKS) indicating key scan timing

• Timing in which key scan data is output can be detected by key scan flag (KSF).

• Whether key scan timing is inserted or not can be selected.

(5) High-voltage output buffer that can directly drive FIP.

(6) FIP0 through FIP52 pins can be connected to pull-down resistors by mask option (mask ROM model only). The HMS87C2232/48 does not have pull-down resistors)

Of the 53 FIP output pins of the HMS81C2232/48, FIP24 through FIP52 are multiplexed with port pins. FIP0 through FIP23 are dedicated output pins.

FIP24 through FIP52 can be used as port pins when FIP display is disabled by bit 7 (DSPEN) of the display mode

register 0 (DSPM0). Even when FIP display is enabled, the FIP output pins not used for display signal output can e used as port pins.

FIP Pin Name	Multiplexed Port Name	I/O
FIP24-FIP31	P30-P37	Output only port
FIP32-FIP39	P40-P47	Output only port
FIP40-FIP47	P50-P57	I/O port
FIP48-FIP52	P60-P64	I/O port

Table 19-1 FIP Output Pins and Multiplexed Port Pins

19.2 Configuration of FIP Controller/Driver

The FIP controller/driver consists of the following hardware.

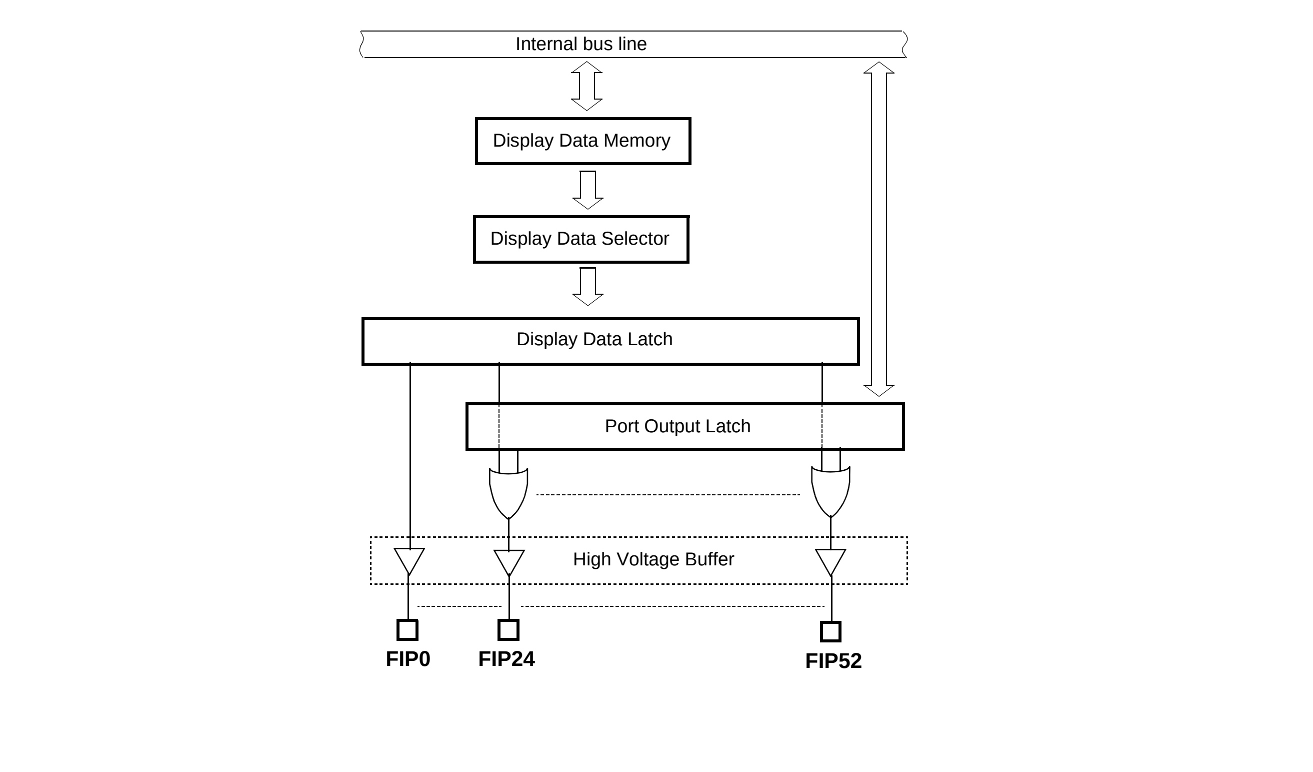


Figure 19-1 . Block Diagram of FIP Controller/Driver

19.3 Registers Controlling FIP Controller/ Driver

The following three types of registers control the FIP controller/ driver.

- Display Mode Register 0 (DSPM0)
- Display Mode Register 1 (DSPM1)
- Display Mode Register 2 (DSPM2)

Item	Configuration
Display	53
Control Register	Display mode register 0 (DSPM0) Display mode register 1 (DSPM1) Display mode register 2 (DSPM2)

Table 19-2 Configuration of FIP Controller/Driver

(1) Display mode register 0 (DSPM0)

DSPM0 performs the following setting.

- Enables or disables display
- Number of FIP output pins

DSPM0 is set by using a 1-bit or 8-bit memory manipulation instruction. The value of this register is set to 10_H by $\overline{\text{RESET}}$ input.

(2) Display mode register 1 (DSPM1)

DSPM1 performs the following setting:

- Blanking width of FIP output signal
- Number of display patterns

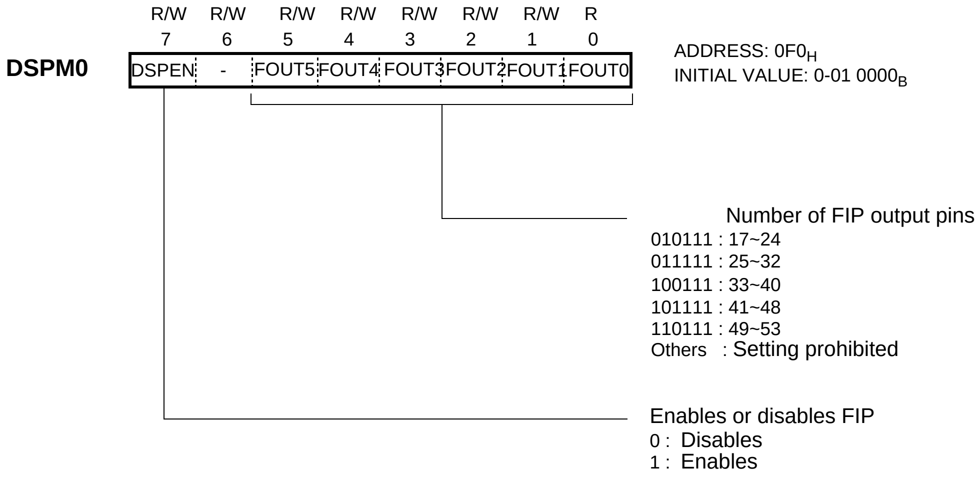
DSPM1 is set by using a 1-bit or 8-bit memory manipulation instruction. The value of this register is set to 01_H by $\overline{\text{RESET}}$ input.

(3) Display mode register 2 (DSPM2)

DSPM2 performs the following setting. It also indicates the status of the display timing/key scan.

- Insertion of key scan timing
- Display cycle (TDSP)

DSPM2 is set by using a 1-bit or 8-bit memory manipulation instruction. However, only bit 7 (KSF) can be read by a 1-bit memory manipulation instruction. The value of this register is initialized to 00H by $\overline{\text{RESET}}$ input.


R/W R/W R/W R/W R/W R/W R/W R
7 6 5 4 3 2 1 0
DSPM0
DSPEN - FOUT5 FOUT4 FOUT3 FOUT2 FOUT1 FOUT0
ADDRESS: 0F0H
INITIAL VALUE: 0-01 0000B

Number of FIP output pins
010111 : 17~24
011111 : 25~32
100111 : 33~40
101111 : 41~48
110111 : 49~53
Others : Setting prohibited

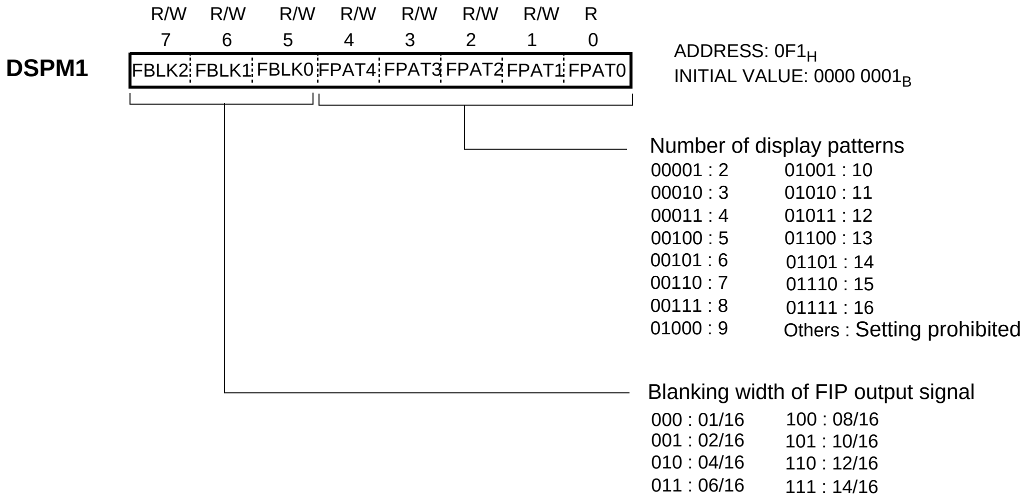
Enables or disables FIP
0 : Disables
1 : Enables

Cautions : 1. Be sure to set bit 6 to “0”.

2. Do not write data to the bits other than DSPEN when bit 7 (DSPEN) = 1.

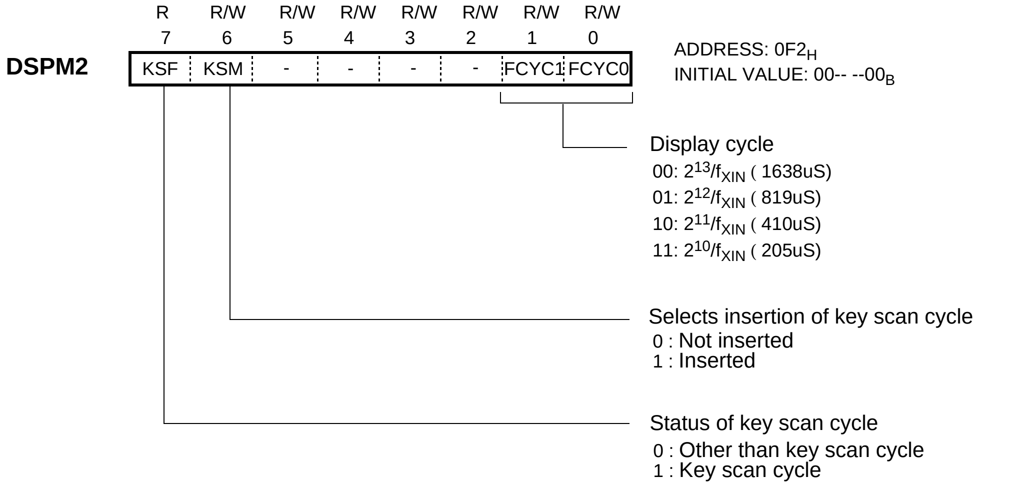
3. Be sure to set the output latch of the multiplexed port of a pin used for FIP output to “0”.

Figure 19-2 Format of Display Mode Register 0



Cautions : **Do not write data to the display mode register 1 (DSPM1) when bit 7 (DSPEN) of the display mode register 0 (DSPM0) is 1.**

Figure 19-3 Format of Display Mode Register 1


R R/W R/W R/W R/W R/W R/W R/W
7 6 5 4 3 2 1 0
DSPM2
KSF KSM - - - - FCYC1 FCYC0
ADDRESS: 0F2$_H$
INITIAL VALUE: 00-- --00$_B$
Display cycle
00: $2^{13}/f_{XIN}$ (1638uS)
01: $2^{12}/f_{XIN}$ (819uS)
10: $2^{11}/f_{XIN}$ (410uS)
11: $2^{10}/f_{XIN}$ (205uS)
Selects insertion of key scan cycle
0 : Not inserted
1 : Inserted
Status of key scan cycle
0 : Other than key scan cycle
1 : Key scan cycle

Cautions : **1. Be sure to set bits 2 through 5 to “0”.**

2. Do not write data to the display mode register 2 (DSPM2) when bit 7 (DSPEN) of the display mode register 0 (DSPM0) is 1.

Remarks : **1.** fx: Main system clock oscillation frequency

2. ( ): fx = 5.0 MHz

Figure 19-4 Format of Display Mode Register 2

19.4 One display period and blanking width

The FIP output signals are blanked equally at the beginning and end of the display period by the blanking width set by bits 0 through 2 (FBLK0 through FBLK2) of the display mode register 1 (DSPM1).

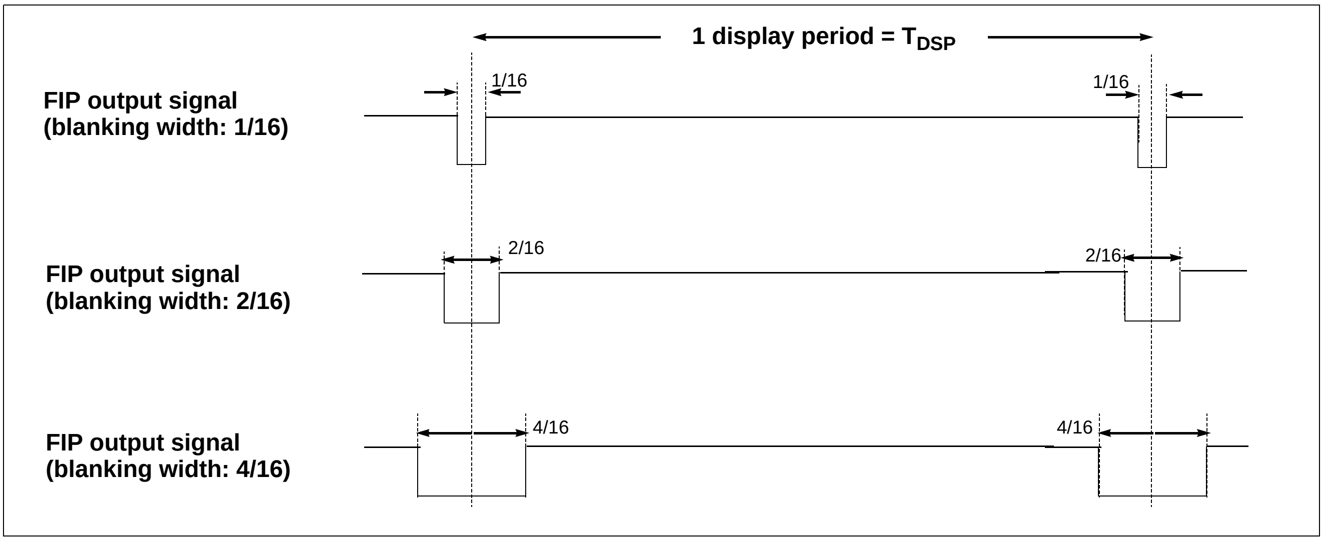
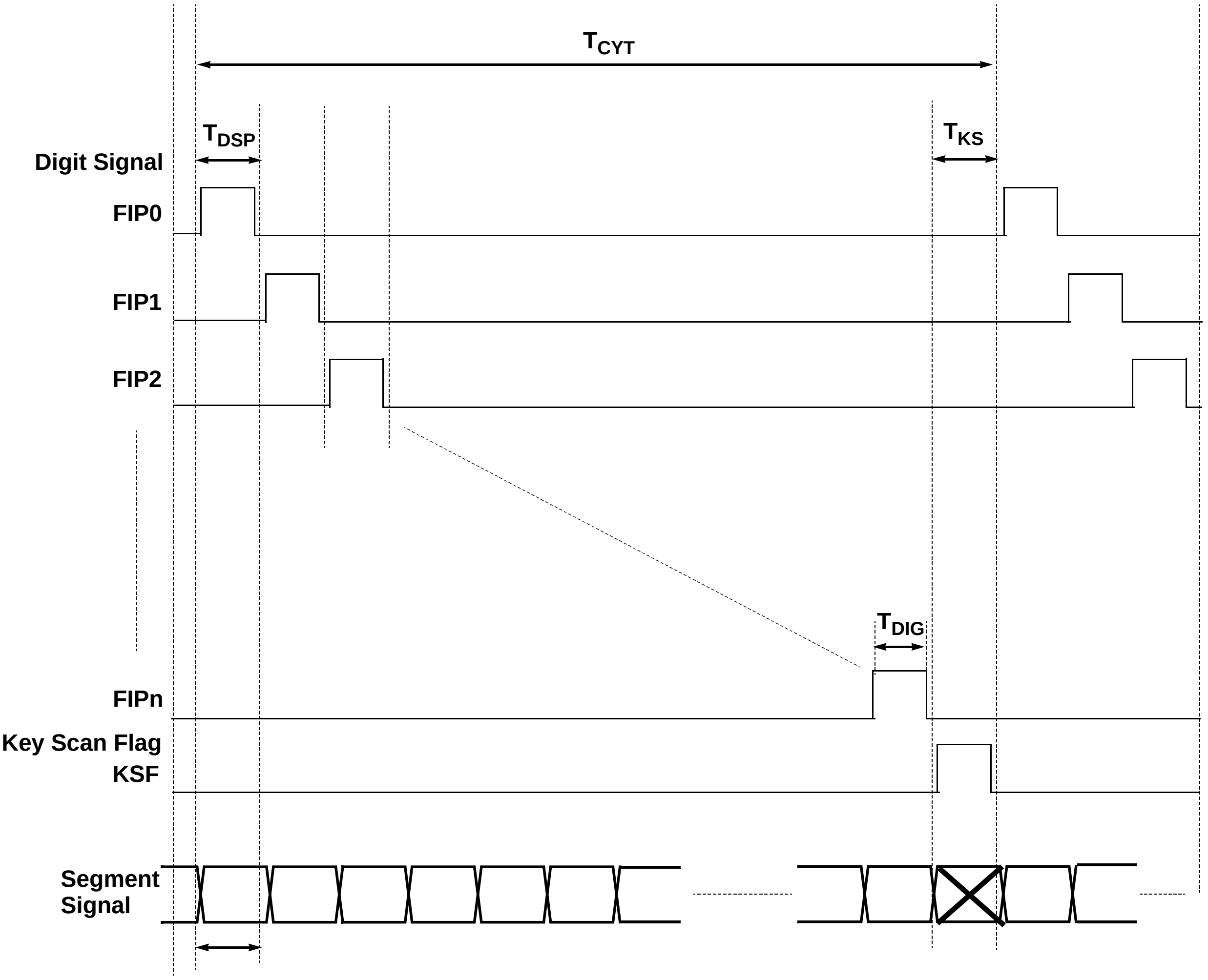

1 display period = T_{DSP}
FIP output signal (blanking width: 1/16)
1/16
1/16
FIP output signal (blanking width: 2/16)
2/16
2/16
FIP output signal (blanking width: 4/16)
4/16
4/16

Figure 19-5 Blanking Width of FIP Output Signal



n : Displayed Digit -1 (Digits 2 to 16 can be selected using the DSPM1)

T_{DSP} : 1 Display Cycle

T_{CYT} : Display Cycle (T_{CYT} = T_{DSP} X Displayed Digit + 1)

T_{KS} : Key Scan Timing (T_{KS} = T_{DSP})

T_{DIG} : Width of Digit Singanl Pulse

Figure 19-6 VFD Output Operation Timings in Display Mode

19.5 Display Data Memory

The display data memory is a 112-byte RAM area that stores data to be displayed, and is mapped to addresses 0400_H through $046F_H$. The FIP controller reads the data stored in the display data memory independently of the CPU operation for FIP display (DMA operation).

The area of the display data memory not used for display can be used as a normal RAM area.

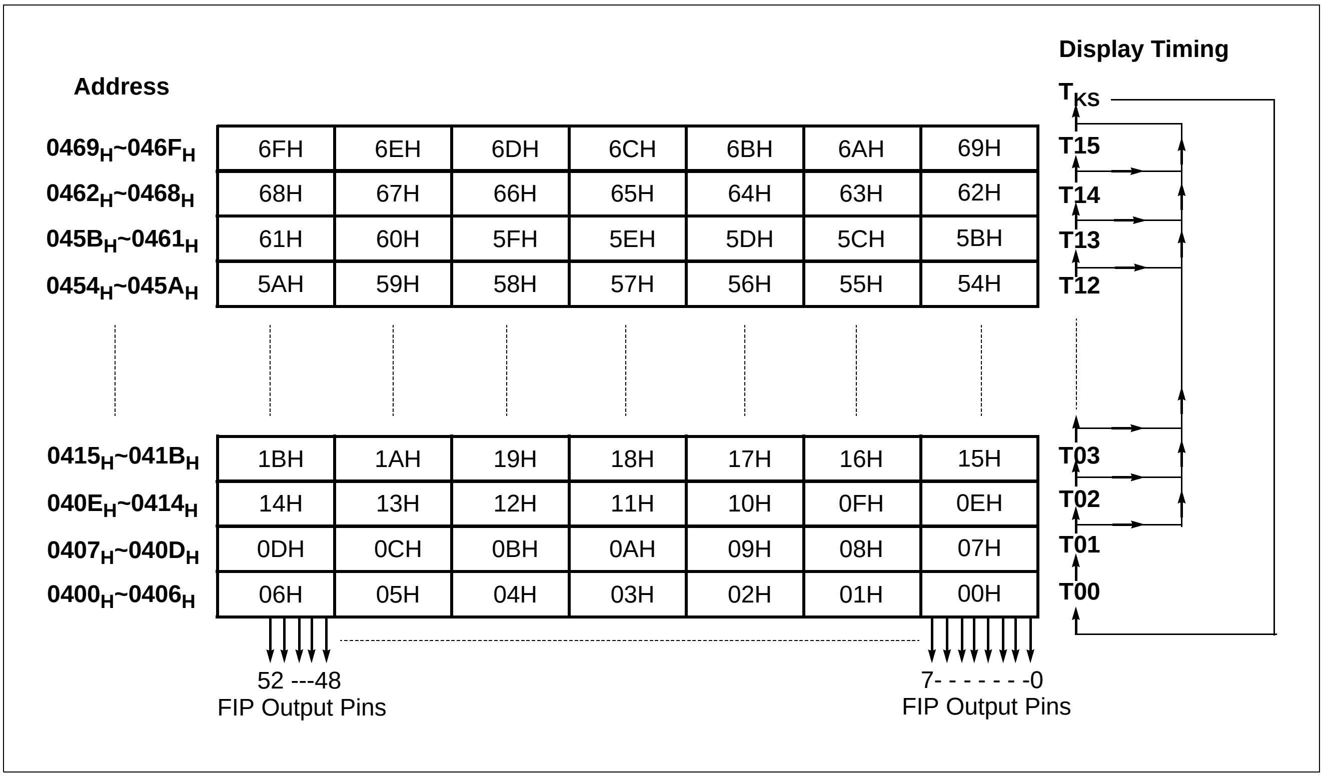
At key scan timing (T_{KS}), all the FIP output pins are cleared to “0”, and the data of the output latches of ports 3 through 6 are output to FIP24/P30 through FIP52/P64.

The address location of the display data memory is as follows:

- **With 53 FIP output pins and 16 patterns**

The addresses of the display data memory corresponding to the data output at each display timing (T0 through T15) are as shown in Figure 19-7 (for example, T0 = 0400_H through 0406_H, and T1 = 0407_H through $040D_H$).

When 53 FIP output pins (FIP0 through FIP52) are used, one block of display data consists of 7 bytes. FIP output pins 0 (FIP0) through 52 (FIP52) correspond to one block of display data sequentially, starting from the least significant bit toward the most significant bit.



Address								Display Timing
								T_{KS}
0469_H~$046F_H$	6FH	6EH	6DH	6CH	6BH	6AH	69H	T15
0462_H~0468_H	68H	67H	66H	65H	64H	63H	62H	T14
$045B_H$~0461_H	61H	60H	5FH	5EH	5DH	5CH	5BH	T13
0454_H~$045A_H$	5AH	59H	58H	57H	56H	55H	54H	T12
0415_H~$041B_H$	1BH	1AH	19H	18H	17H	16H	15H	T03
$040E_H$~0414_H	14H	13H	12H	11H	10H	0FH	0EH	T02
0407_H~$040D_H$	0DH	0CH	0BH	0AH	09H	08H	07H	T01
0400_H~0406_H	06H	05H	04H	03H	02H	01H	00H	T00

Figure 19-7 Relation between Address Location of Display Data Memory and FIP Output HMS81C2232/48 (with 53 FIP output pins and 16 patterns)

19.6 Key Scan Flag and Key Scan Data

Key scan flag

The key scan flag (KSF) is set to 1 during key scan timing, and is automatically reset to 0 at display timing.

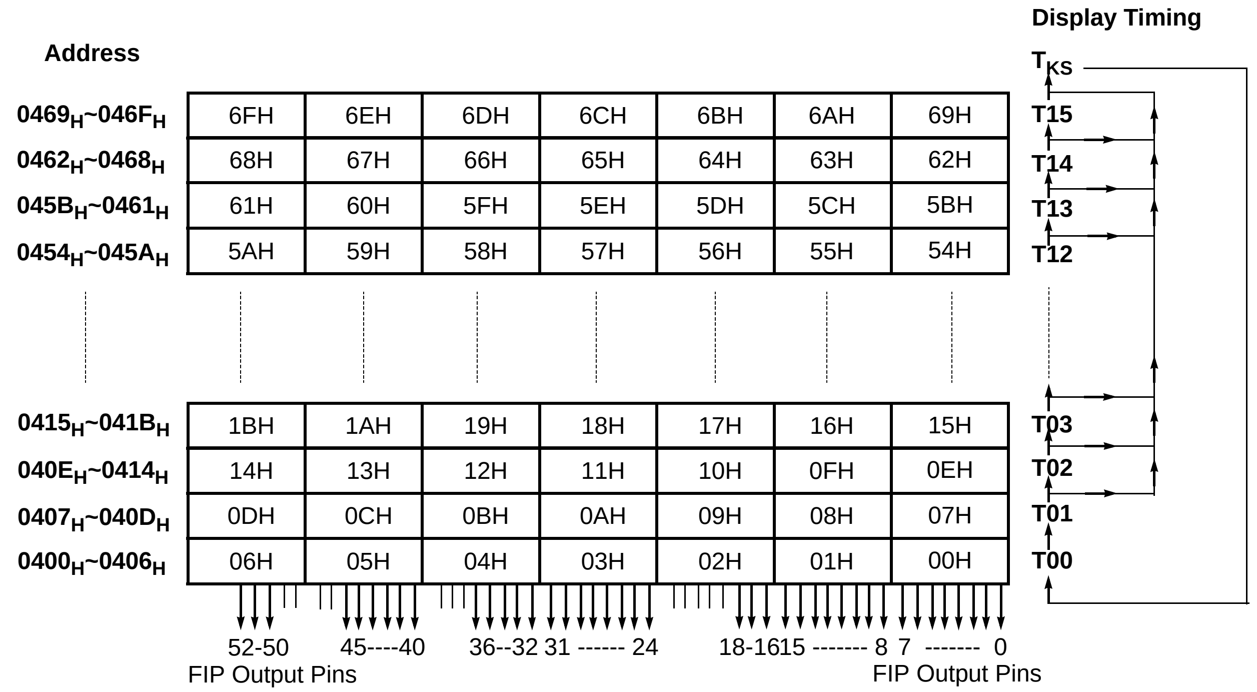
KSF is mapped to bit 7 of the display mode register 2 (DSPM2) and can be tested in 1-bit units. It cannot be written, however. By testing KSF, it can be determined whether key scan timing is in progress, and whether key input data is correct can be checked.

Whether key scan timing is inserted or not can be selected by using the key scan timing insertion specification flag(KSM) (bit 6 of the display mode register 2 (DSPM2)).

Key scan data

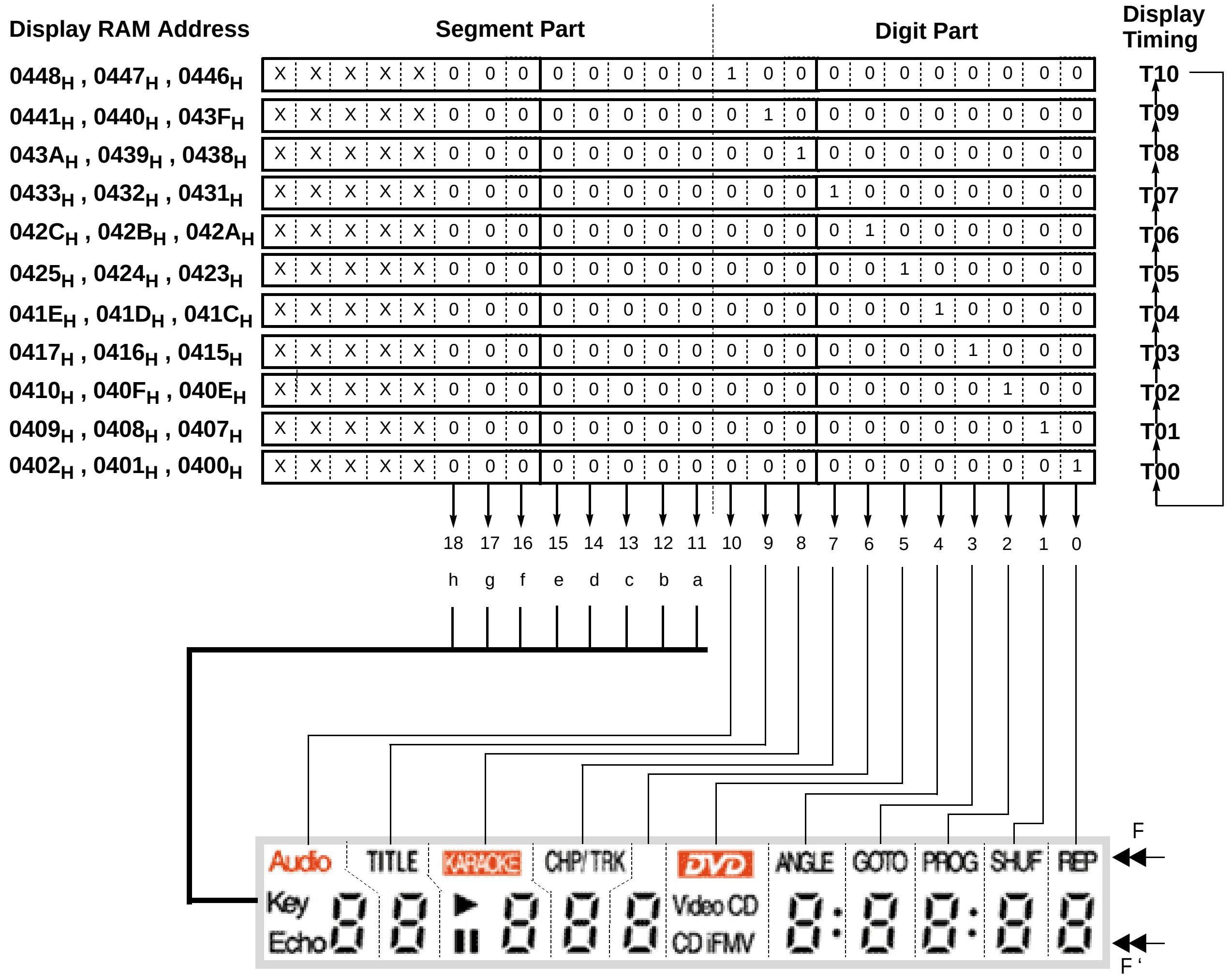
Data stored to ports 3 through 6 are output from the FIP24 through FIP52 pins during key scan timing.

Note: *If scanning is performed in such a manner that both a segment and a digit turn ON during keyscan timing, the display may flicker.*



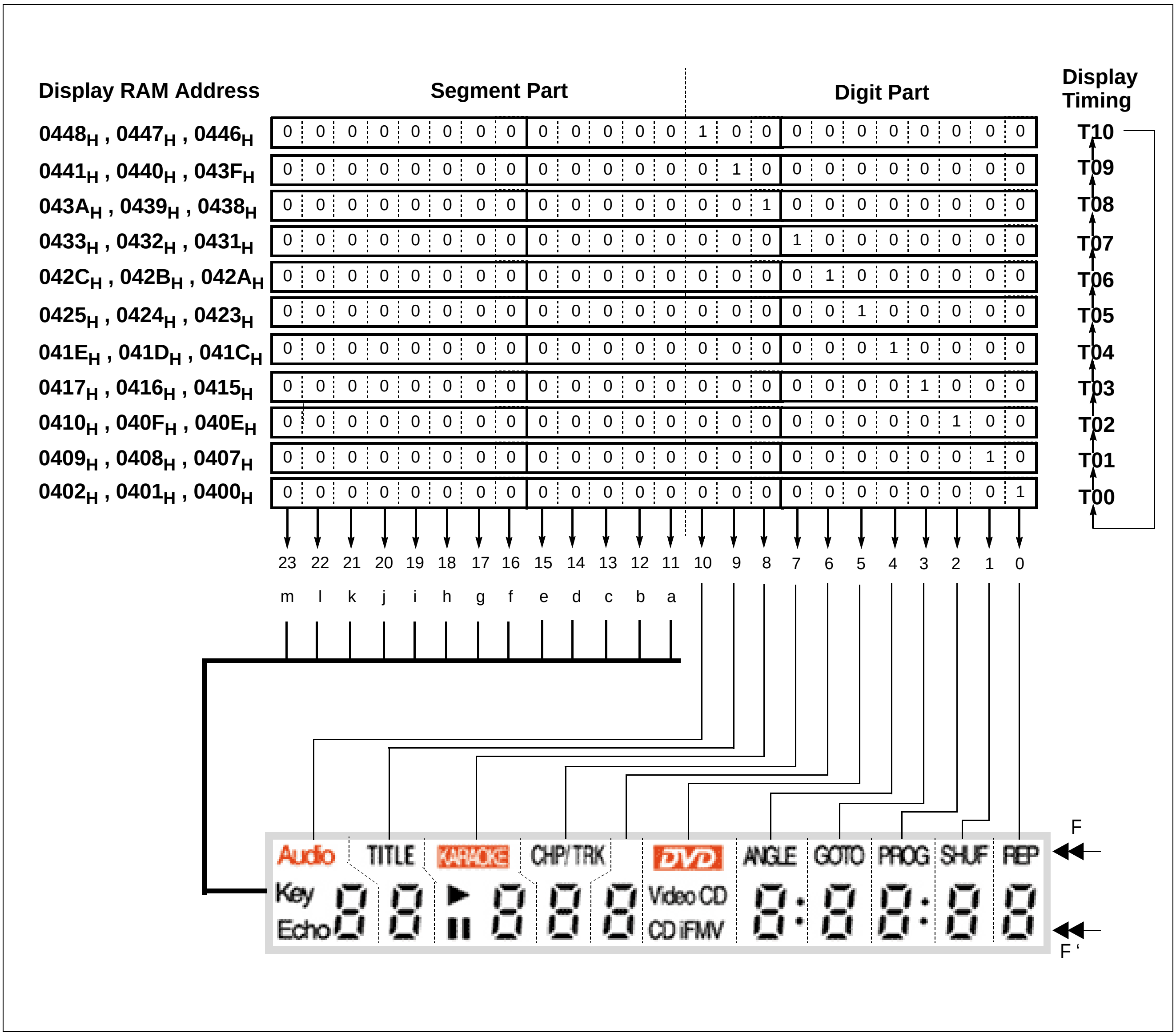
Address							
0469_H~$046F_H$	6FH	6EH	6DH	6CH	6BH	6AH	69H
0462_H~0468_H	68H	67H	66H	65H	64H	63H	62H
$045B_H$~0461_H	61H	60H	5FH	5EH	5DH	5CH	5BH
0454_H~$045A_H$	5AH	59H	58H	57H	56H	55H	54H
0415_H~$041B_H$	1BH	1AH	19H	18H	17H	16H	15H
$040E_H$~0414_H	14H	13H	12H	11H	10H	0FH	0EH
0407_H~$040D_H$	0DH	0CH	0BH	0AH	09H	08H	07H
0400_H~0406_H	06H	05H	04H	03H	02H	01H	00H

Figure 19-8 Relation between Address Location of Display Data Memory and FIP Output HMS81C2332/48 (with 41 FIP output pins and 16 patterns)



Display RAM Address						18 h	17 g	16 f	15 e	14 d	13 c	12 b	11 a	10	9	8	7	6	5	4	3	2	1	0	Display Timing
0448_H , 0447_H , 0446_H	X	X	X	X	X	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T10
0441_H , 0440_H , $043F_H$	X	X	X	X	X	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	T09
$043A_H$, 0439_H , 0438_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	T08
0433_H , 0432_H , 0431_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T07
$042C_H$, $042B_H$, $042A_H$	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T06
0425_H , 0424_H , 0423_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	T05
$041E_H$, $041D_H$, $041C_H$	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	T04
0417_H , 0416_H , 0415_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	T03
0410_H , $040F_H$, $040E_H$	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	T02
0409_H , 0408_H , 0407_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	T01
0402_H , 0401_H , 0400_H	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	T00

Figure 19-9 Relationship between Display Data Memory and FIP Output with 8 Segments-11 Digits Displayed



Display RAM Address	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Display Timing
	m	l	k	j	i	h	g	f	e	d	c	b	a												
0448_H , 0447_H , 0446_H	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T10
0441_H , 0440_H , $043F_H$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	T09
$043A_H$, 0439_H , 0438_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	T08
0433_H , 0432_H , 0431_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T07
$042C_H$, $042B_H$, $042A_H$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T06
0425_H , 0424_H , 0423_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	T05
$041E_H$, $041D_H$, $041C_H$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	T04
0417_H , 0416_H , 0415_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	T03
0410_H , $040F_H$, $040E_H$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	T02
0409_H , 0408_H , 0407_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	T01
0402_H , 0401_H , 0400_H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	T00

Figure 19-10 Relationship between Display Data Memory and FIP Output with 13 Segments-11 Digits Displayed

20. INTERRUPTS

The HMS81C2232/48 interrupt circuits consist of Interrupt enable register (IENH, IENL), Interrupt request flags of IRQH, IRQL, Priority circuit, and Master enable flag ("I" flag of PSW). Nine interrupt sources are provided. The configuration of interrupt circuit is shown in Figure 20-2.

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

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

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

The interrupts are controlled by the interrupt master enable flag I-flag (bit 2 of PSW on page 27), the interrupt enable register (IENH, IENL), and the interrupt request flags (in IRQH and IRQL) except Power-on reset and software BRK interrupt. Below table shows the Interrupt priority.

Reset/Interrupt	Symbol	Priority
Hardware Reset	RESET	-
External Interrupt 0	INTP0	1
External Interrupt 1	INTP1	2
Remote Timer Rising	RTR	3
Remote Timer Falling	RTF	4
Remte Timer Overflow	RTO	5
Key Scan Interrupt	KS	6
SIO1 Interrupt	SIO1	7
SIO3 Interrupt	SIO3	8
Timer/Counter 0	TIMER0	9
Timer/Counter 1	TIMER1	10
ADC Interrupt	ADC	11
Watchdog Timer	WDT	12
Basic Interval Timer	BIT	13

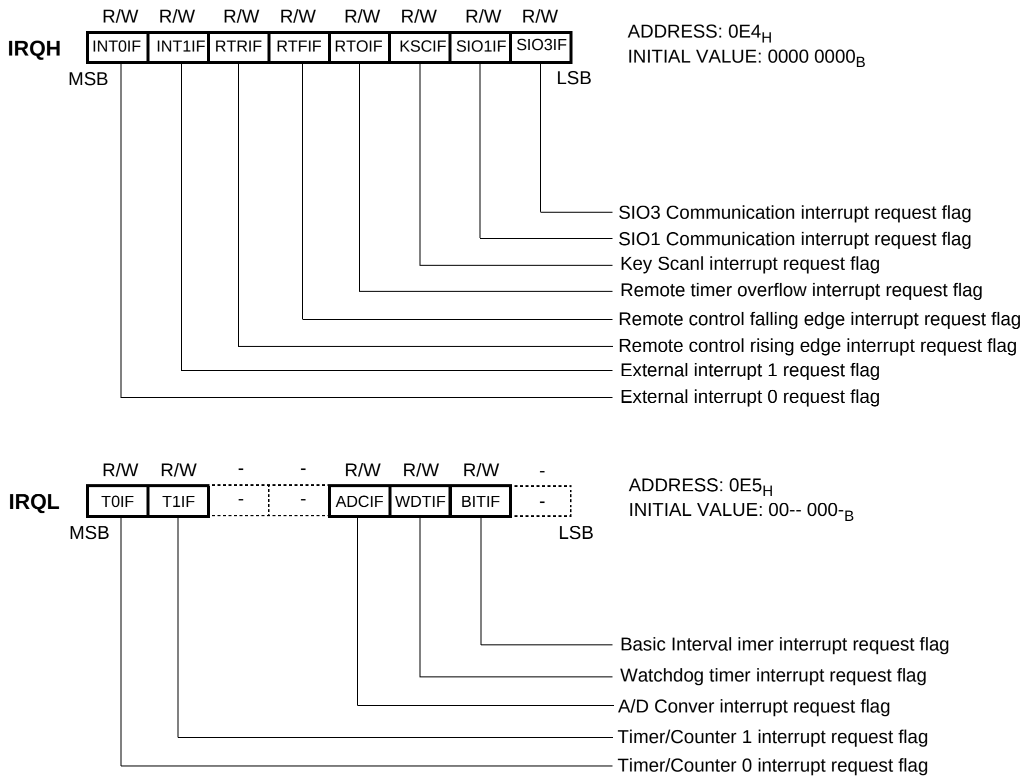

R/W R/W R/W R/W R/W R/W R/W R/W
IRQH
INT0IF INT1IF RTRIF RTFIF RTOIF KSCIF SIO1IF SIO3IF
MSB
LSB
ADDRESS: 0E4H
INITIAL VALUE: 0000 0000B
SIO3 Communication interrupt request flag
SIO1 Communication interrupt request flag
Key Scanl interrupt request flag
Remote timer overflow interrupt request flag
Remote control falling edge interrupt request flag
Remote control rising edge interrupt request flag
External interrupt 1 request flag
External interrupt 0 request flag
R/W R/W - - R/W R/W R/W -
IRQL
T0IF T1IF - - ADCIF WDTIF BITIF -
MSB
LSB
ADDRESS: 0E5H
INITIAL VALUE: 00-- 000-B
Basic Interval imer interrupt request flag
Watchdog timer interrupt request flag
A/D Conver interrupt request flag
Timer/Counter 1 interrupt request flag
Timer/Counter 0 interrupt request flag

Figure 20-1 Interrupt Request Flag

Vector addresses are shown in Figure 8-6 on page 29. Interrupt enable registers are shown in Figure 20-3. These registers are composed of interrupt enable flags of each interrupt source and these flags determines whether an interrupt will be accepted or not. When enable flag is “0”, a corresponding interrupt source is prohibited. Note that PSW contains also a master enable bit, I-flag, which disables all interrupts at once.

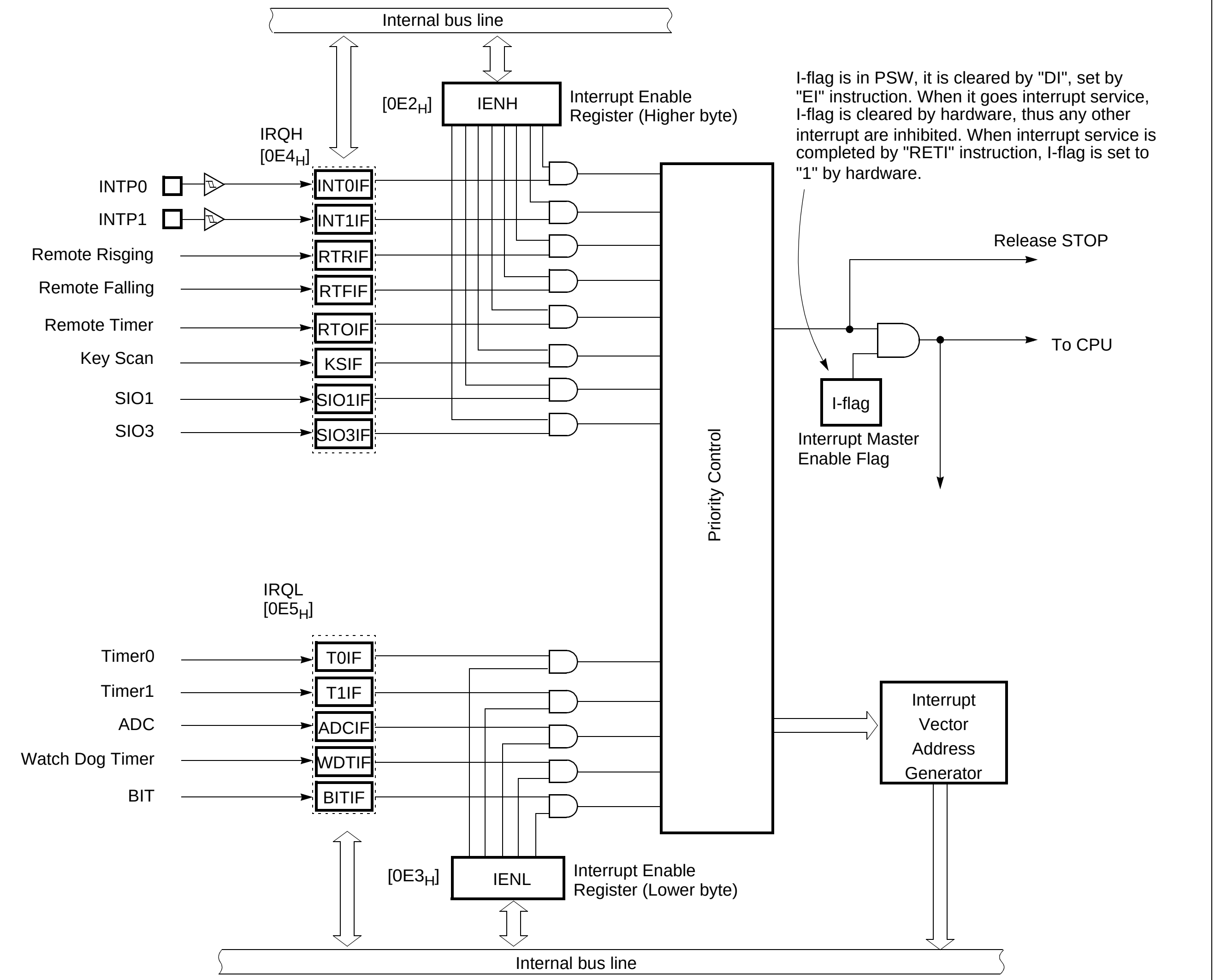

Internal bus line
[0E2H]
IENH
Interrupt Enable Register (Higher byte)
IRQH [0E4H]
I-flag is in PSW, it is cleared by "DI", set by "EI" instruction. When it goes interrupt service, I-flag is cleared by hardware, thus any other interrupt are inhibited. When interrupt service is completed by "RETI" instruction, I-flag is set to "1" by hardware.
INTP0
INTP1
Remote Risging
Remote Falling
Remote Timer
Key Scan
SIO1
SIO3
INT0IF
INT1IF
RTRIF
RTFIF
RTOIF
KSIF
SIO1IF
SIO3IF
Release STOP
To CPU
I-flag
Interrupt Master Enable Flag
Priority Control
IRQL [0E5H]
Timer0
Timer1
ADC
Watch Dog Timer
BIT
T0IF
T1IF
ADCIF
WDTIF
BITIF
Interrupt Vector Address Generator
[0E3H]
IENL
Interrupt Enable Register (Lower byte)
Internal bus line

Figure 20-2 Block Diagram of Interrupt

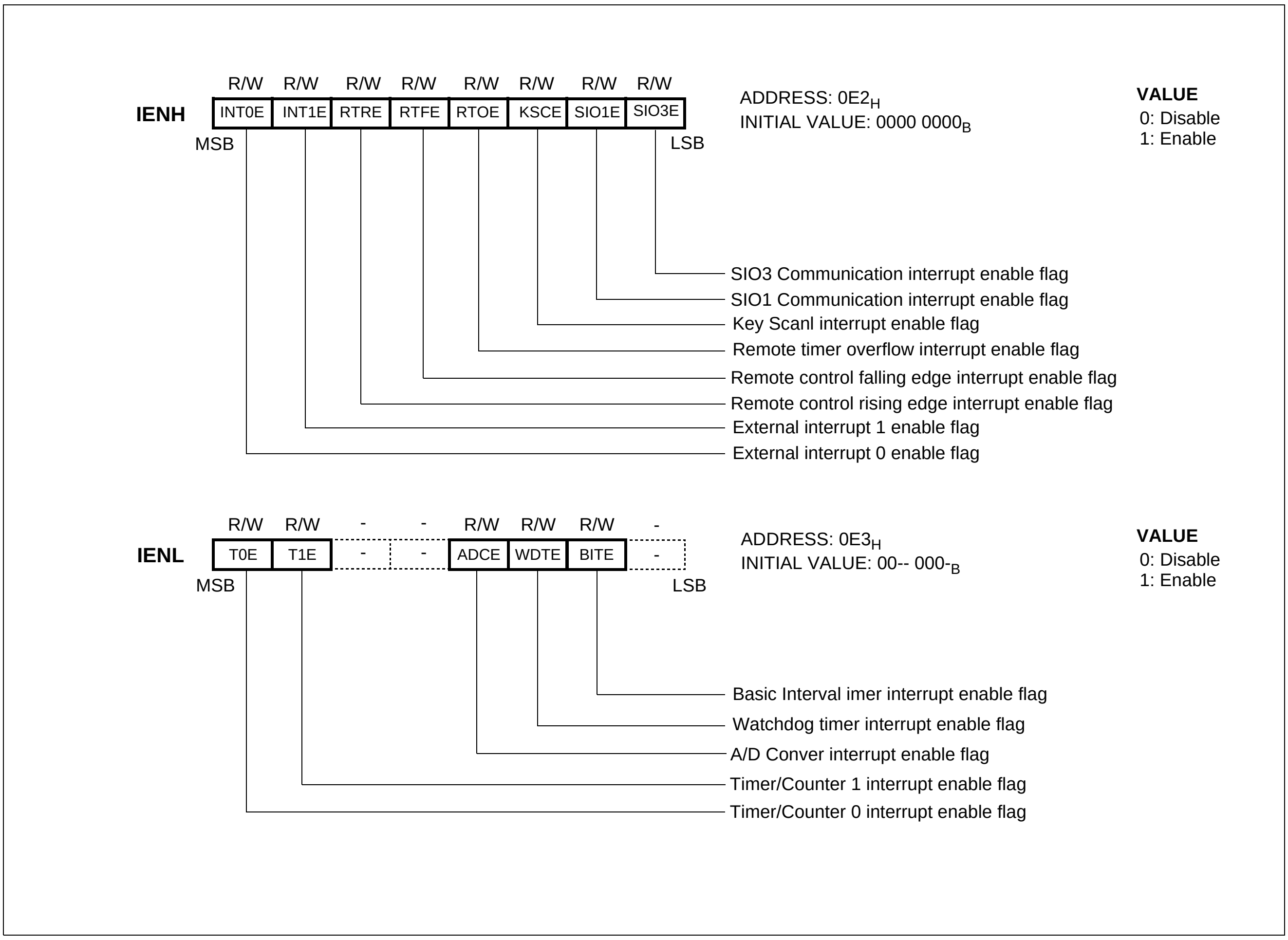

R/W R/W R/W R/W R/W R/W R/W R/W
IENH
INT0E
INT1E
RTRE
RTFE
RTOE
KSCE
SIO1E
SIO3E
MSB
LSB
ADDRESS: 0E2$_H$
INITIAL VALUE: 0000 0000$_B$
VALUE
0: Disable
1: Enable
SIO3 Communication interrupt enable flag
SIO1 Communication interrupt enable flag
Key Scanl interrupt enable flag
Remote timer overflow interrupt enable flag
Remote control falling edge interrupt enable flag
Remote control rising edge interrupt enable flag
External interrupt 1 enable flag
External interrupt 0 enable flag
R/W R/W - - R/W R/W R/W -
IENL
T0E
T1E
-
-
ADCE
WDTE
BITE
-
MSB
LSB
ADDRESS: 0E3$_H$
INITIAL VALUE: 00-- 000-$_B$
VALUE
0: Disable
1: Enable
Basic Interval imer interrupt enable flag
Watchdog timer interrupt enable flag
A/D Conver interrupt enable flag
Timer/Counter 1 interrupt enable flag
Timer/Counter 0 interrupt enable flag

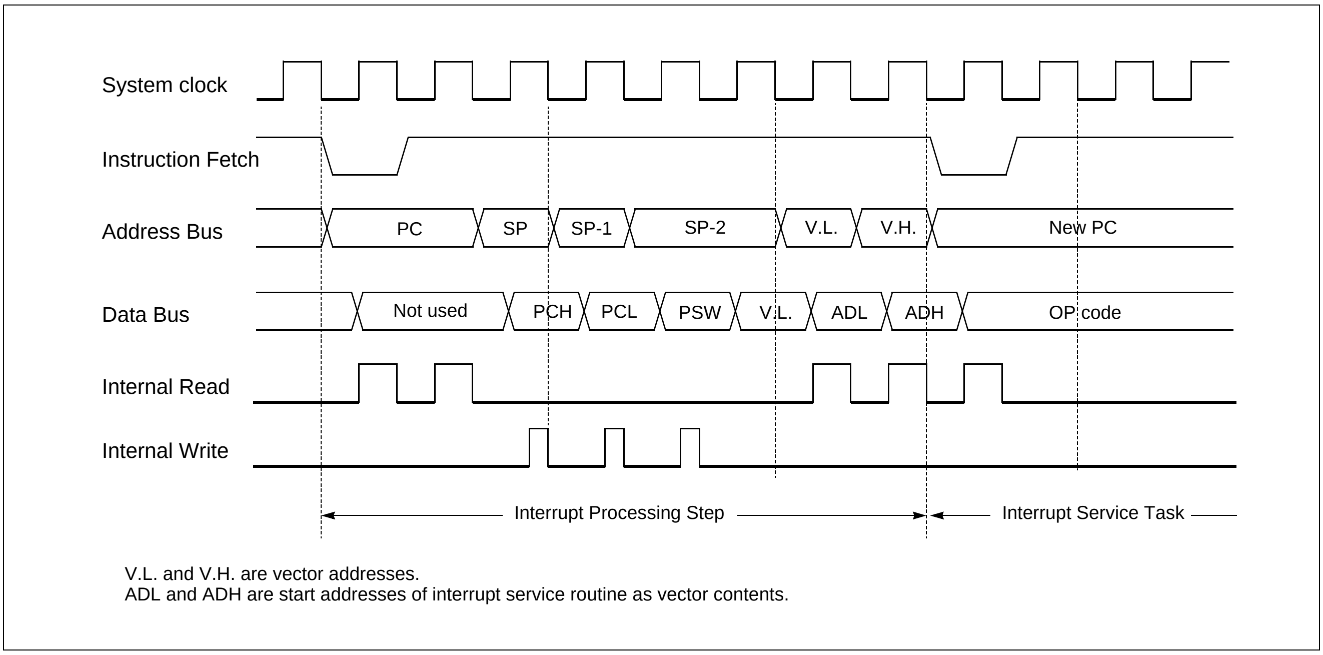
Figure 20-3 Interrupt Enable Flag

20.1 Interrupt Sequence

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

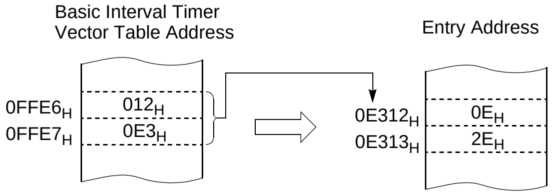
Interrupt acceptance

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



V.L. and V.H. are vector addresses.
ADL and ADH are start addresses of interrupt service routine as vector contents.

Figure 20-4 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



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

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

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

Saving/Restoring General-purpose Register

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

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

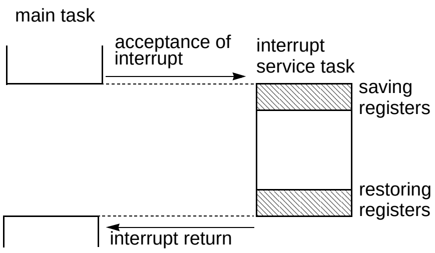
Example: Register save using push and pop instructions

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

        interrupt processing

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

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



20.2 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 20-5.

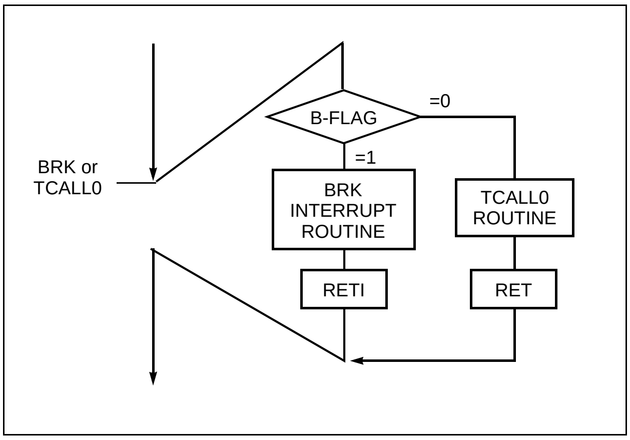
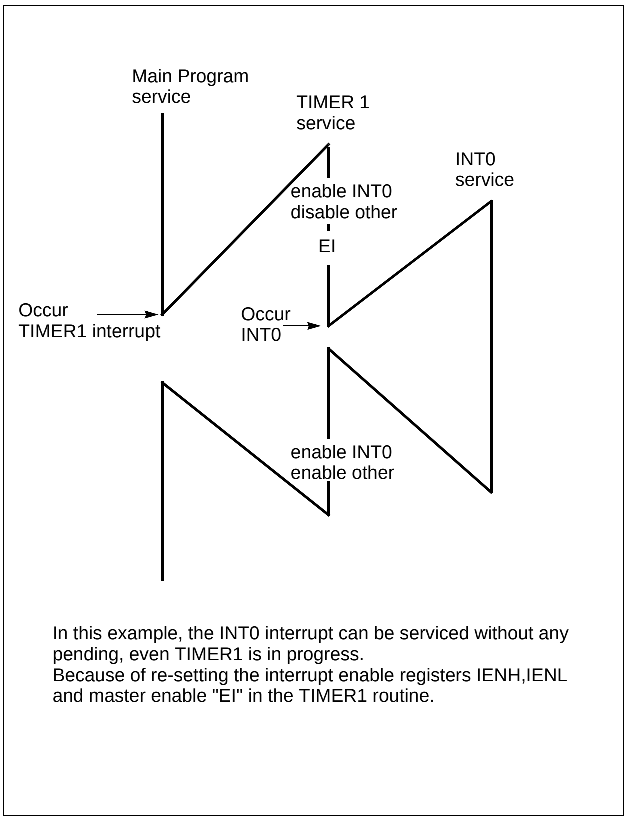


Figure 20-5 Execution of BRK/TCALL0

20.3 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.



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 20-6 Execution of Multi Interrupt

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

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

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


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

20.4 External Interrupt

The external interrupt on INT0 and INT1 pins are edge triggered depending on the edge selection register IEDS (address $0F8_H$) as shown in Figure 20-7.

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

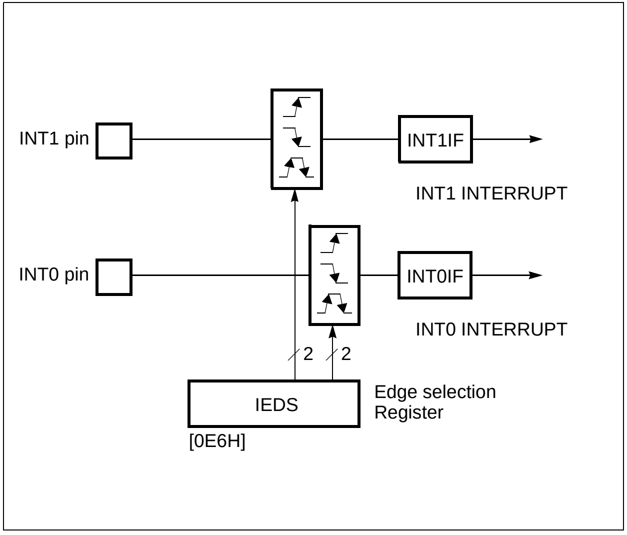


Figure 20-7 External Interrupt Block Diagram

INT0 and INT1 are multiplexed with general I/O ports (P00 and P01). To use as an external interrupt pin, the bit of PSR should be set to “1” correspondingly.

Example: To use as an INT0 and INT1

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

Response Time

The INT0 and INT1 edge are latched into INT0IF and INT1IF at every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The DIV itself takes twelve cycles. Thus, a minimum of twelve complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine.

Figure 20-8 shows interrupt response timings.

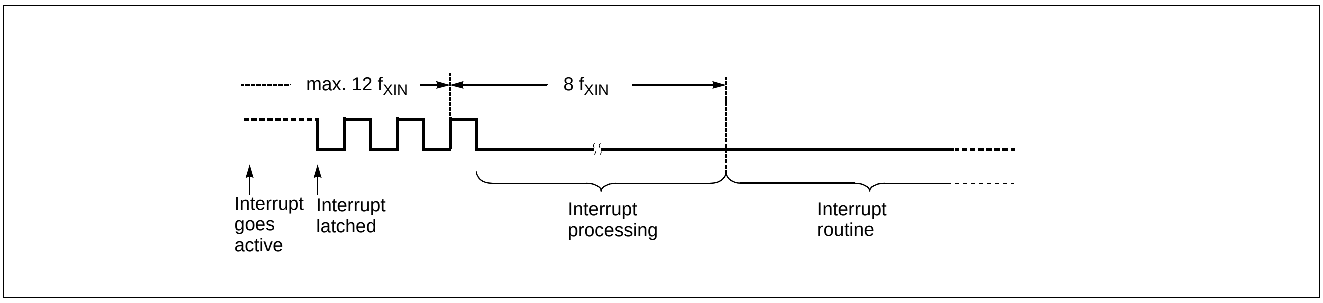


Figure 20-8 Interrupt Response Timing Diagram

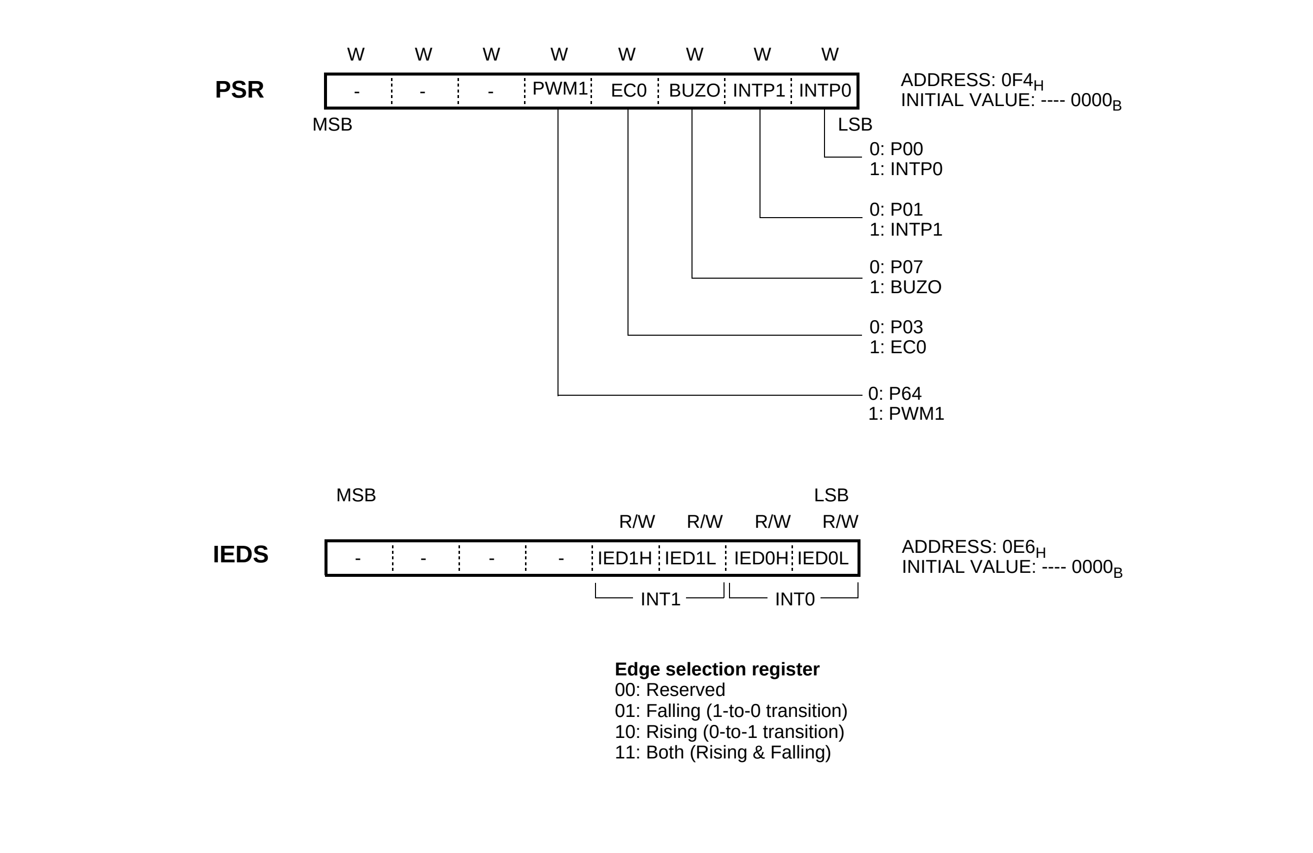

PSR
W W W W W W W W
- - - PWM1 EC0 BUZO INTP1 INTP0
MSB
LSB
ADDRESS: 0F4H
INITIAL VALUE: ---- 0000B
0: P00
1: INTP0
0: P01
1: INTP1
0: P07
1: BUZO
0: P03
1: EC0
0: P64
1: PWM1
IEDS
MSB
LSB
R/W R/W R/W R/W
- - - - IED1H IED1L IED0H IED0L
INT1
INT0
ADDRESS: 0E6H
INITIAL VALUE: ---- 0000B
Edge selection register
00: Reserved
01: Falling (1-to-0 transition)
10: Rising (0-to-1 transition)
11: Both (Rising & Falling)

Figure 20-9 PSR and IEDS Registers

21. Power Saving Mode

The GMS81C2232/48 has two power-down modes. In power-down mode, power consumption is reduced considerably that in Battery operation Battery life can be extended a lot. For applications where power consumption is a critical factor, device provides two kinds of power saving functions, STOP mode and SLEEP mode. Table 21-1 shows the status of each Power Saving Mode. SLEEP mode is entered by setting bit 0 of SMR(Sleep Mode Register), and STOP mode is entered by STOP instruction.

21.1 Sleep Mode

In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operate normally but CPU stops. Movement of all peripherals is shown in Table 20-1. SLEEP mode is entered by setting the bit SLP of SMR to “1”. (This register should be written by byte operation. If this register is set by bit manipulation instruction, for example "set1" or "clr1" instruction, it may be undesired operation) . It is released by RESET or interrupt. To be release by interrupt, interrupt should be enabled before SLEEP mode.

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

```
Ex)     LDM SMR,#0000_0001B
        NOP
        NOP
        NOP
```

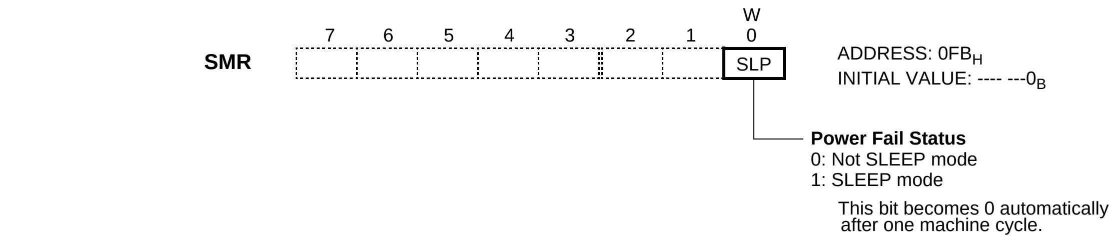

SMR
7
6
5
4
3
2
1
W
0
SLP
ADDRESS: 0FB$_H$
INITIAL VALUE: ---- ---0$_B$
Power Fail Status
0: Not SLEEP mode
1: SLEEP mode
This bit becomes 0 automatically after one machine cycle.

Figure 21-1 SLEEP Mode Register

Release the SLEEP mode

The exit from SLEEP mode is hardware reset or all interrupts. Reset re-defines all the Control registers but does not change the on-chip RAM. 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 SLEEP instruction. It will not vector to interrupt service routine. (refer to Figure 23-5)

When exit from SLEEP mode by reset, enough oscillation stabilization time is required to normal operation. Figure 21-3 shows the timing diagram. When release the SLEEP 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 SLEEP 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 interrupts, exit from SLEEP mode is shown in Figure 21-2. By reset, exit from SLEEP mode is shown in Figure 21-3.

.

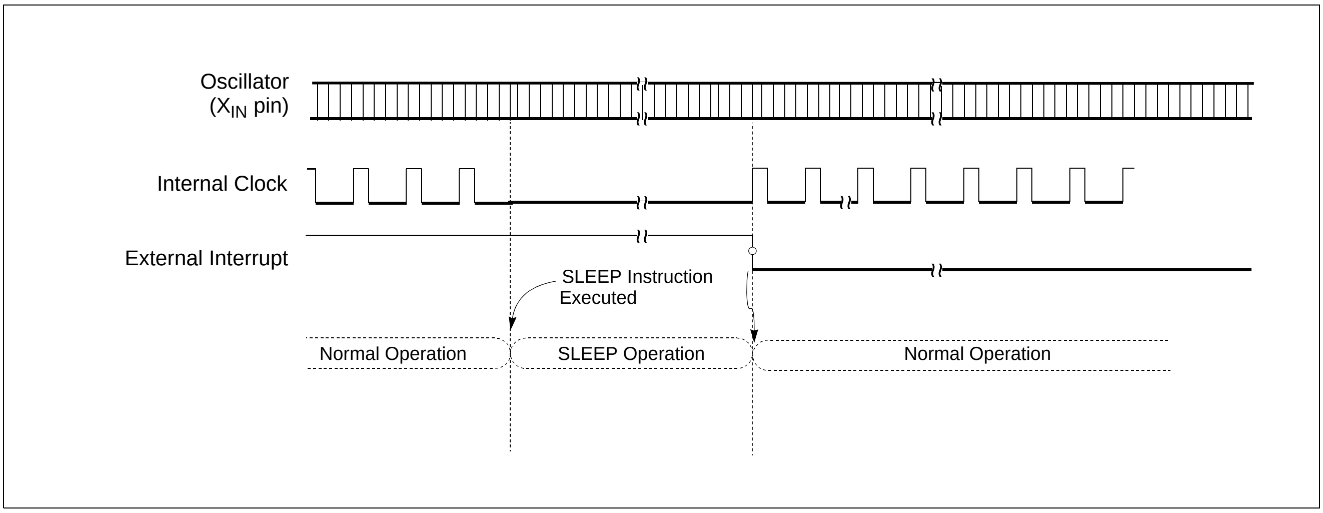


Figure 21-2 SLEEP Mode Release Timing by External Interrupt

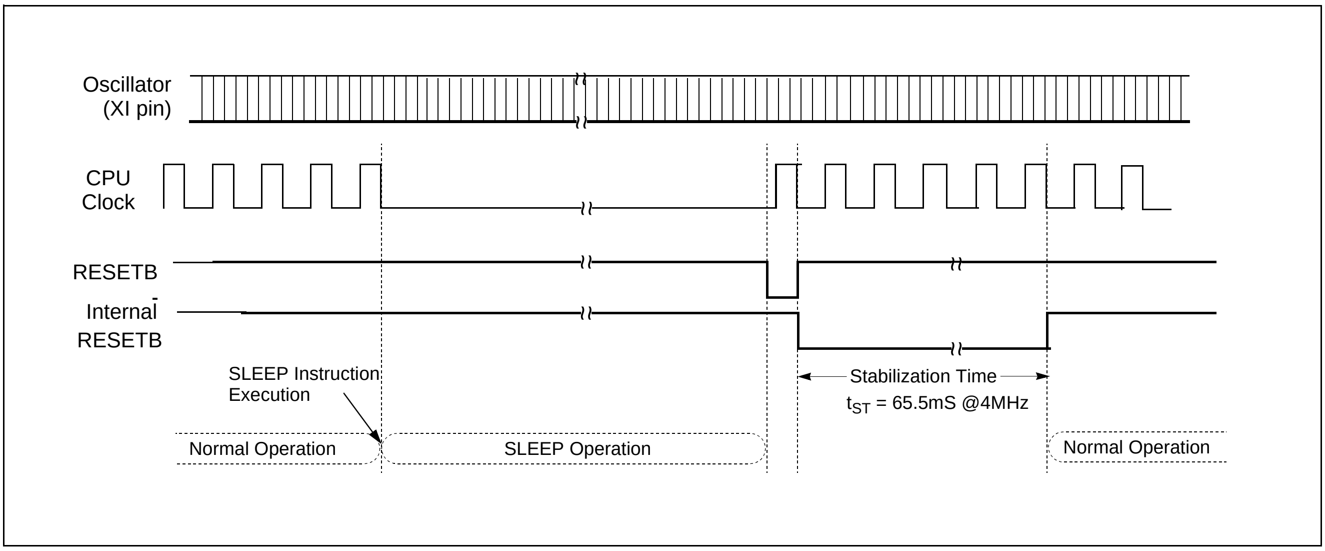


Figure 21-3 Timing of SLEEP Mode Release by RESET

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.

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

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

The reset should not be activated before V_{DD} is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize.

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

```
Ex)     LDM CKCTLR,#0000_1110B
        NOP
        LDM STPC,#0101_1010B
        NOP
        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.

	W 7	W 6	W 5	W 4	W 3	W 2	W 1	W 0
STPC	0	1	0	1	1	0	1	0

ADDRESS: $0FF_H$
INITIAL VALUE: $0000\ 0000_B$

Cautions : 1. To get into STOP mode, STOP Control Registor must be enabled just before STOP instruction.

2. When STOP mode is released, STOP Control Registor(STPC) value is cleared automatically.

3. It is prohibited to wirte another value into STPC.

Figure 21-4 STOP Control Register

Peripheral	STOP Mode	SLEEP Mode
CPU	Stop	Stop
RAM	Retain	Retain
Basic Interval Timer	Halted	Operates Continuously
Watchdog Timer	Stop	Stop
Timer/Event0,1	Halted(Only when the event counter mode is enable, timer operates normaly)	Operates Continuously
VFD Controller	Stop	Stop
SIO3,BUZ,ADC,Remote Timer	Stop	Stop
SIO1	Only operate with external clock	Only operate with external clock
Oscillation	Stop(Xin=L, Xout=H)	Oscillation
I/O Ports	Retain	Retain
Control Registers	Retain	Retain
Internal Circuit	Stop mode	Sleep mode
Prescaler	Retain	Active
Address Data Bus	Retain	Retain
Release Source	**Reset, Timer Interrupt(EC0), SIO1(External Clock), External Interrupt**	**Reset, All Interrupts**

Table 21-1Peripheral Operation During Power Saving Mode

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 21-5)

When exit from Stop mode by external interrupt, enough oscillation stabilization time is required to normal operation. Figure 21-6 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 .

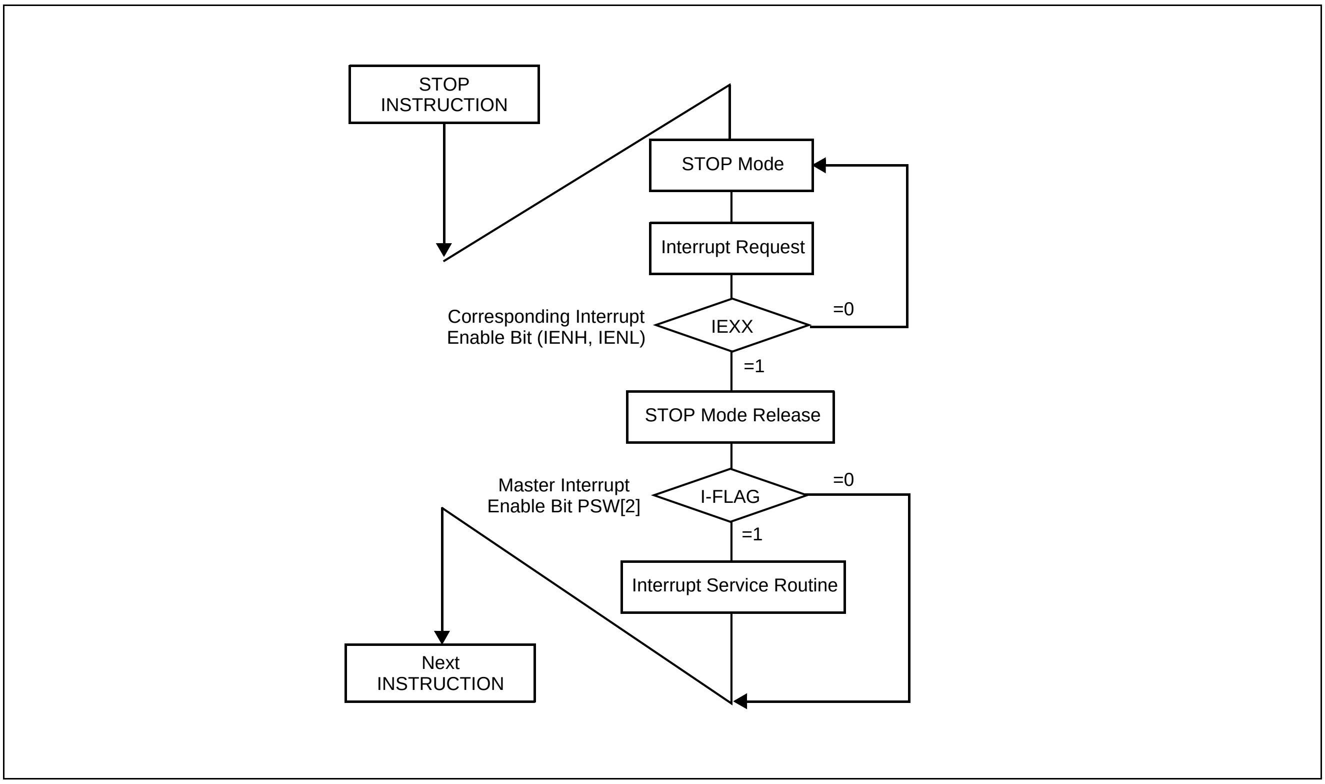
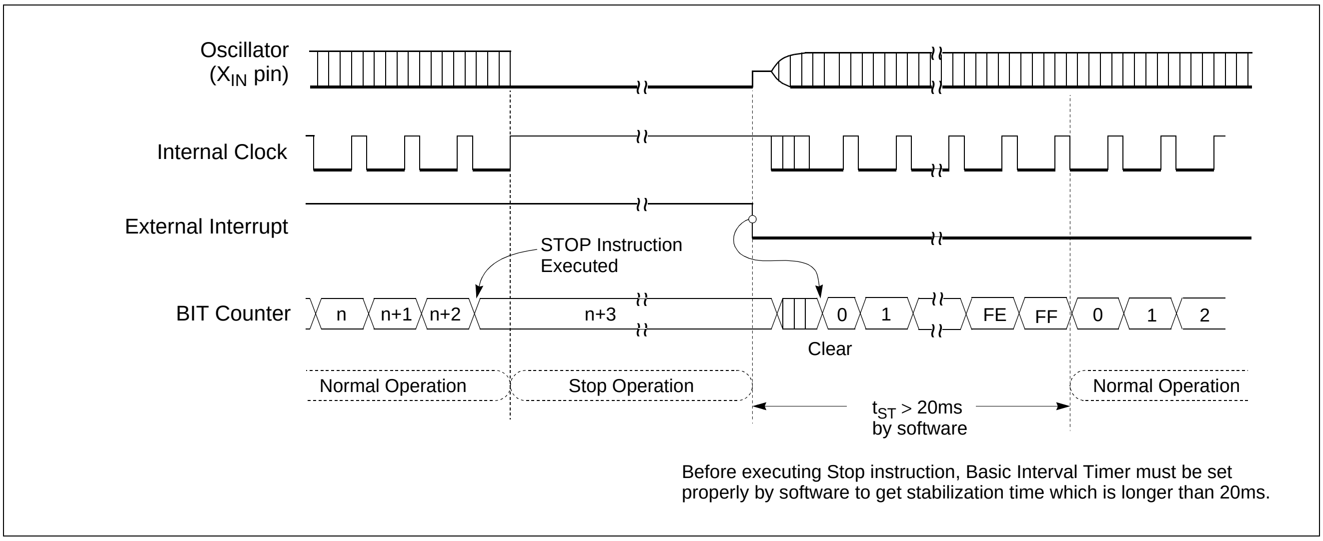

STOP INSTRUCTION
STOP Mode
Interrupt Request
Corresponding Interrupt Enable Bit (IENH, IENL)
IEXX
=0
=1
STOP Mode Release
Master Interrupt Enable Bit PSW[2]
I-FLAG
=0
=1
Interrupt Service Routine
Next INSTRUCTION

Figure 21-5 STOP Releasing Flow by Interrupts

.



Before executing Stop instruction, Basic Interval Timer must be set properly by software to get stabilization time which is longer than 20ms.

Figure 21-6 STOP Mode Release Timing by External Interrupt

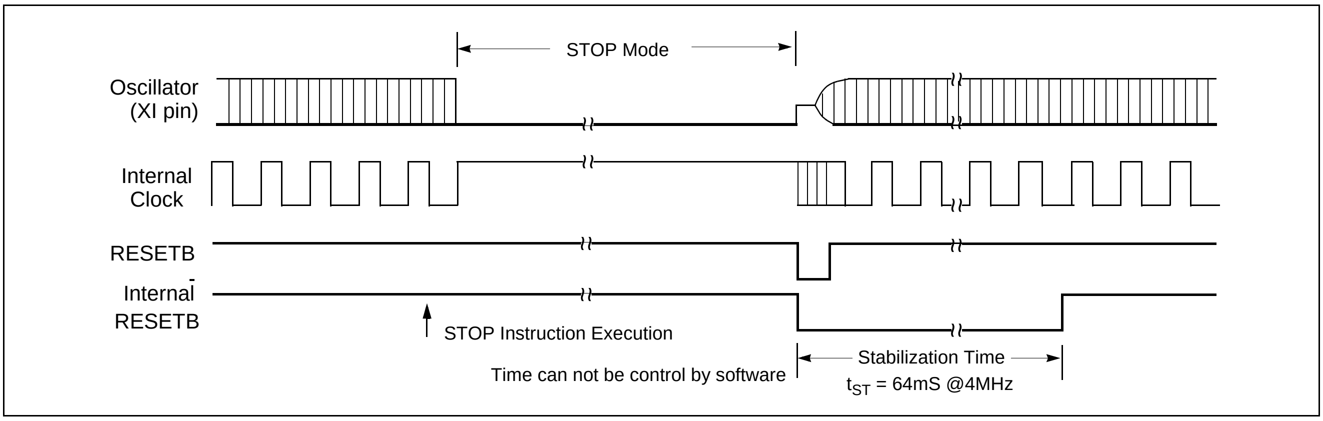


Figure 21-7 Timing of STOP Mode Release by RESET

21.3 Internal RC-Oscillated Watchdog Timer Mode

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

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

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

```
Ex)     LDM  WDTR,#1111_1111B
        LDM  CKCTLR,#0010_1110B
        NOP
        LDM STPC,#0101_1010B
        NOP
        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 21-8) 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 21-9) -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 21-5)

When exit from Internal RC-Oscillated Watchdog Timer mode by external interrupt, the oscillation stabilization time is required to normal operation. Figure 21-8 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 21-9.

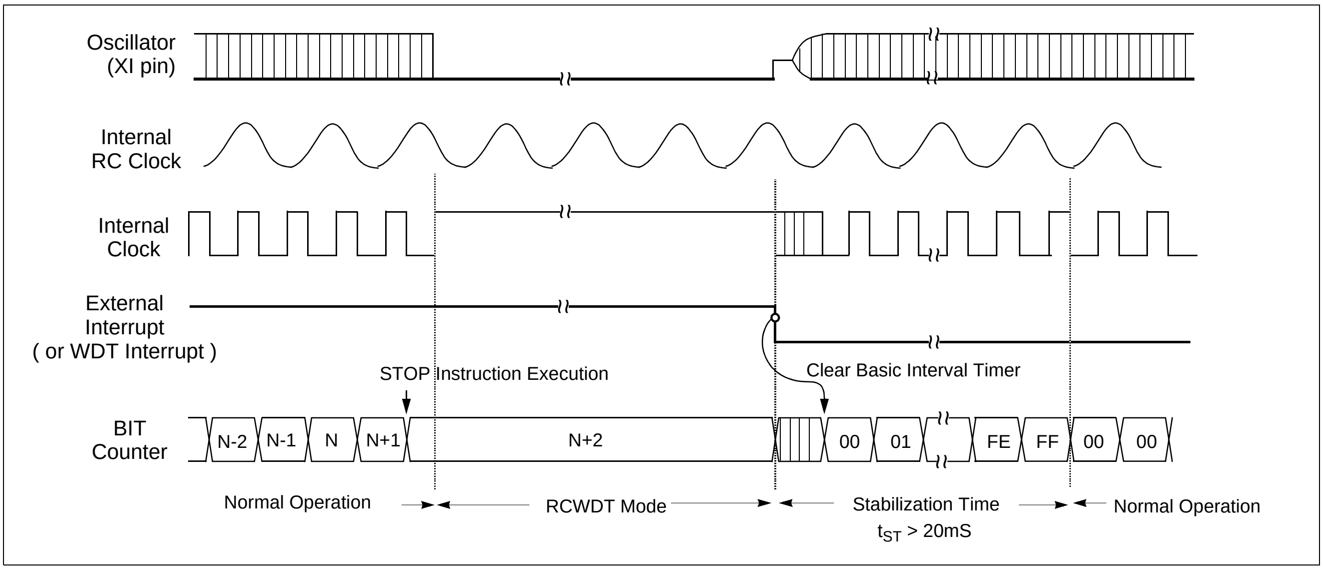


Figure 21-8 Internal RCWDT Mode Releasing by External Interrupt or WDT Interrupt

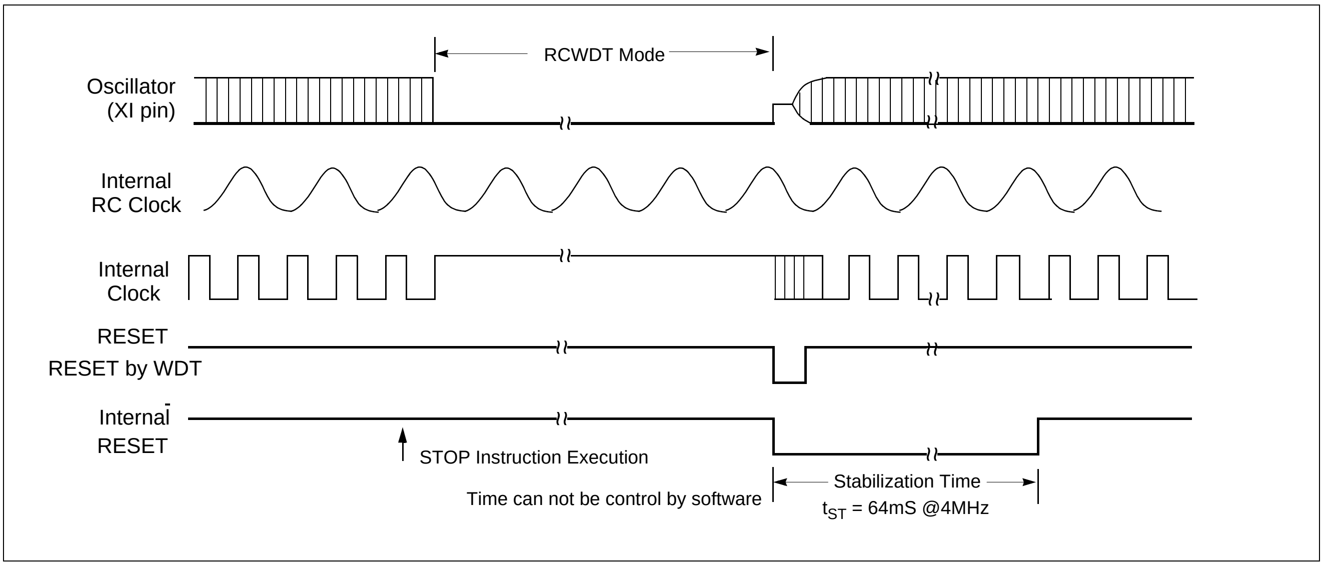


Figure 21-9 Internal RCWDT Mode Releasing by RESET

21.4 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. .

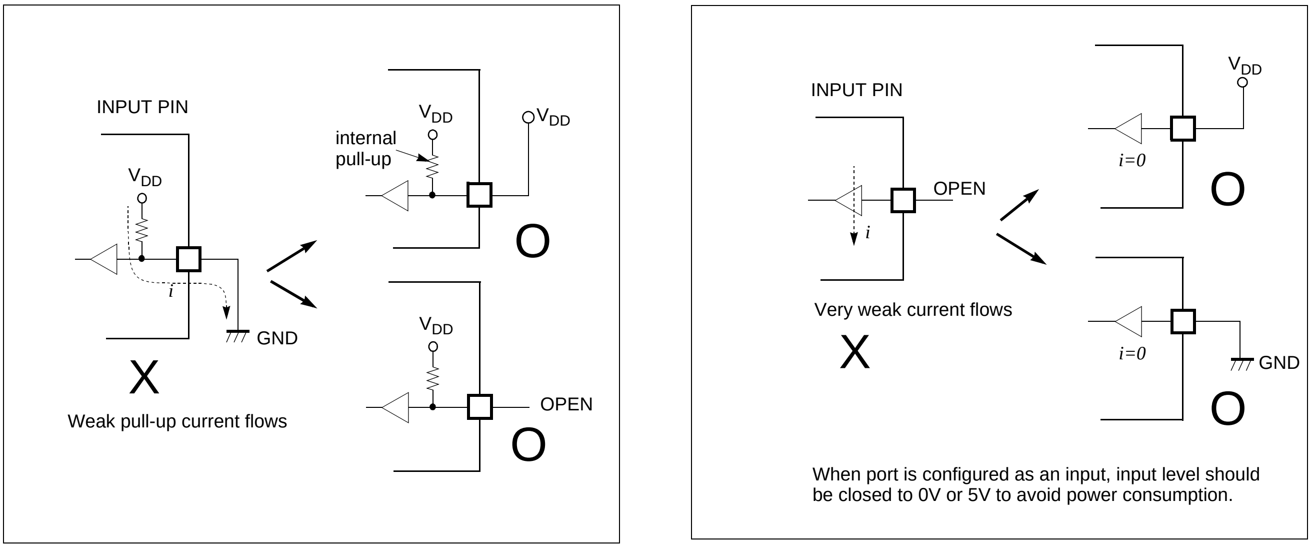

INPUT PIN
V_{DD}
GND
X
Weak pull-up current flows
internal
pull-up
V_{DD}
V_{DD}
O
V_{DD}
OPEN
O
INPUT PIN
OPEN
i
Very weak current flows
X
V_{DD}
i=0
O
i=0
GND
O
When port is configured as an input, input level should be closed to 0V or 5V to avoid power consumption.

Figure 21-10 Application Example of Unused Input Port

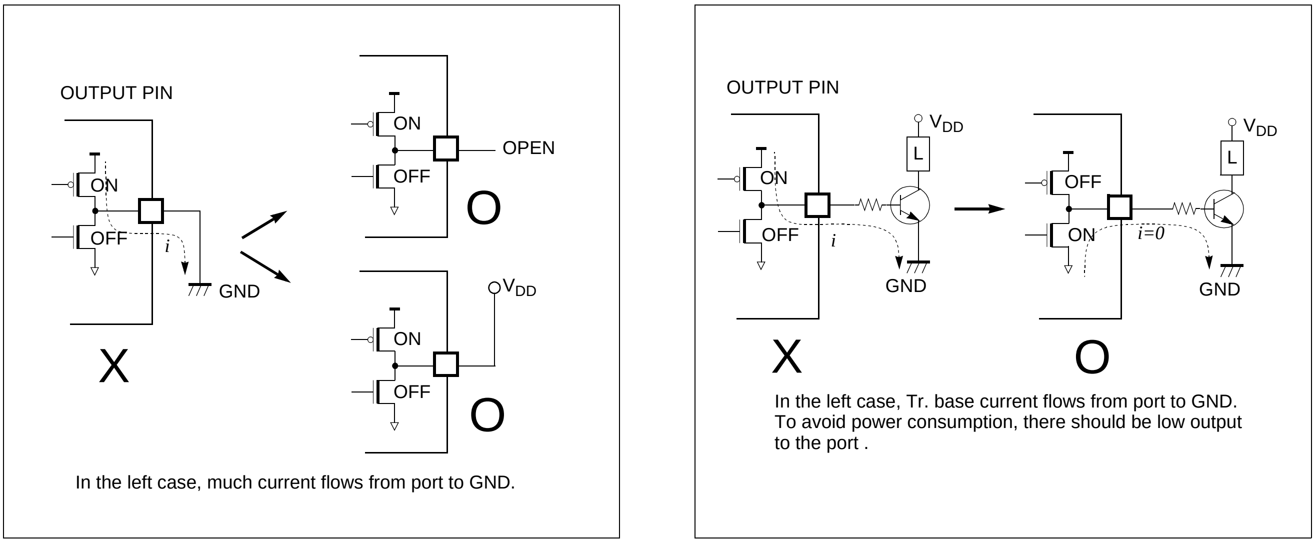

OUTPUT PIN
ON
OFF
i
GND
X
ON
OFF
OPEN
O
ON
OFF
V_{DD}
O
In the left case, much current flows from port to GND.
OUTPUT PIN
ON
OFF
i
V_{DD}
L
GND
X
OFF
ON
i=0
V_{DD}
L
GND
O
In the left case, Tr. base current flows from port to GND. To avoid power consumption, there should be low output to the port .

Figure 21-11 Application Example of Unused Output Port

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

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

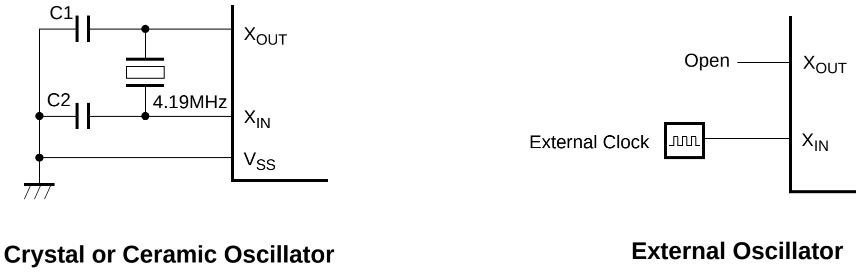
First conseider the setting to input mode. Be sure that there is no current flow after considering its relationship with external circuit. In input mode, the pin impedance viewing from external MCU is very high that the current doesn’t flow.

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

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

22. OSCILLATOR CIRCUIT

The HMS81C2232/48 has two oscillation circuits internally. X_{IN} and X_{OUT} are input and output for main frequency. Respectively, inverting amplifier which can be configured for being used as an on-chip oscillator, as shown in Figure 22-1.



Crystal or Ceramic Oscillator

External Oscillator

Recommend

Crystal Oscillator	C1,C2 = 30pF
Ceramic Resonator	C1,C2 = 30pF

Figure 22-1 Oscillation Circuit

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

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

In addition, see Figure 22-2 for the layout of the crystal.

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

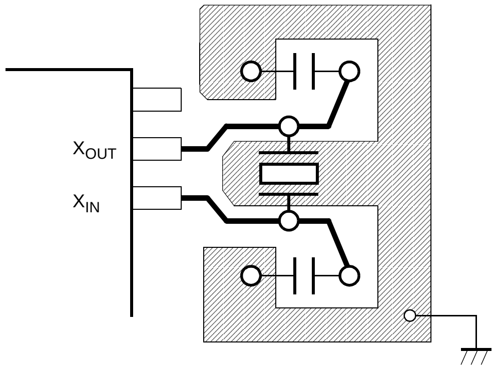


Figure 22-2 Layout of Oscillator PCB circuit

23. RESET

The HMS81C20xxA have two types of reset generation procedures; one is an external reset input, the other is a watch-dog timer reset. Table 23-1 shows on-chip hardware initialization by reset action.

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	$(FFFF_H) - (FFFE_H)$	Peripheral clock	Off
RAM page register (RPR)	0	Watchdog timer	Disable
G-flag (G)	0	Control registers	Refer to Table 8-1 on page 28
Operation mode	Main-frequency clock	Power fail detector	Disable

Table 23-1 Initializing Internal Status by Reset Action

23.1 External Reset Input

The reset input is the RESET pin, which is the input to a Schmitt Trigger. A reset in accomplished by holding the RESET pin low for at least 8 oscillator periods, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset, 64ms (at 4 MHz) add with 7 oscillator periods are required to start execution as shown in Figure 23-2.

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

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

A connection for simple power-on-reset is shown in Figure 23-1.

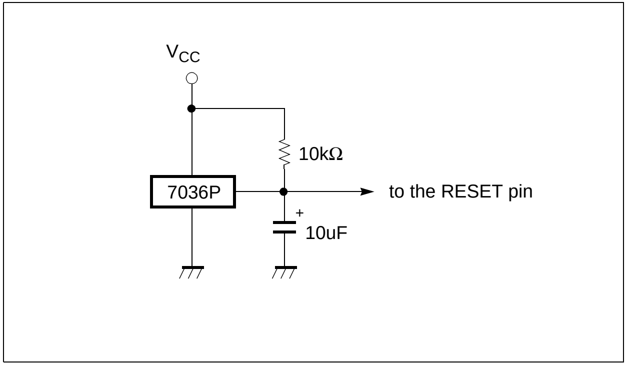


Figure 23-1 Simple Power-on-Reset Circuit

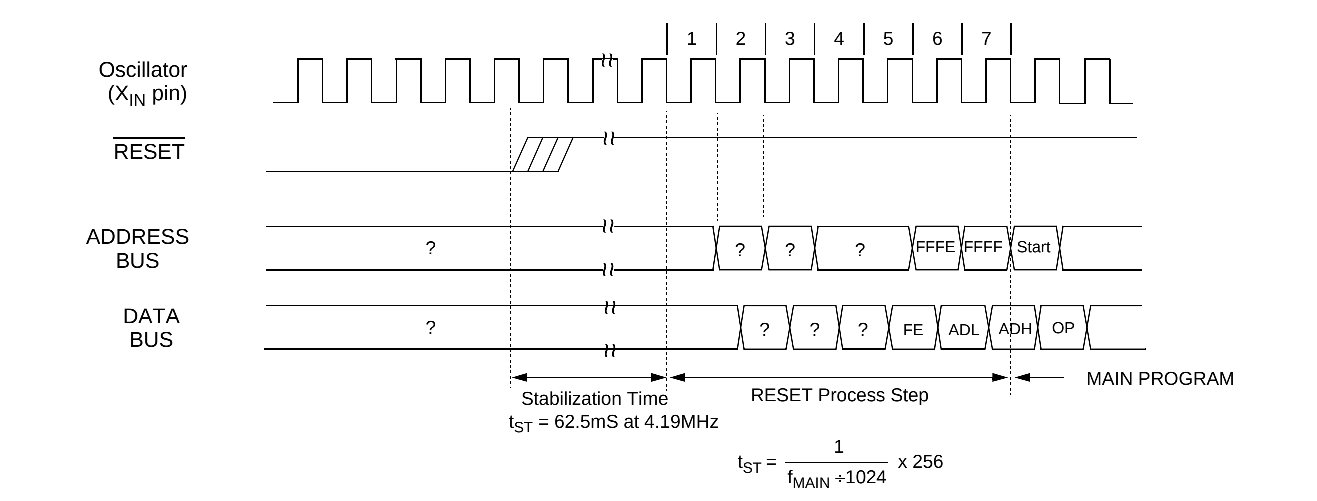


Figure 23-2 Timing Diagram after RESET

23.2 Watchdog Timer Reset

Refer to “12. WATCHDOG TIMER” on page 48.

24. POWER FAIL PROCESSOR

The HMS81C2232/48 has an on-chip power fail detection circuitry to immunize against power noise. A configuration register, PFDR, can enable or disable the power fail detect circuitry. Whenever V_{DD} falls close to or below power fail voltage for 100ns, the power fail situation may reset or freeze MCU according to PFDM bit of PFDR. Refer to “7.4 DC Electrical Characteristics” on page 20.

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

Note: *User can select power fail voltage level according to PFD0, PFD1 bit of CONFIG register(703F$_H$) at the OTP (HMS87C20xxA) but **must select** the power fail voltage level to define PFD option of “Mask Order & Verification Sheet” at the mask chip(HMS81C2232/48).*
Because the power fail voltage level of mask chip (HMS81C20xxA) is determined according to mask option.

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

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

Table 24-1 Power fail processor

.

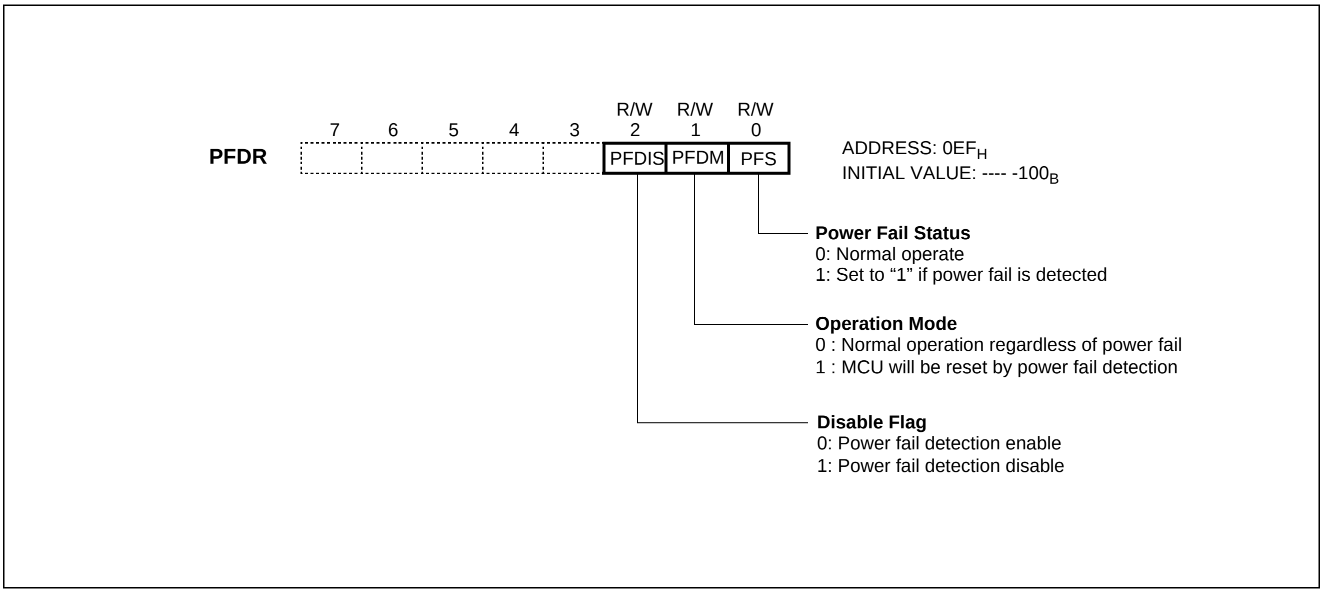

PFDR
7
6
5
4
3
R/W
2
R/W
1
R/W
0
PFDIS
PFDM
PFS
ADDRESS: 0EF$_H$
INITIAL VALUE: ---- -100$_B$
Power Fail Status
0: Normal operate
1: Set to “1” if power fail is detected
Operation Mode
0 : Normal operation regardless of power fail
1 : MCU will be reset by power fail detection
Disable Flag
0: Power fail detection enable
1: Power fail detection disable

Figure 24-1 Power Fail Voltage Detector Register

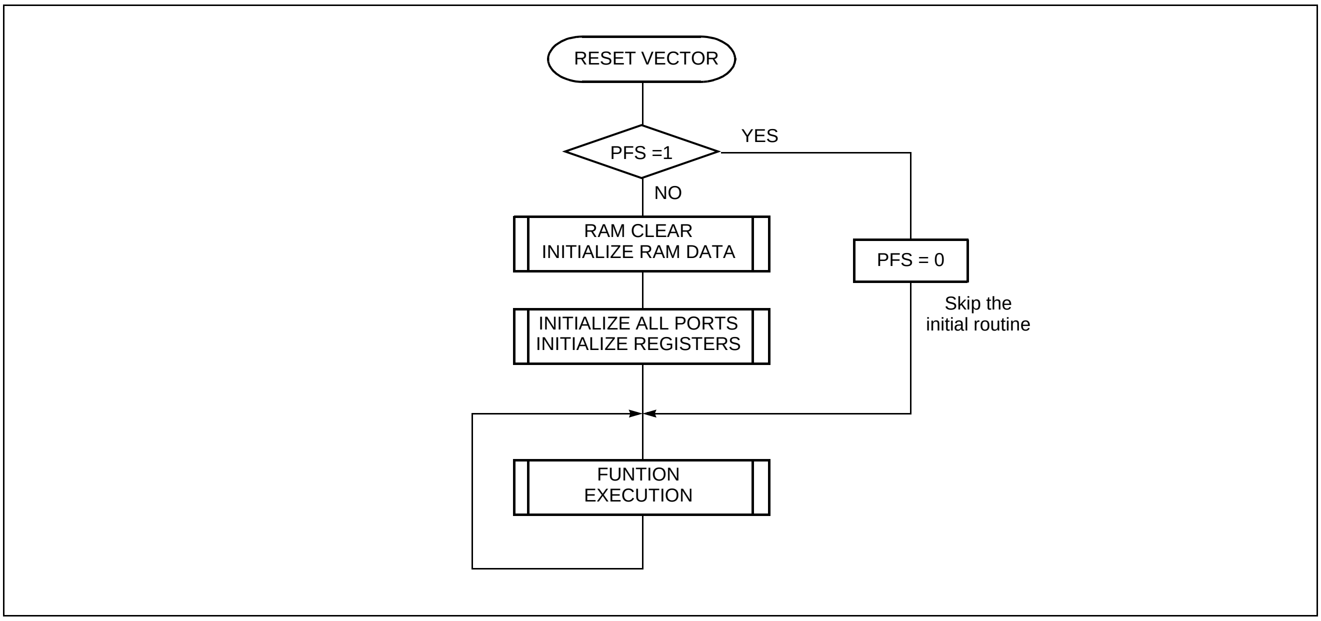


Figure 24-2 Example S/W of RESET flow by Power fail

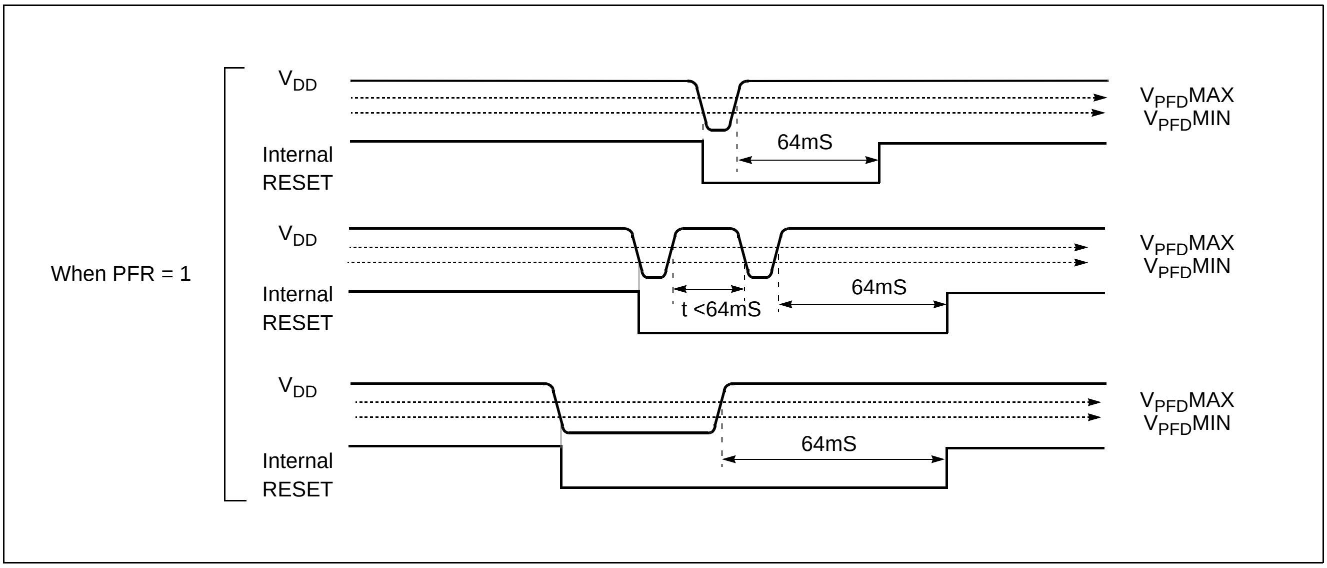


Figure 24-3 Power Fail Processor Situations

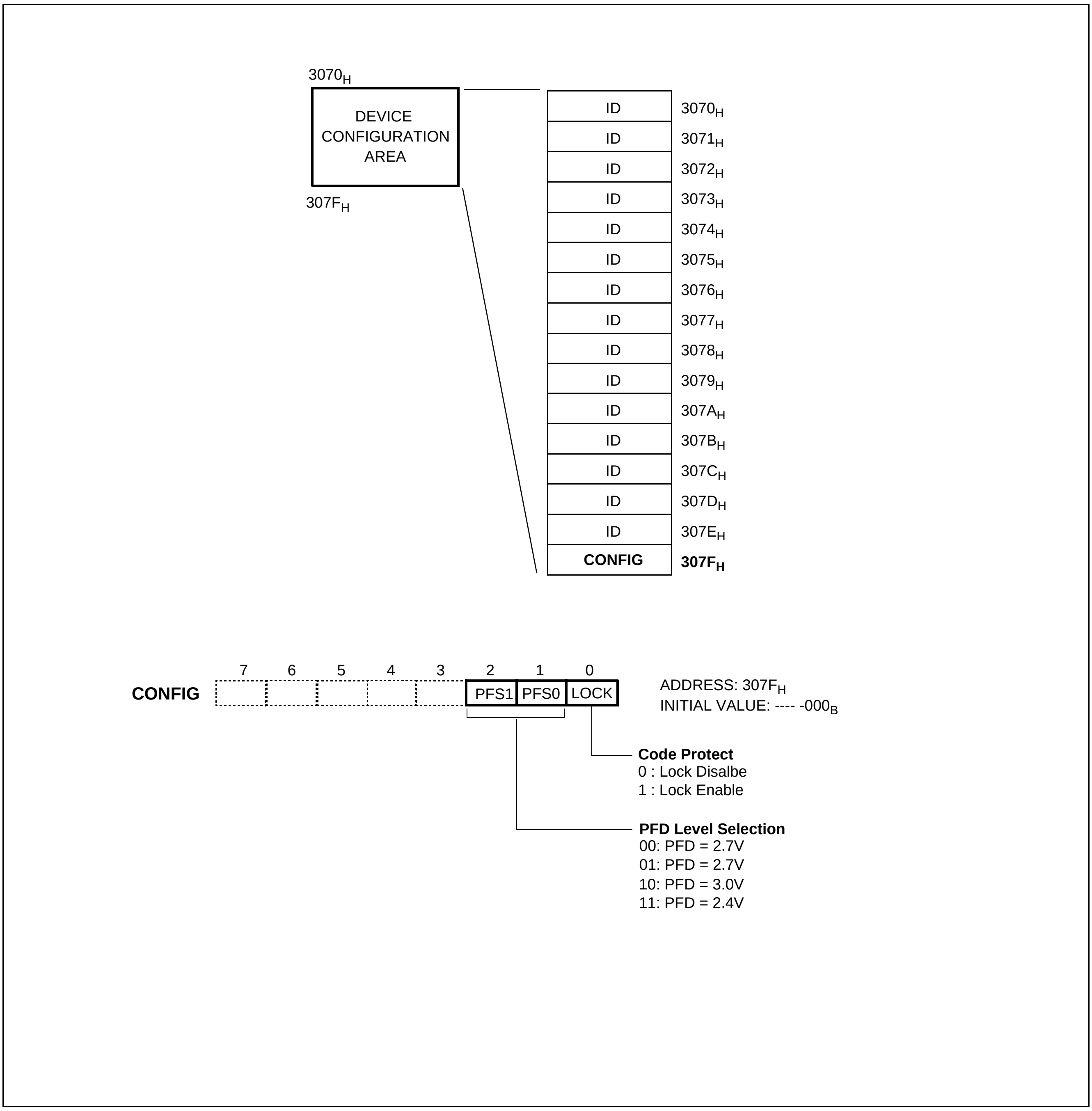
25. OTP PROGRAMMING

25.1 DEVICE CONFIGURATION AREA

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

Sixteen memory locations (3070_H ~ $307F_H$) 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.



ID	3070_H
ID	3071_H
ID	3072_H
ID	3073_H
ID	3074_H
ID	3075_H
ID	3076_H
ID	3077_H
ID	3078_H
ID	3079_H
ID	$307A_H$
ID	$307B_H$
ID	$307C_H$
ID	$307D_H$
ID	$307E_H$
CONFIG	**$307F_H$**

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

ADDRESS: $307F_H$
INITIAL VALUE: ---- -000_B

Code Protect
0 : Lock Disalbe
1 : Lock Enable

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

Figure 25-1 Device Configuration Area

Pin No.				User Mode	EPROM MODE				
80MQFP 80TQFP	64SDIP	64MQFP	64LQFP	Pin Name	Pin Name	Description			
6	14	7	6	$\overline{\text{RESET}}$	VPP	Programming Power (0V, 11.5V)			
3	11	4	3	XIN	PXEN	OTP Clock			
7	15	8	7	P27	CTL3	Program/Verify/Read Control			
8	16	9	8	P26	CTL2				
9	17	10	9	P25	CTL1	Program Mode Control			
10	18	11	10	P24	CTL0				
15	19	12	11	P00	A_D0	Address Input Data Input/Output	A8	A0	D0
16	20	13	12	P01	A_D1		A9	A1	D1
17	21	14	13	P02	A_D2		A10	A2	D2
19	23	16	15	P03	A_D3		A11	A3	D3
20	24	17	16	P04	A_D4		A12	A4	D4
21	25	18	17	P05	A_D5		A13	A5	D5
22	26	19	18	P06	A_D6		A14	A6	D6
5	13	6	5	P07	A_D7		A15	A7	D7
23	27	20	19	VSS0	VSS0	Connect to 0V			
2	10	3	2	VSS1	VSS1	Connect to 0V			
25	29	22	21	VDD0	VDD0	Connect to 5.0V (Port Power)			
1	9	2	1	VDD1	VDD1	Connect to 5.0V (Logic Power)			

Table 25-1 Pin Description in EPROM Mode

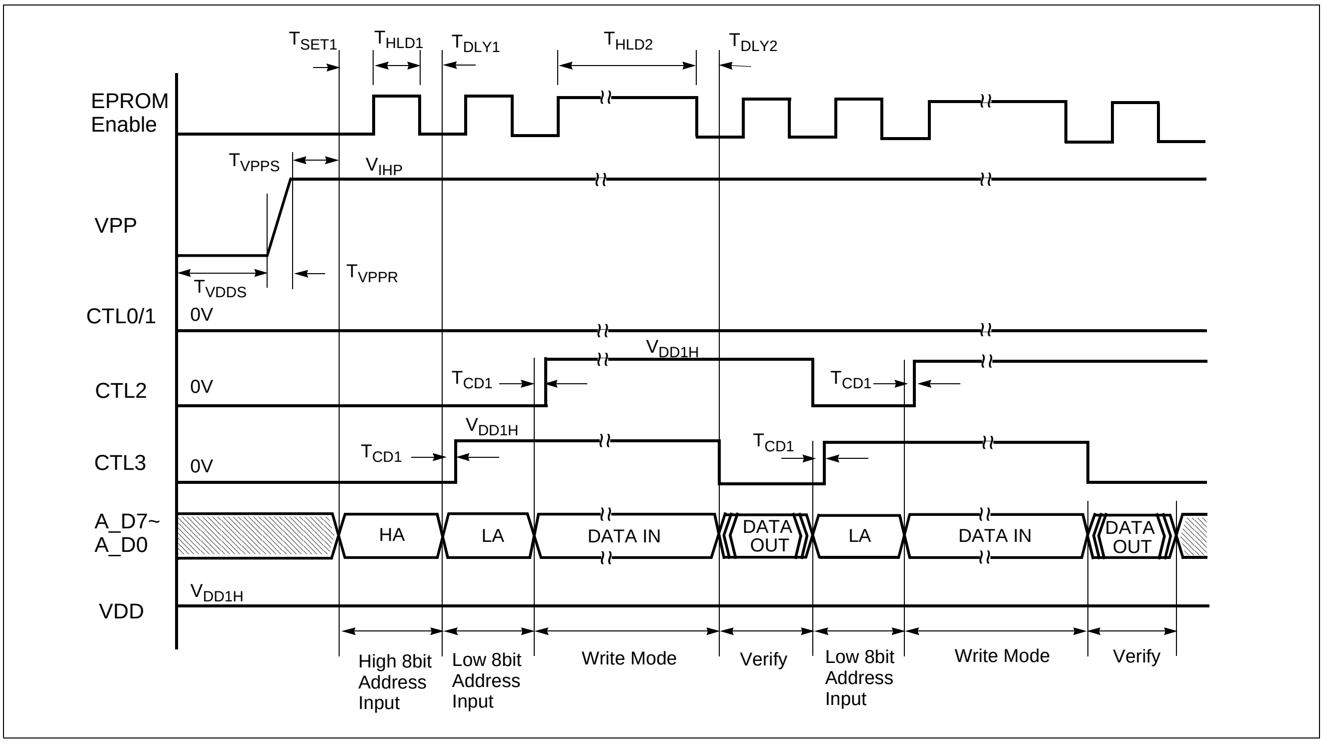


Figure 25-2 Timing Diagram in Program (Write & Verify) Mode

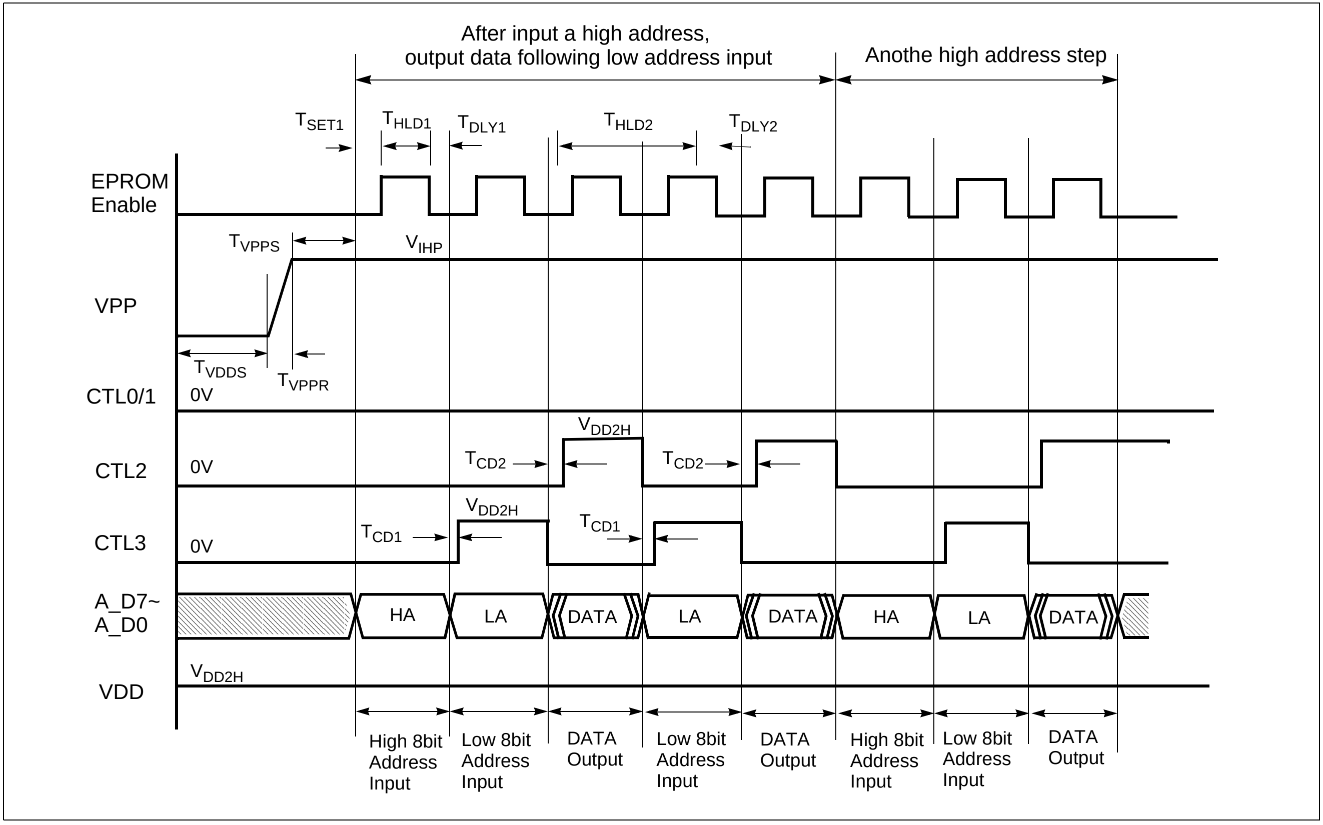


Figure 25-3 Timing Diagram in READ Mode

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}	11.2	11.5	11.8	V
VDD Level in Program Mode	V_{DD1H}	4.8	5.0	5.2	V
VDD Level in Read Mode	V_{DD2H}	-	2.7	-	V
CTL3~0 High Level in EPROM Mode	V_{IHC}	$0.9V_{DD}$	-	-	V
CTL3~0 Low Level in EPROM Mode	V_{ILC}	-	-	$0.1V_{DD}$	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}		500		nS
EPROM Enable Hold Time after T_{SET1}	T_{HLD1}		500		nS
EPROM Enable Delay Time after T_{HLD1}	T_{DLY1}		500		nS
EPROM Enable Hold Time in Write Mode	T_{HLD2}		100		uS
EPROM Enable Delay Time after T_{HLD2}	T_{DLY2}		500		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 25-2 AC/DC Requirements for Program/Read Mode

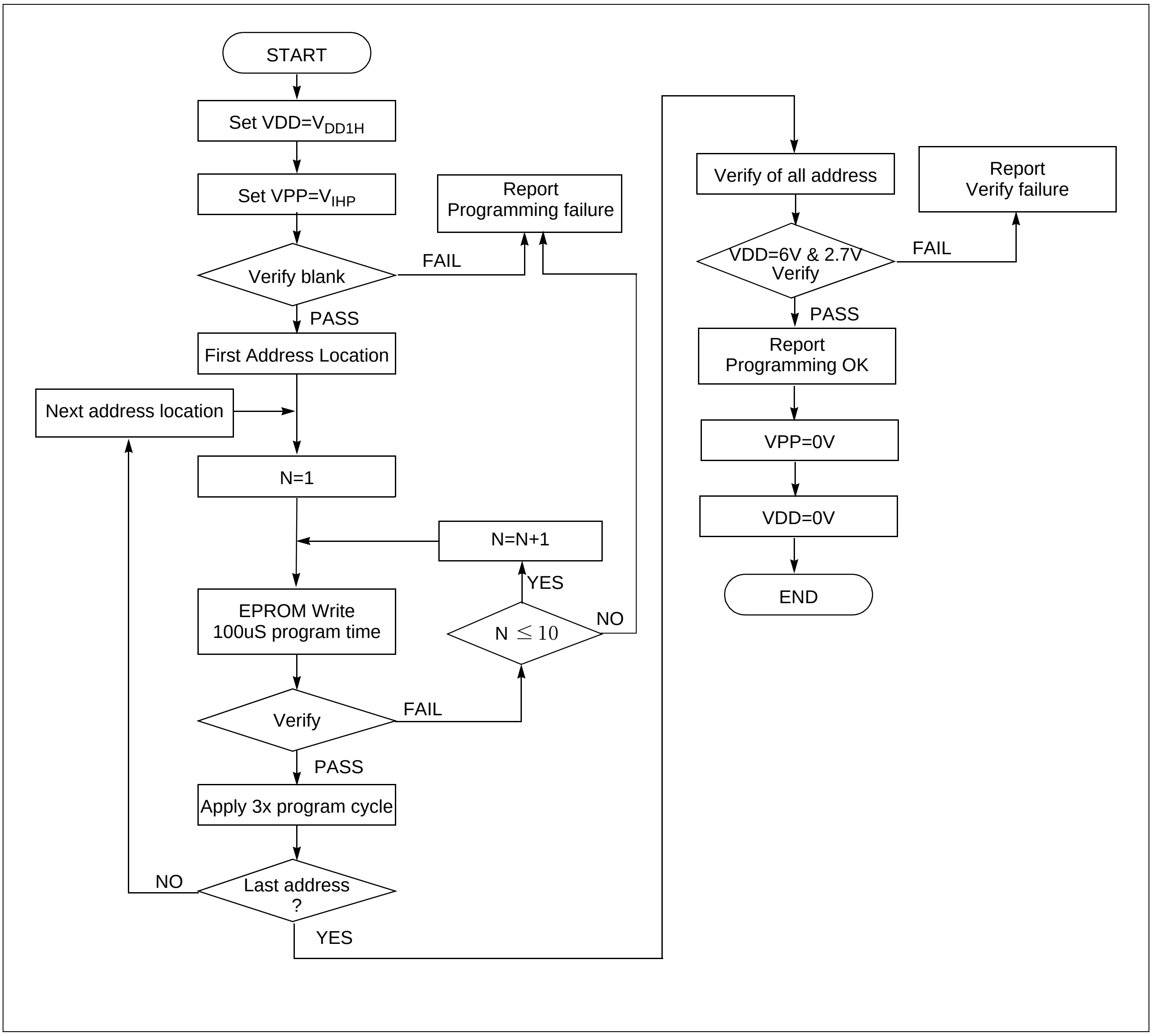


Figure 25-4 Programming Flow Chart