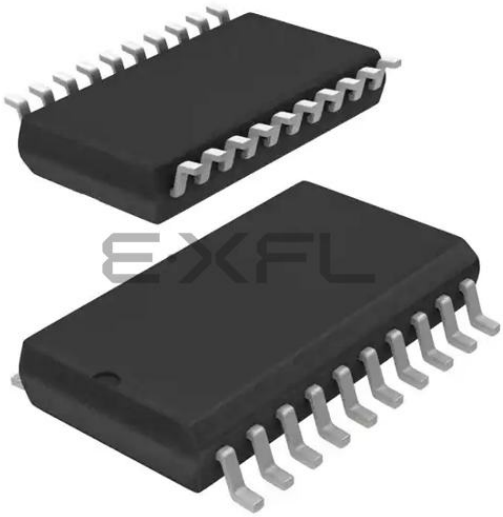


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

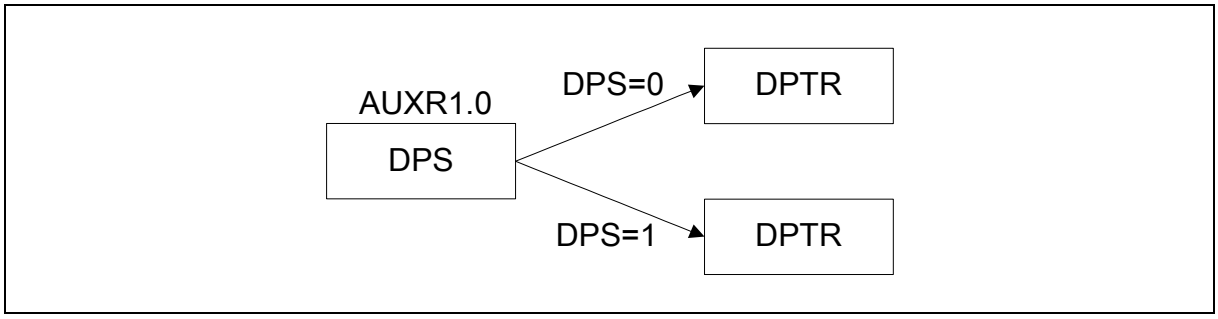
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	18
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w79e825asg

6.6 Data Pointers

The data pointers of W79E82X series are same as 8052 that has dual 16-bit Data Pointers (DPTR) by setting DPS of AUXR1.0. The figure of dual DPTR is as below diagram.



6.7 Architecture

The W79E82X series are based on the standard 8052 device. It is built around an 8-bit ALU that uses internal registers for temporary storage and control of the peripheral devices. It can execute the standard 8052 instruction set.

6.7.1 ALU

The ALU is the heart of the W79E82X series. It is responsible for the arithmetic and logical functions. It is also used in decision making, in case of jump instructions, and is also used in calculating jump addresses. The user cannot directly use the ALU, but the Instruction Decoder reads the op-code, decodes it, and sequences the data through the ALU and its associated registers to generate the required result. The ALU mainly uses the ACC which is a special function register (SFR) on the chip. Another SFR, namely B register is also used in Multiply and Divide instructions. The ALU generates several status signals which are stored in the Program Status Word register (PSW).

6.7.2 Accumulator

The Accumulator (ACC) is the primary register used in arithmetic, logical and data transfer operations in the W79E82X series. Since the Accumulator is directly accessible by the CPU, most of the high speed instructions make use of the ACC as one argument.

6.7.3 B Register

This is an 8-bit register that is used as the second argument in the MUL and DIV instructions. For all other instructions it can be used simply as a general purpose register.

6.7.4 Program Status Word:

This is an 8-bit SFR that is used to store the status bits of the ALU. It holds the Carry flag, the Auxiliary Carry flag, General purpose flags, the Register Bank Select, the Overflow flag, and the Parity flag.

6.7.5 Scratch-pad RAM

The W79E82X series have a **256/128** bytes on-chip scratch-pad RAM. These can be used by the user for temporary storage during program execution. A certain section of this RAM is bit addressable, and can be directly addressed for this purpose.

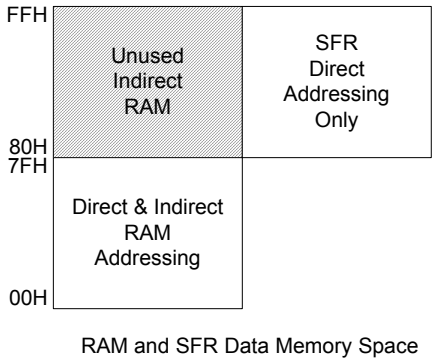
6.7.6 Stack Pointer

The W79E82X series have an 8-bit Stack Pointer which points to the top of the Stack. This stack resides in the Scratch Pad RAM in the W79E82X series. Hence the size of the stack is limited by the size of this RAM.

6.8 Power Management

Power Management like the standard 8052, the W79E82X series also have the IDLE and POWER DOWN modes of operation. In the IDLE mode, the clock to the CPU is stopped while the timers, serial ports and interrupt lock continue to operate. In the POWER DOWN mode, all clocks are stopped and the chip operation is completely stopped. This is the lowest power consumption state.

The W79E823, W79E822 and W79E821 RAM and SFR memory map as below figure:



W79E823/W79E822/W79E821 RAM and SFR Memory Map

Since the scratch-pad RAM is only **256/128** bytes it can be used only when data contents are small. There are several other special purpose areas within the scratch-pad RAM. These are described as follows.

8. SPECIAL FUNCTION REGISTERS

The W79E82X series uses Special Function Registers (SFRs) to control and monitor peripherals and their Modes. The SFRs reside in the register locations 80-FFh and are accessed by direct addressing only. Some of the SFRs are bit addressable. This is very useful in cases where we wish to modify a particular bit without changing the others. The SFRs that are bit addressable are those whose addresses end in 0 or 8. The W79E82X series contain all the SFRs present in the standard 8052. However some additional SFRs are added. In some cases the unused bits in the original 8052, have been given new functions. The list of the SFRs is as follows.

Table 1 Special Function Register Location Table

F8	IP1							
F0	B						P0ID	IP1H
E8	IE1							
E0	ACC	ADCCON	ADCH					
D8	WDCON	PWMPL	PWM0L	PWM1L	PWMCON1	PWM2L	PWM3L	PWMCON2
D0	PSW	PWMPH	PWM0H	PWM1H		PWM2H	PWM3H	PWMCON3
C8							NVMCON	NVMDAT
C0	I2CON	I2ADDR					NVMADDR	TA
B8	IP0	SADEN			I2DAT	I2STATUS	I2CLK	I2TIMER
B0		P0M1	P0M2	P1M1	P1M2	P2M1	P2M2	IP0H
A8	IE	SADDR			CMP1	CMP2		
A0	P2	KBI	AUXR1					
98	SCON	SBUF						
90	P1					DIVM		
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
80	P0	SP	DPL	DPH				PCON

Note: 1. The SFRs in the column with dark borders are bit-addressable

2. The table is condensed with eight locations per row. Empty locations indicate that these are no registers at these addresses. When a bit or register is not implemented, it will read high.

M1, M0: Mode Select bits:

M1	M0	MODE
0	0	Mode 0: 8-bits with 5-bit prescale.
0	1	Mode 1: 16-bits, no prescale.
1	0	Mode 2: 8-bits with auto-reload from THx
1	1	Mode 3: (Timer 0) TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is a 8-bit timer only controlled by Timer 1 control bits. (Timer 1) Timer/counter is stopped.

TIMER 0 LSB

Bit:	7	6	5	4	3	2	1	0
	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0

Mnemonic: TL0 Address: 8Ah

TL0.7-0: Timer 0 LSB

TIMER 1 LSB

Bit:	7	6	5	4	3	2	1	0
	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

Mnemonic: TL1 Address: 8Bh

TL1.7-0: Timer 1 LSB

TIMER 0 MSB

Bit:	7	6	5	4	3	2	1	0
	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

Mnemonic: TH0 Address: 8Ch

TH0.7-0: Timer 0 MSB

TIMER 1 MSB

Bit:	7	6	5	4	3	2	1	0
	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

Mnemonic: TH1 Address: 8Dh

TH1.7-0: Timer 1 MSB

Clock Control

Bit:	7	6	5	4	3	2	1	0
	-	-	-	T1M	T0M	-	-	-

Mnemonic: CKCON Address: 8Eh

Continued.

BIT	NAME	FUNCTION
5	BOI	Brown Out Interrupt: 0: Disable Brownout Detect Interrupt function and it will cause chip reset when BOF is set. 1: This prevents Brownout Detection from causing a chip reset and allows the Brownout Detect function to be used as an interrupt.
4	LPBOV	Low Power Brown Out Detect control: 0: When BOD is enable, the Brown Out detect is always turned on by normal run or Power Down mode. 1: When BOD is enable, the 1/16 time will be turned on Brown Out detect circuit by Power Down mode. When uC is in Power Down mode, the BOD will enable internal RC OSC(2MHz~0.5MHZ)
3	SRST	Software reset: 1: reset the chip as if a hardware reset occurred.
2	ADCEN	0: Disable ADC circuit. 1: Enable ADC circuit.
1	0	Reserved
0	DPS	Dual Data Pointer Select 0: To select DPTR of standard 8051. 1: To select DPTR1

INTERRUPT ENABLE

Bit:	7	6	5	4	3	2	1	0
	EA	EADC	EBO	ES	ET1	EX1	ET0	EX0

Mnemonic: IE Address: A8h

BIT	NAME	FUNCTION
7	EA	Global enable. Enable/Disable all interrupts.
6	EADC	Enable ADC interrupt.
5	EBO	Enable Brown Out interrupt.
4	ES	Enable Serial Port interrupt.
3	ET1	Enable Timer 1 interrupt.
2	EX1	Enable external interrupt 1.
1	ET0	Enable Timer 0 interrupt.
0	EX0	Enable external interrupt 0.

SLAVE ADDRESS

Bit:	7	6	5	4	3	2	1	0
	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0

Mnemonic: SADDR Address: A9h

BIT	NAME	FUNCTION
0~7	I2DAT	The data register of I2C.

I2C Status Register

Bit:	7	6	5	4	3	2	1	0
						-	-	-

Mnemonic: I2STATUS Address: BDh

BIT	NAME	FUNCTION
0~7	I2STATUS	The status register of I2C: The three least significant bits are always 0. The five most significant bits contain the status code. There are 23 possible status codes. When I2STATUScontains F8H, no serial interrupt is requested. All other I2STATUS values correspond to defined I2C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2STATUS one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software. In addition, states 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the formation frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit.

I2C Baud Rate Control Register

Bit:	7	6	5	4	3	2	1	0
	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0

Mnemonic: I2CLK Address: BEh

BIT	NAME	FUNCTION
7~ 0	I2CLK	The I2C clock rate bits.

I2C Timer Counter Register

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	ENTI	DIV4	TIF

Mnemonic: I2TIMER Address: BFh

Interrupt Enable Register 1

Bit:	7	6	5	4	3	2	1	0
	-	-	EPWM	EWDI	EC2	EC1	EKB	EI2

Mnemonic: IE1 Address: E8h

BIT	NAME	FUNCTION
7	-	Reserved.
6	-	Reserved.
5	EPWM	0: Disable PWM Interrupt when external brake pin was brake. 1: Enable PWM Interrupt when external brake pin was brake.
4	EWDI	0: Disable Watchdog Timer Interrupt. 1: Enable Watchdog Timer Interrupt.
3	EC2	0: Disable Comparator 2 Interrupt. 1: Enable Comparator 2 Interrupt.
2	EC1	0: Disable Comparator 1 Interrupt. 1: Enable Comparator 1 Interrupt.
1	EKB	0: Disable Keypad Interrupt. 1: Enable Keypad Interrupt.
0	EI2	0: Disable I2C Interrupt. 1: Enable I2C Interrupt.

B REGISTER

Bit:	7	6	5	4	3	2	1	0
	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0

Mnemonic: B Address: F0h

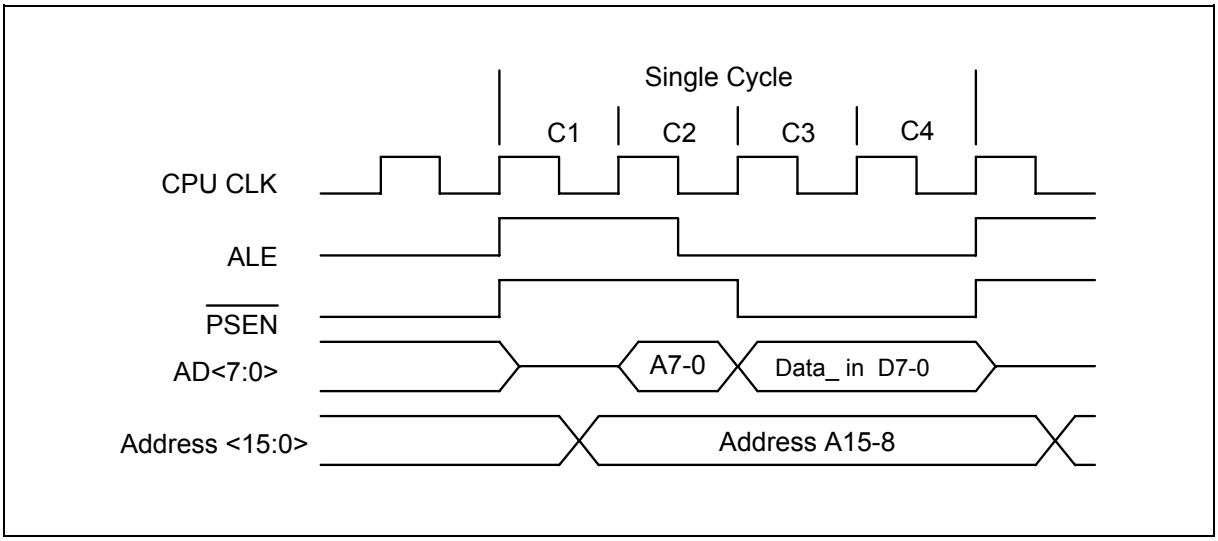
B.7-0: The B register is the standard 8052 register that serves as a second accumulator.

Port 0 Digital Input Disable

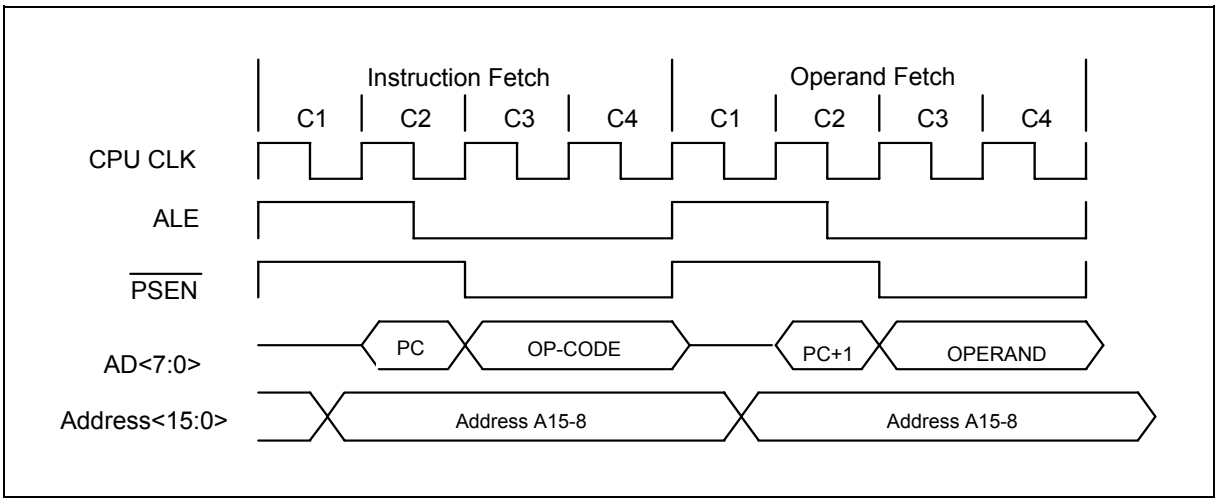
Bit:	7	6	5	4	3	2	1	0
	P0ID.7	P0ID.6	P0ID.5	P0ID.4	P0ID.3	P0ID.2	P0ID.1	P0ID.0

Mnemonic: P0ID Address: F6h

BIT	NAME	FUNCTION
7~0	P0ID.7 ~P0ID.0	Enable/Disable Port 0 digital inputs. 0: Enable Port 0 digital inputs. 1: Disable Port 0 digital inputs.



Single Cycle Instruction Timing



Two Cycle Instruction Timing

The W79E82X series use a four priority level interrupt structure. This allows great flexibility in controlling the handling of the W79E82X series many interrupt sources. The W79E82X series supports up to 13 interrupt sources.

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in registers IE0 or IE1. The IE0 register also contains a global disable bit, EA, which disables all interrupts at once.

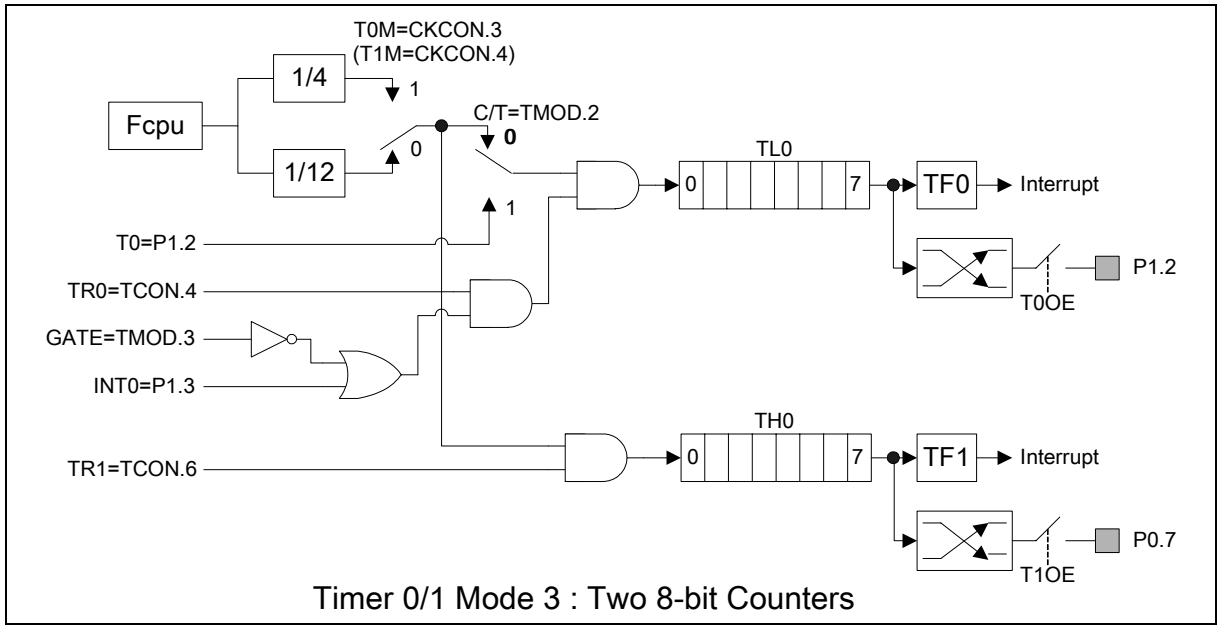
Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the IP0, IP0H, IP1, and IP1H registers. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. So, if two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve simultaneous requests of the same priority level.

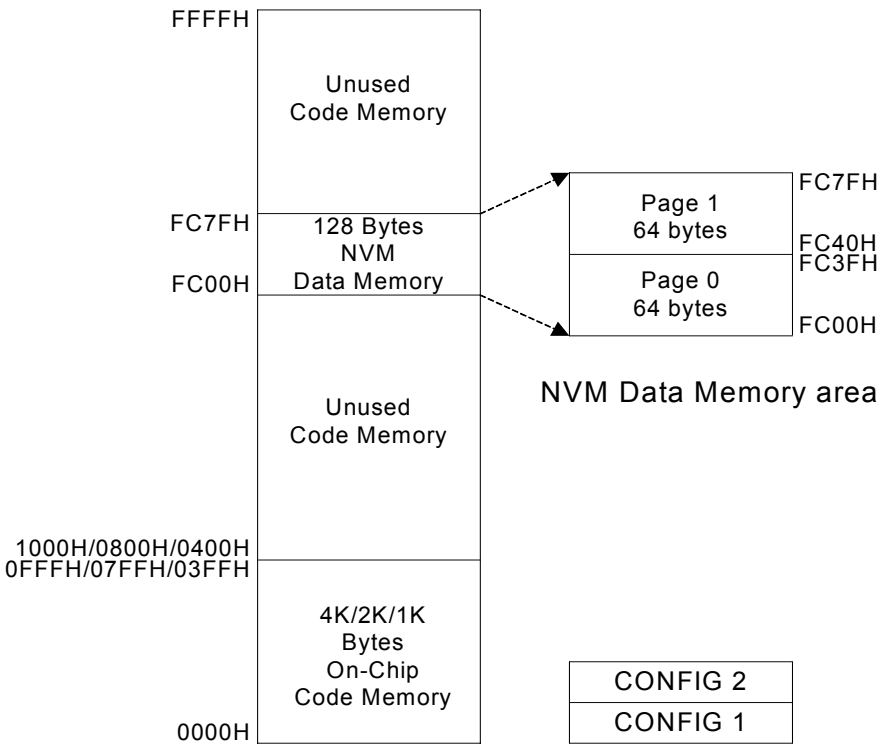
As below Table summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from Power Down mode.

DESCRIPTION	INTERRUPT FLAG BIT(S)	VECTOR ADDRESS	INTERRUPT ENABLE BIT(S)	INTERRUPT PRIORITY	ARBITRATION RANKING	POWER DOWN WAKEUP
External Interrupt 0	IE0	0003H	EX0 (IE0.0)	IP0H.0, IP0.0	1(highest)	Yes
Brownout Detect	BOF	002BH	EBO (IE.5)	IP0H.5, IP0.5	2	Yes
Watchdog Timer	WDIF	0053H	EWDI (IE1.4)	IP1H.4, IP1.4	3	Yes(1)
Timer 0 Interrupt	TF0	000BH	ET0 (IE.1)	IP0H.1, IP0.1	4	No
I2C Interrupt	SI	0033H	EI2 (IE1.0)	IP1H.0, IP1.0	5	No
ADC Converter	ADCI	005BH	EAD (IE.6)	IP0H.6, IP0.6	6	Yes(1)
External Interrupt 1	IE1	0013H	EX1 (IE.2)	IP0H.2, IP0.2	7	Yes
KBI Interrupt	KBF	003BH	EKB (IE1.1)	IP1H.1, IP1.1	8	Yes
Comparator 1 Interrupt	CMF1	0063H	EC1 (IE1.2)	IP1H.2, IP1.2	9	Yes
Timer 1 Interrupt	TF1	001BH	ET1 (IE.3)	IP0H.3, IP0.3	10	No
Comparator 2 Interrupt	CMF2	0043H	EC2 (IE1.3)	IP1H.3, IP1.3	11	Yes
Serial Port Tx and Rx	TI & RI	0023H	ES (IE.4)	IP0H.4, IP0.4	12	No
PWM Interrupt	BKF	0073H	EPWM (IE1.5)	IP1H.5, IP1.5	13 (lowest)	No

Note: 1. The Watchdog Timer and ADC Converter can wake up Power Down Mode when its clock source is used internal RC.



Timer/Counter Mode 3.



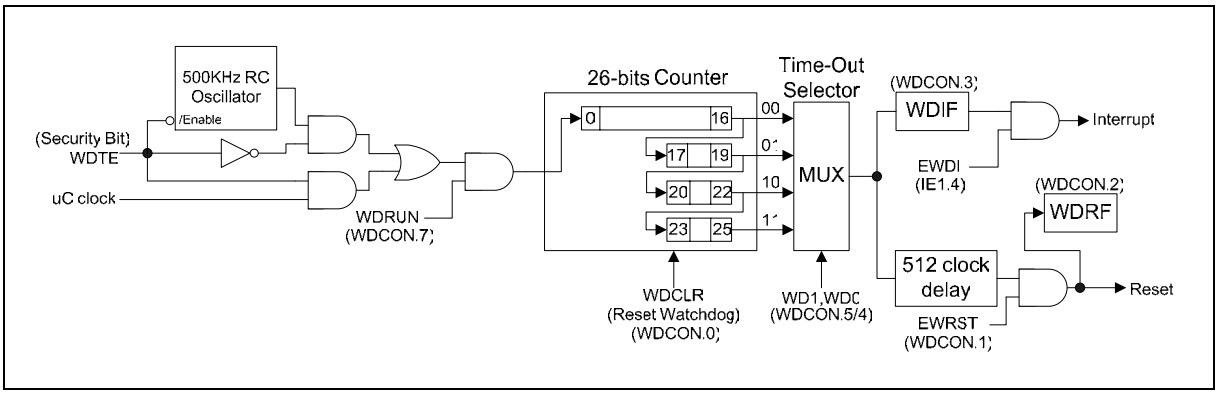
W79E823/W79E822/W79E821 Memory Map

BIT	NAME	FUNCTION
7~0	NVMADDR.7 ~ NVMADDR.0	The NVM address: The register is indicated NVM data memory of low byte address on On-Chip code memory space.

Mnemonic: NVMADDR Address: C6h

15. WATCHDOG TIMER

The Watchdog Timer is a free-running Timer which can be programmed by the user to serve as a system monitor, a time-base generator or an event timer. It is basically a set of dividers that divide the system clock. The divider output is selectable and determines the time-out interval. When the time-out occurs a flag is set, which can cause an interrupt if enabled, and a system reset can also be caused if it is enabled. The interrupt will occur if the individual interrupt enable and the global enable are set. The interrupt and reset functions are independent of each other and may be used separately or together depending on the users software.

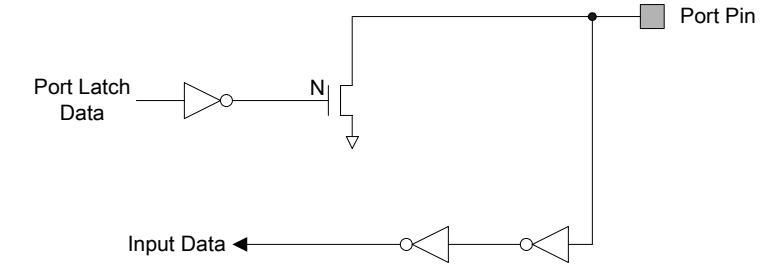


Watchdog Timer

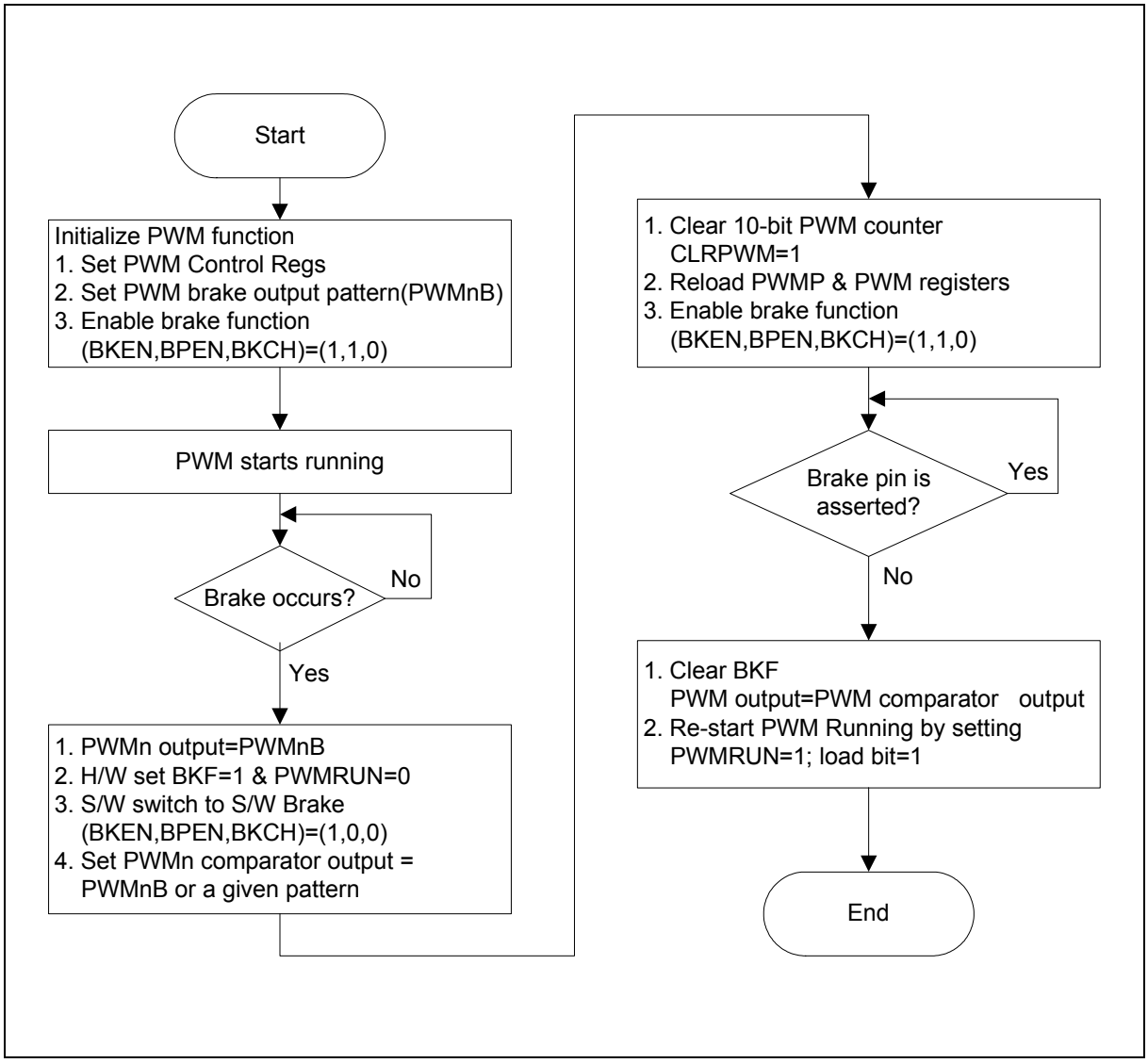
The Watchdog Timer should first be restarted by using WDCLR. This ensures that the timer starts from a known state. The WDCLR bit is used to restart the Watchdog Timer. This bit is self clearing, i.e. after writing a 1 to this bit the software will automatically clear it. The Watchdog Timer will now count clock cycles. The time-out interval is selected by the two bits WD1 and WD0 (WDCON.5 and WDCON.4). When the selected time-out occurs, the Watchdog interrupt flag WDIF (WDCON.3) is set. After the time-out has occurred, the Watchdog Timer waits for an additional 512 clock cycles. If the Watchdog Reset EWRST (WDCON.1) is enabled, then 512 clocks after the time-out, if there is no WDCLR, a system reset due to Watchdog Timer will occur. This will last for two machine cycles, and the Watchdog Timer reset flag WDRF (WDCON.2) will be set. This indicates to the software that the Watchdog was the cause of the reset.

When used as a simple timer, the reset and interrupt functions are disabled. The timer will set the WDIF flag each time the timer completes the selected time interval. The WDIF flag is polled to detect a time-out and the WDCLR allows software to restart the timer. The Watchdog Timer can also be used as a very long timer. The interrupt feature is enabled in this case. Every time the time-out occurs an interrupt will occur if the global interrupt enable EA is set.

The main use of the Watchdog Timer is as a system monitor. This is important in real-time control applications. In case of some power glitches or electro-magnetic interference, the processor may begin to execute errant code. If this is left unchecked the entire system may crash. Using the watchdog timer interrupt during software development will allow the user to select ideal watchdog reset locations. The code is first written without the watchdog interrupt or reset. Then the Watchdog interrupt is enabled to identify code locations where interrupt occurs. The user can now insert instructions to reset the Watchdog Timer, which will allow the code to run without any Watchdog Timer



Open Drain Output


Start
Initialize PWM function
1. Set PWM Control Regs
2. Set PWM brake output pattern(PWMnB)
3. Enable brake function
(BKEN,BPEN,BKCH)=(1,1,0)
PWM starts running
Brake occurs?
No
Yes
1. PWMn output=PWMnB
2. H/W set BKF=1 & PWMRUN=0
3. S/W switch to S/W Brake
(BKEN,BPEN,BKCH)=(1,0,0)
4. Set PWMn comparator output =
PWMnB or a given pattern
1. Clear 10-bit PWM counter
CLRPWM=1
2. Reload PWMP & PWM registers
3. Enable brake function
(BKEN,BPEN,BKCH)=(1,1,0)
Brake pin is asserted?
Yes
No
1. Clear BKF
PWM output=PWM comparator output
2. Re-start PWM Running by setting
PWMRUN=1; load bit=1
End

24. ANALOG-TO-DIGITAL CONVERTER

The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (Vin). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCCON.5 (ADCEX) =0. A conversion is then started by setting control bit ADCCON.3 (ADCS) The hardware or software start mode is selected when ADCCON.5 =1, and a conversion may be started by setting ADCCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with tpw flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set end a value of “1” will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

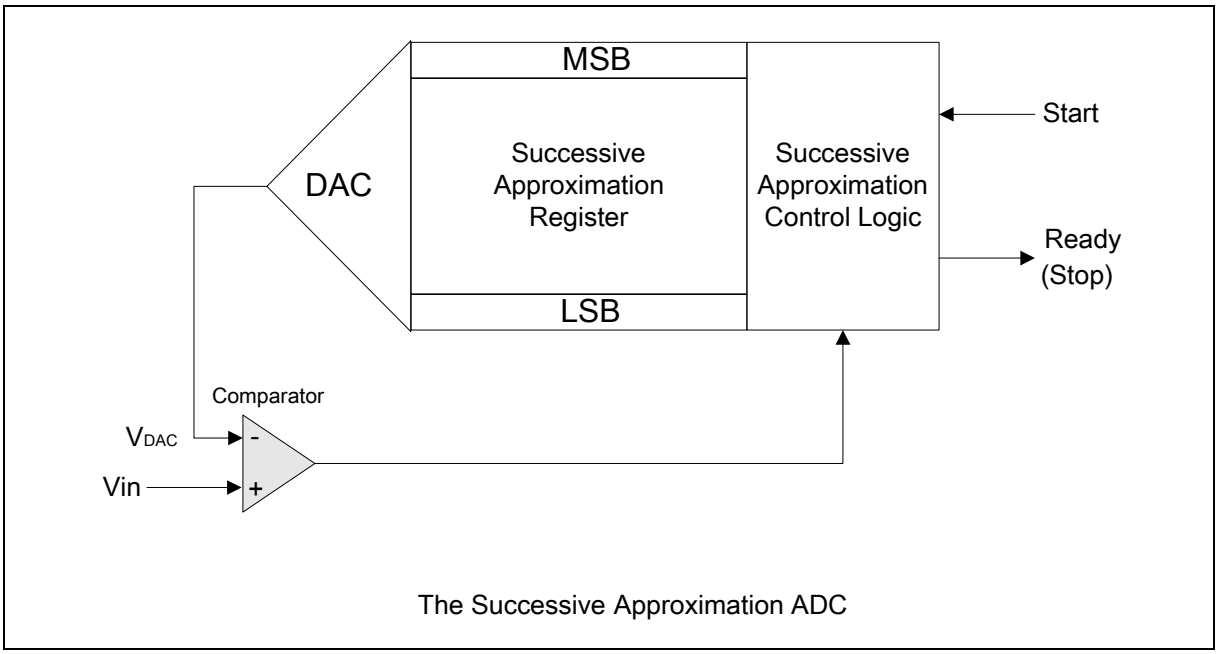
During the next eight machine cycles, the voltage at the previously selected pin of one of analog input pin is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000b). The output of the DAC (50% full scale) is compared to the input voltage Vin. If the input voltage is greater than VDAC, then the bit remains set; otherwise if is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000b or 01 0000 0000b, depending on the previous result), and the VDAC is compared to Vin again. If the input voltage is greater then VDAC, then the bit remains set; otherwise it is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCCON.7 (ADC.1) and ADCCON.6 (ADC.0). The user may ignore the two least significant bits in ADCCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 52 machine cycles. ADCI will be set and the ADCS status flag will be reset 52 cycles after the ADCS is set. Control bits ADCCON.0 and ADCCON.1 are used to control an analog multiplexer which

selects one of 4 analog channels. An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.



Successive Approximation ADC

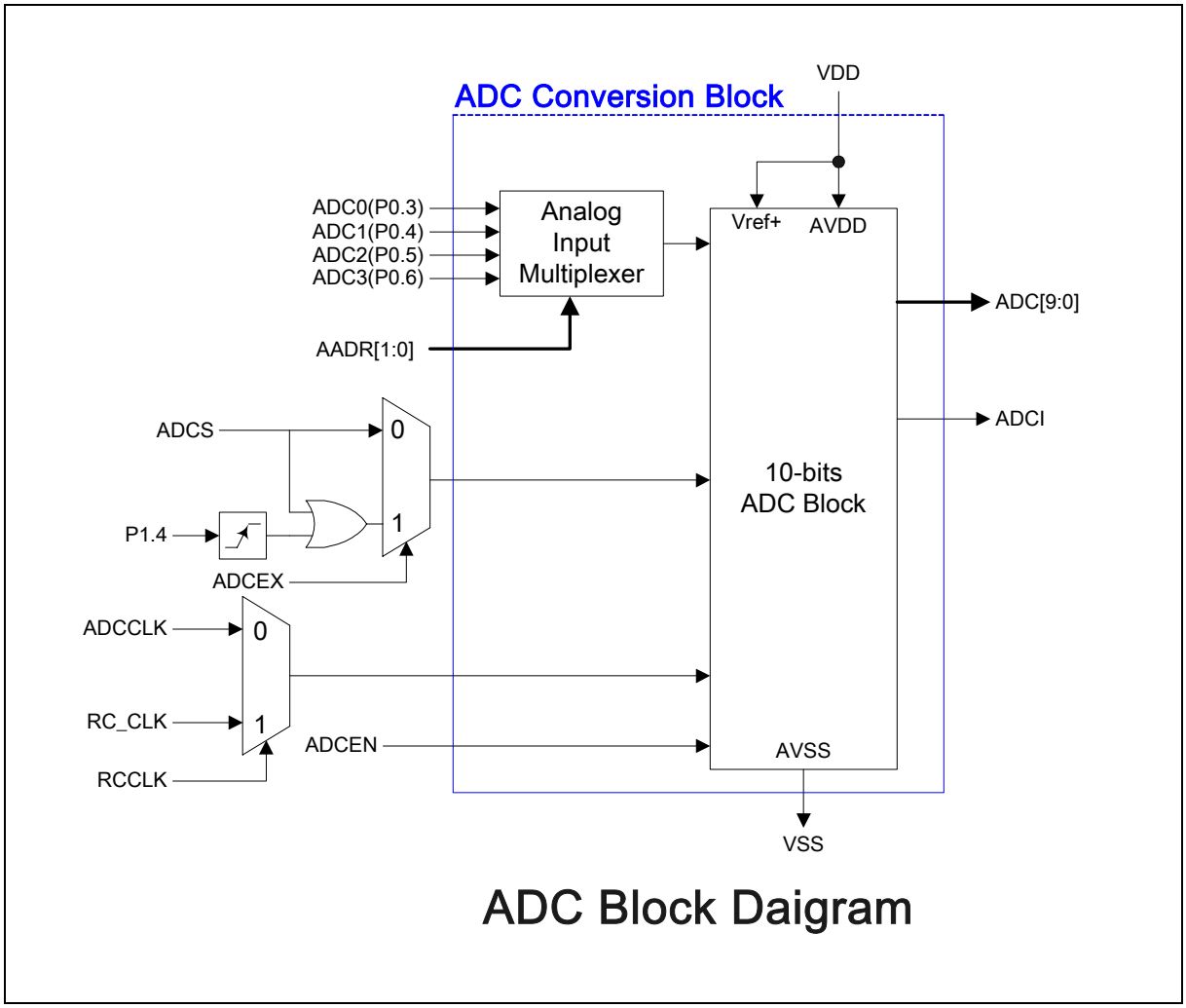
24.1 ADC Resolution and Analog Supply:

The ADC circuit has its own supply pins (AVDD and AVSS) and one pins (Vref+) connected to each end of the DAC's resistance-ladder that the AVDD and Vref+ are connected to VDD and AVSS is connected to VSS. The ladder has 1023 equally spaced taps, separated by a resistance of "R". The first tap is located 0.5×R above AVSS, and the last tap is located 0.5×R below Vref+. This gives a total ladder resistance of 1024×R. This structure ensures that the DAC is monotonic and results in a symmetrical quantization error.

For input voltages between VSS and [(Vref+) + ½ LSB], the 10-bit result of an A/D conversion will be 0000000000B = 000H. For input voltages between [(Vref+) – 3/2 LSB] and Vref+, the result of a conversion will be 1111111111B = 3FFH. Vref+ and AVSS may be between AVDD + 0.2V and VSS – 0.2 V. Vref+ should be positive with respect to VSS, and the input voltage (Vin) should be between Vref+ and VSS.

The result can always be calculated from the following formula:

$$\text{Result} = 1024 \times \frac{\text{Vin}}{\text{Vref}+} \quad \text{or Result} = 1024 \times \frac{\text{Vin}}{\text{VDD}}$$

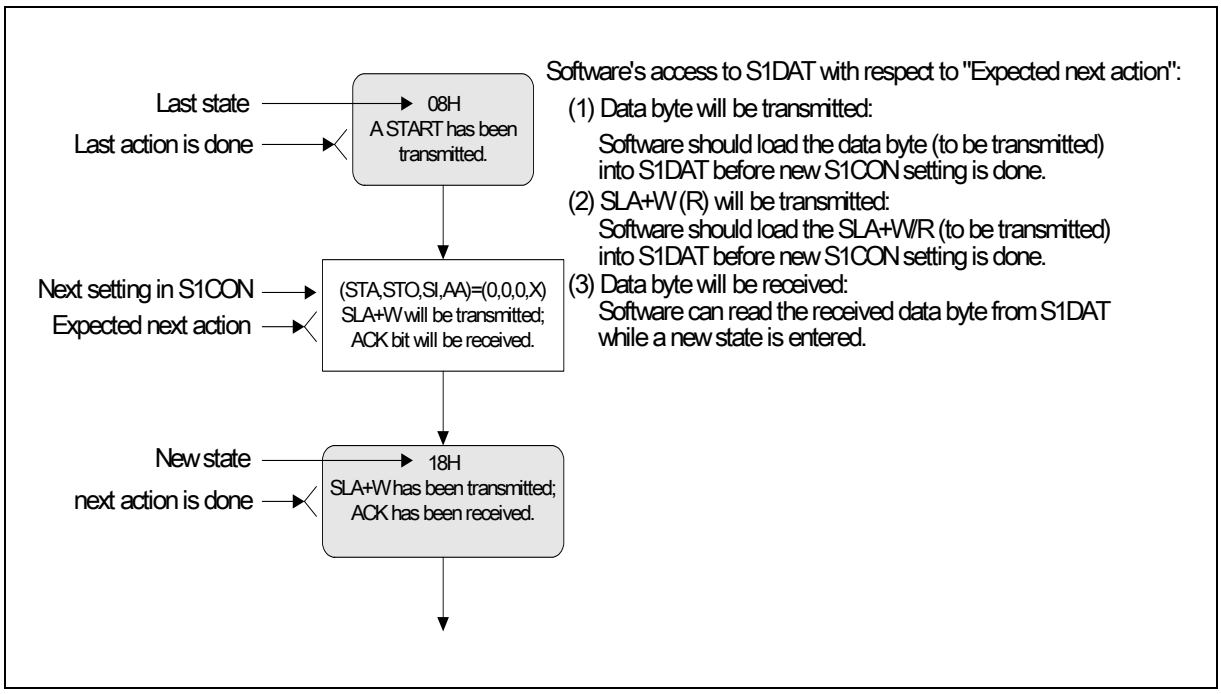


The ADC Block Diagram

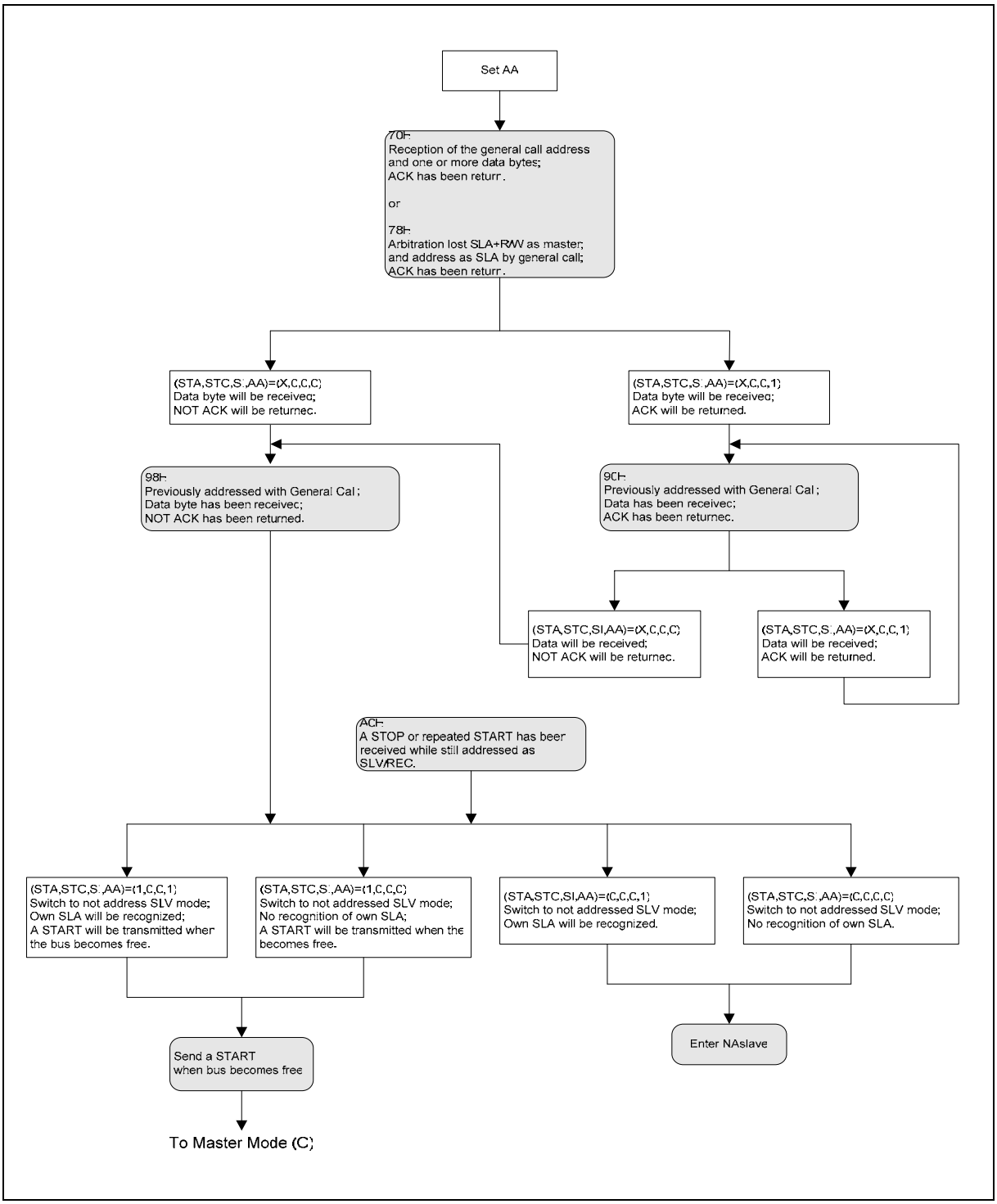
25.3 Operating Modes of I2C

The four operating modes are: Master/Transmitter, Master/Receiver, Slave/Transmitter and Slave/Receiver. Bits STA, STO and AA in I2CON decide the next action the SIO1 hardware will take after SI is cleared. When the next action is completed, a new status code in I2STATUS will be updated and SI will be set by hardware in the same time. Now, the interrupt service routine is entered (if the SI interrupt is enabled), the new status code can be used to decide which appropriate service routine the software is to branch. Data transfers in each mode are shown in the following figures.

*** Legend for the following four figures:



GC Mode


Set AA
70H
Reception of the general call address and one or more data bytes;
ACK has been returr.
or
78H
Arbitration lost SLA+R/W as master;
and address as SLA by general call;
ACK has been returr.
(STA,STC,S:,AA)=(X,C,C,C)
Data byte will be receiveo;
NOT ACK will be returnec.
(STA,STC,S:,AA)=(X,C,C,1)
Data byte will be receiveo;
ACK will be returned.
98H
Previously addressed with General Cal;
Data byte has been receivec;
NOT ACK has been returned.
9CH
Previously addressed with General Cal;
Data has been receivec;
ACK has been returnec.
(STA,STC,SI,AA)=(X,C,C,C)
Data will be received;
NOT ACK will be returnec.
(STA,STC,S:,AA)=(X,C,C,1)
Data will be received;
ACK will be returned.
A0H
A STOP or repeated START has beer
received while still addressed as
SLV/REC.
(STA,STC,S:,AA)=(1,C,C,1)
Switch to not address SLV mode;
Own SLA will be recognized;
A START will be transmitted wher
the bus becomes free.
(STA,STC,S:,AA)=(1,C,C,C)
Switch to not addressed SLV mode;
No recognition of own SLA;
A START will be transmitted when the
becomes free.
(STA,STC,SI,AA)=(C,C,C,1)
Switch to not addressed SLV mode;
Own SLA will be recognized.
(STA,STC,S:,AA)=(C,C,C,C)
Switch to not addressed SLV mode;
No recognition of own SLA.
Send a START
when bus becomes free
Enter NAslave
To Master Mode (C)