December 29, 2009 Ver 0.3

4-BIT SINGLE CHIP MICROCOMPUTERS

ADAM40X272X

USER`S MANUAL

- ADAM40S2728
- ADAM40S2724
- ADAM40P2728
- ADAM40P2724

ABOV SEMICONDUCTOR

1. OVERVIEW

The ADAM40X272X is the high speed and low voltage operating 4-bit single chip microcomputer. This chip contains ADAM40 core, ROM, RAM, input/output ports , two timer/counters, etc. The ADAM40P272X is MTP version and the ADAM40S272X is Masked ROM version.

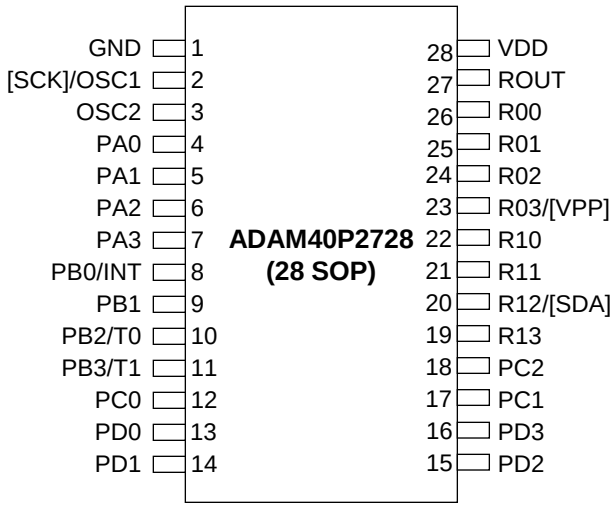
1.1. Features

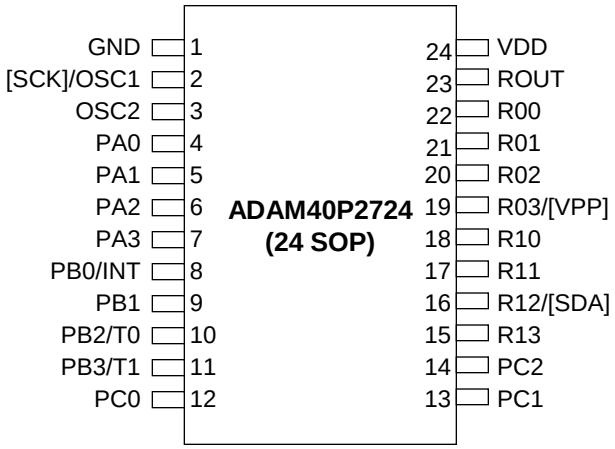
- Instruction Execution Time
  - 3us @ Xin=4MHz
- Program Memory
  - 48K Bytes (24,576 x 16bit)
    [Multi-programmable by 16K-Byte or 24K-Byte in MTP Version]
- Data Memory (RAM)
  - 256 nibble (256 x 4bit)
- 16-Bit Table read Instruction.
- Timer
  - Timer / Counter : 8Bit * 2ch
  - Carrier Generator : 6Bit * 1ch
  - Watch Dog Timer : 17Bit * 1ch
- Oscillator Type
  - Crystal
  - Ceramic Resonator
- Power On Reset
- Power Saving Operation Modes
  - STOP
- 3 Interrupt source
- Operating Voltage Range
  - 1.8V ~ 3.6 V @ 4MHz for ADAM40P2728, ADAM40P2724
  - 1.2V ~ 3.6 V @ 4MHz for ADAM40S2728, ADAM40S2724
- Low Voltage Detection Circuit
- Package
  - 24SOP / 28SOP

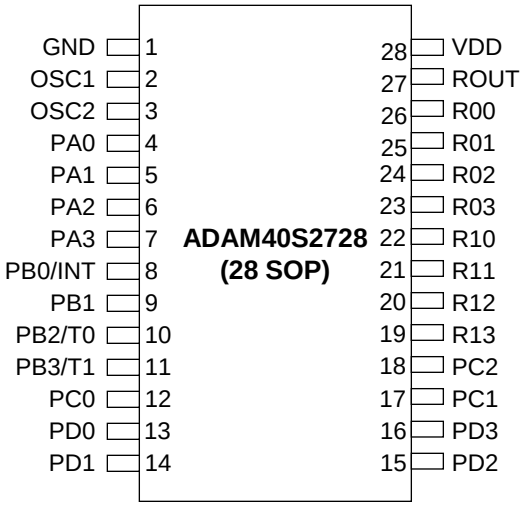
Series	ADAM40P2728	ADAM40S2728	ADAM40P2724	ADAM40S2724
Program memory	24,576 x 16 (24KB*2, 16KB*3)	24,576 x 16	24,576 x 16 (24KB*2, 16KB*3)	24,576 x 16
Data memory	256 x 4	256 x 4	256 x 4	256 x 4
I/O ports	8	8	8	8
Input ports	5	5	5	5
Output ports	13	13	9	9
Package	28SOP	28SOP	24SOP	24SOP

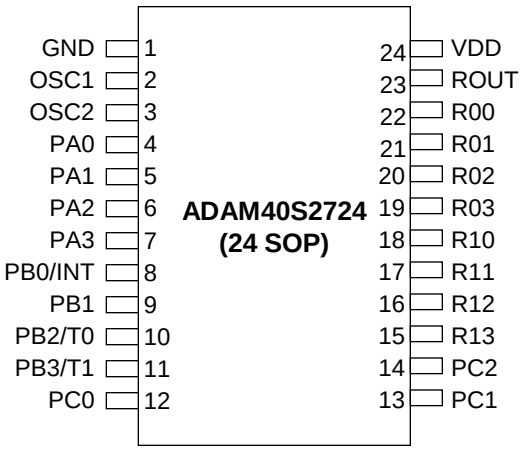
Table 1.1 ADAM40X272X series members

1.2. Pin Assignments (top view)

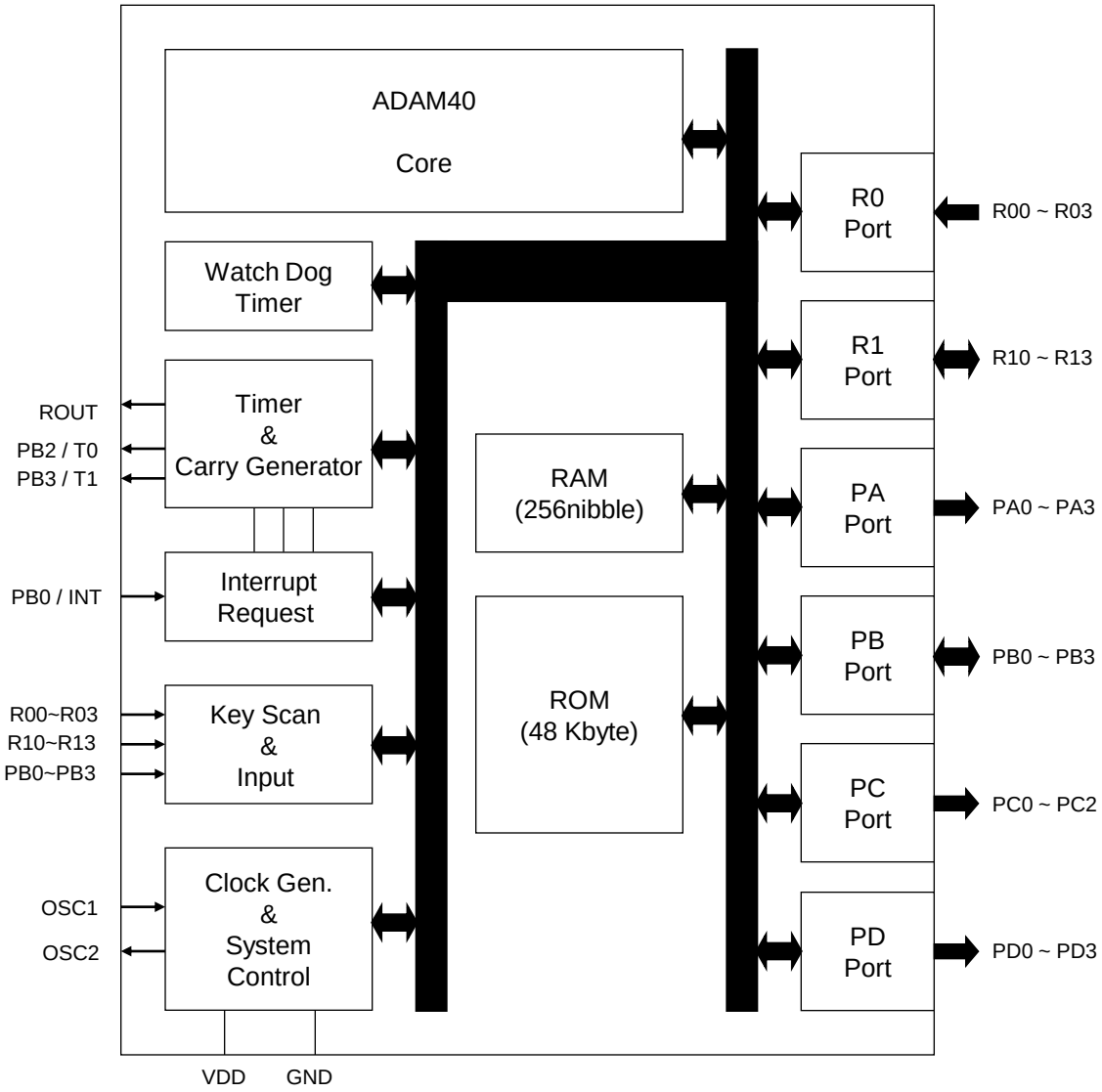






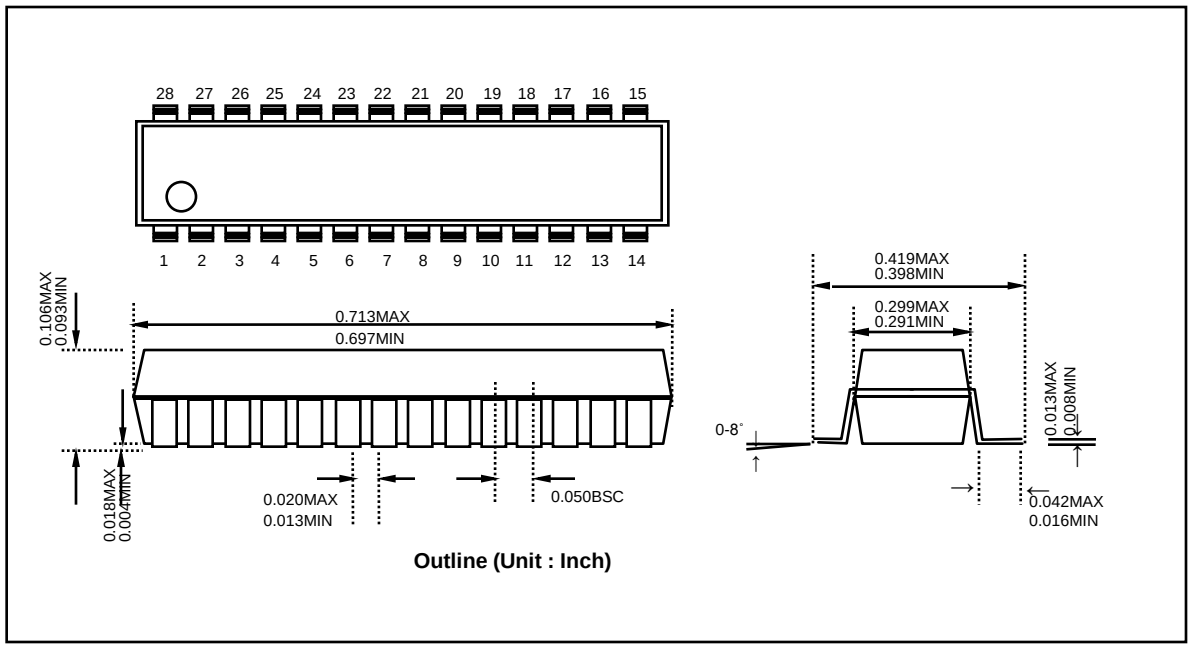


1.3. Block Diagram

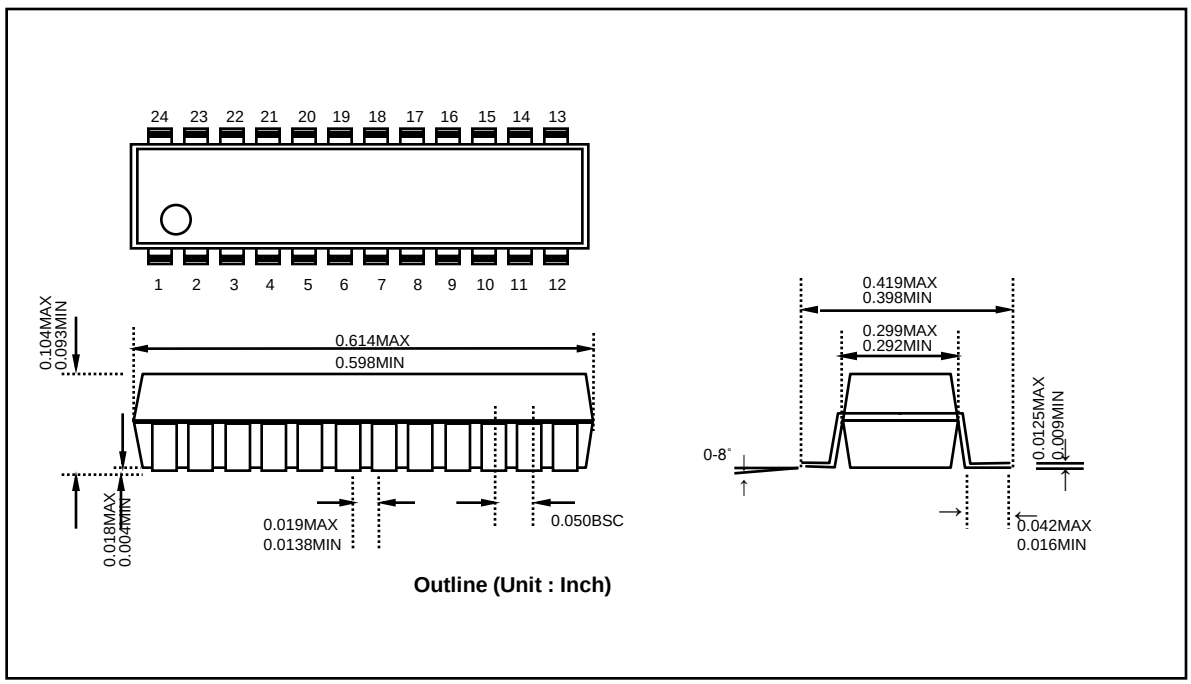


NOTE > PD0~PD3 ports are not available in ADAM40P2724 and ADAM40S2724

1.4. Package Dimension



28 SOP Pin Dimension (dimensions in inch)



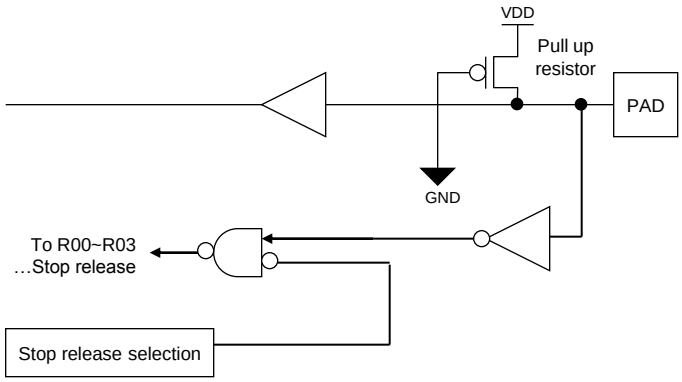
24 SOP Pin Dimension (dimensions in inch)

1.5. Pin Function

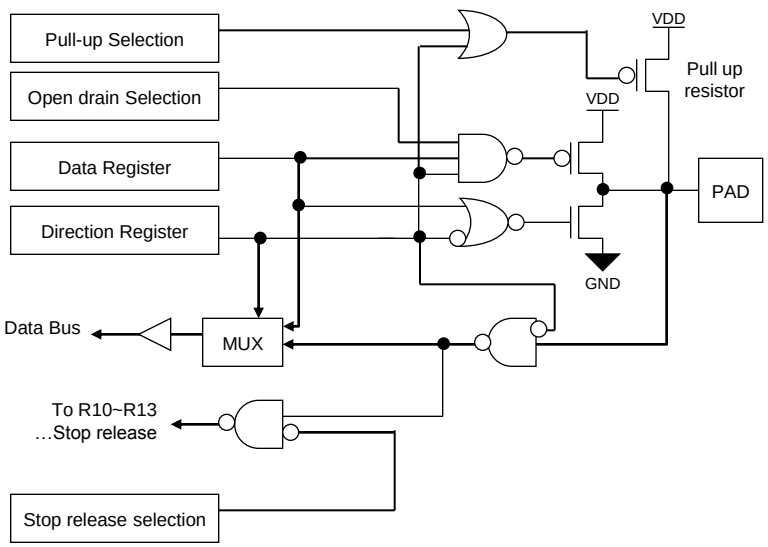
PIN NAME	INPUT OUTPUT	FUNCTION	@RESET	@STOP
R00 R01 R02 R03	INPUT INPUT INPUT INPUT	- 4-bit input Port. - CMOS input with pull-up resistor. - Can be selectable as STOP release Input pin individually by user program. (It is released by “L” input at STOP mode)	Input (with Pull-up)	-
R10 R11 R12 R13	I/O I/O I/O I/O	- 4-bit I/O Port. - CMOS input with pull-up resistor. - Pull-ups can be disabled by user Program. - Can be selectable as STOP release Input pin individually by user program. (It is released by “L” input at STOP mode) - N-ch open drain output. - Can be programmable as Push-pull output individually. - Each pin can be set and reset by R1 Data register value.	Input (with Pull-up)	State of before STOP
PA0 PA1 PA2 PA3	OUTPUT OUTPUT OUTPUT OUTPUT	- N-ch open drain output. - Each pin can be set and reset by PA Data register value.	Hi-Z	“L” level output
PB0/INT PB1 PB2/T0 PB3/T1	I/O I/O I/O I/O	- 4-bit I/O Port. - CMOS input with pull-up resistor. - Pull-ups can be disabled by user Program. - Can be selectable as STOP release Input pin individually by user program. (It is released by “L” input at STOP mode) - N-ch open drain output. - Can be programmable as Push-pull output individually. - Direct Driving of LED(N-TR). - Each pin can be set and reset by PB Data register value.	Input (with Pull-up)	State of before STOP
PC0 PC1 PC2	OUTPUT OUTPUT OUTPUT	- N-ch open drain output. - Can be programmable as Push-pull output individually. - Direct Driving of LED(N-TR). - Each pin can be set and reset by PC Data register value.	Hi-Z	State of before STOP
PD0 PD1 PD2 PD3	OUTPUT OUTPUT OUTPUT OUTPUT	- N-ch open drain output. - Can be programmable as Push-pull output individually. - Direct Driving of LED(N-TR). - Each pin can be set and reset by PD Data register value.	Hi-Z	State of before STOP
OSC1	INPUT	- Oscillator Input.	oscillation	Low
OSC2	OUTPUT	- Oscillator Output.	oscillation	High
ROUT	OUTPUT	- High Current Output.	“L” output	“L” output
VDD	POWER	- Positive power supply.	-	-
GND	POWER	- Ground	-	-

1.6. Port Structure

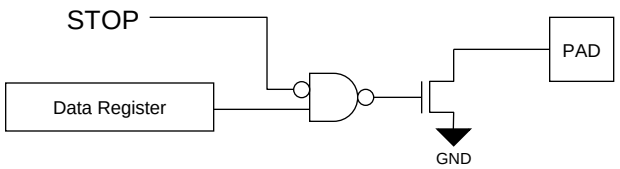
1.6.1. R0 Ports

Pin Name	Circuit Type	@ RESET
R00 R01 R02 R03		High-Input (with pull-up)

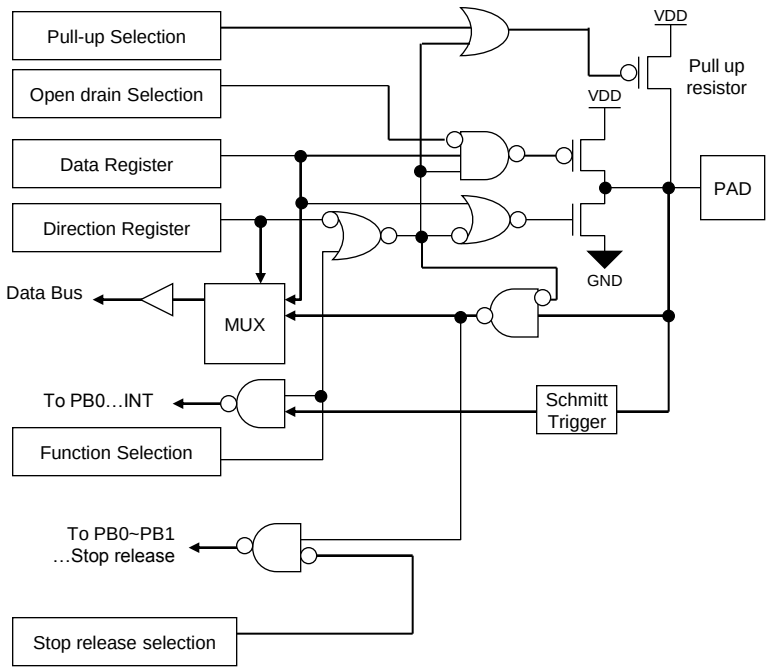
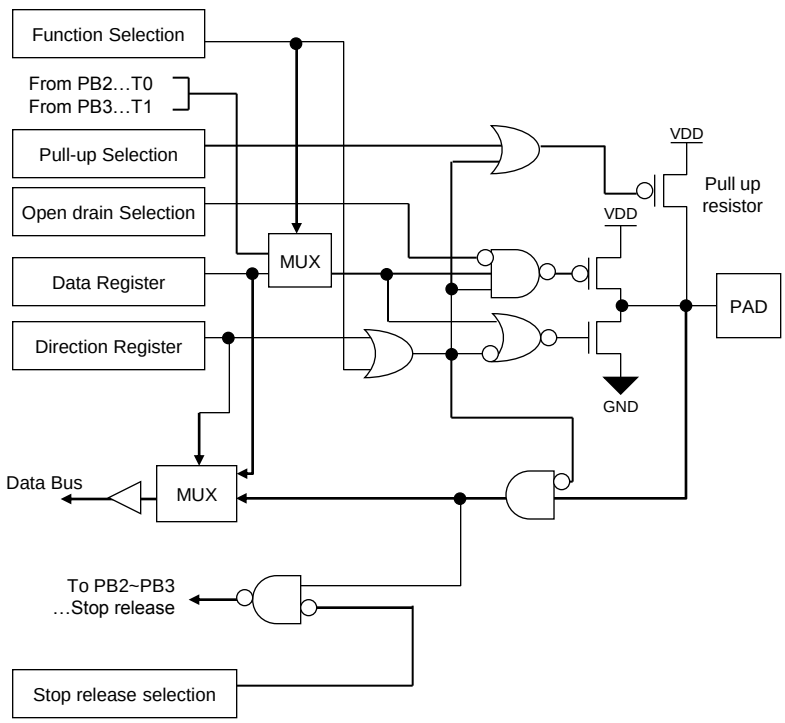
1.6.2. R1 Ports

Pin Name	Circuit Type	@ RESET
R10 R11 R12 R13		High-Input (with pull-up)

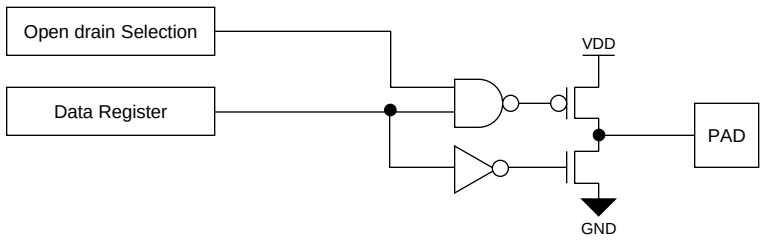
1.6.3. PA Ports

Pin Name	Circuit Type	@ RESET
PA0~PA3		Hi-Z

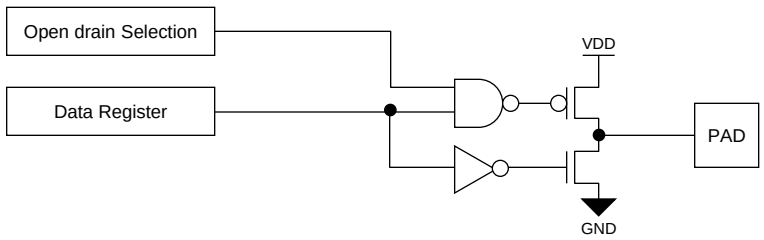
1.6.4. PB Ports

Pin Name	Circuit Type	@ RESET
PB0 / INT PB1		High-Input (with pull-up)
PB2 / T0 PB3 / T1		High-Input (with pull-up)

1.6.5. PC Ports

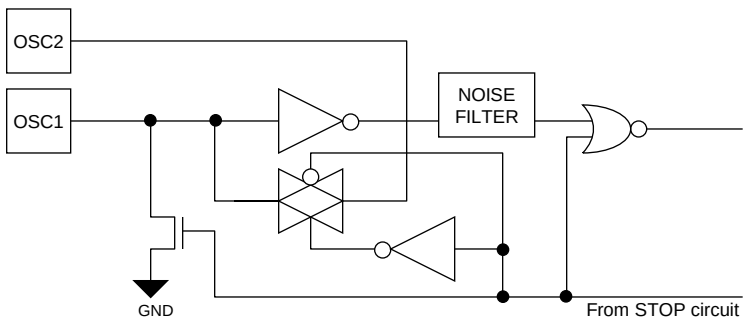
Pin Name	Circuit Type	@ RESET
PC0 ~ PC2		Hi -Z

1.6.6. PD Ports

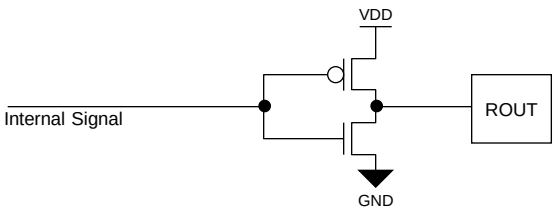
Pin Name	Circuit Type	@ RESET
PD0 ~ PD3		Hi- Z

NOTE > PD0~PD3 ports are not available in ADAM40P2724 and ADAM40S2724

1.6.7. OSC1, OSC2 Port

Pin Name	Circuit Type	@ RESET
OSC1 OSC2		Oscillation

1.6.8. ROUT Port

Pin Name	Circuit Type	@ RESET
ROUT		Hi-Z

1.7. Electrical Characteristics

1.7.1. Absolute Maximum Ratings (Ta = 25℃)

PARAMETER	SYMBOL	RATINGS	UNIT
Supply Voltage	V_{DD}	-0.3 ~ +6.0	V
Input Voltage	V_I	-0.3 ~ V_{DD} + 0.3	V
Output Voltage	V_O	-0.3 ~ V_{DD} + 0.3	V
Storage Temperature	T_{STG}	-65 ~ 150	℃
Power Dissipation	P_D	700	mW

* Thermal derating above 25℃ : 6mW per degree ℃ rise in temperature.

1.7.2. Recommended Operating Ranges

PARAMETER	SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Voltage	V_{DD}	f_{OSC} = 4MHz (ADAM40P272X)	1.8		3.6	V
		f_{OSC} = 4MHz (ADAM40S272X)	1.2		3.6	V
Oscillation Frequency	f_{OSC}		1.0		4.0	MHz
Operating Temperature	T_{OPR}		-20		70	℃

1.7.3. DC Characteristics (VDD = 3.6V~1.8V/1.2V , GND = 0V, Ta = -20℃ ~ 70℃)

PARAMETER	Symbol	Condition			Specification			UNIT
					MIN.	TYP.	MAX.	
High level input voltage	VIH1	INT			0.8VDD		VDD	V
	VIH2	R0, R1, PB			0.7VDD		VDD	V
Low level input voltage	VIL1	INT			0		0.2VDD	V
	VIL2	R0, R1, PB			0		0.3VDD	V
High level input leakage current	IIH	R0, R1, PB		VIH = VDD			1	μA
Low level input leakage current	IIL	R0, R1, PB (without Pull-up)		VIL = 0V			-1	μA
High level output voltage	VOH1	R1, PB, PC, PD		IOH = -1mA	VDD-0.4			V
	VOH2	OSC2		IOH = -200μA	VDD-0.9			V
Low level output voltage	VOL1	R1, PA		IOL = 1mA			0.8	V
	VOL2	PB, PC, PD		IOL = 5mA			0.8	V
	VOL3	OSC2		IOL = 200μA			0.8	V
High level output leakage current	IOHL	R1, PA, PB, PC, PD,		VOH = VDD			1	μA
Low level output leakage current	IOLL	R1, PA, PB, PC, PD		VOL = 0V			-1	μA
High level output current	IOH	ROUT		VDD = 3V VOH = 2V	-30	-12	-5	mA
Low level output current	IOL	ROUT		VDD = 3V VOL = 1V	0.5		5	mA
Input Pull-up current	IP	R0, R1, PB		VDD =3V	10	30	60	μA
Power supply current	IDD	Operating current	f_{XIN} = 4MHz	VDD = 3.6V		1.2	3	mA
				VDD = 2.0V		0.5	1.5	mA
				VDD = 1.3V		0.1	0.5	mA
	ISTOP	Stop mode current	Oscillator stop	VDD = 3.6V		2.2	8	μA
				VDD = 2.0V		0.5	1.5	μA
				VDD = 1.3V		0.1	1	μA
RAM retention supply voltage	VRET				0.7			V

1.7.4. AC Characteristics (VDD = 3.6V ~ 1.8V/1.2V, GND = 0V, Ta = -20℃ ~ 70℃)

No.	Parameter	Symbol	Pin	Specification			UNIT
				MIN.	TYP.	MAX.	
1	External clock input cycle time	t_{CP}	XIN	250	500	1000	ns
2	System clock cycle time	t_{sys}		3000	6000	12000	ns
3	Interrupt pulse width High	t_{IH}	INT	2			t_{sys}
4	Interrupt pulse width Low	t_{IL}	INT	2			t_{sys}

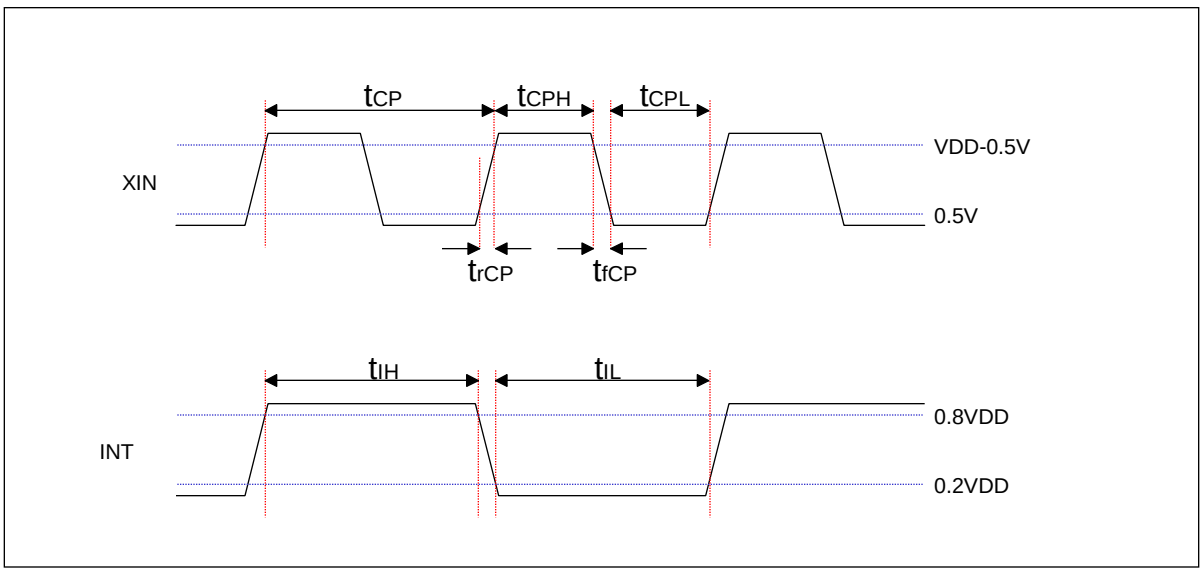


Fig. 1.1 Timing Chart

2. FUNCTION DESCRIPTION

2.1. Program Memory

The ADAM40X272X can incorporate maximum 48K bytes (24K × 16bits) for program memory. Program counter PC (A0~A13) and page address register(A14) are used to address the whole area of program memory having an instruction (16bits) to be executed.

The program memory consists of 16K words on 0-page and 8K words on 1-page.

The program memory is composed as shown below.

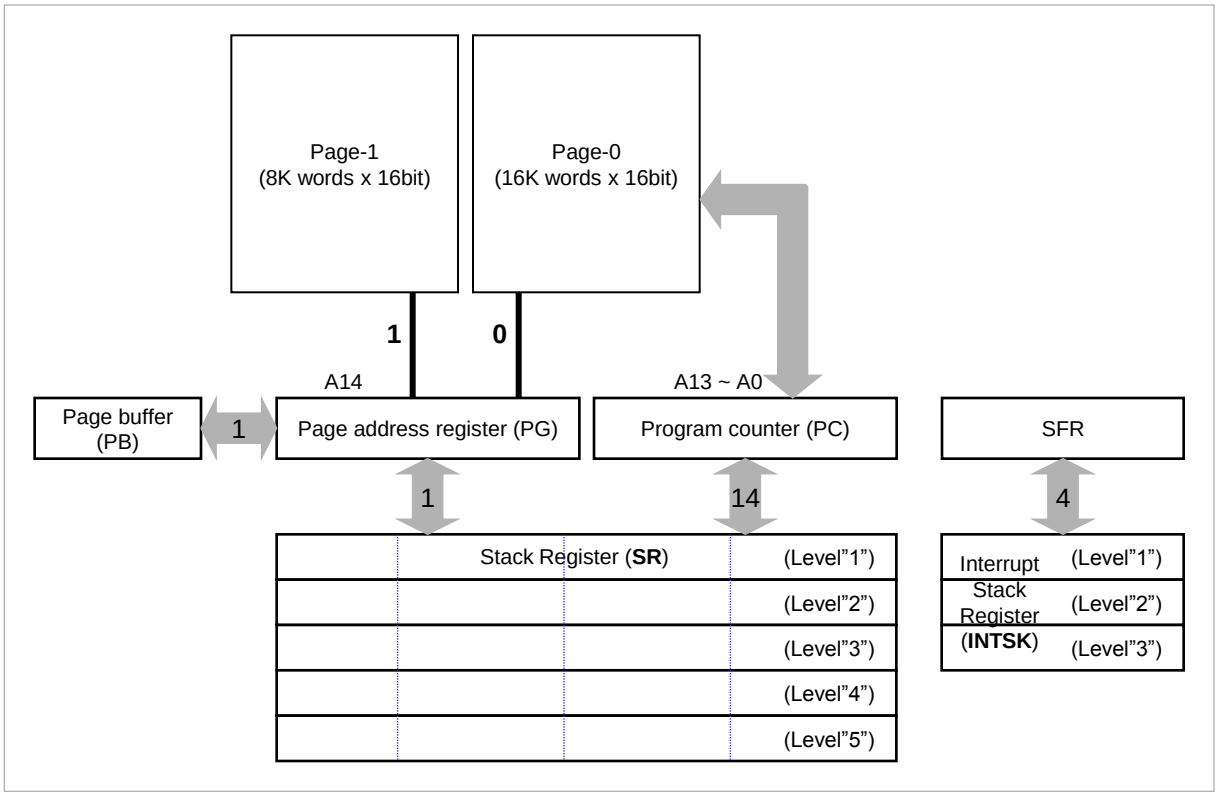


Fig. 2.1 Configuration of Program Memory

2.2. Address Register

The following registers are used to address the ROM.

- Page address register (PG) :
  Holds ROM's page number (0page, 1page) to be addressed.
- Page buffer register (PB) :
  Value of PB is loaded by an LPG command when newly addressing a page.
  Then it is shifted into the PG when rightly executing a branch instruction (BR)
  and a subroutine call (CAL).
- Program counter (PC) :
  Available for addressing word on each page.
- Stack register (SR) :
  Stores returned-word address in the subroutine call mode.

2.2.1. Page address register (PG) and page buffer register (PB)

Address one of 0 page to 1 page in the ROM by the 1-bit page address register.
Unlike the program counter, the page address register is usually unchanged so that the program will repeat on the same page unless a page changing command is issued. To change the page address, take two steps such as (1) writing in the page buffer what page to jump (execution of LPG) and (2) execution of BR or CAL, because instruction code is of eight bits so that page and word can not be specified at the same time.
In case a return instruction (RET) is executed within the subroutine that has been called in the other page, the page address will be changed at the same time.

2.2.2. Program counter (PC)

This 14-bit binary counter increments is for fetching a word to be addressed in the currently addressed page having an instruction to be next executed.
For easier programming, at turning on the power, the program counter is reset to the zero location(0000H). The PG is also set to "0". Then the program counter specifies the next address.
When BR, CAL or RET instructions are decoded, the switches on each step are turned off not to update the address.
Then, for BR or CAL, address data are taken in from the instruction operands (A0 to A13), or for RET, and address including page address is fetched from stack register No. 1.

2.2.3. Stack register (SR)

The stack register provides two stages each for the program counter (14bits) and the page address register (1bit) so that subroutine nesting can be made on five levels.

The address stack register (ADS) stores a return address when the subroutine call instruction is executed or interrupt is acknowledged.
If subroutine or interrupts are nested to more than 5 levels, internal reset is occurred.

The interrupt stack register(INTSK) saves the contents of Status Flag Register (SFR) when an interrupt is acknowledged.
The saved contents are restored when an interrupt return(RETI) instruction is executed.

INTSK saves data each time an interrupt is acknowledged.
The programmer must keep in mind that the level of INTSK is only 3. So, if more over 3 levels of interrupt occur, the first stored data is lost. There is different result between Stack overflow and interrupt stack overflow.
When clearing SP (Stack Pointer) with using “SPC” instruction, interrupt processing must be inhibited before “SPC”.

2.3. Data Memory (RAM)

256 nibbles (256 × 4bits) is incorporated for storing data.

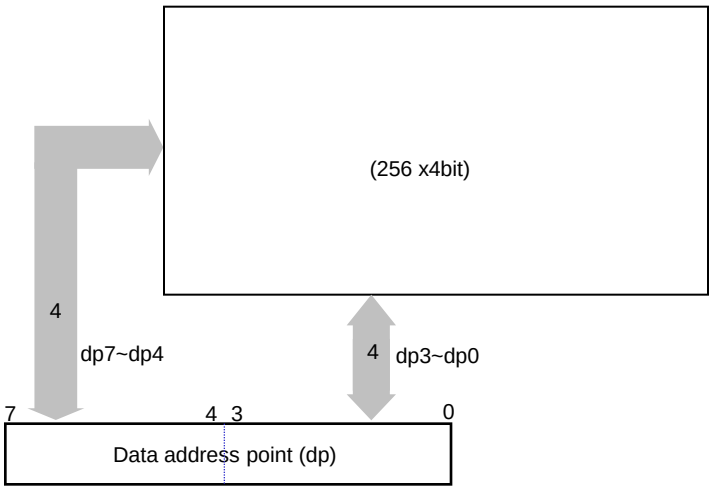
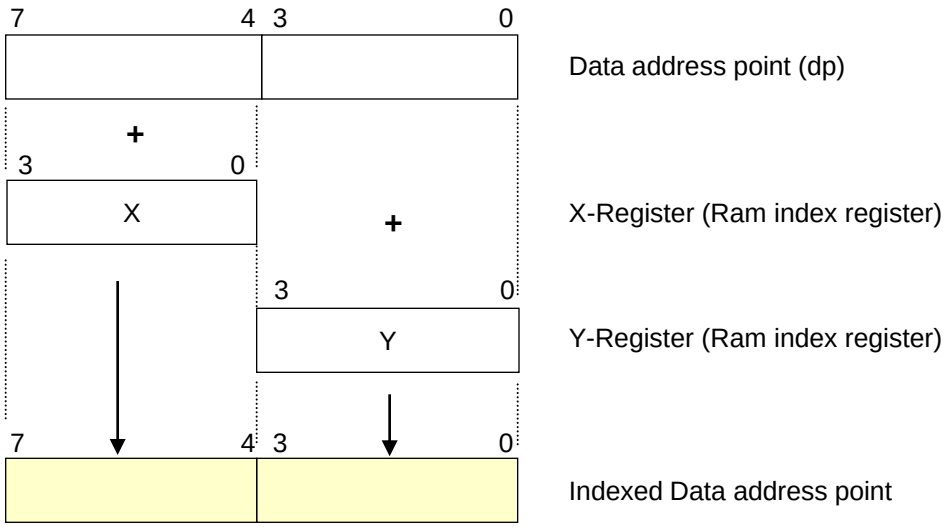


Fig. 2.2 Data Memory

2.3.1. Data memory (RAM) addressing method

The whole data memory area is directly addressed by 8-bit ram data address point (dp).
Index data memory addressing is available using X-register and Y-register.
In this case, X-register is added upper 4bit of data point and Y-register is added lower 4bit of data point.



2.3.2. Data memory(RAM) data addressing example Program

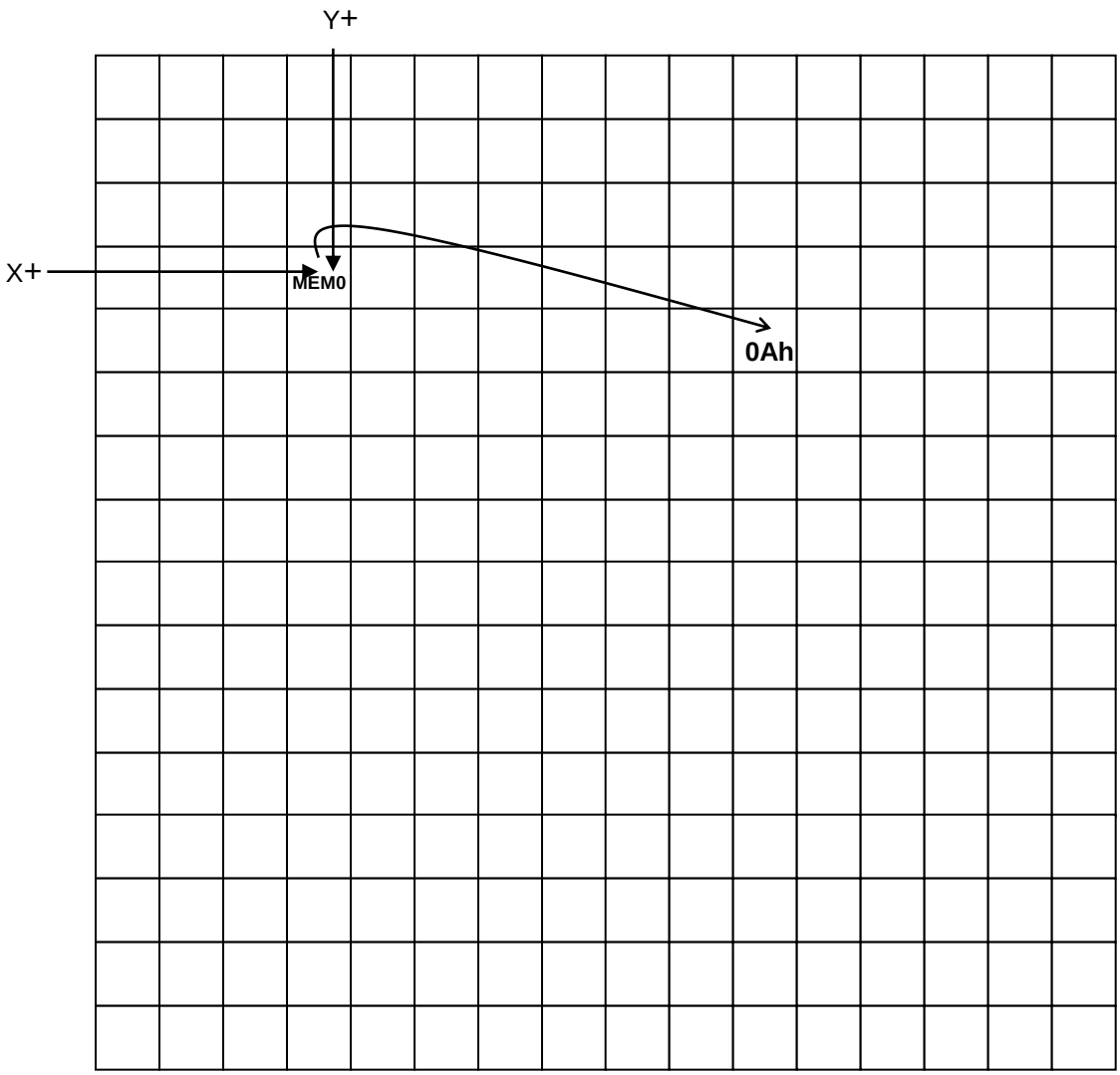


Fig. 2.3 Data Memory Map

Below program example is guidance for understanding the flow of index data memory addressing.

```
MEM0 EQU    033h                ; Defining RAM Address

       LDM    MEM0,#0Ah         ; [33h]  = #0Ah
       LYI    #7                ; Setting Y register as #07h
       LXI    #1                ; Setting X register as #01h
       LDA    MEM0              ; A = #0Ah == [MEM0]
       EIX                      ; Index Enable
       LDM    MEM0,A            ; [4Ah] == [Indexed Addressed Ram] = A
       DIX                       ; Index Disable
```

Result after executing is
MEM0 = #0Ah
[MEM0 + X + Y] = = [4Ah] = #0Ah

2.4. General Function Registers

2.4.1. X-register (X)

X-register consist of 4 bits. It can be used for a general-purpose register and also for data memory indexing register.

2.4.2. Y-register (Y)

Y-register consist of 4 bits. It can be used for a general-purpose register and also for data memory indexing register.

2.4.3. Accumulator (ACC)

The 4-bit register for holding data and calculation results.

2.4.4. Peripheral Address Register (PAR)

The 6-bit address register for addressing peripheral registers including address buffer register (ABR), data buffer register (DBR).

2.4.5. Address Buffer Register (ABR)

The 15-bit register for address buffer.

It is composed of 3 registers(ABR0, ABR1, ABR2) x 4bit and 1 register(ABR3) x 3bit.

The address of Address Buffer Register (ABR) is 38h ~3Bh on the peripheral register (PAR).

2.4.5.1. Function of Address Buffer Register (ABR)

The most important function of ABR is ROM address pointer.

If ROM address pointer is 3456h, each ABR must be written as ABR3=3h, ABR2=4h, ABR1=5h, ABR0=6h.

ABR must be used for reading data from ROM. The data pointed by ABR is read to DBR.

ABR value is varied through peripheral control instruction and “INC ABR”.

2.4.6. Data Buffer Register (DBR)

The 16-bit register for data buffer.

It is composed by 4 registers (DBR0, DBR1, DBR2, DBR3) and each register is 4bit.

Data buffer register on addresses 3Ch ~ 3Fh of the peripheral register address (PAR) is used for data transfer to and from peripheral hardware (Timer, Interrupt, port) via Accumulator.

2.4.6.1. Function of Data Buffer Register (DBR)

The most important function of DBR is intermediate (window) buffer for transferring data between peripheral registers and reading data from ROM.
When the data of ROM is read by "LDW @ABR", one word of ROM is fetched to DBR.
The MSB of ROM data is written to DBR3 and LSB to DBR0.
If the data of pointed ROM is 1234h, each DBR has the data as DBR0 = 4h, DBR1 = 3h, DBR2 = 2h and DBR3 = 1h.
DBR is also used for reading some peripheral register data by 8bit unit. The peripheral registers are T0CR and T1CR.

Note > HEX. File maps the data as big endian type. Be careful to read the ROM data. When the programmer assigns the data like below, the ROM data is mapped as below
DB 12h,34h → ROM data = 1234h

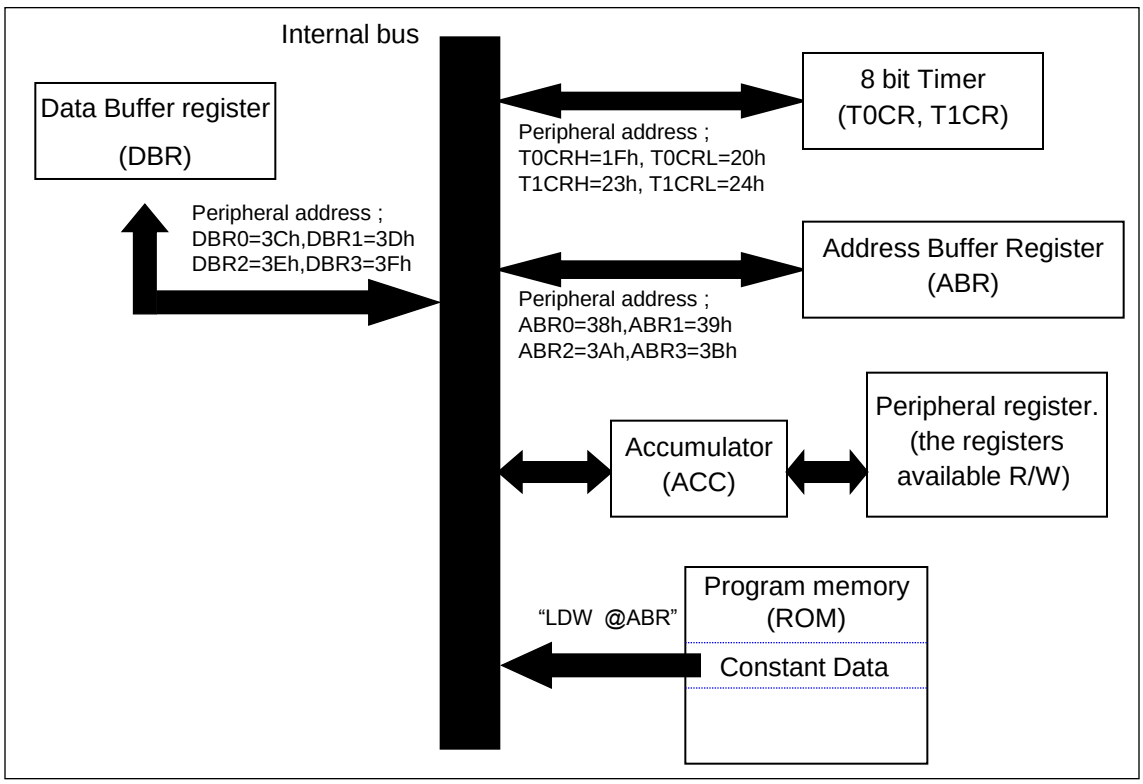


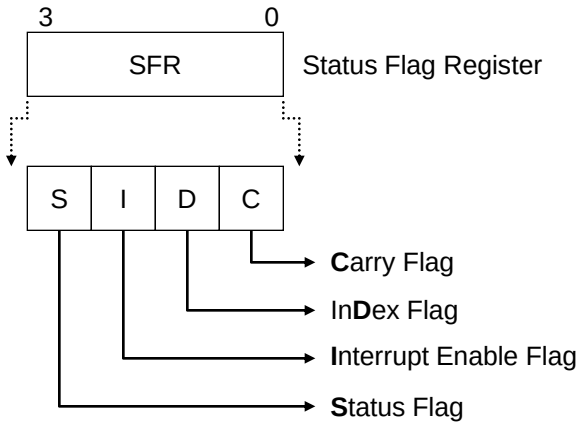
Fig. 2.4 The internal Data flow among DBR, ABR , registers and ROM

2.6. Status Flag Register (SFR)

Status Flag Register (SFR) consists of 4-bit register.

Each of the flags show the post state of operation and the flags determining the CPU operation, initialized as 0h in reset state.

When an interrupt is occurred, the value of SFR keep the value of pre- interrupt except for I flag. So, be careful to initialize the SFR status for getting reliable result in Interrupt sub-routine.



2.6.1. **C**arry flag (**C**)

- Carry flag bit is set when there is carry or borrow After executing ADDC / SUBC / ARRC / ARLC instructions.
- Set by SETC and clear by CLRC.

2.6.2. In**D**ex flag (**D**)

- The control bit of ram data address point indexed or not.
- X-register and Y-register is used for index addressing.
- Set and cleared by EIX, DIX.

2.6.3. **I**nterrupt enable flag (**I**)

- Master enable flag of interrupt.
- Set and cleared by EI, DI
- This flag immediately becomes “0” when an interrupt is served.

2.6.4. **S**tatus flag (**S**)

- According to the condition after executing an instruction , set or clear.
- Can not be set or clear by any instruction.
- This flag decides whether operation of BR and CALL would be done or not.

2.7. Peripheral Registers

Peripheral Address	Function Registers	Read Write	Symbol	RESET Value			
				3	2	1	0
00 h	PORT R0 STOP RELEASE SELECTION REG.	W	R0ST	0			
	PORT R0 DATA REG.	R	R0	F			
01 h	PORT R1 STOP RELEASE SELECTION REG.	W	R1ST	0			
02 h	PORT R1 PULL UP RESISTOR SELECTION REG.	W	R1PC	0			
03 h	PORT R1 OPEN DRAIN SELECTION REG.	W	R1OD	0			
04 h	PORT R1 DATA REG.	R/W	R1	F			
05 h	PORT R1 DIRECTION REG.	W	R1DD	0			
06 h	PORT PA DATA REG.	R/W	PA	F			
07 h	PORT PB STOP RELEASE SELECTION REG.	W	PBST	F			
08 h	PORT PB PULL UP RESISTOR SELECTION REG.	W	PBPC	0			
09 h	PORT PB OPEN DRAIN SELECTION REG.	W	PBOD	0			
0A h	PORT PB DATA REG.	R/W	PB	F			
0B h	PORT PB FUNCTION SELECTION REG.	W	PBFN	0	0	-	0
0C h	PORT PB DIRECTION REG.	W	PBDD	0			
0D h	PORT PC OPEN DRAIN SELECTION REG.	W	PCOD	0			
0E h	PORT PC DATA REG.	R/W	PC	F			
0F h	PORT PD OPEN DRAIN SELECTION REG.	W	PDOD	0			
10 h	PORT PD DATA REG.	R/W	PD	F			
11 h	*Reserved*						
12 h	*Reserved*						
13 h	*Reserved*						
14 h	*Reserved*						
15 h	*Reserved*						
16 h	*Reserved*						
17 h	EXTERNAL INT. EDGE SELECTION REG.	W	IEDS	-	-	0	0
18 h	INT. ENABLE REG.	R/W	IENR	-	0	0	0
19 h	INT. REQUEST FLAG REG.	R/W	IRQR	-	0	0	0
1A h	TIMER0 MODE REG.	R/W	T0MR	0			
1B h	*Reserved*						
1C h	TIMER1 MODE REG.	R/W	T1MR	0			
1D h	ROUT CONTROL REG .	R/W	RCR	0			
1E h	CARRY MODE REG.	R/W	CGMR	0			
1F h	TIMER 0 DATA0 HIGH REG.	W	T0D0H	undefined			
	TIMER 0 COUNT REG. HIGH.	R	T0CRH	undefined			

Note1> '-' is reserved bit , it must be read to "0".

Peripheral Address	Function Registers	Read Write	Symbol	RESET Value			
				3	2	1	0
20 h	TIMER 0 DATA 0 LOW REG.	W	T0D0L	undefined			
	TIMER 0 COUNT REG. LOW.	R	T0CRL	undefined			
21 h	TIMER 0 DATA 1 HIGH REG.	W	T0D1H	undefined			
22 h	TIMER 0 DATA 1 LOW REG.	W	T0D1L	undefined			
23 h	TIMER 1 HIGH DATA REG.	W	T1HD	undefined			
	TIMER 1 COUNT REG. HIGH.	R	T1CRH	undefined			
24 h	TIMER 1 LOW DATA REG.	W	T1LD	undefined			
	TIMER 1 COUNT REG LOW.	R	T1CRL	undefined			
25 h	CARRY GENERATOR HIGH-MSB DATA REG.	W	CGHMD	undefined			
26 h	CARRY GENERATOR HIGH-LSB DATA REG.	W	CGHLD	undefined			
27 h	CARRY GENERATOR LOW-MSB DATA REG.	W	CGLMD	undefined			
28 h	CARRY GENERATOR LOW-LSB DATA REG.	W	CGLLD	undefined			
29 h	*Reserved*						
2A h	*Reserved*						
2B h	*Reserved*						
2C h	*Reserved*						
2D h	*Reserved*						
2E h	*Reserved*						
2F h	*Reserved*						
30 h	*Reserved*						
31 h	*Reserved*						
32 h	*Reserved*						
33 h	*Reserved*						
34 h	*Reserved*						
35 h	*Reserved*						
36 h	*Reserved*						
37 h	*Reserved*						
38 h	ADDRESS BUFF REGISTER 0	R/W	ABR0	undefined			
39 h	ADDRESS BUFF REGISTER 1	R/W	ABR1	undefined			
3A h	ADDRESS BUFF REGISTER 2	R/W	ABR2	undefined			
3B h	ADDRESS BUFF REGISTER 3	R/W	ABR3	undefined			
3C h	DATA BUFF REGISTER 0	R/W	DBR0	undefined			
3D h	DATA BUFF REGISTER 1	R/W	DBR1	undefined			
3E h	DATA BUFF REGISTER 2	R/W	DBR2	undefined			
3F h	DATA BUFF REGISTER 3	R/W	DBR3	undefined			

3. I/O Ports

The ADAM40X272X has maximum 23 Input or output ports which are R0 (4 Input), R1 (4 I/O), PA (4 Output), PB (4 I/O), PC (3 Output), PD (4 Output).
R0, R1 and PB input Port have Stop Release selection register.
Pull-up resistors of R1 and PB port can be selectable by program. R1 and PB port contains data direction register which controls I/O and data register which stores port data.
R1, PA, PB, PC, PD Ports have Open Drain output selection register.

3.1. Port R0

3.1.1. R0 Stop Release Selection Register (R0ST)

bit	3	2	1	0	
R0ST	R0ST3	R0ST2	R0ST1	R0ST0	*00h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

R0 Stop Release Selection Register (R0ST) is 4-bit register and can assign stop release pin or not.
If R0ST is selected as “0”, stop release function is enabled and if selected as “1”, it is disabled. R0ST is write-only register and initialized as “0h” in reset state.

3.1.1. R0 DATA Register (R0)

bit	3	2	1	0	
R0	R03	R02	R01	R00	*00h*
Initial value	1	1	1	1	
R/W	R	R	R	R	

R0 data register (R0) is 4-bit register to store data of port R0.
Since R0 port is input only port, input state of pin is read.
The initial value of R0 is “Fh” in reset state.

3.2. Port R1

3.2.1. R1 Stop Release Selection Register (R1ST)

bit	3	2	1	0	
R1ST	R1ST3	R1ST2	R1ST1	R1ST0	*01h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

R1 Stop Release Selection Register (R1ST) is 4-bit register and can assign stop release pin or not.
If R1ST is selected as “0”, stop release function is enabled and if selected as “1”, it is disabled. R1ST is write-only register and initialized as “0h” in reset state.

3.2.2. R1 Pull-up Resistor Control Register (R1PC)

bit	3	2	1	0	
R1PC	R1PC3	R1PC2	R1PC1	R1PC0	*02h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

R1 pull-up resistor control register (R1PC) is 4-bit register and can control pull-up on or off each port, if corresponding port is selected as input. If R1PC is selected as “0”, pull-up is enabled and if selected as “1”, it is disabled. R1PC is write-only register and initialized as “0h” in reset state.
The pull-up is automatically disabled, if corresponding port is selected as output.

3.2.3. R1 Open Drain Assign Register (R1OD)

bit	3	2	1	0	
R1OD	R1OD3	R1OD2	R1OD1	R1OD0	*03h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

R1 Open Drain Assign Register (R1OD) is 4-bit register, and can assign R1 port as open drain output port for each bit. If R1OD is selected as “0”, port R1 is open drain output, and if selected as “1”, it is push-pull output. R1OD is write-only register and initialized as “0h” in reset state.

3.2.4. R1 Data Register (R1)

bit	3	2	1	0	
R1	R13	R12	R11	R10	*04h*
Initial value	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	

R1 data register (R1) is 4-bit register to store data of port R1.
When set as the output state by R1DD, written data in R1 is outputted through R1 pin.
When set as the input state, input state of pin is read to R1.
The initial value of R1 is “Fh” in reset state.

3.2.5. R1 I/O Data Direction Register (R1DD)

bit	3	2	1	0	
R1DD	R1DD3	R1DD2	R1DD1	R1DD0	*05h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

R1 I/O Data Direction Register (R1DD) is 4-bit register, and can assign input state or output state to each bit. If R1DD is “0”, port R1 is in the input state, and if “1”, it is in the output state. R1DD is write-only register. Since R1DD is initialized as “0h” in reset state, the whole port R1 becomes input state.

3.3. Port PA

3.3.1. PA Data Register (PA)

bit	3	2	1	0	
PA	PA3	PA2	PA1	PA0	*06h*
Initial value	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	

PA data register (PA) is 4-bit register to store data of port PA.
The initial value of PA is “Fh” in reset state.

3.4. Port PB

Pin Name	Port Selection	Function Selection
PB0 / INT	PB0 (I/O)	INT input
PB1	PB1 (I/O)	-
PB2 / T0	PB2 (I/O)	Timer 0 output
PB3 / T1	PB3 (I/O)	Timer 1 output

3.4.1. PB Stop Release Selection Register (PBST)

bit	3	2	1	0	
PBST	PBST3	PBST2	PBST1	PBST0	*07h*
Initial value	1	1	1	1	
R/W	W	W	W	W	

PB Stop Release Selection Register (PBST) is 4-bit register, and can assign stop release pin or not.

If PBST is selected as “0”, stop release function is enabled and if selected as “1”, it is disabled. PBST is write-only register and initialized as “Fh” in reset state.

3.4.2. PB Pull-up Resistor Control Register (PBPC)

bit	3	2	1	0	
PBPC	PBPC3	PBPC2	PBPC1	PBPC0	*08h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

PB pull-up resistor control register (PBPC) is 4-bit register and can control pull-up on or off each bit, if corresponding port is selected as input. If PBPC is selected as “0”, pull-up is enabled and if selected as “1”, it is disabled. PBPC is write-only register and initialized as “0h” in reset state.

The pull-up is automatically disabled, if corresponding port is selected as output.

3.4.3. PB Open Drain Assign Register (PBOD)

bit	3	2	1	0	
PBOD	PBOD3	PBOD2	PBOD1	PBOD0	*09h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

PB Open Drain Assign Register (PBOD) is 4-bit register, and can assign PB port as open drain output port for each bit. If PBOD is selected as “0”, port PB is open drain output, and if selected as “1”, it is push-pull output. PBOD is write-only register and initialized as “0h” in reset state.

3.4.4. PB Data Register (PB)

bit	3	2	1	0	
PB	PB3	PB2	PB1	PB0	*0Ah*
Initial value	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	

PB data register (PB) is 4-bit register to store data of port PB.

The initial value of PB is “Fh” in reset state.

3.4.5. PB Function selection Register (PBFN)

bit	3	2	1	0	
PBFN	T1S	T0S	-	INTS	*0Bh*
Initial value	0	0	0	0	
R/W	W	W	W	W	

Note> Reserved bit must be 0.

PB Function selection Register (PBFN) is 4-bit register, and can assign the Function mode for each bit.

When set as “0”, corresponding bit of PBFN acts as port PB selection mode, and when set as “1”, it becomes function selection mode.

Selection Mode of PBFN

Bit Name	PMR1	Selection Mode	Remarks
T1S	0	PB3 select (I/O)	-
	1	Timer1 output select (output)	The output status is toggled (L<>H) every T1 Output (refer to Fig. 4.5)
T0S	0	PB2 select (I/O)	-
	1	Timer0 output select (output)	The output status is same as T0 Output (refer to Fig. 4.4)
-			-
			-
INTS	0	PB0 select (I/O)	-
	1	External Interrupt select (input)	-

PBFN is write-only register and initialized as “0h” in reset state.
Therefore, becomes I/O Port mode.

3.4.6. PB I/O Data Direction Register (PBDD)

bit	3	2	1	0	
PBDD	PBDD3	PBDD2	PBDD1	PBDD0	*0Ch*
Initial value	0	0	0	0	
R/W	W	W	W	W	

PB I/O Data Direction Register (PBDD) is 4-bit register, and can assign input state or output state to each bit. If PBDD is “0”, port PB is in the input state, and if “1”, it is in the output state. PBDD is write-only register. Since PBDD is initialized as “0h” in reset state, the whole port PB becomes input state.

3.5. Port PC

3.5.1. PC Open Drain Assign Register (PCOD)

bit	3	2	1	0	
PCOD	-	PCOD2	PCOD1	PCOD0	*0Dh*
Initial value	-	0	0	0	
R/W	-	W	W	W	

PC Open Drain Assign Register (PCOD) is 3-bit register, and can assign PC port as open drain output port for each bit. If PCOD is selected as “0”, port PC is open drain output, and if selected as “1”, it is push-pull output. PCOD is write-only register and initialized as “0h” in reset state.

3.5.2. PC Data Register (PC)

bit	3	2	1	0	
PC	-	PC2	PC1	PC0	*0Eh*
Initial value	-	1	1	1	
R/W	-	R/W	R/W	R/W	

PC data register (PC) is 3-bit register to store data of port PC.
The initial value of PC is “Fh” in reset state.

3.6. Port PD

3.6.1. PD Open Drain Assign Register (PDOD)

bit	3	2	1	0	
PDOD	PDOD3	PDOD2	PDOD1	PDOD0	*0Fh*
Initial value	0	0	0	0	
R/W	W	W	W	W	

PD Open Drain Assign Register (PDOD) is 4-bit register, and can assign PD port as open drain output port for each bit. If PDOD is selected as “0”, port PD is open drain output, and if selected as “1”, it is push-pull output. PDOD is write-only register and initialized as “0h” in reset state.

3.6.2. PD Data Register (PD)

bit	3	2	1	0	
PD	PD3	PD2	PD1	PD0	*10h*
Initial value	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	

PD data register (PD) is 4-bit register to store data of port PD.
The initial value of PD is “Fh” in reset state.

4. PERIPHERAL HARDWARE

4.1. Oscillation Circuit

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Fig. 4.1 shows circuit diagrams using a crystal (or ceramic) oscillator and external clock. As shown in the diagram, oscillation circuits can be constructed by connecting a oscillator Between OSC1 and OSC2. Clock from oscillation circuit makes CPU clock via clock pulse generator, and then provide peripheral hardware clock. Alternately, the oscillator may be driven from an external source as shown is Fig. 4.1. In the Standby (STOP) mode, oscillation stop, OSC2 state goes to "High", OSC1 state goes to "Low", and built-in feedback resistor is disabled.

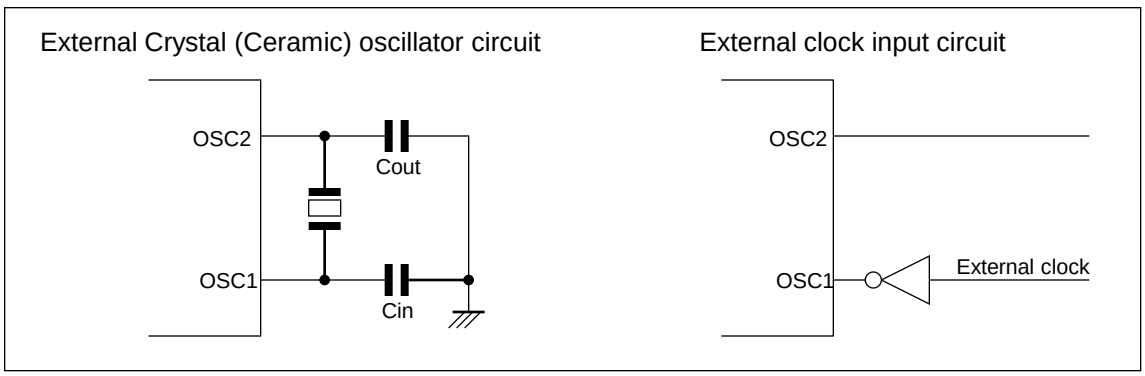


Fig. 4.1 Oscillator configurations

4.2. Watch Dog Timer (WDT)

Watch Dog Timer (WDT) is organized binary of 17 steps. The signal of $f_{osc}/12$ cycle comes in the first step of WDT after WDT reset. If the last step would be "1", reset signal automatically comes out and internal circuit is initialized. The overflow time is $2^{16} \times 12/f_{osc}$ (196.608ms at f_{osc} = 4.0MHz). Normally, the binary counter must be reset before the overflow by using reset instruction (WDTC), Power-on reset pulse or Low VDD detection pulse.

* It is constantly reset in STOP mode. When STOP is released, counting is restarted.

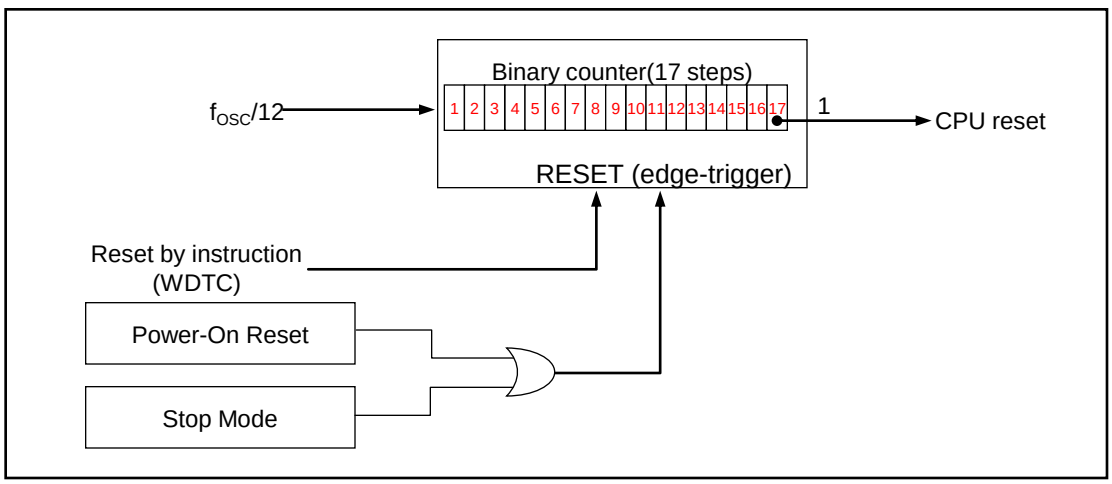


Fig. 4.2 Block Diagram of Watch Dog Timer (WDT)

4.3. Timer

4.3.1. Timer operation mode

Timer is basically made of Timer Data Register, Timer Mode Register and control circuit. The types of Timer are 8bit binary counter Timer0 (T0), 8bit binary counter Timer1 (T1), Carrier Generator (CG).

Timer0 Data Register consists of Timer0 Data 0 High Register (T0D0H), Timer0 Data 0 Low Register (T0D0L), Timer0 Data 1 High Register (T0D1H) and Timer0 Data 1 Low Register (T0D1L).
Timer1 Data Register consists of Timer1 Low Data Register (T1LD), Timer1 High Data Register (T1HD).
Carrier Generator consists of Carrier Generator High MSB Data Register (CGHMD), Carrier Generator High LSB Data Register (CGHLD), Carrier Generator Low MSB Data Register (CGLMD) and Carrier Generator Low LSB Data Register (CGLLD).

Timer0	- 8-bit Interval Timer - 8-bit rectangular-wave output
Timer1	- 8-bit Interval Timer - 8-bit rectangular-wave output
Carrier Generator	- 6-bit up-Counter - 6-bit rectangular-wave output

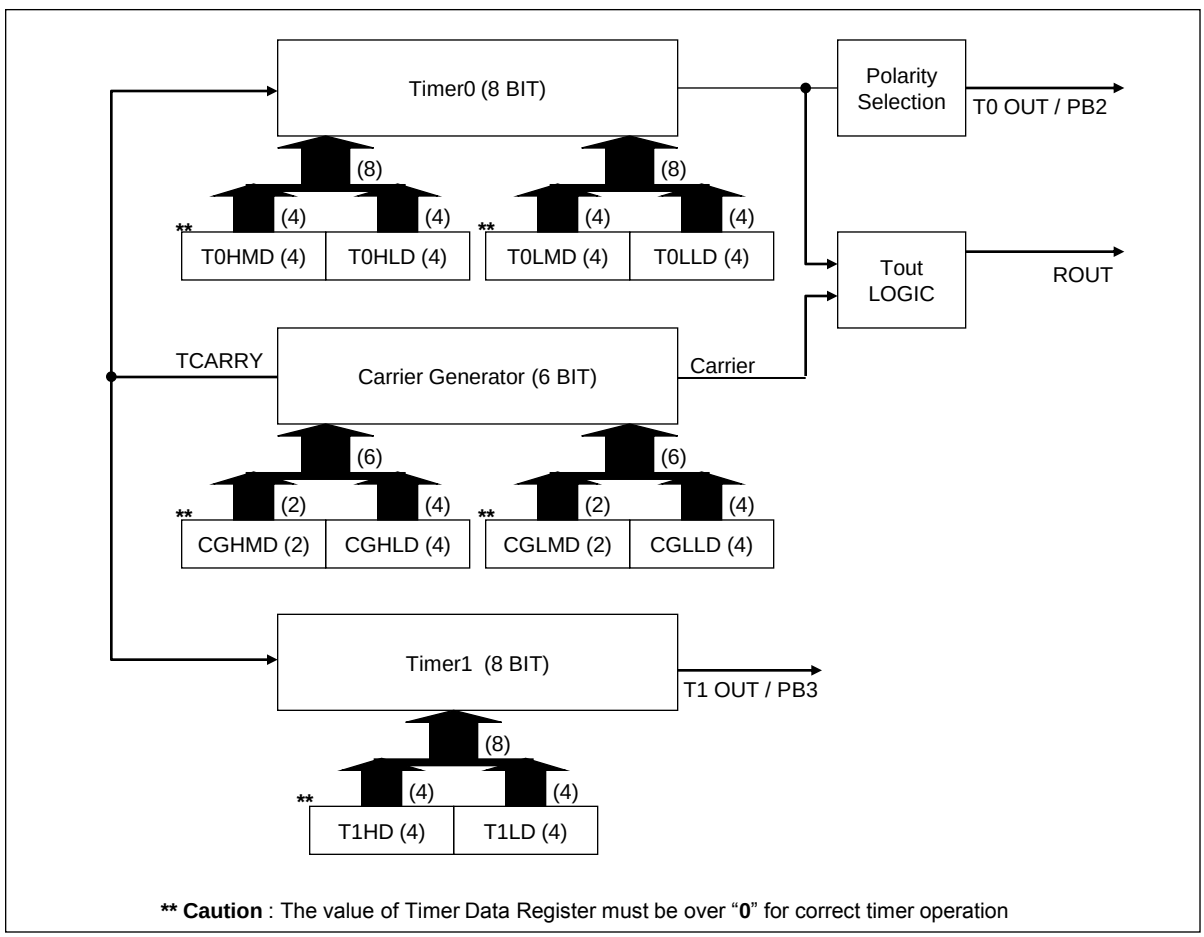


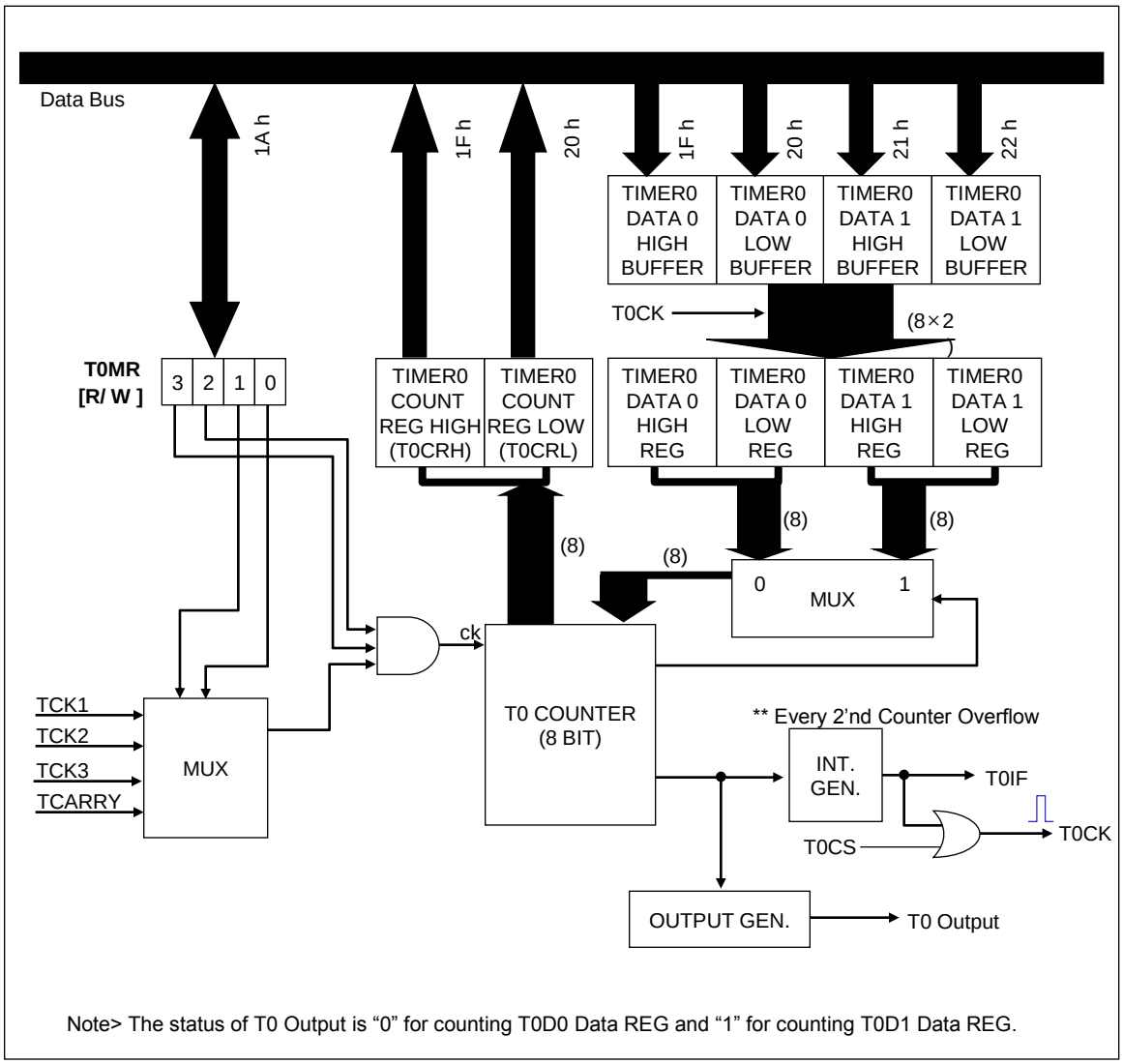
Fig. 4.3 Timer / Counter Block diagram

4.3.2. Function of Timer & Counter

fosc = 4MHz

8bit Timer (Timer0)		8bit Timer (Timer1)		Carrier Generator (CG)	
Resolution (CK)	Max. Count	Resolution (CK)	Max. Count	Resolution (CK)	Max. Count
TCK1 : 0. 5 us	128 us	TCK1 : 0.5 us	128 us	TCK1 : 0. 5 us	32 us
TCK2 : 1 us	256 us	TCK2 : 1 us	256 us	TCK2 : 1 us	64 us
TCK3 : 2 us	512 us	TCK3 : 2 us	512 us	TCK3 : 2 us	128 us
TCARRY(*)		TCARRY(*)		TCK4 : 4 us	256 us

(*) Resolution & Max. count of TCARRY clock is decided by output of Carrier Generator



Note> The status of T0 Output is "0" for counting T0D0 Data REG and "1" for counting T0D1 Data REG.

Fig. 4.4 Block Diagram of Timer0

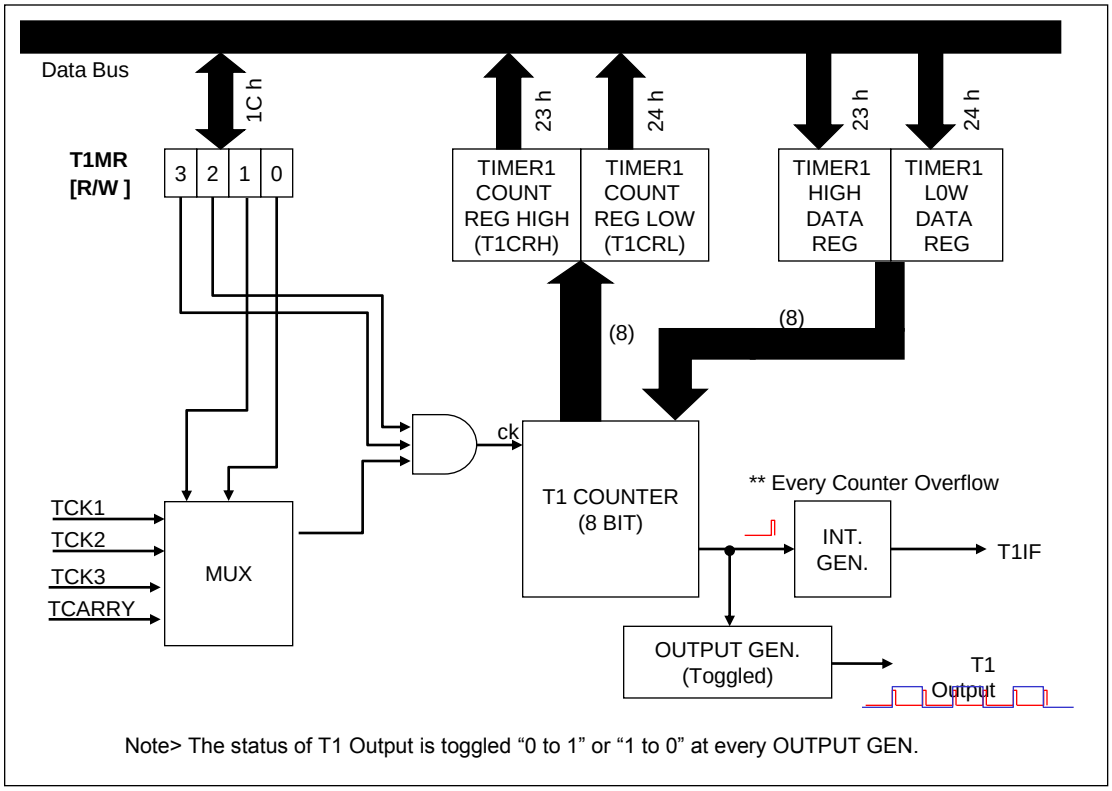


Fig. 4.5 Block Diagram of Timer1

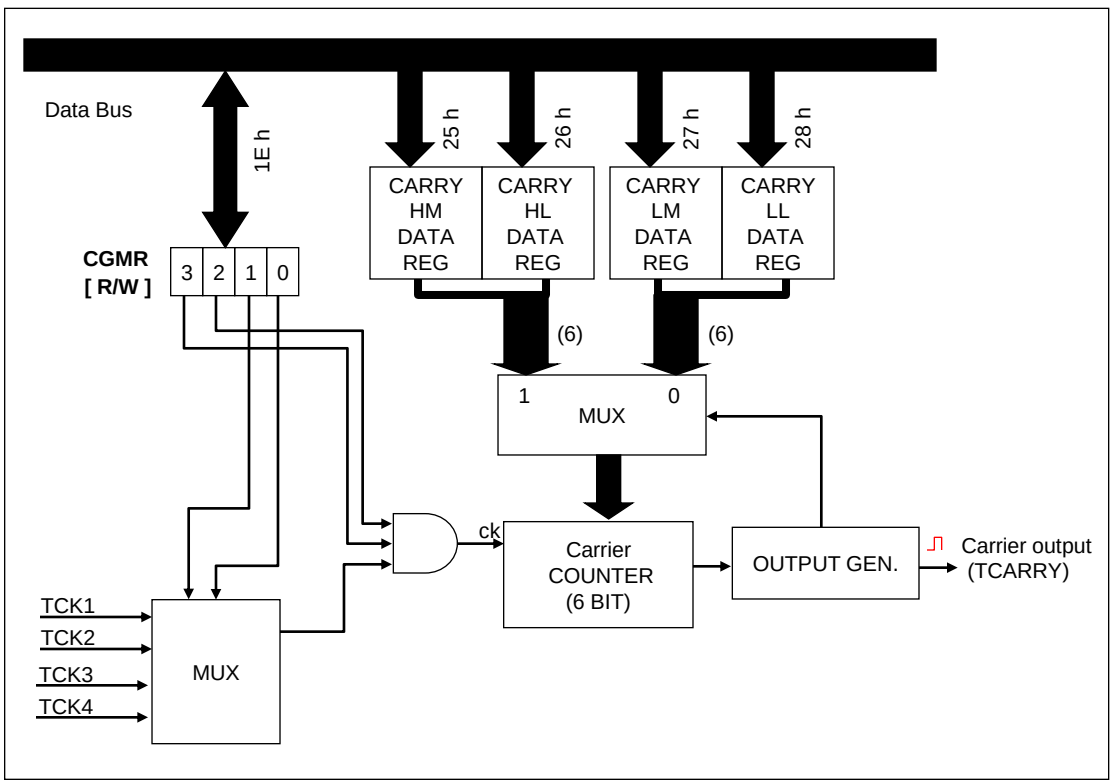


Fig. 4.6 Block Diagram of Carrier Generator

4.3.2.1. Timer0 Mode Register

bit	3	2	1	0	
T0MR [R/W]	T0CS	T0CN	T0CK1	T0CK0	*1Ah*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

T0CK1 & T0CK0	Input clock selection	00	TCK1 (500ns)
		01	TCK2 (1us)
		10	TCK3 (2us)
		11	TCARRY (output of Carrier Generator)
T0CN	Timer0 Pause / Continue Control	0	Timer0 Pause
		1	Timer0 continue
T0CS	Timer0 Clear / start Control	0	Timer0 Stop
		1	Timer0 Clear and Start

* Timer 0 counts with 'T0D0H+T0D0L' first , after overflowed counts with 'T0D1H+T0D1L'

4.3.2.2. Timer1 Mode Register

bit	3	2	1	0	
T1MR [R/W]	T1CS	T1CN	T1CK1	T1CK0	*1Ch*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

T1CK1 & T1CK0	Input clock selection	00	TCK1 (500ns)
		01	TCK2 (1us)
		10	TCK3 (2us)
		11	TCARRY (output of Carrier Generator)
T1CN	Timer1 Pause / Continue Control	0	Timer1 Pause
		1	Timer1 continue
T1CS	Timer1 Clear / start Control	0	Timer1 Stop
		1	Timer1 Clear and Start

4.3.2.3. Carrier Generator Mode Register

CGMR [R/W]	CGCS	CGON	CGCK1	CGCK0	*1Eh*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

CGCK1 & CGCK0	Input clock selection	00	TCK1 (500ns)
		01	TCK2 (1us)
		10	TCK3 (2us)
		11	TCK4 (4us)
CGON	Carrier Generator Output Control	0	Output of ROUT without Carrier Pulse
		1	Output of ROUT with Carrier Pulse
CGCS	Carrier Generator Clear / start Control	0	Carrier Generator Stop
		1	Carrier Generator Clear and Start

4.3.2.4. ROUT Control Register

bit	3	2	1	0	
RCR [R/W]	PR1	PR0	PRON	REM	*1Dh*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

REM	Output of ROUT Bit Control (When PRON=0)	0	ROUT output low
		1	ROUT output high
PRON	PR0 / PR1 Function Control	0	PR0 / PR1 Function Disable (“REM” bit active)
		1	PR0 / PR1 Function Enable (“REM” bit inactive)
PR0	Preset of ROUT Bit Control (When PRON=1)	0	ROUT “L” on counting Timer0 DATA0 (T0D0H + T0D0L)
		1	ROUT “H” on counting Timer0 DATA0 (T0D0H + T0D0L)
PR1	Preset of ROUT Bit Control (When PRON=1)	0	ROUT “L” on counting Timer0 DATA1 (T0D1H + T0D1L)
		1	ROUT “H” on counting Timer0 DATA1 (T0D1H + T0D1L)

* ROUT Pin is Controlled by REM when PRON bit =0
* ROUT Pin is Controlled by PR0, PR1 when PRON bit =1
Because Timer0 counts with Data0 (T0D0H+T0D0L) first , ROUT pin is controlled by PR0 initially.

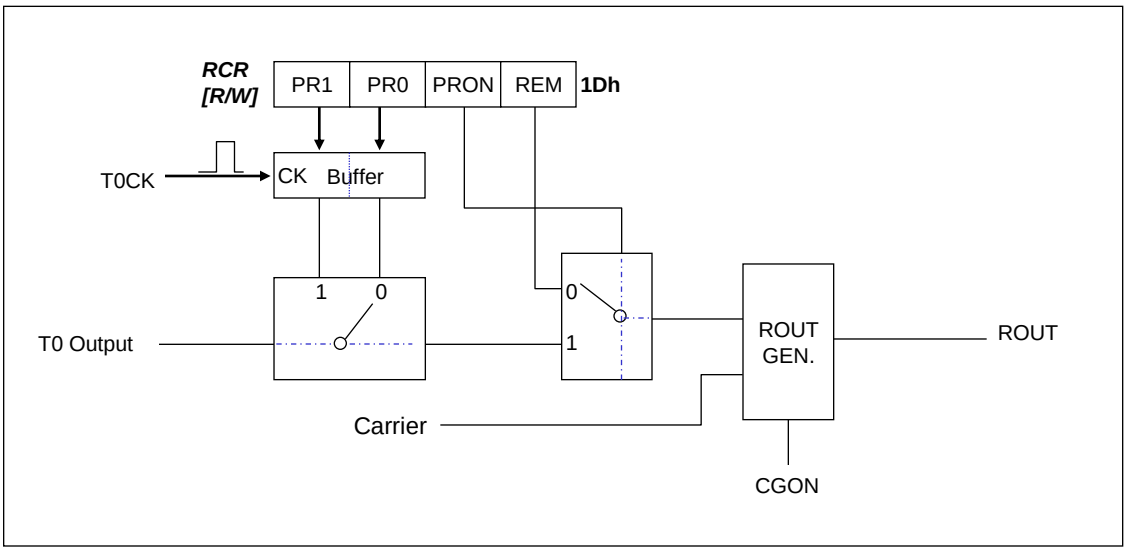
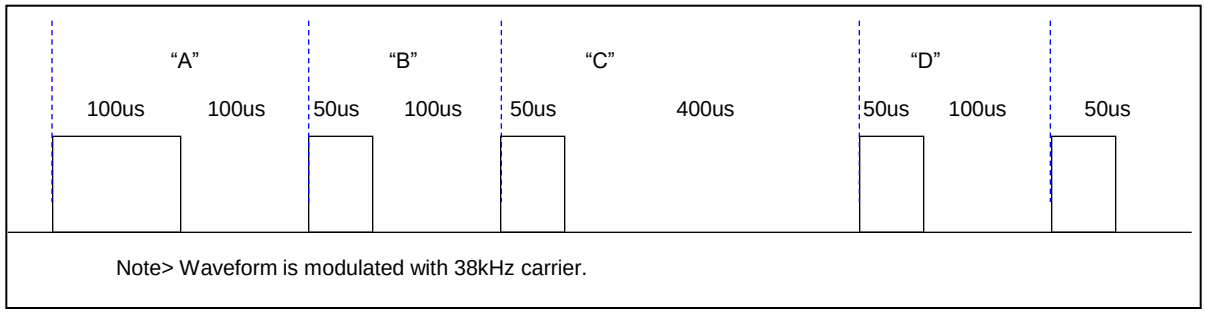
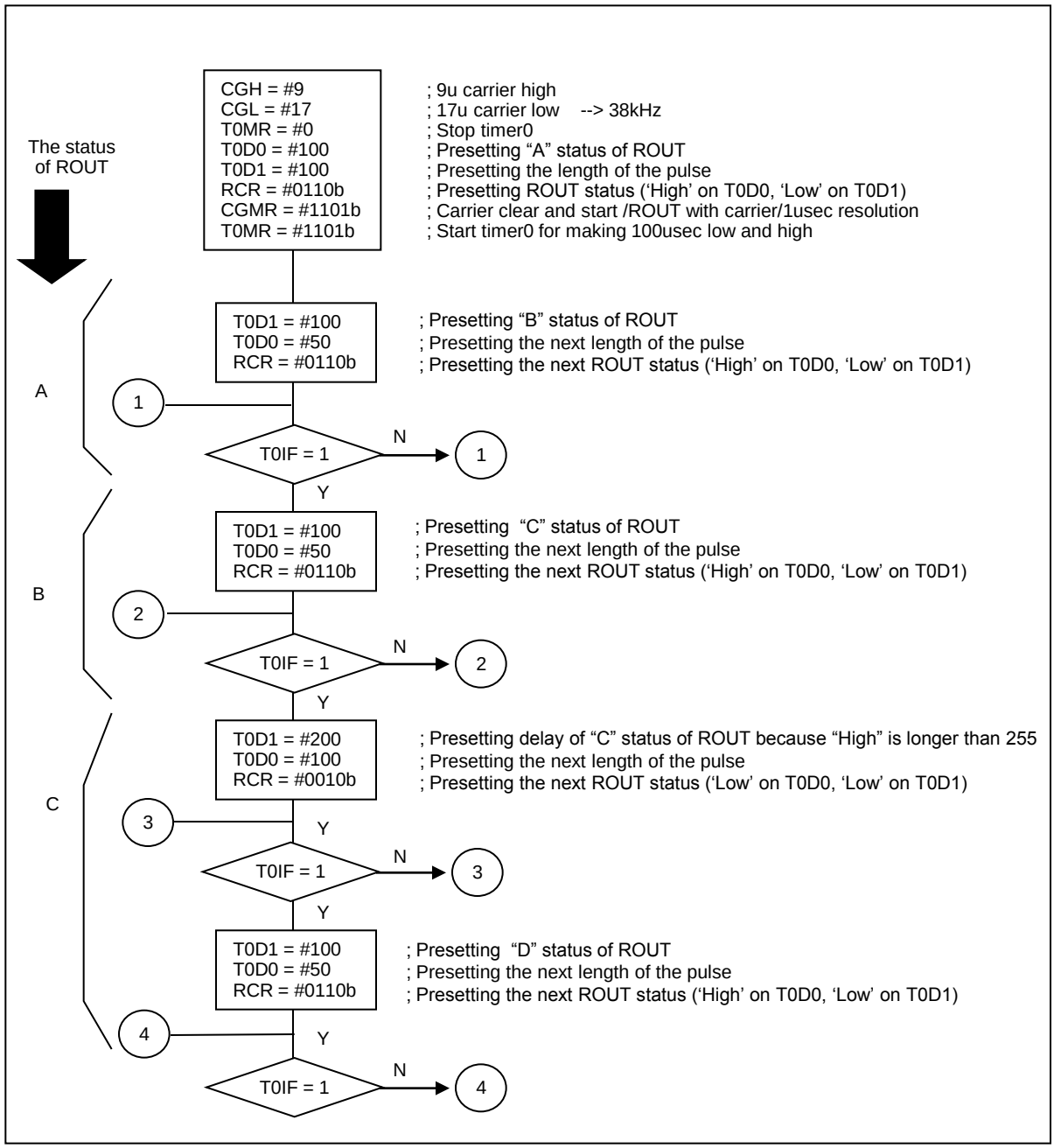


Fig. 4.7 Flow of controlling ROUT

4.3.2.5. Control of ROUT with using Timer0





4.3.3. Timer0

TIMER0 operates as a up-counter with two-8bit data register (T0D0H+T0D0L, T0D1H+T0D1L). When the value of the up-counter reaches the content of Timer Data Register, the up-counter is cleared to ``00 h``, and interrupt (T0IF) is occurred at the next clock.
Internal clock (TCK) and the output of Carrier Generator (TCARRY) is used as counter clock.

The counter execution is controlled by T0CS, T0CN, of T0MR.
T0CN is used to stop and start Timer0 without clearing the counter and T0CS does to clear and start the counter. During counting-up, value of counter can be read.
Timer execution is stopped by the reset signal.

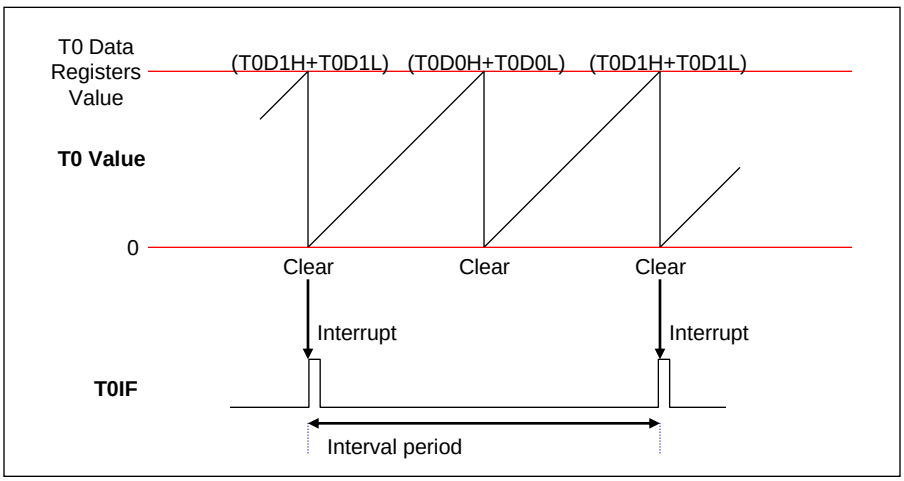


Fig. 4.8 Operation of Timer0

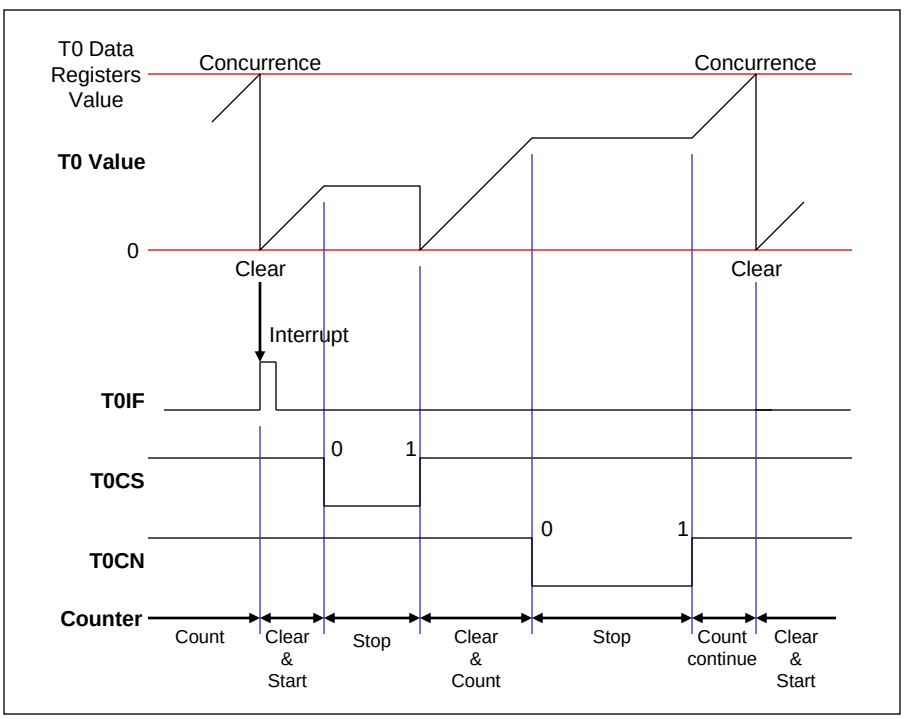


Fig. 4.9 Start / Stop operation of Timer0

4.3.4. Timer1

TIMER1 operates as a up-counter. Timer1 has 8bit data register (T1HD+T1LD).
When the value of the up-counter reaches the content of Timer Data Register, the up-counter is cleared to “00h”, and interrupt (T1IF) is occurred at the next clock.
Internal clock (TCK) and the output of Carrier Generator (TCARRY) is used as counter clock.

The counter execution is controlled by T1CS, T1CN, of T1MR.
T1CN is used to stop and start Timer1 without clearing the counter and T1CS does to clear and start the counter. During counting-up, value of counter can be read.
Timer execution is stopped by the reset signal.

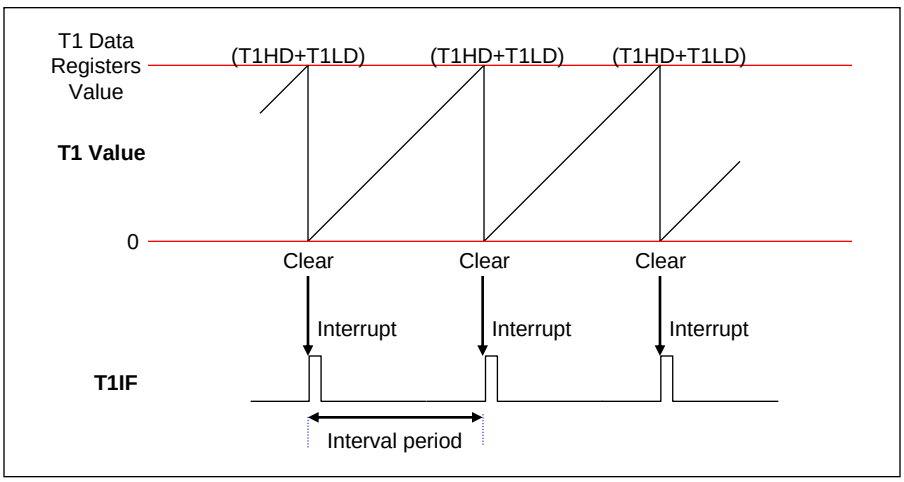


Fig. 4.10 Operation of Timer1

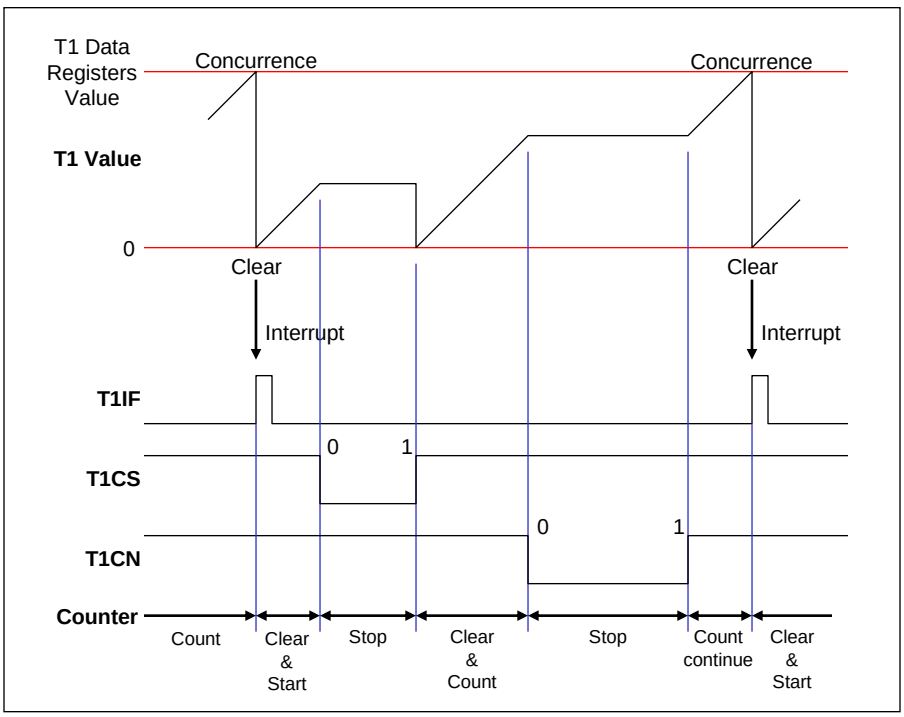


Fig. 4.11 Start / Stop operation of Timer1

4.3.5. Carrier Generator

Carrier Generator operates as a up-counter.
Carrier Generator has two 6bit-data register(CGHMD+CGHLD, CGLMD+CGLLD).
The execution of Carrier generator is controlled by CGF0,CGF1,CGON,CGCS of Carrier Generator Mode Register (CGMR).
When CGCS is set to ``1``, count value of Carrier Generator is cleared and starts counting-up.
Carrier Generator first counts CGLMD+CGLLD and next CGHMD+CGHLD.
CGLMD+CGLLD are for the pulse of the carrier (BURST) and CGLMD+CGLLD are for the low of the carrier (PAUSE).

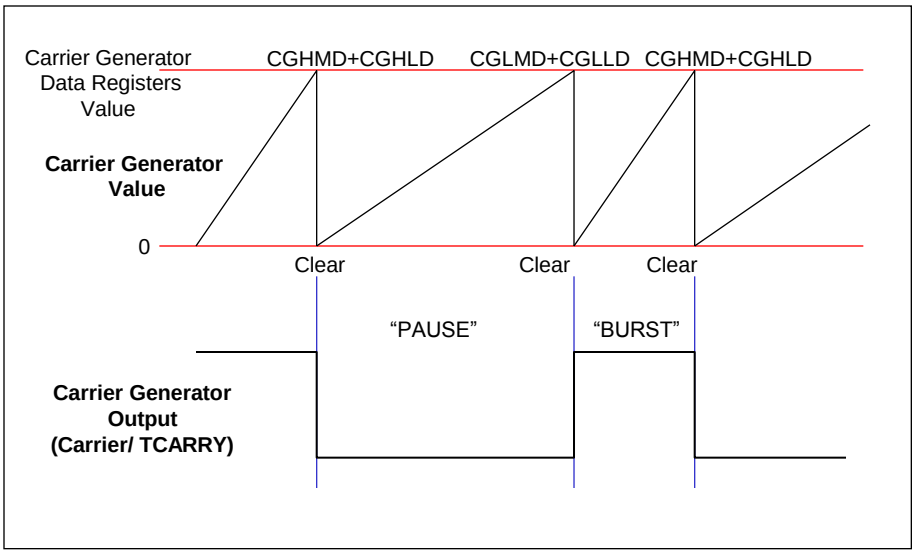


Fig. 4.12 Operation of Carrier Generator

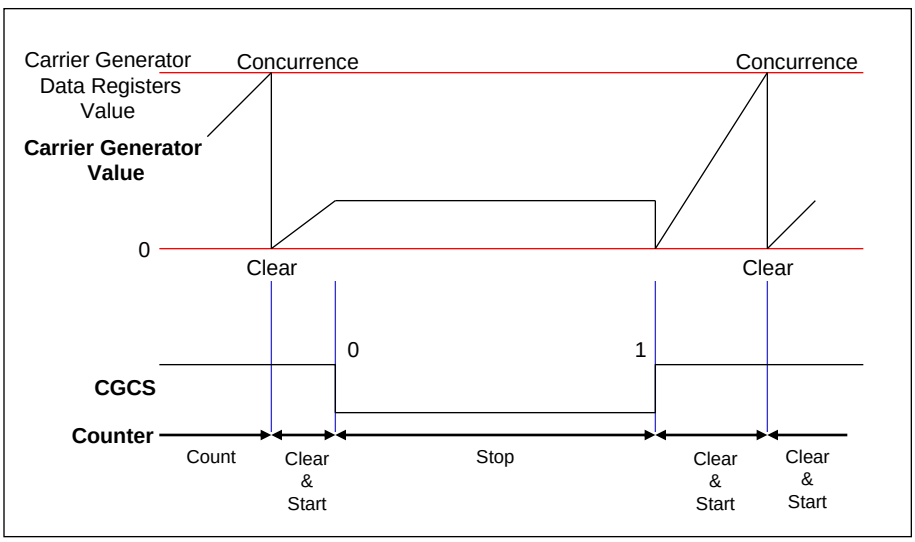


Fig. 4.13 Start/Stop of Carrier Generator

5. INTERRUPT

The ADAM40X272X contains 3 interrupt sources; 1 externals and 2 internals. Nested interrupt services with priority control is also possible.

- 3 interrupt source (1Ext, 2Timer)
- 3 interrupt vector
- 3 level nested interrupt control is possible.
- Read of interrupt request flag are possible.
- In interrupt accept, request flag is automatically cleared.

Interrupt Enable Register (IENR), Interrupt Request Register (IRQR) and priority circuit. Interrupt function block diagram is shown in Fig. 5.1.

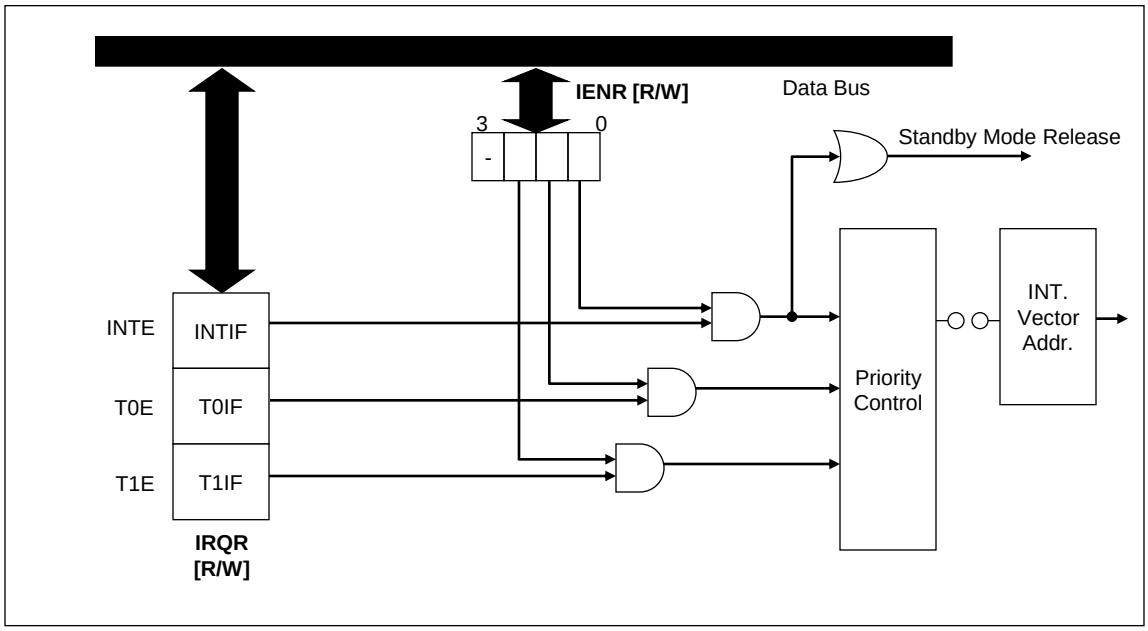


Fig. 5.1 Interrupt Source

5.1. Interrupt Source

Each interrupt vector is independent and has its own priority.

	Mask	Priority	Interrupt Source	INT Vector Addr.
Hardware Interrupt	Non-maskable	-	Reset	0000H
	maskable	1	INTE (External Interrupt)	0002H
		2	T0E (Timer0)	0004H
		3	T1E (Timer1)	0006H

Table 5.1 Interrupt Source

5.2. Interrupt Control Register

I flag of SFR is a interrupt mask enable flag. When I flag = “0”, all interrupts become disable. When I flag = “1”, interrupts can be selectively enabled and disabled by contents of corresponding Interrupt Enable Register(IENR).

When interrupt is occurred, interrupt request flag is set,and Interrupt request is detected at the edge of interrupt signal. The accepted interrupt request flag is automatically cleared during interrupt cycle process. The interrupt request flag maintains “1” until the interrupt is accepted or is cleared in program. In reset state, interrupt request flag register (IRQR) is cleared to “0”.

It is possible to read the state of interrupt register and to manipulate the contents of register.

External Interrupt Edge selection Register

bit	3	2	1	0	
IEDS [W]	-	-	IED1H	IED1L	*17h*
Initial value	0	0	0	0	
R/W	W	W	W	W	

Note> Reserved bit must be 0.

IEDH & IEDL	00	-
	01	Falling Edge Selection
	10	Rising Edge Selection
	11	Both Edge Selection

Interrupt Enable Register

bit	3	2	1	0	
IENR [R/W]	-	T1E	T0E	INTE	*18h*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

Note> Reserved bit must be 0.

-	-
T1E	Timer1 interrupt Enable bit
T0E	Timer0 interrupt Enable bit
INTE	Ext. interrupt Enable bit

Interrupt Request Flag Register

bit	3	2	1	0	
IRQR [R/W]	-	T1IF	T0IF	INTIF	*19h*
Initial value	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	

Note> Reserved bit must be 0.

-	-
T1IF	Timer1 interrupt Request Flag bit
T0IF	Timer0 interrupt Request Flag bit
INTIF	Ext. interrupt Request Flag bit

5.2.1. Interrupt Timing

Interrupt Request Sampling Time

- Maximum 2 machine cycles (When execute LDW @ABR Instruction)
- Minimum 0 machine cycle

Interrupt preprocess step is 1 machine cycle.

5.2.2. The valid timing after executing Interrupt control instructions

I flag is valid just after executing of EI/DI on the contrary.

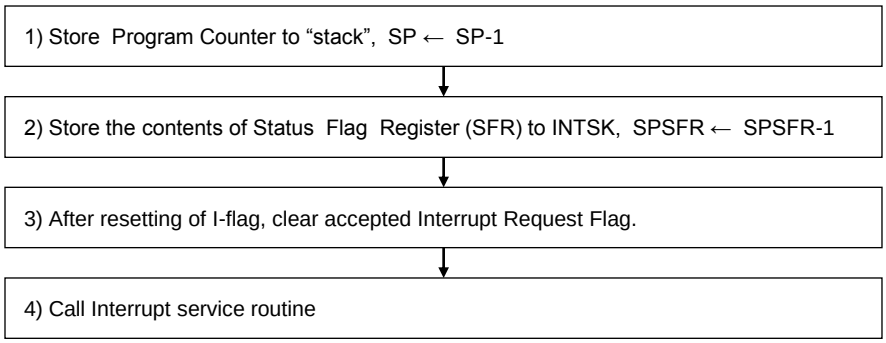
5.3. Interrupt Processing Sequence

When an interrupt is accepted, the on-going process is stopped and the interrupt service routine is executed. After the interrupt service routine is completed it is necessary to restore everything to the state before the interrupt occurred.

As soon as an interrupt is accepted, the content of the program counter is saved in the stack register and the contents of status flag register (SFR) is saved into the interrupt stack register (INTSK) which is 3 level stack area.

At the same time, the content of the vector address corresponding to the accepted interrupt, which is in the interrupt vector table, enters into the program counter and interrupt service is executed. In order to execute the interrupt service routine, it is necessary to write the jump addresses in the vector table (0002 h ~ 0006 h) corresponding to each interrupt.

Interrupt Processing Step



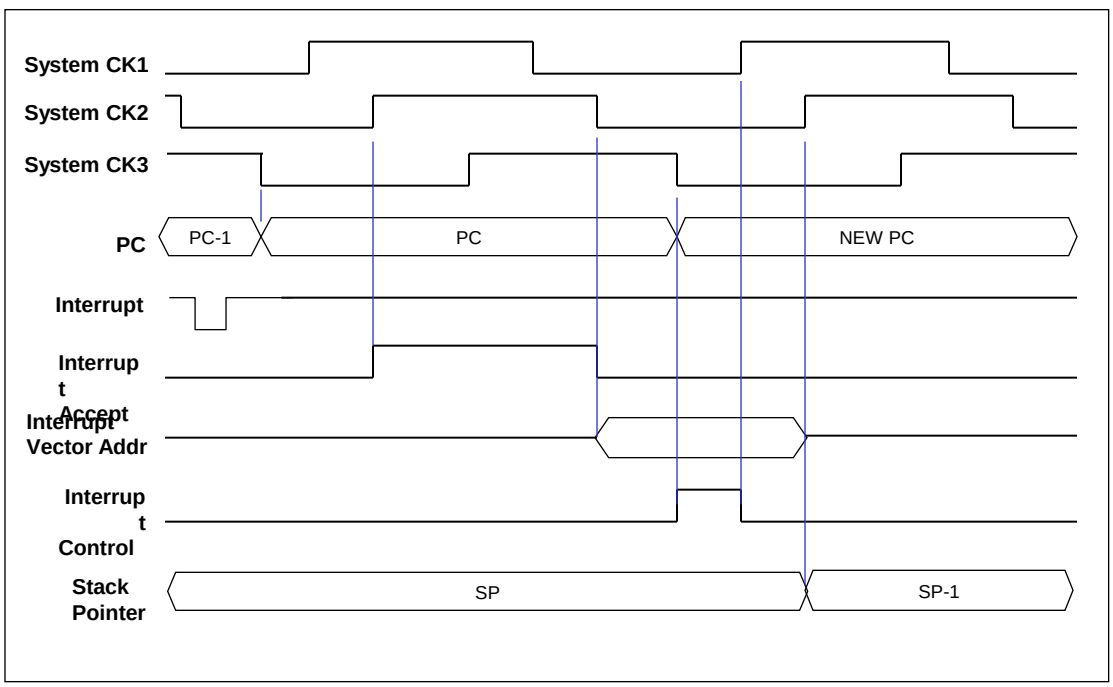


Fig. 5. 2 Interrupt Processing Step Timing

5.4. Multiple Interrupt

If there is an interrupt, Interrupt Mask Enable Flag is automatically cleared before entering the Interrupt Service Routine. After then, no interrupt is accepted. If EI instruction is executed, interrupt mask enable bit becomes ``1``, and each enable bit can accept interrupt request.

When two or more interrupts are generated simultaneously, the highest priority interrupt is accepted.

6. STOP FUNCTION

6.1. Stop Mode

STOP mode can be entered by STOP instruction during program.
In STOP mode, oscillator is stopped to make all clocks stop, which leads to less power consumption. All registers and RAM data are preserved.
"NOP" instruction should be follows STOP instruction for pre-charge time of Data Bus line.

ex) STOP : STOP instruction execution
NOP : NOP instruction

Internal circuit		STOP mode
Oscillator		Stop
Internal CPU clock		Stop
Register		Retained
RAM		Retained
Ports	R1,PB,PC,PD ports	Retained
	PA port	Low
	ROUT port	Low
Timer		Stop
Watch dog Timer		Reset and restart at stop release
Address Bus, Data Bus		Retained

Table 6.1 Operation State in Stop Mode

6.2. Stop Mode Release

Release of STOP mode is executed by Power on reset , inputting Low to Key input Port (one of R0, R1, PB) which is selected by R0ST, R1ST and PBST register for stop release, external interrupt and Low Voltage Detection (LVD) mode release .
When there is a release signal of STOP mode, the instruction execution starts after oscillation stabilization time($2^{12} \times 12/f_{OSC}$ = 12.288ms at f_{OSC} = 4.0MHz)

Release Factor	Release Method
Power on reset	By Power on reset, Stop mode is release and system is initialized
R0, R1, PB Port (key input)	Stop mode is released by low input of pin selected by R0ST, R1ST, PBST register
External interrupt	Stop mode is released by external interrupt input
Release from LVD detection	Stop mode is released by LVD detection.

Table 6.2 Stop Mode Release

7. RESET FUNCTION

7.1. Power on RESET

Power On Reset circuit automatically detects the rise of power voltage (the rising time should be within 50ms). Until the power voltage reaches a certain voltage level, internal reset signal is maintained at ″L″ Level until oscillator is stable.

After power applies and starting of oscillation, this reset state is maintained for about oscillation stabilization time of 2^{15} x 12/fosc (about 98.304ms : at 4MHz).

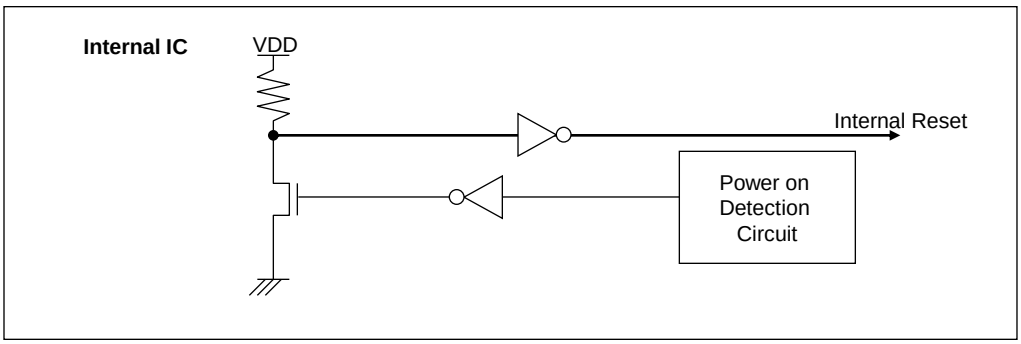


Fig. 7.1 Block Diagram of Power On Reset Circuit

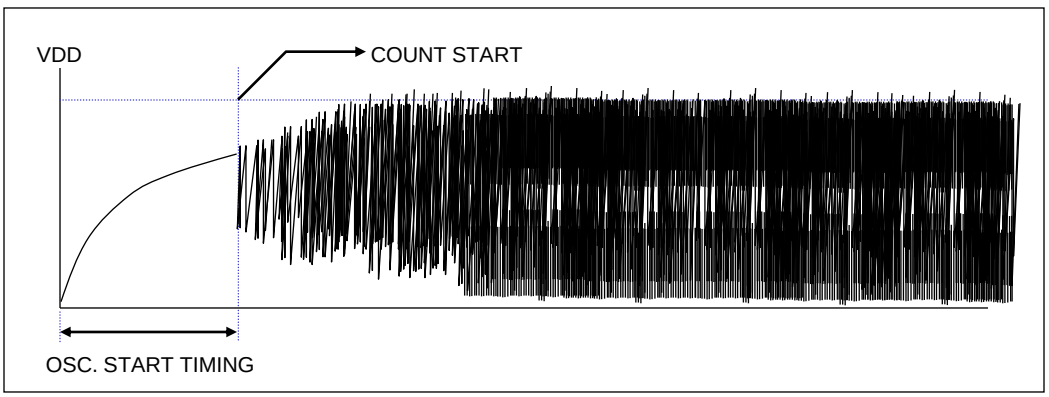


Fig. 7.2 Oscillator stabilization diagram

Note> When Power On Reset, oscillator stabilization time dose not include OSC. Start time.

8. LVD (Low Voltage Detection) Mode

8.1. Low Voltage Detection Condition

An on-board voltage comparator checks that V_{DD} is at the required level to ensure correct operation of the device. If V_{DD} is below a certain level,
Low voltage detector forces the device into low voltage detection mode.

8.2. Low Voltage Detection Mode

There is no power consumption except stop current, stop mode release function is disabled. I/O port is configured as input mode(with pull-up resistor) and Data memory is retained until Voltage through external capacitor is worn out.
In this mode, output ports(PA, PC, PD) are configured open drain “H” output (PC, PD Ports have Push-pull “H” output by Option) and I/O ports(R1 , PB) are fixed input with pull-up enabled (Pull-up disable is option in LVD mode).

8.3. Release of Low Voltage Detection Mode

Reset signal result from new battery (normally 3V) wakes the low voltage detection mode and come into normal reset state. It depends on user whether to execute RAM clear routine or not.

9. SRAM DATA BACK-UP

9.1. SRAM DATA BACK-UP after Low Voltage Detection

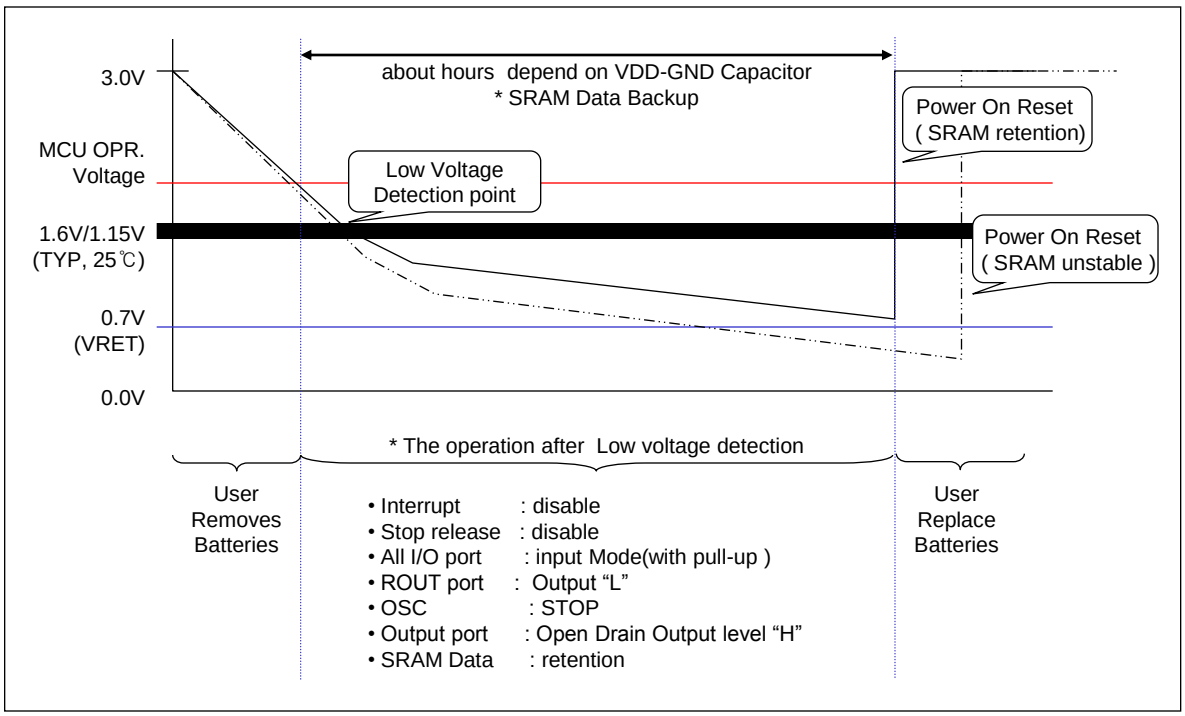


Fig. 9.1 Low Voltage Detection and Protection

9.2. S/W flow chart example after Reset using SRAM DATA Back-up

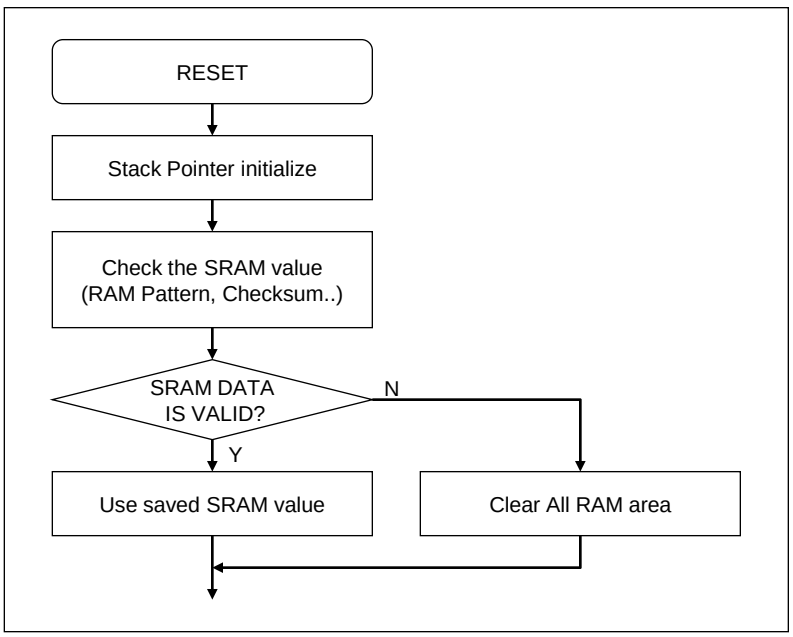
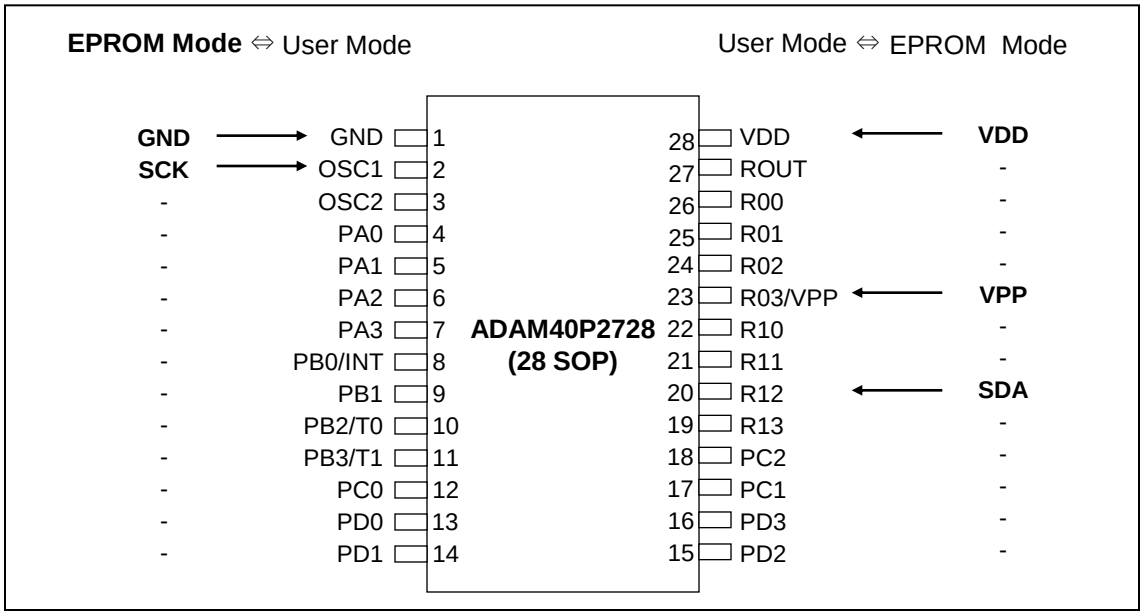


Fig. 9.2 S/W Flow Chart Example for SRAM Back-up

10. MTP PROGRAMMING

10.1. ADAM40P2728

10.1.1. PIN Assignment

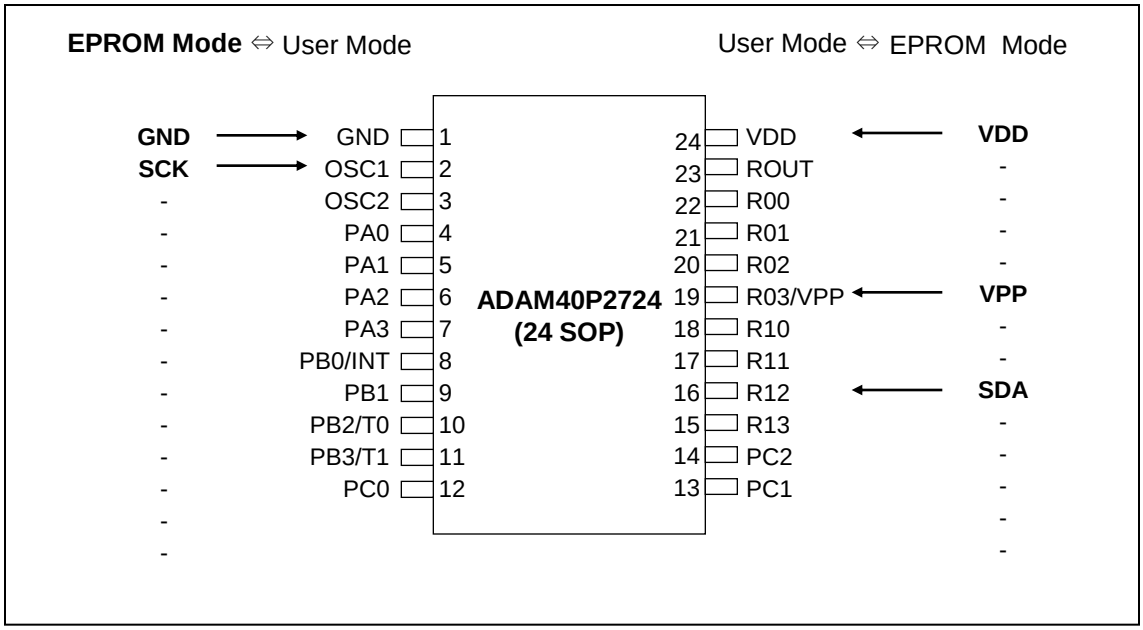


10.1.2. PIN Function

Pin No.	Port Name	User Mode	EPROM Mode
1, 28	GND, VDD	Power	**Power**
2	OSC1	Clock (Input)	**SCK** (Input)
3	OSC2	Clock (Output)	-
4	PA0	PA0 (Output)	-
5	PA1	PA1 (Output)	-
6	PA2	PA2 (Output)	-
7	PA3	PA3 (Output)	-
8	PB0	PB0 (I/O)	-
9	PB1	PB1 (I/O)	-
10	PB2	PB2 (I/O)	-
11	PB3	PB3 (I/O)	-
12	PC0	PC0 (Output)	-
13	PD0	PD0 (Output)	-
14	PD1	PD1 (Output)	-
15	PD2	PD2 (Output)	-
16	PD3	PD3 (Output)	-
17	PC1	PC1 (Output)	-
18	PC2	PC2 (Output)	-
19	R13	R13 (I/O)	-
20	R12	R12 (I/O)	**SDA** (I/O)
21	R11	R11 (I/O)	-
22	R10	R10 (I/O)	-
23	R03	R03 (Input)	**VPP** (11.5V)
24	R02	R02 (Input)	-
25	R01	R01 (Input)	-
26	R00	R00 (Input)	-
27	ROUT	ROUT (Output)	-

10. 2. ADAM40P2724

10.2.1. PIN Assignment



10.2.2. PIN Function

Pin No.	Port Name	User Mode	EPROM Mode
1, 24	GND, VDD	Power	**Power**
2	OSC1	Clock (Input)	**SCK** (Input)
3	OSC2	Clock (Output)	-
4	PA0	PA0 (Output)	-
5	PA1	PA1 (Output)	-
6	PA2	PA2 (Output)	-
7	PA3	PA3 (Output)	-
8	PB0	PB0 (I/O)	-
9	PB1	PB1 (I/O)	-
10	PB2	PB2 (I/O)	-
11	PB3	PB3 (I/O)	-
12	PC0	PC0 (Output)	-
13	PC1	PC1 (Output)	-
14	PC2	PC2 (Output)	-
15	R13	R13 (I/O)	-
16	R12	R12 (I/O)	**SDA** (I/O)
17	R11	R11 (I/O)	-
18	R10	R10 (I/O)	-
19	R03	R03 (Input)	**VPP** (11.5V)
20	R02	R02 (Input)	-
21	R01	R01 (Input)	-
22	R00	R00 (Input)	-
23	ROUT	ROUT (Output)	-

10.3. Code Option Description

10.3.1. Optional Features

- 16 Bit for Option bit are available.
- R1, PB Ports have Pull-up Resistor or not at LVD Mode
- PC, PD Ports are Push-Pull “high” output or High-Z(High- impedance output) at LVD Mode.
- Lock bit is available.

10.3.2. CODE Option Bit mapping List [Address : 8400h]

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	PD3L	PD2L	PD1L	PD0L	PC2L	PC1L	PC0L	PB3L	PB2L	PB1L	PB0L	R13L	R12L	R11L	R10L	*8400h*
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Bit Name	Option Description	Initial Value	Code Write Value	Option Result
-	-	1 (No Writing =default)	-	-
			-	-
PD3L	PD3 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PD3 Port is Push-Pull “High” at LVD Mode
			1	PD3 Port is “High-Z” at LVD Mode
PD2L	PD2 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PD2 Port is Push-Pull “High” at LVD Mode
			1	PD2 Port is “High-Z” at LVD Mode
PD1L	PD1 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PD1 Port is Push-Pull “High” at LVD Mode
			1	PD1 Port is “High-Z” at LVD Mode
PD0L	PD0 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PD0 Port is Push-Pull “High” at LVD Mode
			1	PD0 Port is “High-Z” at LVD Mode
PC2L	PC2 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PC2 Port is Push-Pull “High” at LVD Mode
			1	PC2 Port is “High-Z” at LVD Mode
PC1L	PC1 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PC1 Port is Push-Pull “High” at LVD Mode
			1	PC1 Port is “High-Z” at LVD Mode
PC0L	PC0 is Push-Pull “High” or “High-Z” at LVD MODE	1 (No Writing =default)	0	PC0 Port is Push-Pull “High” at LVD Mode
			1	PC0 Port is “High-Z” at LVD Mode
PB3L	PB3 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	PB3 has no Pull-up resistor at LVD Mode
			1	PB3 has Pull-up resistor at LVD Mode
PB2L	PB2 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	PB2 has no Pull-up resistor at LVD Mode
			1	PB2 has Pull-up resistor at LVD Mode
PB1L	PB1 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	PB1 has no Pull-up resistor at LVD Mode
			1	PB1 has Pull-up resistor at LVD Mode
PB0L	PB0 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	PB0 has no Pull-up resistor at LVD Mode
			1	PB0 has Pull-up resistor at LVD Mode
R13L	R13 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	R13 has no Pull-up resistor at LVD Mode
			1	R13 has Pull-up resistor at LVD Mode
R12L	R12 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	R12 has no Pull-up resistor at LVD Mode
			1	R12 has Pull-up resistor at LVD Mode
R11L	R11 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	R11 has no Pull-up resistor at LVD Mode
			1	R11 has Pull-up resistor at LVD Mode
R10L	R10 has Pull-up Resistor or not at LVD MODE	1 (No Writing =default)	0	R10 has no Pull-up resistor at LVD Mode
			1	R10 has Pull-up resistor at LVD Mode

10.4. AC / DC Timing for MTP Program Write / Read MODE (Ta = 25 ℃)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Programming Supply Current	IVPP	-	-	50	mA
Supply Current in EPROM Mode	IVDDP	-	-	20	mA
VPP Level during Programming	VIHP	11.0	11.5	12.0	V
VDD Level in Program Mode	VDD1H	4.75	5.0	5.25	V
VDD Level in Read Mode	VDD2H	-	3.0 / 5.0	-	V
SCK/SDA Input High Level in EPROM Mode	VIHC	VDD*0.8	-	-	V
SCK/SDA Input Low Level in EPROM Mode	VILC	-	-	VDD*0.2	V
Mode Latch Time	TMODI	50	-	-	μs
Mode Setup Time	TMODS	10	-	-	μs
Mode Hold Time	TMODH	10	-	-	μs
Mode Clock Pulse Period	TMCK	2	-	-	μs
VPP Setup Time	TVPPS	1	-	-	ms
VPP Rise Time	TVPPR	1	-	-	ms
SCK Pulse Width High	TPWH1	2	-	-	μs
SCK Pulse Width Low	TPWL1	2	-	-	μs
SDA Input Setup to SCK High	TSET1	1	-	-	μs
SDA Input Hold after SCK Low	THLD1	1	-	-	μs
SCK High to SDA Output Valid	THLD2	-	-	TPWH	μs
SCK Pulse Width High in Program Mode	TPWH2	-	**50±10%**	-	μs

11. INSTRUCTION SET

11.1. Legend

A : accumulator(4bit)

r : peripheral address register(6bit)

[r] : data addressed by peripheral address register (4bit)

Y : Y register(4bit)

X : X register(4bit)

ABR : address buffer register(15bit)

ABRn : address buffer register #0~2(4bit), #3(3bit)

[@ABR] : data addressed by ABR(16bit)

DBR : data buffer register(16bit)

DBRn : data buffer register #0~3(4bit)

T0CR : Timer 0 count register(8bit)

T1CR : Timer 1 count register(8bit)

#n4 : 0~Fh

#n2 : 0~3

#n1 : 0~1

dp : data address point(8bit)

dp+X+Y : data address point indexed by X-register and Y-register (8bit)

m(dp) : data addressed by dp

m(dp+X+Y) : data addressed by dp+X+Y

PG : Page address(1bit)

ADS : address stack register

!abs : address

11.2. INSTRUCTION SET TABLE

	Instruction Group	MNEMONIC	USAGE	OPEATION	S	CY

1	ARITHMATIC	ADDC	ADDC m(dp),#n4	A = m(dp) + #n4 + CY
				A = m(dp+X+Y) + #n4 + CY at D flag of SFR is set.
				“S” set if overflow
					C	O

2			ADDC A,#n4	A = A + #n4 + CY, “S” set if overflow	C	O
3			ADDC m(dp),A	A = m(dp) + A + CY A = m(dp+X+Y) + A + CY at D flag of SFR is set. “S” set if overflow	C	O
4			ADDC ABRn,#n4	ABRn = ABRn + #n4 + CY, “S” set if overflow	C	O
5			ADDC ABRn,A	ABRn = ABRn + A + CY, “S” set if overflow	C	O
6			ADDC ABRn,Y	ABRn = ABRn + Y + CY, “S” set if overflow	C	O
7			ADDC DBRn,#n4	DBRn = DBRn + #n4 + CY, “S” set if overflow	C	O
8			ADDC DBRn,A	DBRn = DBRn + A + CY, “S” set if overflow	C	O
9			ADDC DBRn,Y	DBRn = DBRn + Y + CY, “S” set if overflow	C	O
10			ADDC Y,#n4	Y = Y + #n4 + CY, “S” set if overflow	C	O
11			ADDC X,#n4	X = X + #n4 + CY, “S” set if overflow	C	O

12		SUBC	SUBC m(dp),#n4	A = m(dp) - #n4 - CY
				A = m(dp+X+Y) - #n4 - CY at D flag of SFR is set.
				“S” clear if underflow
					B	W

13			SUBC A,#n4	A = A - #n4 - CY, “S” clear if underflow	B	W
14			SUBC m(dp),A	A = m(dp) - A - CY A = m(dp+X+Y) - A - CY at D flag of SFR is set. “S” clear if underflow	B	W
15			SUBC ABRn,#n4	ABRn = ABRn - #n4 - CY, “S” clear if underflow	B	W
16			SUBC ABRn,A	ABRn = ABRn - A - CY, “S” clear if underflow	B	W
17			SUBC ABRn,Y	ABRn = ABRn - Y - CY, “S” clear if underflow	B	W
18			SUBC DBRn,#n4	DBRn = DBRn - #n4 - CY, “S” clear if underflow	B	W
19			SUBC DBRn,A	DBRn = DBRn - A - CY, “S” clear if underflow	B	W
20			SUBC DBRn,Y	DBRn = DBRn - Y - CY, “S” clear if underflow	B	W
21			SUBC Y,#n4	Y = Y - #n4 - CY, “S” clear if underflow	B	W
22			SUBC X,#n4	X = X - #n4 - CY, “S” clear if underflow	B	W
23		ARRC	ARRC	A = A rotate right with CY	S	R
24	COMPARE	CALE	CALE #n4	“S” set if A ≤ #n4	E	
25			CALE m(dp)	“S” set if A ≤ m(dp) “S” set if A ≤ m(dp+X+Y) at D flag of SFR is set.	E	
26		CANE	CANE #n4	“S” set if A != #n4	N	
27			CANE m(dp)	“S” set if A != m(dp) “S” set if A != m(dp+X+Y) at D flag of SFR is set.	N	

	Instruction Group	MNEMONIC	USAGE	OPEATION	S	CY
28	COMPARE	CMLE	CMLE m(dp),#n4	“S” set if m(dp) ≤ #n4 “S” set if m(dp+X+Y) ≤#n4 at D flag of SFR is set.	E	
29		CMNE	CMNE m(dp),#n4	“S” set if m(dp) != #n4 “S” set if m(dp+X+Y) != #n4 at D flag of SFR is set.	N	
30		CYNE	CYNE #n4	“S” set if Y != #n4	N	
31			CYNE A	“S” set if Y != A	N	
32		CXNE	CXNE #n4	“S” set if X != #n4	N	
33	BIT MANIPULATION	SET1	SET1 m(dp),#n2	Set bit m(dp).#n2 Set bit m(dp+X+Y).#n2 at D flag of SFR is set.	S	
34		CLR1	CLR1 m(dp),#n2	Clear bit m(dp).#n2 Clear bit m(dp+X+Y).#n2 at D flag of SFR is set.	S	
35		TM	TM m(dp),#n2	“S” set if m(dp).#n2 = 1 “S” set if m(dp+X+Y).#n2 = 1 at D flag of SFR is set.	E	
36		SETR1	SETR1 r,#n2	Set bit [r].#n2	S	
37		CLRR1	CLRR1 r,#n2	Set bit [r].#n2	S	
38		TSTR	TSTR r,#n2	“S” set if [r].#n2 Bit = 1	E	
39	CARRY MANIPULATION	CLRC	CLRC	Carry Bit of SFR is clear	S	
40		SETC	SETC	Carry Bit of SFR is set	S	
41		TSTC	TSTC	“S” set if Carry Test = 1	E	
42	DATA TRANSFER	LDM	LDM m(dp),#n4	m(dp) = #n4 m(dp+X+Y) = #n4 at D flag of SFR is set.	S	
43			LDM m(dp),A	m(dp) ← A m(dp+X+Y) ← A at D flag of SFR is set.	S	
44		LDA	LDA #n4	A = #n4	S	
45			LDA m(dp)	A ← m(dp) A ← m(dp+X+Y) at D flag of SFR is set.	S	
46			LDA X	A ← X	S	
47			LDA Y	A ← Y	S	
48		LDY	LDY #n4	Y = #n4	S	
49			LDY A	Y ← A	S	
50		LDX	LDX #n4	X = #n4	S	
51			LDX A	X ← A	S	
52		XMA	XMA m(dp)	A ⇔ m(dp) A ⇔ m(dp+X+Y) at D flag of SFR is set.	S	
53		LDW	LDW @ABR	DBR ← [@ABR] **[Note]	S	
54			LDW DBR,ABR	DBR ← ABR	S	
55			LDW ABR,DBR	ABR ← DBR	S	
56			LDW DBR,T0CR	DBR0,DRB1 ← T0CR	S	
57			LDW DBR,T1CR	DBR0,DBR1 ← T1CR	S	

	Instruction Group	MNEMONIC	USAGE	OPEATION	S	CY
58	DATA TRANSFER	LPG	LPG #n1	PG = #n1	S	
59		LRA	LRA r	[r] ← A	S	
60		LAR	LAR r	A ← [r]	S	
61		LRI	LRI r,#n4	[r] = #n4	S	
62	INCREMENT	INC	INC ABR	ABR++	C	
63	BRANCH	BR	BR !abs	If S bit of SFR = 1, Absolute branch, PC ← !abs	S	
64			BR @ABR	If S bit of SFR = 1, Indirect branch, PG+PC ← ABR	S	
65	SUBROUTINE	CALL	CALL !abs	If S bit of SFR = 1, ADS ← PG+PC, SP ← SP-1, PC ← !abs	S	
66			CALL @ABR	If S bit of SFR = 1, ADS ← PG+PC, SP ← SP-1, PG+PC ← ABR	S	
67		RET	RET	SP ← SP+1, PG+PC ← ADS	S	
68		RETI	RETI	SPSFR ← SPSFR+1, SFR ← M(SPSFR) SP ← SP+1, PG+PC ← ADS	S	
69	ETC	NOP	NOP		S	
70		STOP	STOP		S	
71		WDTC	WDTC	Watch Dog Timer Clear	S	
72		XOR	XOR m(dp)	A = A⊕m(dp) A = A⊕m(dp+X+Y) at D flag of SFR is set.	S	
73		EIX	EIX	Index bit of SFR is set.	S	
74		DIX	DIX	Index bit of SFR is clear.	S	
75		EI	EI	Interrupt bit of SFR is set.	S	
76		DI	DI	Interrupt bit of SFR is clear.	S	
77		CMPL	CMPL	A = $\overline{A}$ + 1	Z	

**[Note] The Instruction “LDW @ABR” execution time is 2cycle and execution process is as follow.
SP = SP-1 , ADS ← PG+PC ,
PG+PC ← ABR , DBR ← (PG+PC) , PG+PC ← ADS , SP = SP+1
.

** CARRY BIT(CY) hold previous value before execution CLRC/SETC instruction .
Symbols have meaning as follows.
- O : Carry bit is only set when overflow has occurred in operation.
- W : Carry bit is only set when borrow has occurred in operation.
- R : Carry bit is only set or reset according to shift bit.

** STATUS BIT(S) indicates conditions for changing status. Symbols have meaning as follows.
- S : On executing an instruction, status bit is unconditionally set.
- C : Status bit is only set when overflow has occurred in operation.
- B : Status bit is only set when underflow has not occurred in operation.
- E : Status bit is only set when equality is found in comparison.
- N : Status bit is only set when equality is not found in comparison.
- Z : Status bit only set when the result is zero.

11.3. INSTRUCTION SET

▶ *ADDC (Add with Carry)*

Add the contents of General Function Register to the contents of assigned memory address or the direct data value and Add carry bit value to the result.
Store the result in general function register.

(Function) GFR = GFR + M + C

MNEMONIC	OPERAND	S	I	D	C
ADDC	mp(dp),#n4	C	.	.	O
	A,#n4	C	.	.	O
	m(dp),A	C	.	.	O
	ABRn,#n4	C	.	.	O
	ABRn,A	C	.	.	O
	ABRn,Y	C	.	.	O
	DBRn,#n4	C	.	.	O
	DBRn,A	C	.	.	O
	DBRn,Y	C	.	.	O
	T,#n4	C	.	.	O
	X,#n4	C	.	.	O

▶ *SUBC (Subtract with Carry)*

Subtract the contents of General Function Register to the contents of the assigned memory address or the direct data value and subtract carry bit value to the result.
Store the result in general function register.

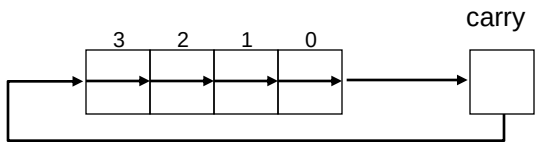
(Function) GFR ← GFR - M - C

MNEMONIC	OPERAND	S	I	D	C
SUBC	mp(dp),#n4	B	.	.	W
	A,#n4	B	.	.	W
	m(dp),A	B	.	.	W
	ABRn,#n4	B	.	.	W
	ABRn,A	B	.	.	W
	ABRn,Y	B	.	.	W
	DBRn,#n4	B	.	.	W
	DBRn,A	B	.	.	W
	DBRn,Y	B	.	.	W
	Y,#n4	B	.	.	W
	X,#n4	B	.	.	W

▶ *ARRC(Accumulator rotate right with carry)*

The contents of A location to the right by 1 bit. Move bit0 to Carry flag, and carry into bit3.

(Function)



MNEMONIC	OPERAND	S	I	D	C
ARRC	-	S	.	.	R

▶ *CALE (Compare to Accumulator)*

Check the contents of A is less or equal compare to the contents of assigned memory address or the direct data value.

(Function) compare A ≤ #n4 , A ≤ m(dp) , A ≤ m(dp+X+Y)

MNEMONIC	OPERAND	S	I	D	C
CALE	#n4	E	.	.	.
	m(dp)	E	.	.	.

▶ *CANE (Compare to Accumulator)*

Check the contents of A is not equal with the contents of assigned memory address or the direct data value.

(Function) compare A != #n4 , A != m(dp) , A != m(dp+X+Y)

MNEMONIC	OPERAND	S	I	D	C
CANE	#n4	N	.	.	.
	m(dp)	N	.	.	.

▶ *CMLE (Compare to memory)*

Check the contents of assigned memory address is less or equal with the direct data value.

(Function) compare m(dp) ≤ #n4 , m(dp+X+Y) ≤ #n4

MNEMONIC	OPERAND	S	I	D	C
CMLE	#n4	E	.	.	.

▶ *CMNE (Compare to memory)*

Check the contents of assigned memory address is not equal with the direct data value.

(Function) compare m(dp) != #n4 , m(dp+X+Y) != #n4

MNEMONIC	OPERAND	S	I	D	C
CMNE	#n4	N	.	.	.

▶ *CYNE (Compare to Y-register)*

Check the contents of Y-register is not equal with the contents of A or the direct data value.

(Function) compare Y != #n4 , Y != A

MNEMONIC	OPERAND	S	I	D	C
CYNE	#n4	N	.	.	.
	A	N	.	.	.

▶ *CXNE(Compare to X-register)*

Check the contents of X-register is not equal with the direct data value.

(Function) compare X != #n4

MNEMONIC	OPERAND	S	I	D	C
CXNE	#n4	N	.	.	.

▶ *SET1 (Set bit)*

Set the bit of assigned memory address

(Function) m(dp).bit ← 1 , m(dp+X+Y).bit ← 1

MNEMONIC	OPERAND	S	I	D	C
SET1	m(dp),#n2	S	.	.	.

▶ *CLR1 (Clear bit)*

Clear the bit of assigned memory address

(Function) m(dp).bit ← 0 , m(dp+X+Y).bit ← 0

MNEMONIC	OPERAND	S	I	D	C
CLR1	m(dp),#n2	S	.	.	.

▶ *TM (Test bit)*

Check the bit of assigned memory address is 1.

(Function) m(dp).bit = 1 ? , m(dp+X+Y).bit = 1 ?

MNEMONIC	OPERAND	S	I	D	C
TM	m(dp),#n2	E	.	.	.

▶ *SETR1 (Set register data bit)*

Set the bit of assigned register data.

(Function) [r].bit ← 1

MNEMONIC	OPERAND	S	I	D	C
SETR1	[r],#n2	S	.	.	.

▶ *CLRR1 (Clear register data bit)*

Clear the bit of assigned register data.

(Function) [r].bit ← 0

MNEMONIC	OPERAND	S	I	D	C
CLRR1	[r],#n2	S	.	.	.

▶ *TSTR (Test register data bit)*

Check the bit of assigned register data is 1.

(Function) [r].bit = 1 ?

MNEMONIC	OPERAND	S	I	D	C
TSTR	[r],#n2	E	.	.	.

▶ *CLRC (Clear Carry Flag)*

Clear the carry bit

(Function) CY ← 0

MNEMONIC	OPERAND	S	I	D	C
CLRC	-	S	.	.	0

▶ *SETC (Set Carry flag)*

Set the carry bit

(Function) CY ← 1

MNEMONIC	OPERAND	S	I	D	C
SETC	-	S	.	.	1

▶ *TSTC (Test Carry Flag)*

Check the content of carry bit is 1.

(Function) CY = 1?

MNEMONIC	OPERAND	S	I	D	C
TSTC	-	E	.	.	.

▶ *LDM (Load memory)*

Load the contents of A or the direct data value on assigned memory.

(Function) m(dp) ← #n4 , m(dp+X+Y) ← #n4, m(dp) ← A , m(dp+X+Y) ← A

MNEMONIC	OPERAND	S	I	D	C
LDM	#n4	S	.	.	.
	A	S	.	.	.

▶ *LDA (Load accumulator)*

Load the contents of memory address or the direct data value , X-register's ,Y-register's on A .

(Function) A ← #n4 , m(dp) , m(dp+X+Y) , X-register , Y-register

MNEMONIC	OPERAND	S	I	D	C
LDA	#n4	S	.	.	.
	m(dp)	S	.	.	.
	X	S	.	.	.
	Y	S	.	.	.

▶ *LDY (Load register Y)*

Load the contents of the A or the direct data value on Y register.

(Function) Y ← #n4 , A

MNEMONIC	OPERAND	S	I	D	C
LDY	#n4	S	.	.	.
	A	S	.	.	.

▶ *LDX (Load register X)*

Load the contents of the A or the direct data value on X- register.

(Function) X ← #n4 , A

MNEMONIC	OPERAND	S	I	D	C
LDX	#n4	S	.	.	.
	A	S	.	.	.

▶ *XMA (Exchange Accumulator and Memory)*

Exchange the contents of A and the contents of assigned memory.

(Function) A ⇔ m(dp), A ⇔ m(dp+X+Y)

MNEMONIC	OPERAND	S	I	D	C
XMA	m(dp)	S	.	.	.

▶ *LDW (Load word)*

Data transfer from function register to function register or from timer count register to function register.

(Function) DBR = @ ABR , DBR ← ABR , ABR ← DBR ,
DBR0+DBR1 ← T0CR , DBR0+DBR1 ← T1CR

MNEMONIC	OPERAND	S	I	D	C
LDW	@ABR	S	.	.	.
	DBR,ABR	S	.	.	.
	ABR,DBR	S	.	.	.
	DBR,T0CR	S	.	.	.
	DBR,T1CR	S	.	.	.

**[Note] The execution time of the Instruction "LDW @ABR" is 2cycle and execution process is as follow.
SP = SP-1 , ADS← PG+PC ,
PG+PC ← ABR , DBR ← (PG+PC) , PG+PC ← ADS , SP = SP+1

▶ *LPG (Load ROM Page register)*

Load 0 or 1 on 1-bit page register.

(Function) PG ← 0 or 1

MNEMONIC	OPERAND	S	I	D	C
LPG	#n1	S	.	.	.

▶ *LRA (Load register from A)*

Load the contents of A on the register assigned by peripheral address register(PAR).

(Function) [r] ← A

MNEMONIC	OPERAND	S	I	D	C
LRA	r	S	.	.	.

▶ *LAR (Load A from register)*

Load the contents of the assigned by peripheral address register(PAR) on A .

(Function) A ← [r]

MNEMONIC	OPERAND	S	I	D	C
LAR	r	S	.	.	.

▶ *LRI (Load register immediate)*

Load the direct data value on the register assigned by peripheral address register(PAR).

(Function) [r] ← #n4

MNEMONIC	OPERAND	S	I	D	C
LRI	r,#n4	S	.	.	.

▶ *INC (Increment ABR)*

Incrementing the address buffer register(ABR) by 1 .

(Function) ABR++

MNEMONIC	OPERAND	S	I	D	C
INC	ABR	C	.	.	.

▶ *BR (Branch to address)*

Branch to the assigned memory address.

(Function) If S bit of SFR =1 , PC ← !abs
PG+PC ← ABR

MNEMONIC	OPERAND	S	I	D	C
BR	!abs	S	.	.	.
	@ABR	S	.	.	.

▶ *CALL (Subroutine Call)*

Call the assigned memory address.

(Function) If S bit of SFR =1 , ADS ← PG+PC, SP ← SP – 1 , PC ← !abs
ADS ← PG+PC, SP ← SP – 1 , PG+PC ← ABR

MNEMONIC	OPERAND	S	I	D	C
CALL	!abs	S	.	.	.
	@ABR	S	.	.	.

▶ *RET (Return from Subroutine)*

By CALL instruction, return to the address pushed onto stack.

(Function) SP ← SP +1, PG+PC ← ADS

MNEMONIC	OPERAND	S	I	D	C
RET	-	S	.	.	.

▶ *RETI (Return from Interrupt)*

Right after finishing the interrupt processing routine, execute from the next to instruction, which is executed before interrupt occur and use always this instruction in the last stage of interrupt processing routine.

(Function) SPSFR ← SPSFR +1, SFR ← M(SPSFR)
SP ← SP +1, PG+PC ← ADS

MNEMONIC	OPERAND	S	I	D	C
RETI	-	S	.	.	.

▶ *NOP (No Operation)*

No operation

MNEMONIC	OPERAND	S	I	D	C
NOP	-	S	.	.	.

▶ *STOP (Stop mode)*

(Function) Oscillation Stop

MNEMONIC	OPERAND	S	I	D	C
STOP	-	S	.	.	.

▶ *WDTC (Watch Dog Timer Clear)*

Watch Dog Timer clear

MNEMONIC	OPERAND	S	I	D	C
WDTC	-	S	.	.	.

▶ *XOR (Logical Exclusive OR)*

EOR operation of the contents of A with the contents of assigned memory address. Store the result in A.

(Function) A ← A EOR M

MNEMONIC	OPERAND	S	I	D	C
XOR	m(dp)	S	.	.	.

▶ *EIX (Enable index flag bit)*

Set the index flag bit of SFR.

(Function) IX ← 1

MNEMONIC	OPERAND	S	I	D	C
EIX	-	S	.	1	.

▶ *DIX (Disable index flag bit)*

Clear the index flag bit of SFR.

(Function) IX ← 0

MNEMONIC	OPERAND	S	I	D	C
DIX	-	S	.	0	.

▶ *EI (Enable interrupt flag bit)*

Set the interrupt flag bit of SFR.

(Function) I ← 1

MNEMONIC	OPERAND	S	I	D	C
EI	-	S	1	.	.

▶ *DI (Disable interrupt flag bit)*

Clear the interrupt flag bit of SFR.

(Function) I ← 0

MNEMONIC	OPERAND	S	I	D	C
DI	-	S	0	.	.

▶ *CMPL (Complement Accumulator)*

Add the 1's complement contents of A to 1. Store the result in A.

(Function) $A \leftarrow \overline{A} + 1$

MNEMONIC	OPERAND	S	I	D	C
CMPL	-	Z	.	.	.

12. ORDERING INFORMATION

ROM Size	Oscillator Type	PKG Type	Ordering Device
48K Bytes (MTP)	Ceramic/Crystal	24 SOP (300mil)	ADAM40P2724
		28 SOP (300mil)	ADAM40P2728
48K Bytes (MASK)	Ceramic/Crystal	24 SOP (300mil)	ADAM40S2724
		28 SOP (300mil)	ADAM40S2728

13. DEVELOPMENT SYSTEM

The ADAM40X272X are supported by In-Circuit Emulators , Assembler and 4-gang MTP Programmer.

In-Circuit Emulator	ADAM4 ICE
Assembler	ADAM Assembler
MTP Writer (Gang4)	ADAM MTP PROGRAMER

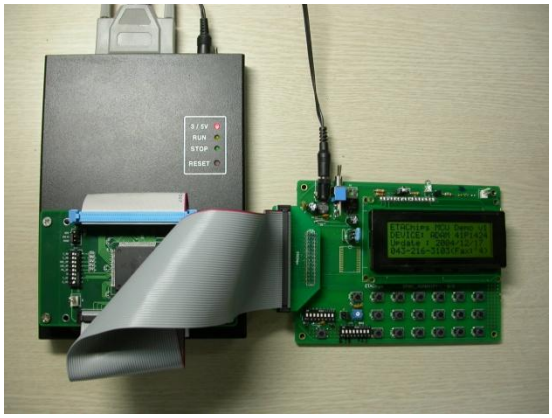


Fig 13.1 ADAM4 ICE Emulator

Fig 13.2 ADAM MTP PROGRAMER – Gang4