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	24MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n79e844awg

22.1 On-Chip RC Oscillator 151
22.2 Crystal/Resonator 151
23 POWER MONITORING 152
23.1 Power-on Detection 152
23.2 Brown-out Detection 152
24 RESET CONDITIONS 155
24.1 Power-on Reset 155
24.2 BOD Reset 156
24.3 RST Pin Reset 157
24.4 Watchdog Timer Reset 157
24.5 Software Reset 158
24.6 Boot Selection 159
24.7 Reset State 160
25 CONFIG BITS (CONFIG) 162
25.1 CONFIG0 162
25.2 CONFIG1 (N79E845 Only) 163
25.3 CONFIG2 164
25.4 CONFIG3 165
26 INSTRUCTION SETS 167
27 IN-CIRCUIT PROGRAM (ICP) 171
28 ELECTRICAL CHARACTERISTICS 173
28.1 Absolute Maximum Ratings 173
28.2 DC Electrical Characteristics 173
28.3 AC Electrical Characteristics 177
28.3.1 10-bits SAR-ADC Specification 177
28.3.2 4 ~ 24 MHz XTAL Specifications 178
28.3.3 Internal RC Oscillator Specifications – 22.1184 MHz/11.0592 MHz 178
28.3.4 Internal RC Oscillator Specifications – 10 kHz 179
29 APPLICATION CIRCUIT FOR EMC IMMUNITY 180
30 PACKAGE DIMENSIONS 181
30.1 20-pin TSSOP - 4.4X6.5mm 181
30.2 16-pin SOP - 150 mil 182
31 DOCUMENT REVISION HISTORY 183

6 Memory Organization

The N79E845/844/8432 has embedded Flash EPROM including 16K/8K/4K bytes Application Program Flash memory (APROM), fixed 4K bytes Data Flash (except the device with 16K APROM), fixed 2K bytes Load ROM Flash memory (LDROM) and CONFIG-bits. The N79E845/844/8432 also provides 256 bytes of on-chip direct/indirect RAM and 256 bytes of XRAM accessed by MOVX instruction.

For the device of 16K-bytes APROM, the APROM block and Data Flash block comprise the 16K bytes embedded Flash. The block size is CONFIG-bits/software configurable.

The N79E845/844/8432 is built with a CMOS page-erase. The page-erase operation erases all bytes within a page of 128 bytes.

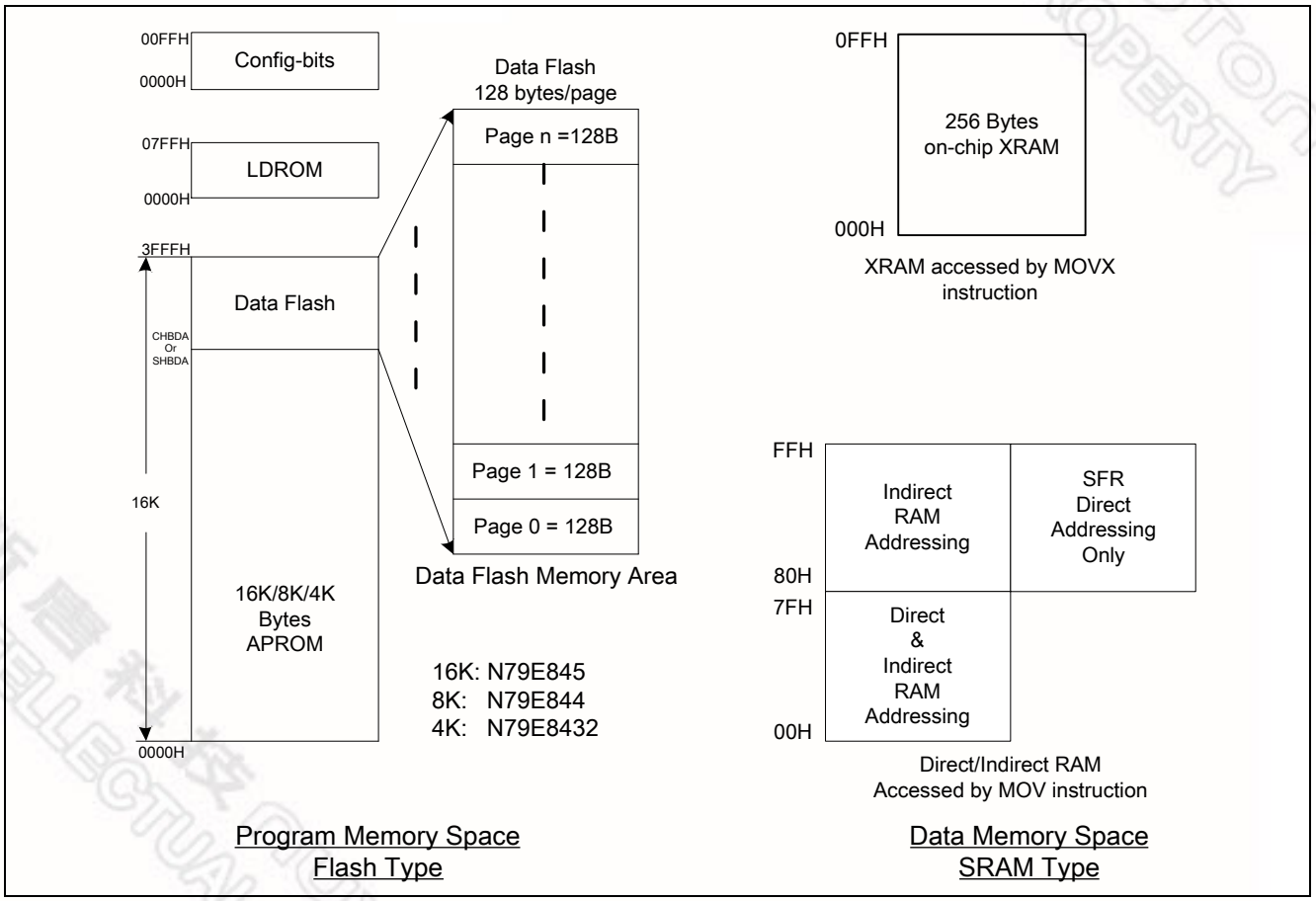

Figure 6-1 N79E845/844/8432 Memory Map

6.1 APROM Flash memory

The N79E845/844/8432 has **16K/8K/4K** Program Memory. All instructions are fetched for execution from this memory area. The MOVC instruction can also read this memory region.

The user application program is located in APROM. When CPU boots from APROM (CHPCON.BS=0), CPU starts executing the program from address 0000H. If the value of program counter (PC) is over the space of APROM, CPU will execute NOP operand and program counter increases one by one until PC reaches 3FFFH then it wraparounds to address 0000H of APROM, the CPU executes the application program again.

6.2 LDROM Flash Memory

Each device of the N79E845/844/8432 is equipped with 2 Kbytes LDROM stored the ISP application program. User may develop the ISP function in LDROM for updating application program or Data Flash. Similarly, APROM can also reprogram LDROM and Data Flash. The start address of LDROM is at 0000H corresponding to the physical address of the Flash memory. However, when CPU runs in LDROM, CPU automatically re-vectors the LDROM start address to 0000H, therefore user program regards the LDROM as an independent program memory, meanwhile, with all interrupt vectors that CPU provides.

6.3 CONFIG-bits

There are several bytes of CONFIG-bits located CONFIG-bits block. The CONFIG-bits define the CPU initial setting after power up or reset. Only hardware parallel writer or hardware ICP writer can erase/program CONFIG-bits. ISP program in LDROM can also erase/program CONFIG-bits.

6.4 On-chip Non-volatile Data Flash

The N79E845/844/8432 additionally has non-volatile Data Flash, which is non-volatile so that it remains its content even after the power is off. Therefore, in general application the user can write or read data which rules as parameters or constants. By the software path, SP mode can erase, written, or read the Data Flash only . Of course, hardware with parallel Programmer/Writer or ICP programmer can also access the Data Flash.

The Data Flash size is software adjustable in N79E845 (16KB) by updating the content of SHBDA. SHBDA[7:0] represents the high byte of 16-bit Data Flash start address and the low byte is hardware set to 00H. The value of SHBDA is loaded from the content of CONFIG1 (CHBDA) after all resets. The application program can dynamically adjust the Data Flash size by resetting SHBDA value. Once the Data Flash size is changed the APROM size is changed accordingly. SHBDA has time access protection while a write to SHBDA is required. The Data Flash size will be 15.75k bytes and there will is 256 bytes APROM.

The Data Flash size is fixed as 4 Kbytes from address 3000H through 3FFFH in N79E844/8432. SHBDA affects nothing.

6.8 Bit-addressable Locations

The Scratch-pad RAM area from location 20h to 2Fh is byte as well as bit-addressable. This means that a bit in this area can be individually addressed. In addition, some of the SFRs are also bit-addressable. The instruction decoder is able to distinguish a bit access from a byte access by the type of the instruction itself. In the SFR area, any existing SFR whose address ends in 0 or 8 is bit-addressable.

6.9 Stack

The scratch-pad RAM can be used for the stack. This area is selected by the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a jump, call or interrupt is invoked the return address is placed on the stack. There is no restriction as to where the stack can begin in the RAM. By default however, the Stack Pointer contains 07h at reset. The user can then change this to any value desired. The SP will point to the last used value. Therefore, the SP will be incremented and then address saved onto the stack. Conversely, while popping from the stack the contents will be read first, and then the SP is decreased.

peripheral-clock time to pull the port pin high quickly. Then it turns off and “weak: pull-up continues remaining the port pin high. The quasi bidirectional port structure is shown below.

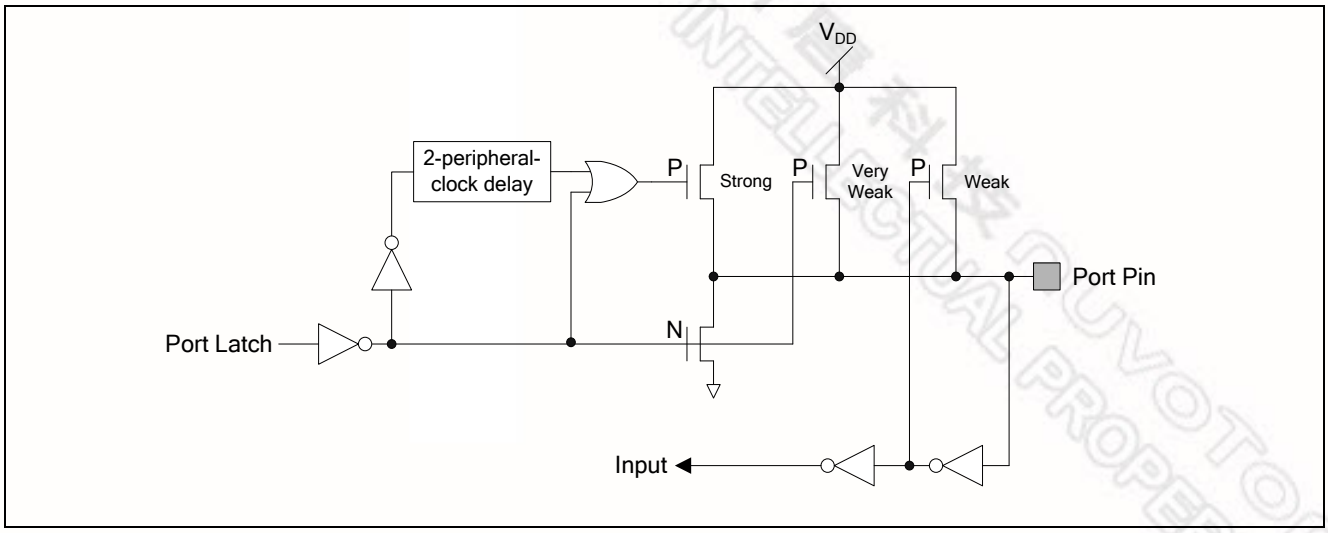

Figure 9–1 Quasi Bi-direction I/O Structure

9.1.1 Read-Modify-Write

In the standard 8051 instruction set, user should watch out for one kind of instructions, read-modify-write instructions. Instead of the normal instructions, the read-modify-write instructions read the internal port latch (Px in SFRs) rather than the external port pin state. This kind of instructions read the port SFR value, modify it and write back to the port SFR. Read-modify-write instructions are listed as follows.

Instruction	Description
ANL	Logical AND. (ANL Px,A and ANL Px,direct)
ORL	Logical OR. (ORL Px,A and ORL Px,direct)
XRL	Logical exclusive OR. (XRL Px,A and XRL Px,direct)
JBC	Jump if bit = 1 and clear it. (JBC Px.y,LABEL)
CPL	Complement bit. (CPL Px.y)
INC	Increment. (INC Px)
DEC	Decrement. (DEC Px)
DJNZ	Decrement and jump if not zero. (DJNZ Px,LABEL)
MOV	Px.y,C Move carry bit to Px.y.
CLR	Px.y Clear bit Px.y.
SETB	Px.y Set bit Px.y.

The last three seems not obviously read-modify-write instructions but actually they are. They read the entire port latch value, modify the changed bit, and then write the new value back to the port latch.

Bit	Name	Description
0	M0	**M1** **M0** **Timer 0 Mode** 0 0 Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL0[4:0]) 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH0 1 1 Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer

TCON – Timer 0 and 1 Control (Bit-addressable)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 88H Reset value: 0000 0000B

Bit	Name	Description
7	TF1	**Timer 1 Overflow Flag** This bit is set when Timer 1 overflows. It is automatically cleared by hardware when the program executes the Timer 1 interrupt service routine. Software can also set or clear this bit.
6	TR1	**Timer 1 Run Control** 0 = Timer 1 is halted. Clearing this bit will halt Timer 1 and the current count will be preserved in TH1 and TL1. 1 = Timer 1 is enabled.
5	TF0	**Timer 0 Overflow Flag** This bit is set when Timer 0 overflows. It is automatically cleared via hardware when the program executes the Timer 0 interrupt service routine. Software can also set or clear this bit.
4	TR0	**Timer 0 Run Control** 0 = Timer 0 is halted. Clearing this bit will halt Timer 0 and the current count will be preserved in TH0 and TL0. 1 = Timer 0 is enabled.

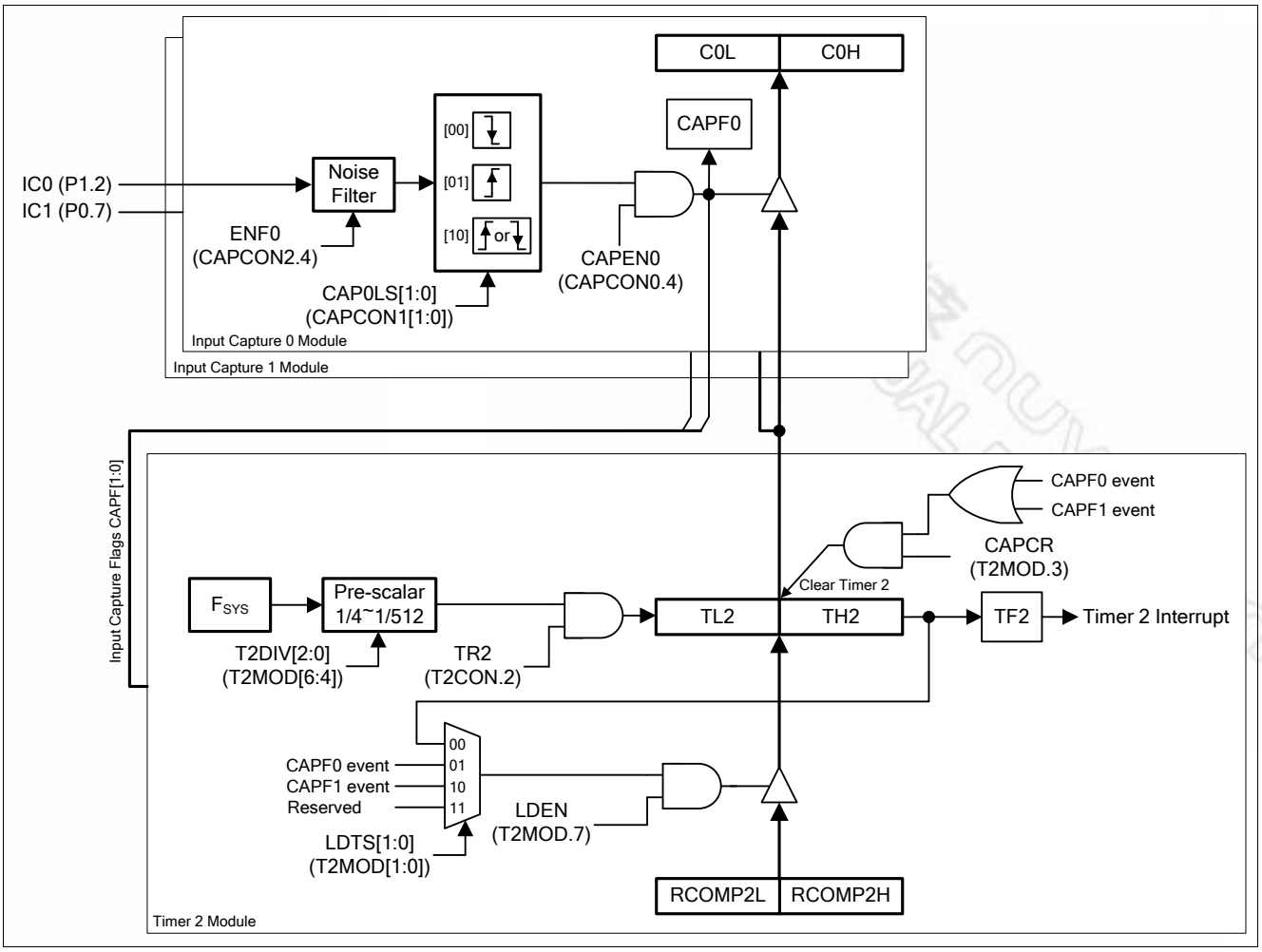

Figure 10–5 Timer 2 Input Capture and Auto-reload Mode Function Block

CAPCON0 – Input Capture Control 0

7	6	5	4	3	2	1	0
-	-	CAPEN1	CAPEN0	-	-	CAPF1	CAPF0
-	-	R/W	R/W	-	-	R/W	R/W

Address: 92H Reset value: 0000 0000B

Bit	Name	Description
7:6	-	**Reserved**
5	CAPEN1	**Input Capture 1 Enable** 0 = Disable input capture channel 1. 1 = Enable input capture channel 1.
4	CAPEN0	**Input Capture 0 Enable** 0 = Disable input capture channel 0. 1 = Enable input capture channel 0.
3:2	-	**Reserved**
1	CAPF1	**Input Capture 1 Flag** This bit is set by hardware if the determined edge of input capture 1 occurs. This bit

EIE – Extensive Interrupt Enable

7	6	5	4	3	2	1	0
ET2	ESPI	EPWM	EWDI	-	ECPTF	EKB	EI2C
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Address: E8H Reset value: 0000 0000B

Bit	Name	Description
4	EWDI	0 = Disable Watchdog Timer Interrupt. 1 = Enable Watchdog Timer Interrupt.

The Watchdog Timer time-out selection will result in different time-out values depending on the clock speed. The reset, when enabled, will occur when 512 clocks after time-out has occurred.

Table 11-1 Time-out Values for the Watchdog Timer

(WPS2,WPS1,WPS0)	Pre-Scalar	WDT Interrupt time-out		Reset time-out	
		Number of Clocks	Time	Number of Clocks	Time
(0,0,0)	1/1	2^6	6.4ms	2^6+512	57.6ms
(0,0,1)	1/2	$2x2^6$	12.8ms	$2x2^6$+512	64ms
(0,1,0)	1/8	$8x2^6$	51.2ms	$8x2^6$+512	102.4ms
(0,1,1)	1/16	$16x2^6$	102.40ms	$16x2^6$+512	153.6ms
(1,0,0)	1/32	$32x2^6$	204.80ms	$32x2^6$+512	256ms
(1,0,1)	1/64	$64x2^6$	409.60ms	$64x2^6$+512	460.8ms
(1,1,0)	1/128	$128x2^6$	819.20ms	$128x2^6$+512	870.4ms
(1,1,1)	1/256	$256x2^6$	1.638s	$256x2^6$+512	1.6892s

11.2 Applications of Watchdog Timer Reset

The main application of the Watchdog Timer with time-out reset enabling is for the 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 erroneous codes and operate in an unpredictable state. If this is left unchecked the entire system may crash. Using the Watchdog Timer during software development will require the user to select ideal Watchdog reset locations for inserting instructions to reset the Watchdog Timer. By inserting the instruction setting WDCLR, it will allow the code to run without any Watchdog Timer reset. However If any erroneous code executes by any power of other interference, the in-

Transmission is initiated by any writing instructions to SBUF. Transmission takes place on TXD pin. First the start bit comes out, the 8-bit data follows to be shifted out and then ends with a stop bit. After the stop bit appears, TI (SCON.1) will be set to indicate one byte transmission complete. All bits are shifted out depending on the rate determined by the baud rate generator.

Once the baud rate generator is activated and REN (SCON.4) is 1, the reception can begin at any time. Reception is initiated by a detected 1-to-0 transition at RXD. Data will be sampled and shifted in at the selected baud rate. In the midst of the stop bit, certain conditions should be met to LOAD SBUF with the received data:

1. RI (SCON.0) = 0, and

2. Either SM2 (SCON.5) = 0, or the received stop bit = 1 while SM2 = 1.

If these conditions are met, the SBUF will be loaded with the received data, the RB8 (SCON.2) with stop bit, and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-0 transition on RXD pin to start next data reception.

12.3 Mode 2

Mode 2 supports asynchronous, full duplex serial communication. Different from Mode1, there are 11 bits to be transmitted or received. They are a start bit (logic 0), 8 data bits (LSB first), a programmable 9^{th} bit TB8 or RB8 bit and a stop bit (logic 1). The most common use of 9^{th} bit is to put the parity bit in it. The baud rate is fixed as 1/32 or 1/64 the clock system frequency depending on SMOD bit. Figure 12–4 shows a simplified functional diagram of the serial port in Mode 2 and associated timings for transmitting and receiving.

12.5 Baud Rates

Table 12–2 UART Baud Rate Formulas

UART Mode	Baud Rate Clock Source	Baud Rate
0	Oscillator	$F_{SYS}/12$ or $F_{SYS}/4$ [1]
2	Oscillator	$\frac{2^{SMOD}}{64} \times F_{SYS}$
1 or 3	Timer/Counter 1 overflow[2]	$\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$ or $\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{4 \times (256 - TH1)}$ [3]

[1] While SM2 (SCON.5) is set as logic 1.
[2] Timer 1 is configured as a timer in auto-reload mode (Mode 2).
[3] While T1M (CKCON.4) is set as logic 1.

Note that in using Timer 1 as the baud rate generator, the interrupt should be disabled. The Timer itself can be configured for either “Timer” or “Counter” operation. And Timer 1 can be in any of its 3 running modes. In the most typical applications, it is configured for “Timer” operation, in the auto-reload mode (Mode2). If Timer 1 is used as the baud rate generator, the reloaded value is stored in TH1. Therefore the baud rate is determined by TH1 value.

Table 12–3 lists various commonly used baud rates and how they can be obtained from Timer 1. In this mode, Timer 1 operates with divided-by-12 pre-scale, as an auto-reload Timer with SMOD (PCON.7) is 0. If SMOD is 1, the baud rate will be doubled.

Table 12–3 Timer 1 Generated Commonly Used Baud Rates

TH1 reload value	Oscillator Frequency (MHz)			
Baud Rate	**11.0592**	**14.7456**	**18.432**	**22.1184**
57600				FFh
38400		FFh		
19200		FEh		FDh
9600	FDh	FCh	FBh	FAh
4800	FAh	F8h	F6h	F4h
2400	F4h	F0h	ECh	E8h
1200	E8h	E0h	D8h	D0h
300	A0h	80h	60h	40h

MOSI pin under the control of SPCLK. Simultaneously, another byte shifts in from the Slave on the MISO pin. After 8-bit data transfer complete, SPIF (SPSR.7) will automatically set via hardware to indicate one byte data transfer complete. At the same time, the data received from the Slave is also transferred in SPDR. The user can clear SPIF and read data out of SPDR.

13.4.2 Slave Mode

When MSTR is 0, the SPI operates in Slave mode. The SPCLK pin becomes input and it will be clocked by another Master SPI device. The $\overline{SS}$ pin be becomes input. The Master device cannot exchange data with the Slave device until the $\overline{SS}$ pin of the Slave device is externally pulled low. Before data transmissions occurs, the $\overline{SS}$ of the Slave device should be pulled and remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state. If the $\overline{SS}$ is force to high at the middle of transmission, the transmission will be aborted and the rest bits of the receiving shifter buffer will be high and goes into idle state.

In Slave mode, data flows from the Master to the Slave on MOSI pin and flows from the Slave to the Master on MISO pin. The data enters the shift register under the control of the SPCLK from the Master device. After one byte is received in the shift register, it is immediately moved into the read data buffer and the SPIF bit is set. A read of the SPDR is actually a read of the read data buffer. To prevent an overrun and the loss of the byte that caused by the overrun, the Slave should read SPDR out and the first SPIF should be cleared before a second transfer of data from the Master device comes in the read data buffer.

13.5 Clock Formats and Data Transfer

To accommodate a wide variety of synchronous serial peripherals, the SPI has a clock polarity bit CPOL (SPCR.3) and a clock phase bit CPHA (SPCR.2). Figure 13–4 SPI Clock Format shows that CPOL and CPHA compose four different clock formats. The CPOL bit denotes the SPCLK line level in ISP idle state. The CPHA bit defines the edge on which the MOSI and MISO lines are sampled. The CPOL and CPHA should be identical for the Master and Slave devices on the same system. Communicating in different data formats with one another will result in undetermined results.

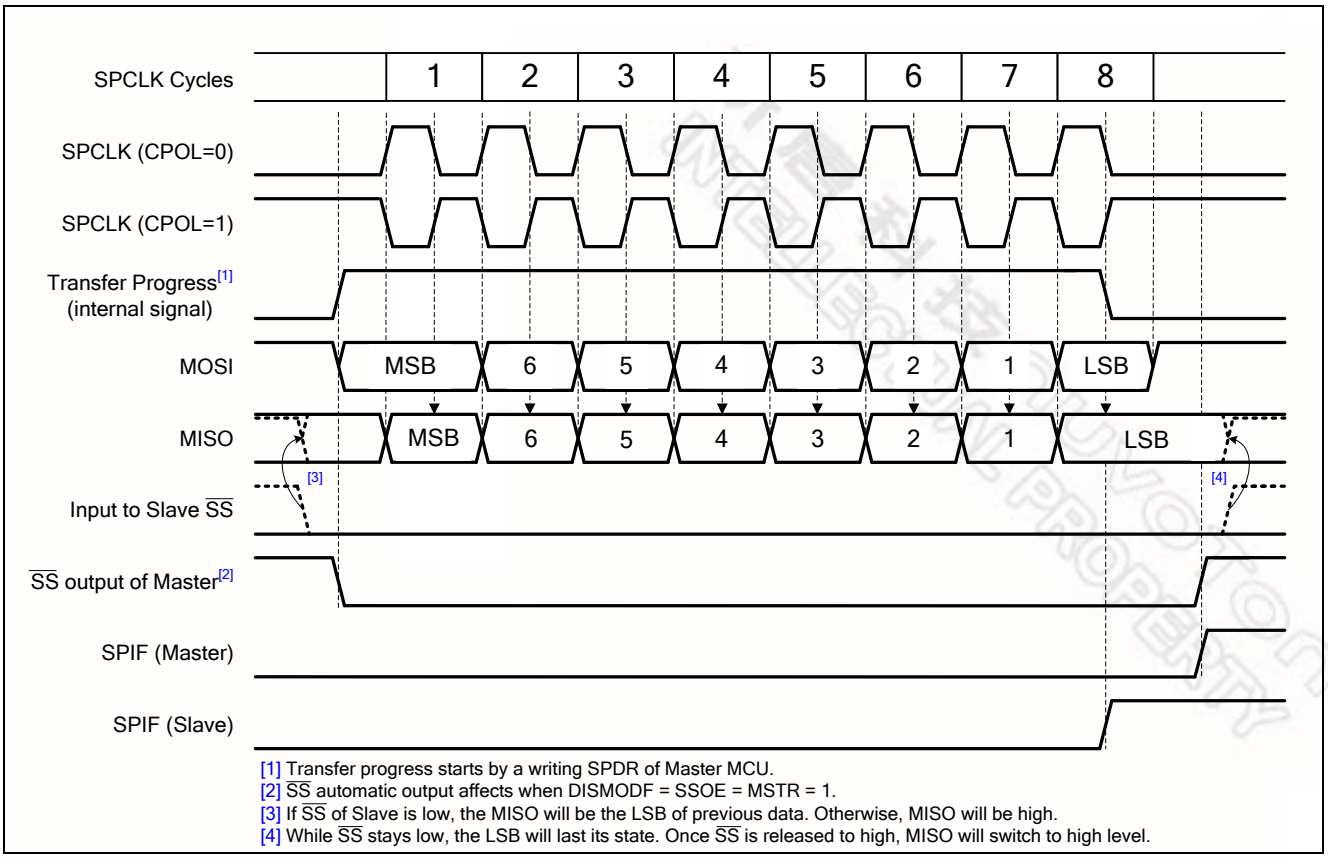

Figure 13–6 SPI Clock and Data Format with CPHA = 1

13.6 Slave Select Pin Configuration

The N79E845/844/8432 SPI provides a flexible $\overline{SS}$ pin feature for different system requirements. When the SPI operates as a Slave, $\overline{SS}$ pin always rules as Slave select input. When the Master mode is enabled, $\overline{SS}$ has three different functions according to DISMODF (SPSR.3) and SSOE (SPCR.7). By default, DISMODF is 0. It means that the Mode Fault detection activates. $\overline{SS}$ is configured as a input pin to check if the Mode Fault appears. On the contrary, if DISMODF is 1, Mode Fault is inactivated and the SSOE bit takes over to control the function of the $\overline{SS}$ pin. While SSOE is 1, it means the Slave select signal will generate automatically to select a Slave device. The $\overline{SS}$ as output pin of the Master usually connects with the $\overline{SS}$ input pin of the Slave device. The $\overline{SS}$ output automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device. While SSOE is 0 and DISMODF is 1, $\overline{SS}$ is no more used by the SPI and reverts to be a general purpose I/O pin.

13.7 Mode Fault Detection

The Mode Fault detection is useful in a system where more than one SPI devices might become Masters at the same time. It may induce data contention. A Mode Fault error occurs once the $\overline{SS}$ is pulled low by others. It indicates that some oth-

15 Analog-To-Digital Converter (ADC)

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 ADCCON0 register. ADCS can be set by software only or by either hardware or software.
Note that when the ADC function is disabled, all ADC related SFR bits will be unavailable and will not effect any other CPU functions. The power of ADC block is approached to zero.

The software only start mode is selected when control bit ADCCON0.5 (ADCEX) =0. A conversion is then started by setting control bit ADCCON0.3 (ADCS) The hardware or software start mode is selected when ADCCON0.5 (ADCEX) =1, and a conversion may be started by setting ADCCON0.3 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 should 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 port 0 is sampled, and this input voltage should be stable to obtain a useful sample. In any event, the input voltage slew rate should be less than 10V/ms 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, 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 than VDAC, 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 demo code of ADC channel 0 with clock source = Fsys/4 is as follows:

```
    ORG     0000H
    LJMP    START

    ORG     005BH           ;ADC Interrupt Service Routine
    CLR     ADCI            ;Clear ADC flag
    reti

START:
    ORL     P0DIDS,#02H     ; Disable digital function for P0.1
    ORL     P0M1,#02H       ; ADC0(P0.1) is input-only mode
    ANL     P0M2,#0FDH
    ANL     ADCCON0,#0F8H    ;ADC0(P0.1) as ADC Channel
    ANL     ADCCON1,#0FDH   ;The FSYS/4 clock is used as ADC clock.
    SETB    EADC            ;Enable ADC Interrupt
    SETB    EA
    ORL     ADCCON1,#80H    ;Enable ADC Function

Convert_LOOP:
    SETB    ADCS            ;Trigger ADC
    ORL     PCON,#01H       ;Enter idle mode
    MOV     P0,ADCH         ;Converted Data put in P0 and P1
    MOV     P1,ADCL
    SJMP    Convert_LOOP

    END
```

19 Interrupt System

The N79E845/844/8432 has four priority level of interrupts structure with 14 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.

19.1 Interrupt Sources

The External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ 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, which 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 timeout count is reached, the Watchdog Timer interrupt flag WDTRF (WDCON0.3) 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, which bits are not automatically cleared by the hardware, and the user will have to clear these bits using software.

I^2C will generate an interrupt due to a new SIO state present in I2STA register, if both EA and ES bits (in IE register) are both enabled.

SPI asserts interrupt flag, SPIF, upon completion of data transfer with an external device. If SPI interrupt is enabled (ESPI at EIE.6), a serial peripheral interrupt is generated. SPIF flag is software clear, by writing 0. MODF and SPIOVF will also generate interrupt if occur. They share the same vector address as SPIF.

The ADC can generate interrupt after finished ADC converter. There is one interrupt source, which is obtained by the ADCI bit in the ADCCON0 SFR. This bit is not automatically cleared by the hardware, and the user will have to clear this bit using software.

PWM brake interrupt flag BKF is generated if P0.2 (Brake pin) detects a high (BKPS=1) or low (BKPS=0) at port pin. At this moment, BKF (PWMCON2.0) is set by hardware and it should be cleared by software. PWM period interrupt flag CF is set by hardware when its' 10-bit down counter underflow and is only cleared by software. BKF is set the PWM interrupt is requested If PWM interrupt is enabled (EPWM=1).

19.2 Priority Level Structure

There are four priority levels for the interrupts, highest, high, low and lowest. The interrupt sources can be individually set to either high or low levels. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a pre-defined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown in Table 19-3, the interrupts are numbered starting from the highest priority to the lowest.

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 include

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 executed.
3. The current instruction does not involve a write to IE, EIE, IP, IPH, EIP 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 interrupt, INT0 and INT1, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. In the case of Timer 2 interrupt, the flags are not cleared by hardware. 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, which address of vector for the different sources are as follows

Bit	Name	Description
2	PX1	1 = Set interrupt priority of External interrupt 1 as higher priority level.
1	PT0	1 = Set interrupt priority of Timer 0 as higher priority level.
0	PX0	1 = Set interrupt priority of External interrupt 0 as higher priority level.

IPH – Interrupt High Priority Register

7	6	5	4	3	2	1	0
PCAPH	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: B7H Reset value: 0000 0000B

Bit	Name	Description
7	PCAPH	1 = Set interrupt high priority of Capture 0/1/2 as highest priority level.
6	PADCH	1 = Set interrupt high priority of ADC as the highest priority level.
5	PBODH	1 = Set interrupt high priority of BOD Detector as the highest priority level.
4	PSH	1 = Set interrupt high priority of Serial port 0 as the highest priority level.
3	PT1H	1 = Ro set interrupt high priority of Timer 1 as the highest priority level.
2	PX1H	1 = Set interrupt high priority of External interrupt 1 as the highest priority level.
1	PT0H	1 = Set interrupt high priority of Timer 0 as the highest priority level.
0	PX0H	1 = Set interrupt high priority of External interrupt 0 as the highest priority level.

EIP – Interrupt Priority-1 Register

7	6	5	4	3	2	1	0
PT2	PSPI	PPWM	PWDI	-	-	PKB	PI2
R/W	R/W	R/W	R/W	-	-	R/W	R/W

Address: FFH Reset value: 0000 0000B

Bit	Name	Description
7	PT2	1 = Set interrupt priority of Timer 2 as higher priority level.
6	PSPI	1 = Set interrupt priority of SPI as higher priority level.
5	PPWM	1 = Set interrupt priority of PWM's brake as higher priority level.
4	PWDI	1 = Set interrupt priority of Watchdog as higher priority level.
3:2	-	Reserved

```
MOV    ISPCN,#00001100b    ;select “Read Device ID” mode
MOV    ISPAH,#00H          ;fill address with 0000H for low-byte DID
MOV    ISPAL,#00H          ;
CALL   Trigger_ISP
MOV    A,ISPFD             ;now, ISPFD contains low-byte DID, move to ACC for further use
MOV    ISPAH,#00H          ;fill address with 0001H for high-byte DID
MOV    ISPAL,#01H          ;
CALL   Trigger_ISP
MOV    A,ISPFD             ;now, ISPFD contains high-byte DID, move to ACC for further use
CALL   Disable_ISP
```

FLASH Page Erase (target address in APROM/Data Flash/LDROM area)

```
CALL   Enable_ISP
MOV    ISPCN,#00100010b    ;select “FLASH Page Erase” mode, (A17,A16)=(0,0) for APROM/Data
                           ;Flash/LDROM
MOV    ISPAH,#??H          ;fill page address
MOV    ISPAL,#??H
CALL   Trigger_ISP
CALL   Disable_ISP
```

FLASH Program (target address in APROM/Data Flash/LDROM area)

```
CALL   Enable_ISP
MOV    ISPCN,#00100001b    ;select “FLASH Program” mode, (A17,A16)=(0,0) for APROM/Data
                           ;Flash/LDROM
MOV    ISPAH,#??H          ;fill byte address
MOV    ISPAL,#??H
MOV    ISPFD,#??H          ;fill data to be programmed
CALL   Trigger_ISP
CALL   Disable_ISP
```

FLASH Read (target address in APROM/Data Flash/LDROM area)

```
CALL   Enable_ISP
MOV    ISPCN,#00000000b    ;select “FLASH Read” mode, (A17,A16)=(0,0) for APROM/Data
                           ;Flash/LDROM
MOV    ISPAH,#??H          ;fill byte address
MOV    ISPAL,#??H
CALL   Trigger_ISP
MOV    A,ISPFD             ;now, ISPFD contains the Flash data, move to ACC for further use
CALL   Disable_ISP
```

CONFIG Page Erase (target address in CONFIG area)

```
CALL   Enable_ISP
MOV    ISPCN,#11100010b    ;select “CONFIG Page Erase” mode, (A17,A16)=(1,1) for CONFIG
MOV    ISPAH,#00H          ;fill page address #0000H, because there is only one page
MOV    ISPAL,#00H
CALL   Trigger_ISP
CALL   Disable_ISP
```

CONFIG Program (target address in CONFIG area)

```
CALL   Enable_ISP
MOV    ISPCN,#11100001b    ;select “CONFIG Program” mode, (A17,A16)=(1,1) for CONFIG
MOV    ISPAH,#00H          ;fill byte address, 0000H/0001H/0002H/0003H for CONFIG0/1/2/3,
                           ;respectively
```

striction: The KBI pin keeps low (high) before CPU enters Power Down. Then only rising (falling) edge of KBI Interrupt can wake up CPU from Power Down.

25.4 CONFIG3

7	6	5	4	3	2	1	0
CWDTEN	-	-	CKF	OSCFS	-	FOSC1	FOSC0
R/W	-	-	R/W	R/W	-	R/W	R/W

Unprogrammed value: 1111 1111B

Bit	Name	Description
7	CWDTEN	**CONFIG Watchdog Timer Enable** 1 = Disable Watchdog Timer after all resets. 0 = Enable Watchdog Timer after all resets. WDTEN is initialized by inverted CWDTEN (CONFIG3, bit-7) at any other resets.
6	-	**Reserved**
5	-	**Reserved**
4	CKF	**Clock Filter Enable** 1 = Enable clock filter. It increases noise immunity and EMC capacity. 0 = Disable clock filter.
3	OSCFS	**Internal RC Oscillator Frequency Selection** 1 = Select 22.1184 MHz as the clock system if internal RC oscillator mode is used. It bypasses the divided-by-2 path of internal oscillator to select 22.1184MHz output as the clock system source. 0 = Select 11.0592 MHz as the clock system if internal RC oscillator mode is used. The internal RC divided-by-2 path is selected. The internal oscillator is equivalent to 11.0592 MHz output used as the clock system.
2	-	**Reserved**
1	FOSC1	**Oscillator Select Bit**
0	FOSC0	**Chip Clock Source Selection (See the Following Table)**

(FOSC1, FOSC0)	Chip Clock Source
(1, 1)	Internal RC oscillator
(1, 0)	Reserved
(0, 0) (0, 1)	External crystal, 4 MHz ~ 24 MHz