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	Active
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	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
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/n79e824asg

Continued

BIT	NAME	FUNCTION
5	TF0	Timer 0 Overflow Flag. This bit is set when Timer 0 overflows. It is cleared automatically when the program does a timer 0 interrupt service routine. Software can also set or clear this bit.
4	TR0	Timer 0 Run Control. This bit is set or cleared by software to turn timer/counter on or off.
3	IE1	Interrupt 1 Edge Detect Flag: Set by hardware when an edge/level is detected on $\overline{INT1}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin.
2	IT1	Interrupt 1 Type Control. Set/cleared by software to specify falling edge/ low level triggered external inputs.
1	IE0	Interrupt 0 Edge Detect Flag. Set by hardware when an edge/level is detected on $\overline{INT0}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin.
0	IT0	Interrupt 0 Type Control: Set/cleared by software to specify falling edge/ low level triggered external inputs.

TIMER MODE CONTROL

Bit:	7	6	5	4	3	2	1	0
	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
	TIMER1				TIMER0			

Mnemonic: TMOD   Address: 89h

BIT	NAME	FUNCTION
7	GATE	Gating control: When this bit is set, Timer/counter 1 is enabled only while the $\overline{INT1}$ pin is high and the TR1 control bit is set. When cleared, the $\overline{INT1}$ pin has no effect, and Timer 1 is enabled whenever TR1 control bit is set.
6	$C/\overline{T}$	Timer or Counter Select: When clear, Timer 1 is incremented by the internal clock. When set, the timer counts falling edges on the T1 pin.
5	M1	Timer 1 mode select bit 1. See table below.
4	M0	Timer 1 mode select bit 0. See table below.
3	GATE	Gating control: When this bit is set, Timer/counter 0 is enabled only while the $\overline{INT0}$ pin is high and the TR0 control bit is set. When cleared, the $\overline{INT0}$ pin has no effect, and Timer 0 is enabled whenever TR0 control bit is set.
2	$C/\overline{T}$	Timer or Counter Select: When clear, Timer 0 is incremented by the internal clock. When set, the timer counts falling edges on the T0 pin.
1	M1	Timer 0 mode select bit 1. See table below.
0	M0	Timer 0 mode select bit 0. See table below.

Port Output Configuration Settings:

PXM1.Y	PXM2.Y	PORT INPUT/OUTPUT MODE
0	0	Quasi-bidirectional
0	1	Push-Pull
1	0	Input Only (High Impedance) P2M1.PxS=0, TTL input P2M1.PxS=1, Schmitt input
1	1	Open Drain

INTERRUPT HIGH PRIORITY

Bit:

7	6	5	4	3	2	1	0
-	PADCH	PBOH	PSH	PT1H	PX1H	PT0H	PX0H

Mnemonic: IP0H   Address: B7h

BIT	NAME	FUNCTION
7	-	This bit is un-implemented and will read high.
6	PADCH	1: To set interrupt high priority of ADC is highest priority level.
5	PBOH	1: To set interrupt high priority of Brown Out Detector is highest priority level.
4	PSH	1: To set interrupt high priority of Serial port 0 is highest priority level.
3	PT1H	1: To set interrupt high priority of Timer 1 is highest priority level.
2	PX1H	1: To set interrupt high priority of External interrupt 1 is highest priority level.
1	PT0H	1: To set interrupt high priority of Timer 0 is highest priority level.
0	PX0H	1: To set interrupt high priority of External interrupt 0 is highest priority level.

INTERRUPT PRIORITY 0

Bit:

7	6	5	4	3	2	1	0
-	PADC	PBO	PS	PT1	PX1	PT0	PX0

Mnemonic: IP   Address: B8h

BIT	NAME	FUNCTION
7	-	This bit is un-implemented and will read high.
6	PADC	1: To set interrupt priority of ADC is higher priority level.
5	PBO	1: To set interrupt priority of Brown Out Detector is higher priority level.
4	PS	1: To set interrupt priority of Serial port 0 is higher priority level.
3	PT1	1: To set interrupt priority of Timer 1 is higher priority level.
2	PX1	1: To set interrupt priority of External interrupt 1 is higher priority level.
1	PT0	1: To set interrupt priority of Timer 0 is higher priority level.

Continued

BIT	NAME	FUNCTION
6	AC	Auxiliary carry: Set when the previous operation resulted in a carry from the high order nibble.
5	F0	User flag 0: The General purpose flag that can be set or cleared by the user.
4~3	RS1~RS0	Register bank select bits.
2	OV	Overflow flag: Set when a carry was generated from the seventh bit but not from the 8th bit as a result of the previous operation, or vice-versa.
1	F1	User Flag 1: The General purpose flag that can be set or cleared by the user software.
0	P	Parity flag: Set/cleared by hardware to indicate odd/even number of 1's in the accumulator.

RS.1-0: Register Bank Selection Bits:

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00-07h
0	1	1	08-0Fh
1	0	2	10-17h
1	1	3	18-1Fh

PWMP COUNTER HIGH BITS REGISTER

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWMP.9	PWMP.8

Mnemonic: PWMPH   Address: D1h

BIT	NAME	FUNCTION
7-2	-	Reserved.
1-0	PWMP.[9:8]	The PWM Counter Register bits 9~8.

PWM 0 HIGH BITS REGISTER

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWM0.9	PWM0.8

Mnemonic: PWM0H   Address: D2h

BIT	NAME	FUNCTION
7~2	-	Reserved.
1~0	PWM0.[9:8]	The PWM 0 High Bits Register bit 9~8.

BIT	NAME	FUNCTION
7-0	ACC.[7:0]	The A or ACC register is the standard 8052 accumulator

ADC CONTROL REGISTER

Bit:	7	6	5	4	3	2	1	0
	ADC.1	ADC.0	ADCEX	ADCI	ADCS	RCCLK	AADR1	AADR0

Mnemonic: ADCCON   Address: E1h

BIT	NAME	FUNCTION
7	ADC.1	The ADC conversion result.
6	ADC.0	The ADC conversion result.
5	ADCEX	Enable STADC-triggered conversion 0: Conversion can only be started by software (i.e., by setting ADCS). 1: Conversion can be started by software or by a rising edge on STADC (pin P1.4).
4	ADCI	ADC Interrupt flag: This flag is set when the result of an A/D conversion is ready. This generates an ADC interrupt, if it is enabled. The flag may be cleared by the ISR. While this flag is 1, the ADC cannot start a new conversion. ADCI can not be set by software.
3	ADCS	ADC Start and Status: Set this bit to start an A/D conversion. It may also be set by STADC if ADCEX is 1. This signal remains high while the ADC is busy and is reset right after ADCI is set. Notes: 1. It is recommended to clear ADCI *before* ADCS is set. However, if ADCI is cleared and ADCS is set at the same time, a new A/D conversion may start on the same channel. 2. Software clearing of ADCS will abort conversion in progress. 3. ADC cannot start a new conversion while ADCS or ADCI is high.
2	RCCLK	0: The CPU clock is used as ADC clock. 1: The internal RC clock is used as ADC clock.
1	AADR1	The ADC input select. See table below.
0	AADR0	The ADC input select. See table below.

The ADCI and ADCS control the ADC conversion as below:

ADCI	ADCS	ADC STATUS
0	0	ADC not busy; A conversion can be started.
0	1	ADC busy; Start of a new conversion is blocked.
1	0	Conversion completed; Start of a new conversion requires ADCI = 0.
1	1	This is an internal temporary state that user can ignore it.

9.1 Instruction Timing

This section is important because some applications use software instructions to generate timing delays. It also provides more information about timing differences between the N79E825 series and the standard 8051/52.

In N79E825 series, each machine cycle is four clock periods long. Each clock period is called a state, and each machine cycle consists of four states: C1, C2 C3 and C4, in order. Both clock edges are used for internal timing, so the duty cycle of the clock should be as close to 50% as possible to avoid timing conflicts.

The N79E825 series does one op-code fetch per machine cycle, so, in most instructions, the number of machine cycles required is equal to the number of bytes in the instruction. There are 256 available op-codes. 128 of them are single-cycle instructions, so many op-codes are executed in just four clocks period. Some of the other op-codes are two-cycle instructions, and most of these have two-byte op-codes. However, there are some instructions that have one-byte instructions yet take two cycles to execute. One important example is the MOVX instruction.

In the standard 8052, the MOVX instruction is always two machine cycles long. However, in the N79E825 series each machine cycle is made of only 4 clock periods compared to the 12 clock periods for the standard 8052. Therefore, even though the number of categories has increased, each instruction is at least 1.5 to 3 times faster than the standard 8052 in terms of clock periods.
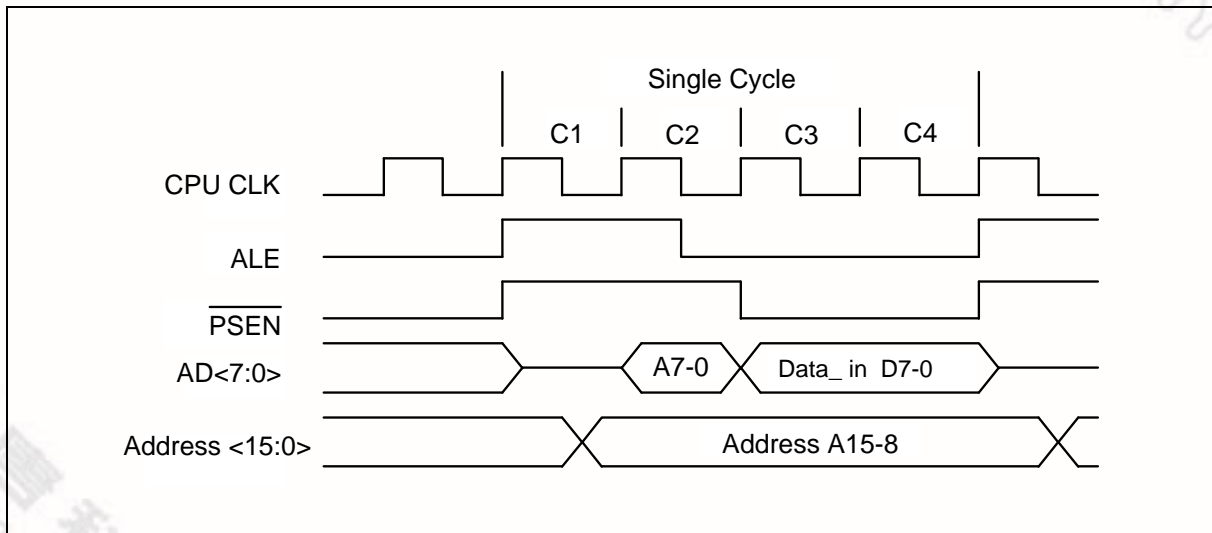

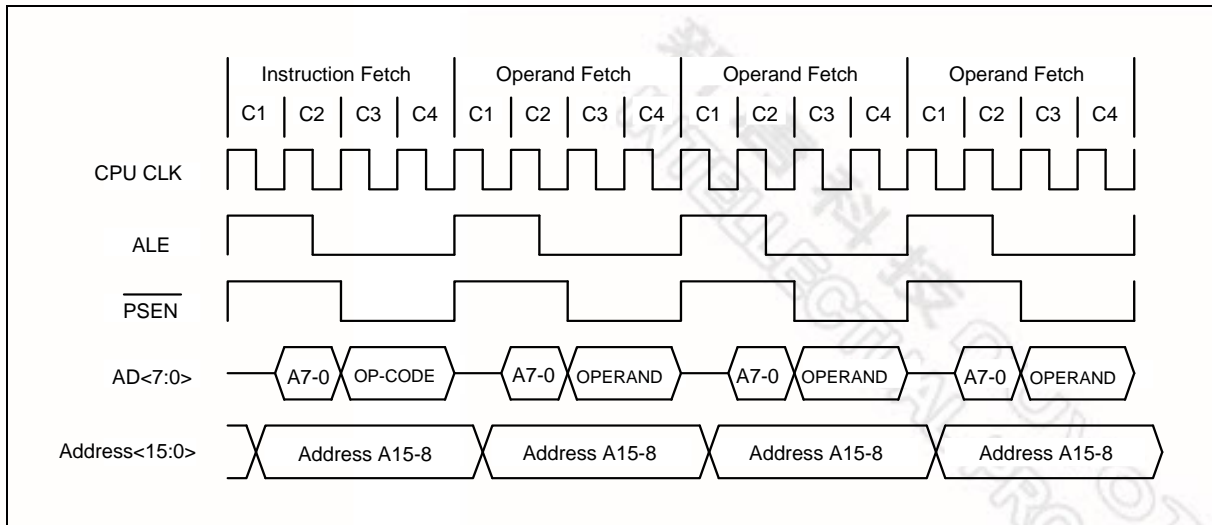Figure 9-1: Single Cycle Instruction Timing
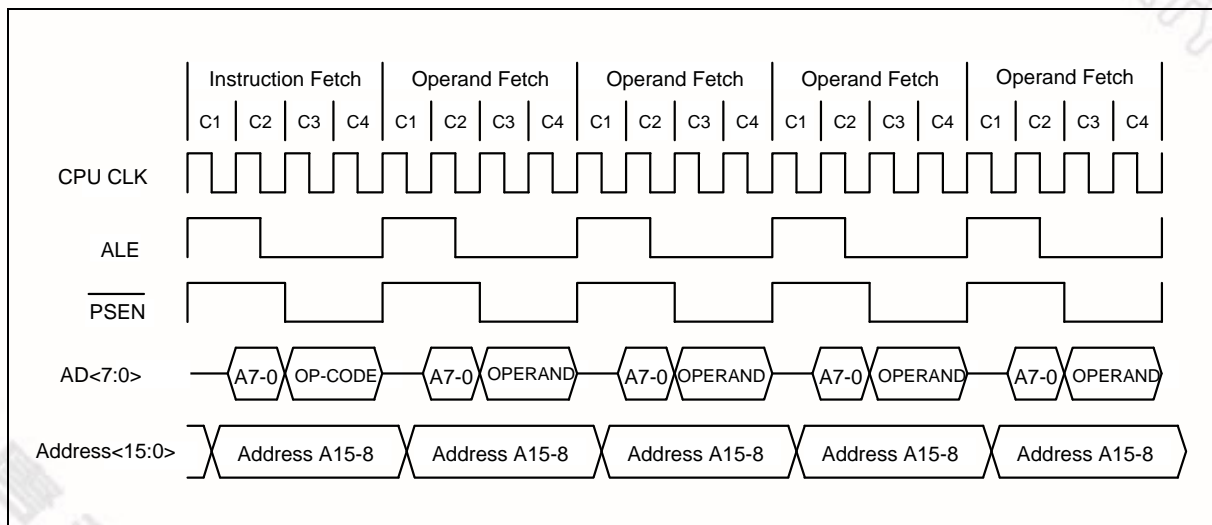
Figure 9-4: Four Cycles Instruction Timing

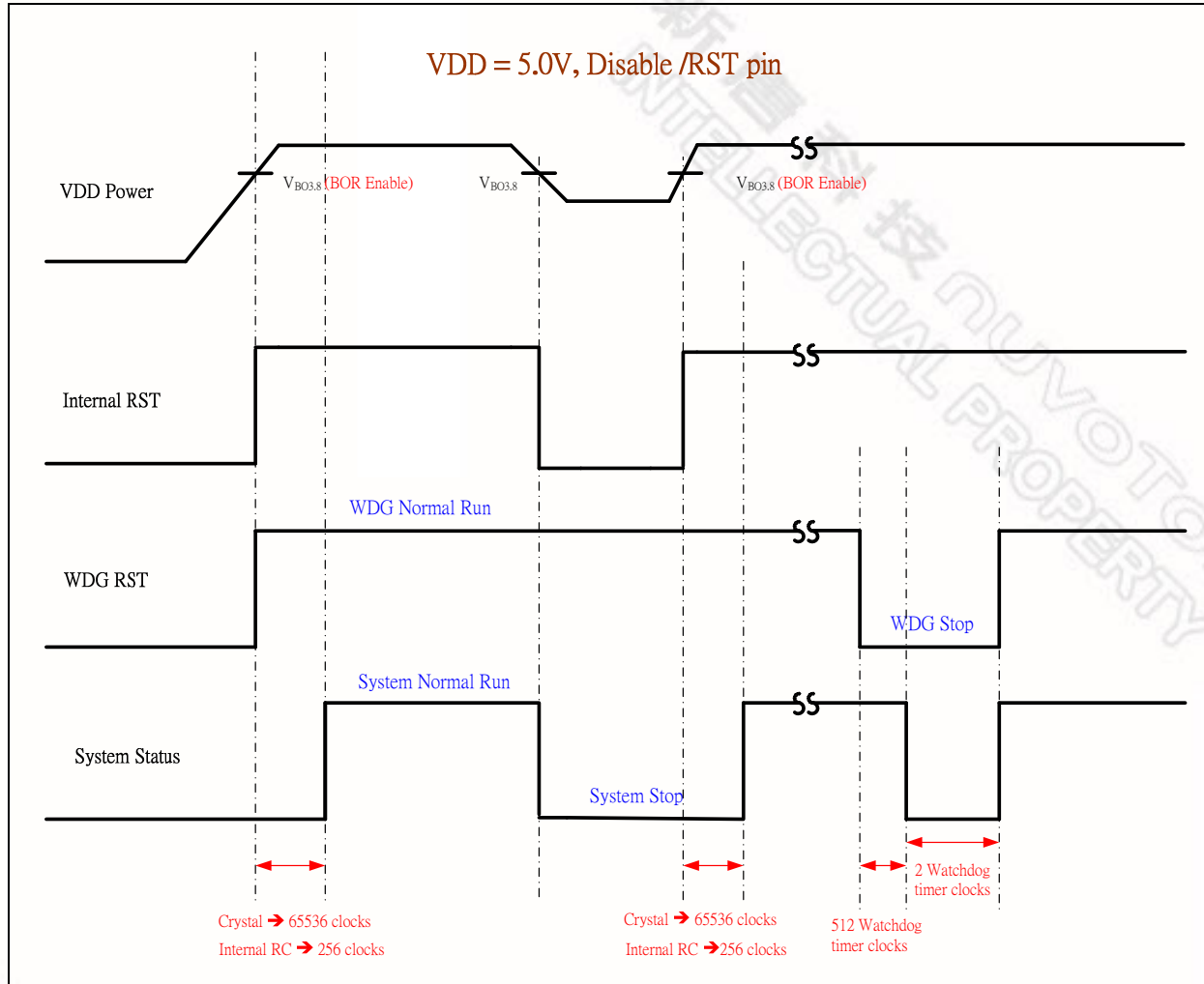Figure 9-5: Five Cycles Instruction Timing

and hence save power.

Figure 11-1: Internal reset and VDD monitor timing diagram

12 INTERRUPTS

The N79E825 series have four priority level interrupts structure with 13 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled.

12.1 Interrupt Sources

The External Interrupts INT0 and INT1 can be either edge triggered or level triggered, depending on bits IT0 and IT1. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. In the edge triggered mode, the INTx inputs are sampled in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IEx in TCON is set. The flag bit requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. The IEx flag is automatically cleared when the service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IEx flag will not be cleared by the hardware on entering the service routine. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source.

The Timer 0 and 1 Interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and Timer 1. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced. The Watchdog timer can be used as a system monitor or a simple timer. In either case, when the time-out count is reached, the Watchdog Timer interrupt flag WDIF (WDCON.3) is set. If the interrupt is enabled by the enable bit EIE.4, then an interrupt will occur.

The Serial block can generate interrupt on reception or transmission. There are two interrupt sources from the Serial block, which are obtained by the RI and TI bits in the SCON SFR. These bits are not automatically cleared by the hardware, and the user will have to clear these bits by software.

All the bits that generate interrupts can be set or reset by software, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IE SFR. IE also has a global enable/disable bit EA, which can be cleared to disable all interrupts.

The ADC can generate interrupt after finished ADC converter. There is one interrupt source, which is obtained by the ADCI bit in the ADCCON SFR. This bit is not automatically cleared by the hardware, and the user will have to clear this bit using software.

The two comparators can generate interrupt after comparator output has toggle occurs by CMF1 and CMF2. These bits are not automatically cleared by the hardware, and the user will have to clear these bits using software.

The I2C function can generate interrupt, if EI2C and EA bits are enabled, when SI Flag is set due to a new I2C status code is generated, SI flag is generated by hardware and must be cleared by software.

The PWM function can generate interrupt by BKF flag, after external brake pin has brake occurred. This bit will be cleared by software.

The interrupt flags are sampled every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are;

1. An interrupt of equal or higher priority is not currently being serviced.

2. The current polling cycle is the last machine cycle of the instruction currently being execute.

3. The current instruction does not involve a write to IE, EIE, IP0, IP0H, IP1 or IPH1 registers and is

not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in the same machine cycle. If an interrupt flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered; every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of Timer interrupts, the TF0 or TF1 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of external interrupts, $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. The Watchdog timer interrupt flag WDIF has to be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack, but does not save the Program Status Word PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL. These address of vector for the different sources are as follows:

VECTOR LOCATIONS FOR INTERRUPT SOURCES

SOURCE	VECTOR ADDRESS	SOURCE	VECTOR ADDRESS
External Interrupt 0	0003h	Timer 0 Overflow	000Bh
External Interrupt 1	0013h	Timer 1 Overflow	001Bh
Serial Port	0023h	Brownout Interrupt	002Bh
I2C Interrupt	0033h	KBI Interrupt	003Bh
Comparator 2 Interrupt	0043h	-	004Bh
Watchdog Timer	0053h	ADC Interrupt	005Bh
Comparator 1 Interrupt	0063h	-	006Bh
PWM Brake Interrupt	0073h	-	007Bh

Table 12-1: Vector locations for interrupt sources

Execution continues from the vectored address till an RETI instruction is executed. On execution of the RETI instruction the processor pops the Stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what it was after the hardware LCALL, if the execution is return to the interrupted program. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the Interrupt Controller that the interrupt service routine is completed, and would leave the controller still thinking that the service routine is underway.

nuvoTon

machine cycles. This is a 50% reduction in terms of clock periods.

12.4 Interrupt Inputs

The N79E825 series have 13 interrupts source, and two individual interrupt inputs sources, one is for IE0, IE1, BOF, KBF, WDT, ADC, CMF1 and CMF2, and other is IF0, IF1, RI+TI ,SI and BKF. Two interrupt inputs are identical to those present on the standard 80C51 microcontroller as show in below figures.

If an external interrupt is enabled when the N79E825 series are put into Power Down or Idle mode, the interrupt will cause the processor to wake up and resume operation.
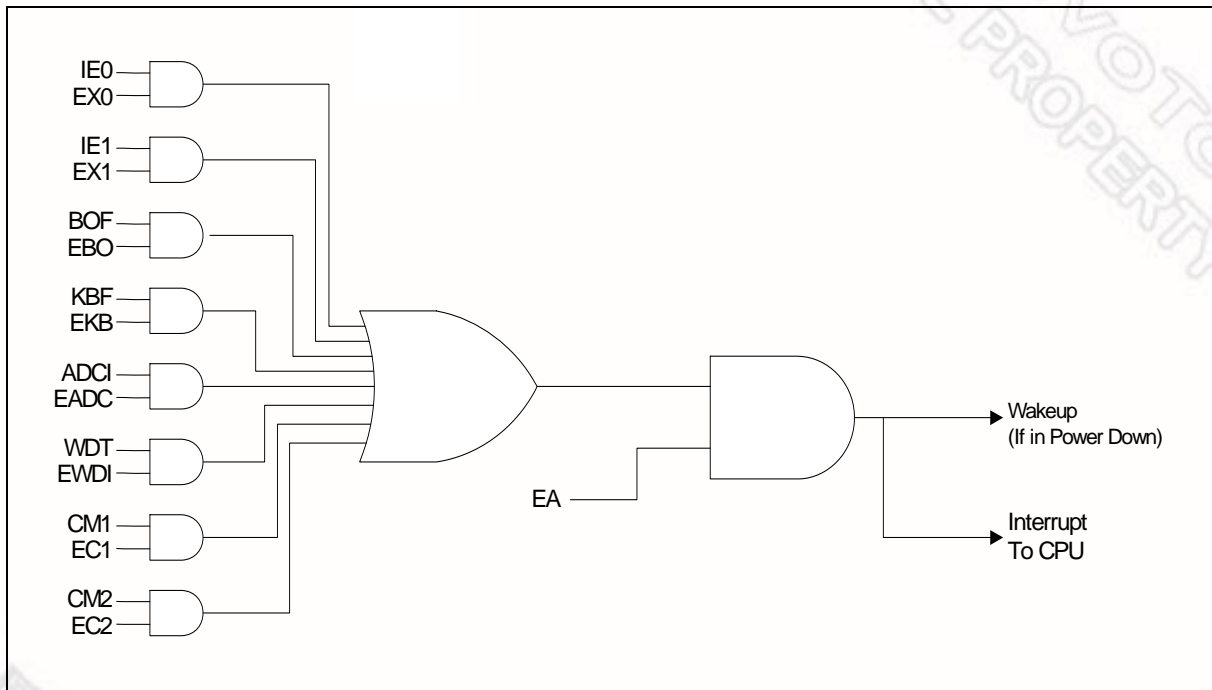
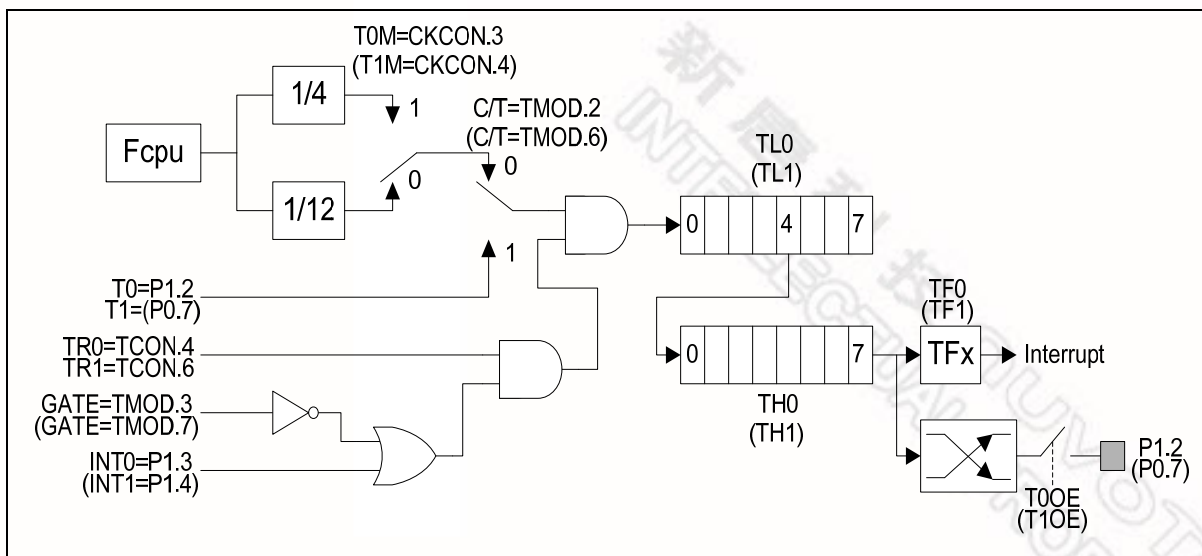Figure 12-1: Interrupt Sources that can wake up from Power Down Mode

Figure 13-1: Timer/Counters 0 & 1 in Mode 0

13.1.3 Mode 1

Mode 1 is similar to Mode 0 except that the counting register forms a 16-bit counter, rather than a 13-bit counter. This means that all the bits of THx and TLx are used. Roll-over occurs when the timer moves from a count of FFFFh to 0000h. The timer overflow flag TFx of the relevant timer is set and if enabled an interrupt will occur. The selection of the time-base in the timer mode is similar to that in Mode 0. The gate function operates similarly to that in Mode 0.
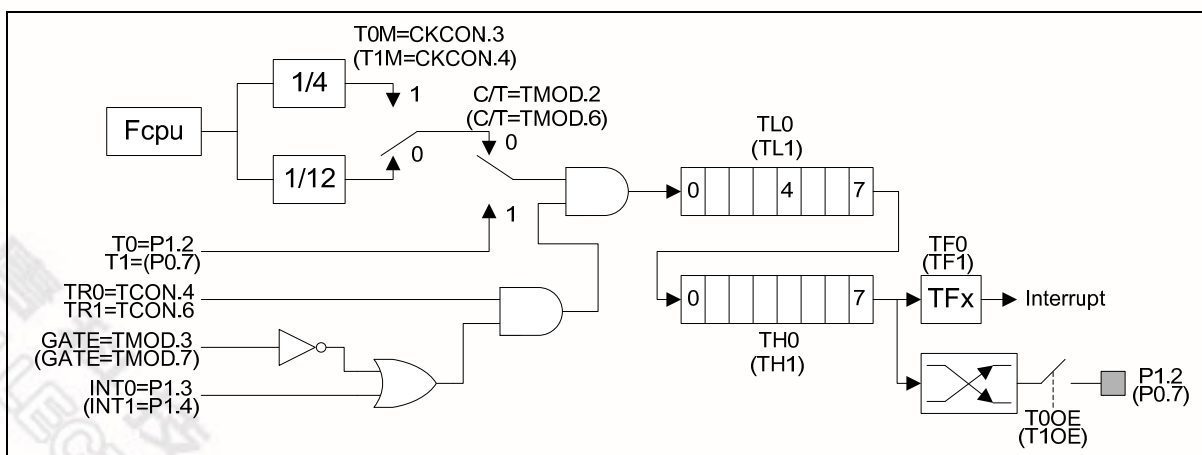

Figure 13-2: Timer/Counters 0 & 1 in Mode 1

13.1.4 Mode 2

In Mode 2, the timer/counter is in the Auto Reload Mode. In this mode, TLx acts as 8-bit count register, while THx holds the reload value. When the TLx register overflows from FFh to 00h, the TFx bit in TCON is set and TLx is reloaded with the contents of THx, and the counting process continues from here. The reload operation leaves the contents of the THx register unchanged. Counting is enabled by the TRx bit and proper setting of GATE and INTx pins. As in the other two modes 0 and 1 mode 2 allows counting of either clock cycles (clock/12 or clock/4) or pulses on pin Tn.

In "Timer" mode, if output toggle enable bit of P2M1.T0OE or P2M1.T1OE is enabled, T0 or T1 output pin will toggle whenever a timer overflow occurs.
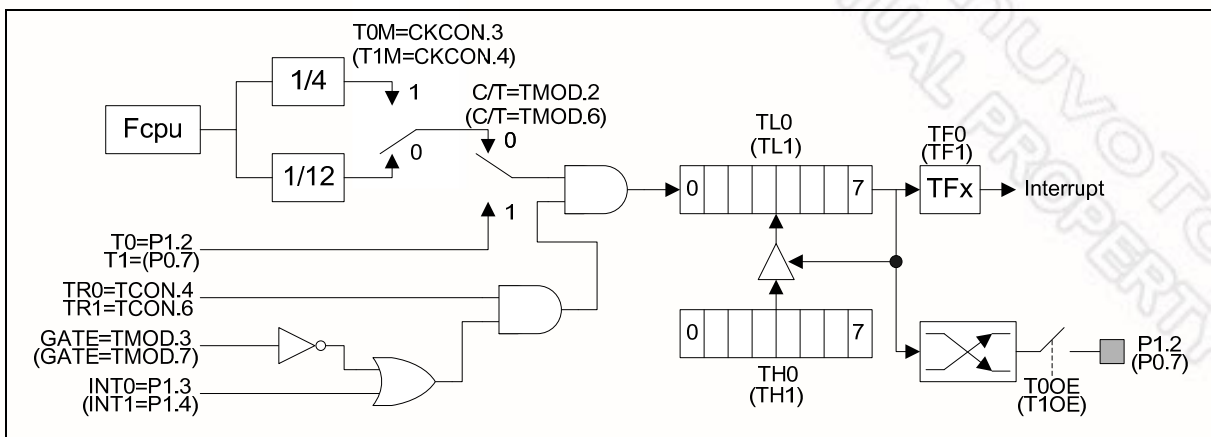

Figure 13-3: Timer/Counter 0 & 1 in Mode 2

13.1.5 Mode 3

Mode 3 has different operating methods for the two timer/counters. For timer/counter 1, mode 3 simply freezes the counter. Timer/Counter 0, however, configures TL0 and TH0 as two separate 8 bit count registers in this mode. The logic for this mode is shown in the figure. TL0 uses the Timer/Counter 0 control bits C/$\overline{\text{T}}$, GATE, TR0, $\overline{\text{INT0}}$ and TF0. The TL0 can be used to count clock cycles (clock/12 or clock/4) or 1-to-0 transitions on pin T0 as determined by C/T (TMOD.2). TH0 is forced as a clock cycle counter (clock/12 or clock/4) and takes over the use of TR1 and TF1 from Timer/Counter 1. Mode 3 is used in cases where an extra 8 bit timer is needed. With Timer 0 in Mode 3, Timer 1 can still be used in Modes 0, 1 and 2, but its flexibility is somewhat limited. While its basic functionality is maintained, it no longer has control over its overflow flag TF1 and the enable bit TR1. Timer 1 can still be used as a timer/counter and retains the use of GATE and INT1 pin. In this condition it can be turned on and off by switching it out of and into its own Mode 3. It can also be used as a baud rate generator for the serial port.

In "Timer" mode, if output toggle enable bit of P2M1.T0OE or P2M1.T1OE is enabled, T0 or T1 output pin will toggle whenever a timer overflow occurs.

Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. The 1-to-0 detector continuously monitors the RxD line, sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide–by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter.

The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three bases. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port. If the first bit detected after the falling edge of RxD pin is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF.

After shifting in 8 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.
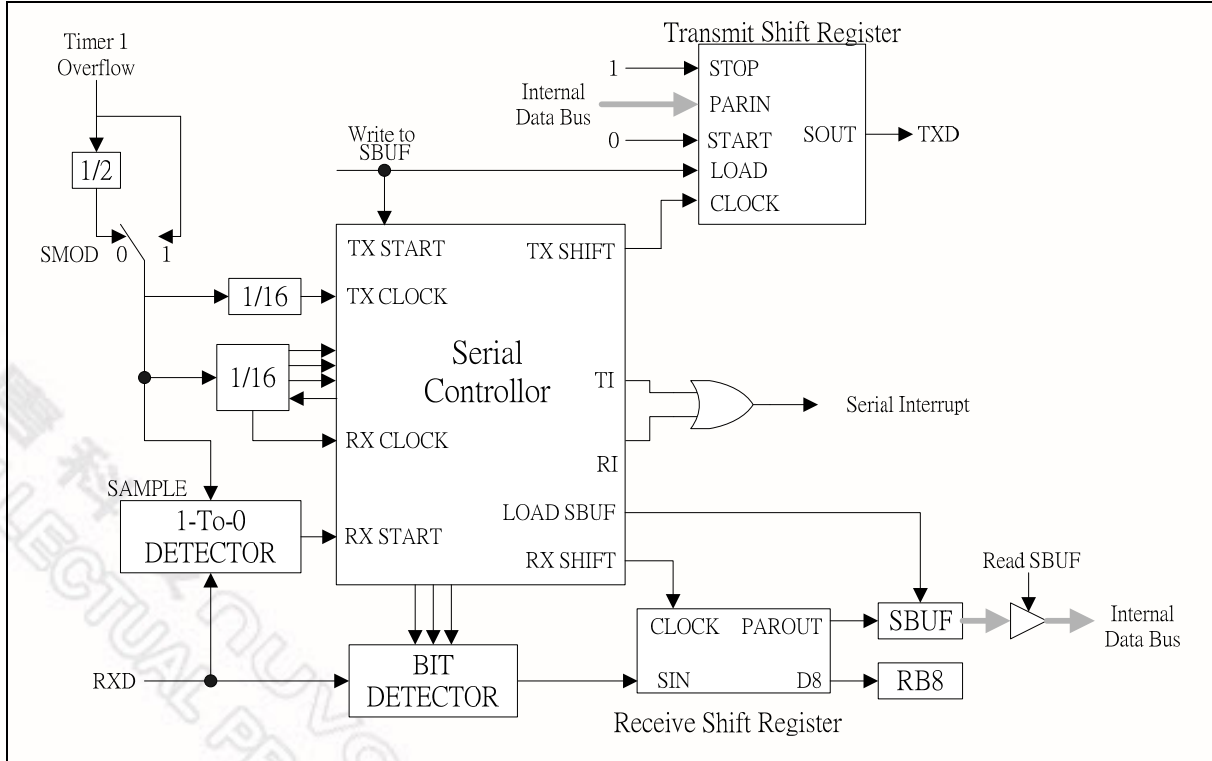

Figure 16-2: Serial Port Mode 1

nuvoTon

17 TIME ACCESS PROCTECTION

The N79E825 series have a new feature, like the Watchdog Timer which is a crucial to proper operation of the system. If left unprotected, errant code may write to the Watchdog control bits resulting in incorrect operation and loss of control. In order to prevent this, the N79E825 series have a protection scheme which controls the write access to critical bits. This protection scheme is done using a timed access.

In this method, the bits which are to be protected have a timed write enable window. A write is successful only if this window is active, otherwise the write will be discarded. This write enable window is open for 3 machine cycles if certain conditions are met. After 3 machine cycles, this window automatically closes. The window is opened by writing AAh and immediately 55h to the Timed Access (TA) SFR. This SFR is located at address C7h. The suggested code for opening the timed access window is

```
TA      REG    0C7h                    ;Define new register TA, located at 0C7h
        MOV    TA, #0AAh
        MOV    TA, #055h
```

When the software writes AAh to the TA SFR, a counter is started. This counter waits for 3 machine cycles looking for a write of 55h to TA. If the second write (55h) occurs within 3 machine cycles of the first write (AAh), then the timed access window is opened. It remains open for 3 machine cycles, during which the user may write to the protected bits. Once the window closes the procedure must be repeated to access the other protected bits.

Examples of Timed Assessing are shown below.

Example 1: Valid access
```
        MOV        TA, #0AAh               ;3 M/C  Note: M/C = Machine Cycles
        MOV        TA, #055h               ;3 M/C
        MOV        WDCON, #00h             ;3 M/C
```
Example 2: Valid access
```
        MOV        TA, #0AAh               ;3 M/C
        MOV        TA, #055h               ;3 M/C
        NOP                                ;1 M/C
        SETB       EWRST                   ;2 M/C
```
Example 3: Valid access
```
        MOV        TA, #0AAh               ;3 M/C
        MOV        TA, #055h               ;3 M/C
        ORL        WDCON, #00000010B       ;3M/C
```
Example 4: Invalid access
```
        MOV        TA, #0AAh               ;3 M/C
        MOV        TA, #055h               ;3 M/C
        NOP                                ;1 M/C
        NOP                                ;1 M/C
```

Example 5: Invalid Access

```
        MOV         TA, #0AAh                    ;3 M/C
        NOP                                      ;1 M/C
        MOV         TA, #055h                    ;3 M/C
        SETB        EWT                          ;2 M/C
```

In the first three examples, the writing to the protected bits is done before the 3 machine cycles window closes. In Example 4, however, the writing to the protected bit occurs after the window has closed, and so there is effectively no change in the status of the protected bit. In Example 5, the second write to TA occurs 4 machine cycles after the first write, therefore the timed access window is not opened at all, and the write to the protected bit fails.

The N79E825 series devices support brake function which can be activated by software or external pin (P0.2).   The Brake function is controlled by the PWMCON2 register. The setting and details description of software brake and external pin brake can be found at the brake condition table at the SFR section.

As for external brake, the user program can poll the brake flag (BKF) or enable PWM's brake interrupt to determine when the external Brake Pin is asserted and causes a brake to occur. The brake pin (P0.2) can be set to trigger the brake function by either low or high level, by clearing or setting the PWMCON2.6 (BKPS) bit respectively. The details description of varies brake functions can be found in the brake condition table.

Since the Brake Pin being asserted will automatically clear the Run bit of PWMCON1.7 and BKF (PWMCON3.0) flag will be set, the user program can poll this bit or enable PWM's brake interrupt to determine when the Brake Pin causes a brake to occur. The other method for detecting a brake caused by the Brake Pin would be to tie the Brake Pin to one of the external interrupt pins. This latter approach is needed if the Brake signal is of insufficient length to ensure that it can be captured by a polling routine. When, after being asserted, the condition causing the brake is removed, the PWM outputs go to whatever state that had immediately prior to the brake. This means that in order to go from brake being asserted to having the PWM run without going through an indeterminate state, care must be taken. If the Brake Pin causes brake to be asserted, the following prototype code will allow the PWM to go from brake and then run smoothly after brake is released.

25 I2C SERIAL CONTROL

The I2C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

– Bidirectional data transfer between masters and slaves

– Multimaster bus (no central master)

– Arbitration between simultaneously transmitting masters without corruption of serial data on the bus

– Serial clock synchronization allows devices with different bit rates to communicate via one serial bus

– Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer

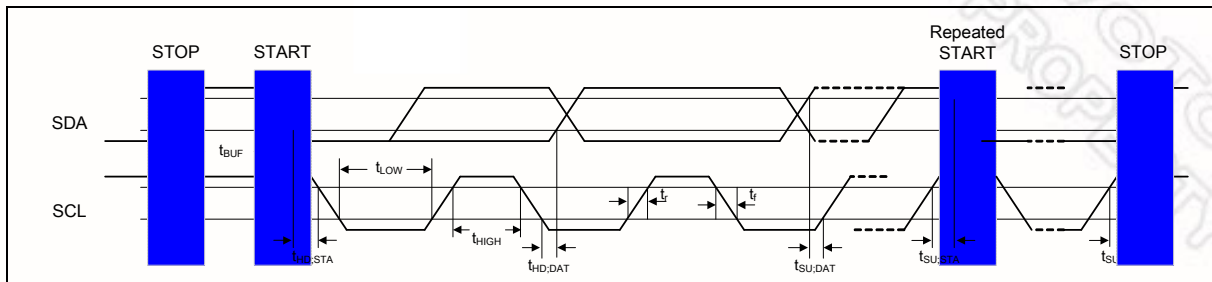–   The I2C bus may be used for test and diagnostic purposes

Figure 25-1: I2C Bus Timing

The device's on-chip I2C logic provides the serial interface that meets the I2C bus standard mode specification. The I2C logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (I2STATUS) reflects the status of the I2C bus.

The I2C port, SCL and SDA are at P1.2 and P1.3. When the I/O pins are used as I2C port, user must set the pins to logic high in advance. When I2C port is enabled by setting ENS to high, the internal states will be controlled by I2CON and I2C logic hardware. Once a new status code is generated and stored in I2STATUS, the I2C interrupt flag (SI) will be set automatically. If both EA and EI2C are also in logic high, the I2C interrupt is requested. The 5 most significant bits of I2STATUS stores the internal state code, the lowest 3 bits are always zero and the content keeps stable until SI is cleared by software.

25.1 SIO Port

The SIO port is a serial I/O port, which supports all transfer modes from and to the I2C bus. The SIO port handles byte transfers autonomously. To enable this port, the bit ENS1 in I2CON should be set to '1'. The CPU interfaces to the SIO port through the following six special function registers: I2CON (control register, C0H), I2STATUS (status register, BDH), I2DAT (data register, BCH), I2ADDR (address registers, C1H), I2CLK (clock rate register BEH) and I2TIMER (Timer counter register, BFH). The SIO H/W interfaces to the I2C bus via two pins: SDA (P1.3, serial data line) and SCL (P1.2, serial clock line). Pull up resistor is needed for Pin P1.2 and P1.3 for I2C operation as these are 2 open drain pins.

Continued

BIT	NAME	FUNCTION
3	-	Reserved.
2	-	Reserved.
1	Fosc1	CPU Oscillator Type Select bit 1
0	Fosc0	CPU Oscillator Type Select bit 0

Oscillator Configuration bits:

FOSC1	FOSC0	OSC SOURCE
0	0	4MHz ~ 20MHz crystal
0	1	Internal RC Oscillator
1	0	Reserved
1	1	External Oscillator in XTAL1

27.2 CONFIG2

CONFIG 2 :

7	6	5	4	3	2	1	0	
C7	C6	-	-	-	-	-	-	CONFIG Bit

C7: **16K/8K/4K/2K** Flash EPROM Code Lock Bit.
C6: **256** byte Data Lock Bit.

CONFIG Register 2

C7: 16K/8K/4K/2K Flash EPROM Lock bit

This bit is used to protect the customer's program code. It may be set after the programmer finishes the programming and verifies sequence. Once this bit is set to logic 0, both the Flash EPROM data and CONFIG Registers can not be accessed again.

C6: 256 byte Data Flash EPROM Lock bit

This bit is used to protect the customer's data code. It may be set after the programmer finishes the programming and verifies sequence. Once this bit is set to logic 0, both the data Flash EPROM and CONFIG Registers can not be accessed again.

29 PACKAGE DIMENSIONS

29.1 20-pin SSOP
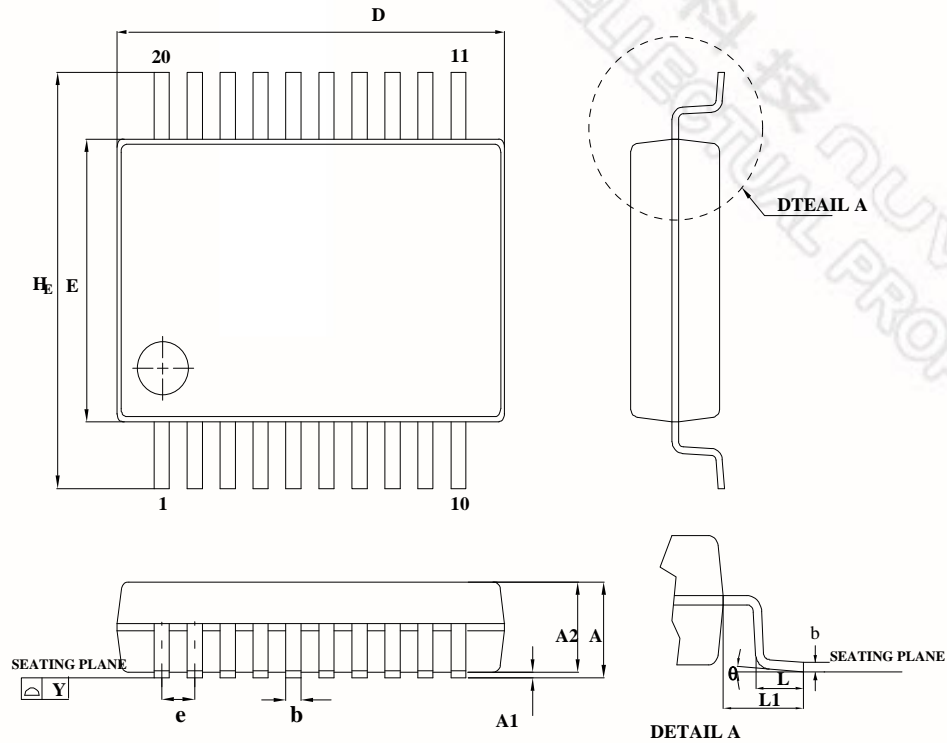
SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM	MAX.	MIN.	NOM	MAX.
A	—	—	2.00	—	—	0.079
A1	0.05	—	—	0.002	—	—
A2	1.65	1.75	1.85	0.065	0.069	0.073
b	0.22	—	0.38	0.009	—	0.015
c	0.09	—	0.25	0.004	—	0.010
D	6.90	7.20	7.50	0.272	0.283	0.295
E	5.00	5.30	5.60	0.197	0.209	0.220
H_E	7.40	7.80	8.20	0.291	0.307	0.323
e	—	0.65	—	—	0.0256	—
L	0.55	0.75	0.95	0.021	0.030	0.037
L1	—	1.25	—	—	0.050	—
Y	—	—	0.10	—	—	0.004
θ	0	—	8	0	—	8

Figure 29-1: 20-Pin SSOP