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, KPI, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOP
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n79e715as28

Table of Contents

4 Block Diagram

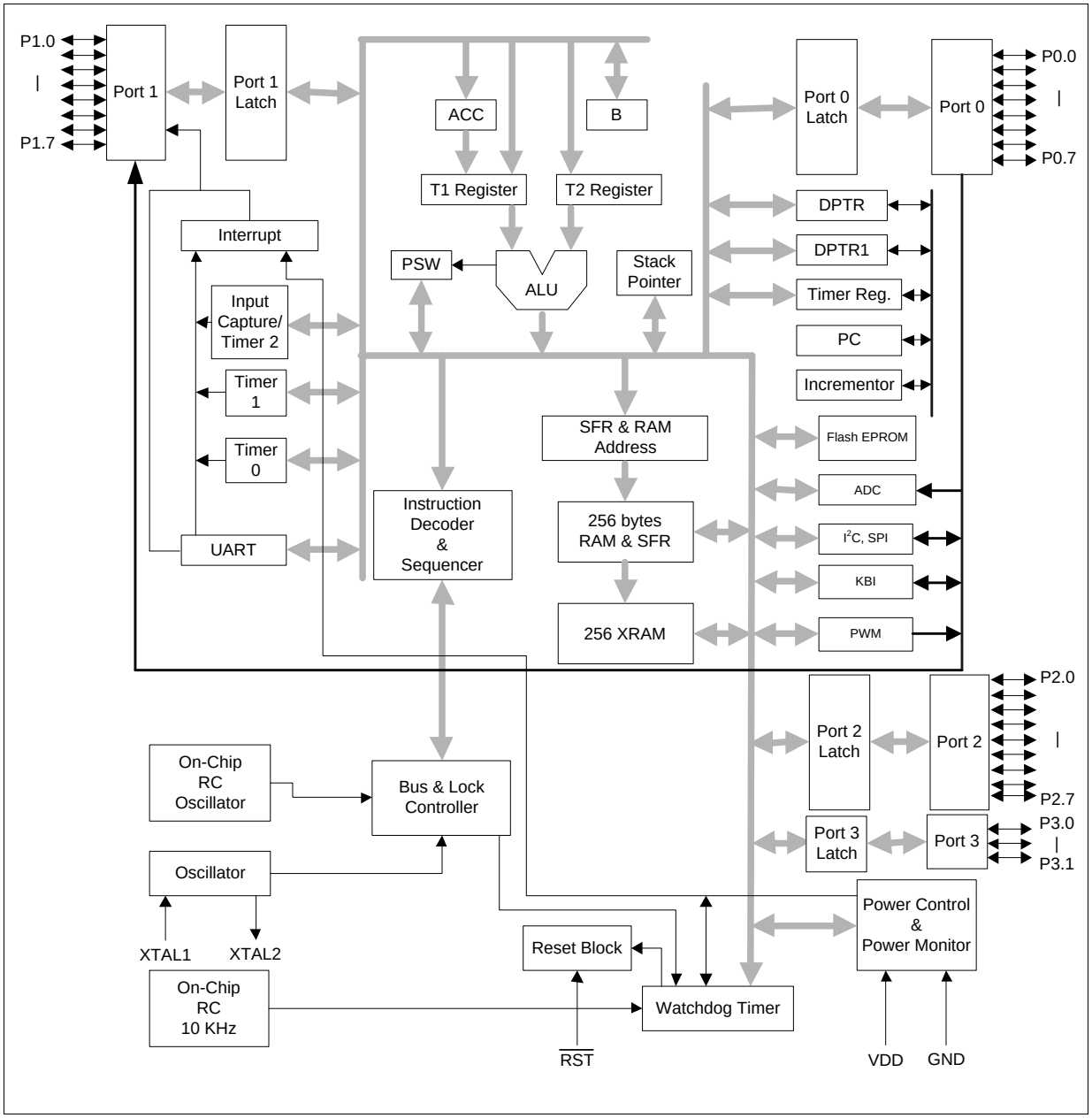


Figure 4-1 N79E715 Function Block Diagram

6.1 APROM Flash Memory

The N79E715 has **16 Kbytes** on-chip 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

The N79E715 has 2 Kbytes LDROM. User may develop the ISP function in LDROM for updating application program or Data Flash. Similarly, APROM can also re-program 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 N79E715 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 N79E715 (16 KB) 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.

Bit	Name	Description
7:0	SHBDA[7:0]	**SFR high byte of Data Flash starting address** This byte is valid only when DFEN (CONFIG0.0) being 0 condition. It is used to dynamic adjust the starting address of the Data Flash when the application program is executing.

[1] SHBDA is loaded from CONFIG1 after all resets.

6.5 On-chip XRAM

The N79E715 provides additional on-chip 256 bytes auxiliary RAM called XRAM to enlarge the RAM space. It occupies the address space from 00H through FFH. The 256 bytes of XRAM are indirectly accessed by move external instruction MOVX @DPTR or MOVX @Ri. (See the demo code below.) Note that the stack pointer may not be located in any part of XRAM. Figure 6-1 shows the memory map for this product series.

XRAM demo code:

```
MOV     R0,#23H             ;write #5AH to XRAM with address @23H
MOV     A,#5AH
MOVX    @R0,A

MOV     R1,#23H             ;read from XRAM with address @23H
MOVX    A,@R1

MOV     DPTR,#0023H         ;write #5BH to XRAM with address @0023H
MOV     A,#5BH
MOVX    @DPTR,A

MOV     DPTR,#0023H         ;read from XRAM with address @0023H
MOVX    A,@DPTR
```

6.6 On-chip scratch-pad RAM and SFR

The N79E715 provides the on-chip 256 bytes scratch pad RAM and Special Function Registers (SFRs) which are accessed by software. The SFRs are accessed only by direct addressing, while the on-chip RAM is accessed by either direct or indirect addressing.

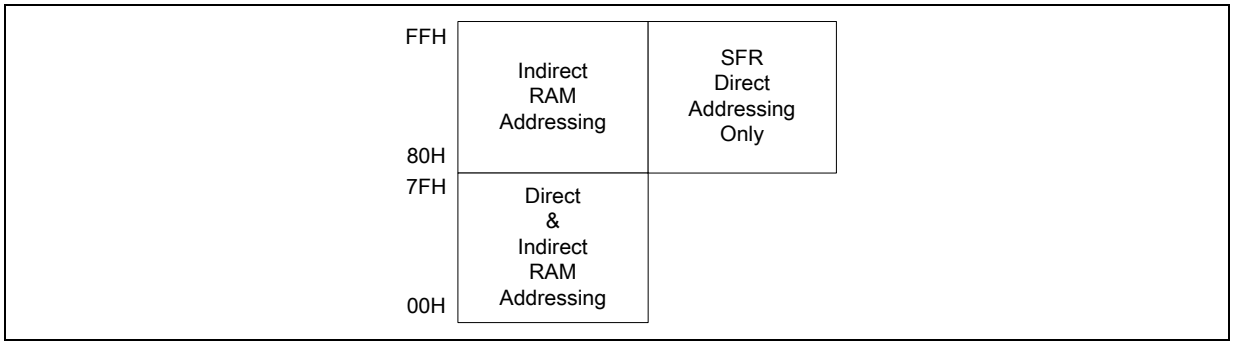


Figure 6-3 256 bytes RAM and SFR

Table 7–2 N79E715 SFR Description and Reset Values

Symbol	Definition	Address	MSB							LSB	Reset Value[1]
PSW	Program status word	D0H	(D7) CY	(D6) AC	(D5) F0	(D4) RS1	(D3) RS0	(D2) OV	(D1) F1	(D0) P	0000 0000B
TH2	Timer 2 MSB	CDH	TH2[7:0]								0000 0000B
TL2	Timer 2 LSB	CCH	TL2[7:0]								0000 0000B
RCOMP2H	Timer 2 Reload MSB	CBH	RCOMP2H[7:0]								0000 0000B
RCOMP2L	Timer 2 Reload LSB	CAH	RCOMPL2[7:0]								0000 0000B
T2MOD	Timer 2 Mode	C9H	LDEN	T2DIV[2:0]			CAPCR	COMPCR	LDTS[1:0]		0000 0000B
T2CON	Timer 2 Control	C8H	(CF) TF2	-	-	-	-	(CA) TR2	-	(C8) CP/$\overline{RL2}$	0000 0000B
TA	Timed Access Protection	C7H									1111 1111B
I2ADDR	I2C address	C1H	ADDR[7:1]							GC	0000 0000B
I2CON	I2C Control register	C0H	(C7) -	(C6) I2CEN	(C5) STA	(C4) STO	(C3) SI	(C2) AA	(C1) -	(C0) -	0000 0000B
I2TOC	I2C Time-out Counter register	BFH	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000B
I2CLK	I2C Clock Rate	BEH	I2CLK[7:0]								0000 0000B
I2STA	I2C Status Register	BDH	I2STA[7:3]					0	0	0	1111 1000B
I2DAT	I2C Data Register	BCH	I2DAT[7:0]								0000 0000B
SADEN	Slave address mask	B9H	SADEN[7:0]								0000 0000B
IP	Interrupt priority	B8H	(BF) PCAP	(BE) PADC	(BD) PBOD	(BC) PS	(BB) PT1	(BA) PX1	(B9) PT0	(B8) PX0	0000 0000B
IPH	Interrupt high priority	B7H	PCAPH	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000B
P2M2	Port 2 output mode 2	B6H	P2M2[7:0]								0000 0000B
P2M1	Port 2 output mode 1	B5H	P2M1[7:0]								0000 0000B
P1M2	Port 1 output mode 2	B4H	P1M2.7	P1M2.6	-	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	0000 0000B
P1M1	Port 1 output mode 1	B3H	P1M1.7	P1M1.6	-	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	0000 0000B
P0M2	Port 0 output mode 2	B2H	P0M2[7:0]								0000 0000B
P0M1	Port 0 output mode 1	B1H	P0M1[7:0]								0000 0000b
P3	Port3	B0H	-	-	-	-	-	-	(B1) X1	(B0) X2 CLKOUT	0000 0011B
ISPCN	ISP Control Register	AFH	ISPA17	ISPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0011 0000B
ISPFD	ISP Flash Data Register	AEH	ISPFD[7:0]								0000 0000B
WDCON1[4]	Watch-Dog control1	ABH	-	-	-	-	-	-	-	EWRST	0000 0000B
SADDR	Slave address	A9H	SADDR[7:0]								00000000B
IE	Interrupt enable	A8H	(AF) EA	(AE) EADC	(AD) EBOD	(AC) ES	(AB) ET1	(AA) EX1	(A9) ET0	(A8) EX0	0000 0000B
ISPAH	ISP Flash Address High-byte	A7H	ISPAH[7:0]								0000 0000B
ISPAL	ISP Flash Address Low-byte	A6H	ISPAL[7:0]								0000 0000B
ISPTRG[4]	ISP Trigger Register	A4H	-	-	-	-	-	-	-	ISPGO	0000 0000B
PMCR[2][4]	Power Monitor Control Register	A3H	BODEN	BOV	-	BORST	BOF	-	-	BOS	Power-on CC0C 100XB BOR reset UU0U 100XB Other reset UU0U 000XB
AUXR1	AUX function register	A2H	SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS	0000 0000B
P2	Port 2	A0H	(97)	(96)	(95)	(94)	(93)	(92)	(91)	(90)	1111 1111B

Instruction	CY	OV	AC	Instruction	CY	OV	AC
RLC A	X			CJNE	X		
SETB C	1						

[1] X indicates the modification is dependent on the result of the instruction

PCON – Power Control

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Address: 87H Reset value: see Table 7–2 N79E715 SFR Description and Reset Values

Bit	Name	Description
3	GF1	**General Purpose Flag 1** The general purpose flag that can be set or cleared by the user.
2	GF0	**General Purpose Flag 0** The general purpose flag that can be set or cleared by the user.

General 80C51 support one DPTR but the N79E715 support two DPTRs by switching AUXR1.DPS. The setting is as follows.

AUXR1 – AUX Function Resgister-1

7	6	5	4	3	2	1	0
SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS
R/W	R/W	-	-	R/W	-	R	R/W

Address: A2H Reset value: 0000 0000B

Bit	Name	Description
0	DPS	**Dual Data Pointer Selection** 0 = Select DPTR of standard 8051. 1 = Select DPTR1

10 Timers/Counters

The N79E715 has three 16-bit programmable timers/counters.

10.1 Timers/Counters 0 and 1

Timer/Counter 0 and 1 in N79E715 are two 16-bit Timers/Counters. Each of them has two 8-bit registers that form the 16-bit counting register. For Timer/Counter 0 they are TH0, the upper 8-bit register, and TL0, the lower 8-bit register. Similar Timer/Counter 1 has two 8-bit registers, TH1 and TL1. TCON and TMOD can configure modes of Timer/Counter 0 and 1.

They have additional timer 0 or timer 1 overflow toggle output enable feature as compare to conventional timers/counters. This timer overflow toggle output can be configured to automatically toggle T0 or T1 pin output whenever a timer overflow occurs.

When configured as a “Timer”, the timer counts clock cycles. The timer clock can be programmed to be thought of as 1/12 of the clock system or 1/4 of the clock system. In the “Counter” mode, the register is incremented on the falling edge of the external input pin, T0 in case of Timer 0, and T1 for Timer 1. The T0 and T1 inputs are sampled in every machine-cycle at C4. If the sampled value is high in one machine-cycle and low in the next, then a valid high to low transition on the pin is recognized and the count register is incremented. Since it takes two machine-cycles to recognize a negative transition on the pin, the maximum rate at which counting will take place is 1/24 of the master clock frequency. In either the “Timer” or “Counter” mode, the count register will be updated at C3. Therefore, in the “Timer” mode, the recognized negative transition on pin T0 and T1 can cause the count register value to be updated only in the machine-cycle following the one in which the negative edge was detected.

The “Timer” or “Counter” function is selected by the “ $C/\overline{T}$ ” bit in the TMOD Special Function Register. Each Timer/Counter has one selection bit for its own; bit 2 of TMOD selects the function for Timer/Counter 0 and bit 6 of TMOD selects the function for Timer/Counter 1. In addition each Timer/Counter can be set to operate in any one of four possible modes. The mode selection is done by bits M0 and M1 in the TMOD SFR.

The N79E715 can operate like the standard 8051/52 family, counting at the rate of 1/12 of the clock speed, or in turbo mode, counting at the rate of 1/4 clock speed. The speed is controlled by the T0M and T1M bits in CKCON, and the default value is zero, which uses the standard 8051/52 speed.

Bit	Name	Description
2	C/$\overline{T}$	**Timer 0 Counter/Timer Selection** 0 = Timer 0 is incremented by internal peripheral clocks. 1 = Timer 0 is incremented by the falling edge of the external pin T0.
1	M1	**Timer 0 Mode Selection**
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.

10.1.2 Mode 1 (16-bit Timer)

Mode 1 is similar to Mode 0 except that the counting registers are fully used as a 16-bit counter. Rollover occurs when a count moves FFFFH to 0000H. The Timer overflow flag TFx of the relevant Timer/Counter is set and an interrupt will occurs if enabled.

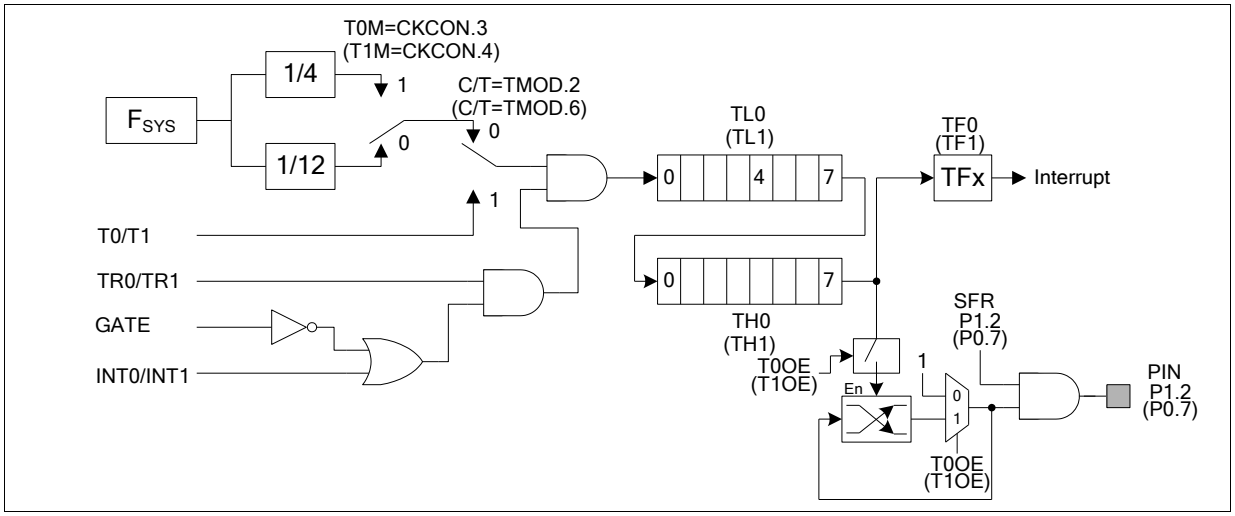
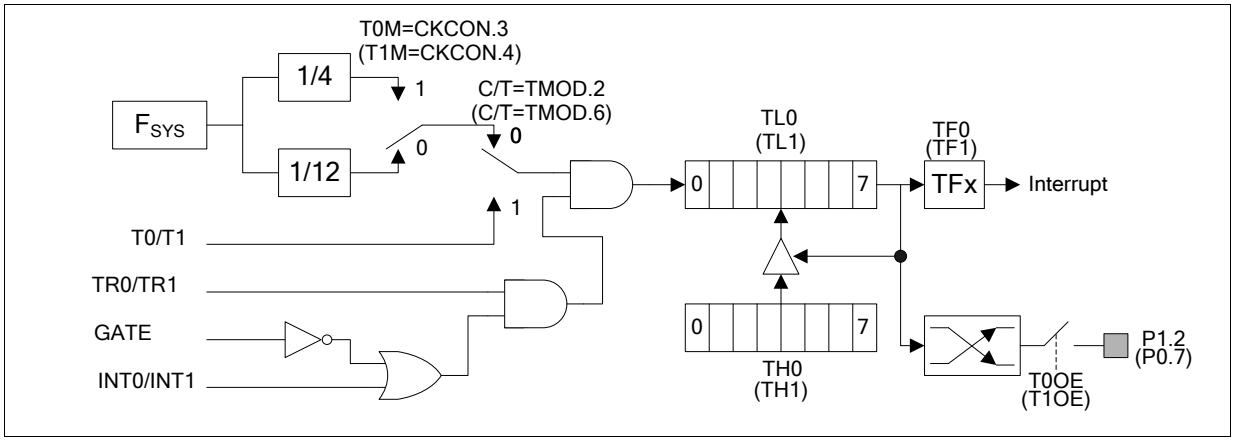


Figure 10-2 Timers/Counters 0 and 1 in Mode 1

10.1.3 Mode 2 (8-bit Auto-Reload Timer)

In Mode 2, the Timer/Counter is in auto-reload mode. In this mode, TLx acts as an 8-bit count register whereas 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. This feature is best suitable for UART baud rate generator for it runs without continuous software intervention. Note that only Timer1 can be the baud rate source for UART. Counting is enabled by the TRx bit and proper setting of GATE and $\overline{INTx}$ pins. The functions of GATE and $\overline{INTx}$ pins are just the same as Mode 0 and 1.



While any input capture channel is enabled and the selected edge trigger occurs, the content of the free running Timer 2 counter, TH2 and TL2, will be captured, transferred, and stores into the capture registers CnH and CnL. The edge triggering also causes CAPFn (CAPCON0.n) is set by hardware. The interrupt will also be generated if ECPTF (EIE.2) and EA bit are both set. For three input capture flags shares the same interrupt vector, the user should check CAPFn to confirm which channel comes the input capture edge. These flags should be cleared by software.

The bit CAPCR (T2MOD.3) benefits the implement of period calculation. Setting CAPCR makes the hardware clear Timer 2 as 0000H automatically after the value of TH2 and TL2 have been captured after an input capture edge event occurs. It eliminates the routine software overhead of writing 16-bit counter or an arithmetic subtraction.

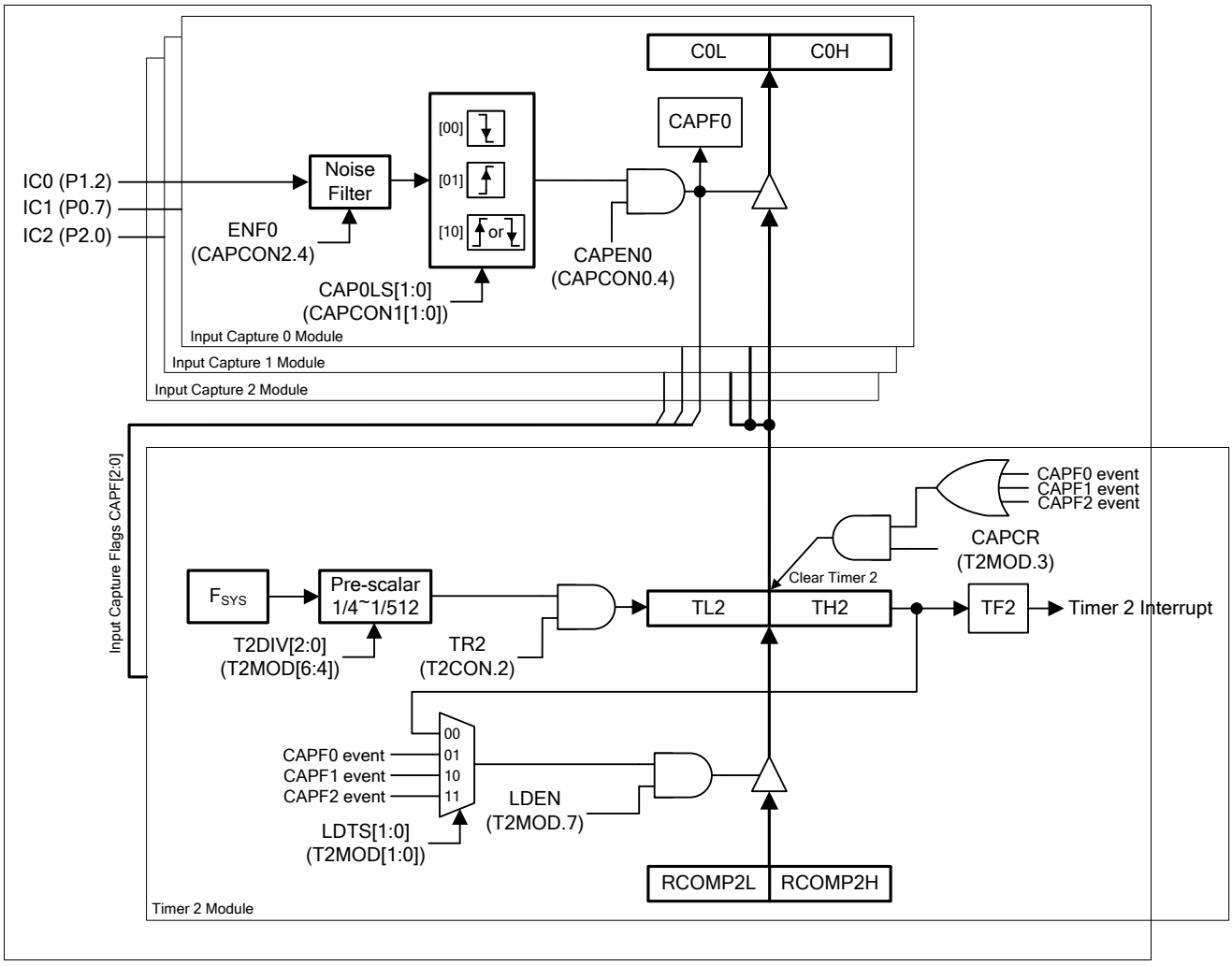


Figure 10-5 Timer 2 Input Capture and Auto-reload Mode Function Block

without any Watchdog Timer reset. However If any erroneous code executes by any power of other interference, the instructions to clear the Watchdog Timer counter will not be executed at the required instants. Thus the Watchdog Timer reset will occur to reset the system start from an erroneously executing condition. The user should remember that WDCON0 requires a timed access writing.

11.3 Applications of Watchdog Timer Interrupt

There is another application of the Watchdog Timer, which is used as a simple timer. The WDTF flag will be set while the Watchdog Timer completes the selected time interval. The software polls the WDTF flag 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. Every time the time-out occurs, an interrupt will occur if the individual interrupt EWDI (EIE.4) and global interrupt enable EA is set.

In some application of low power consumption, the CPU usually stays in Idle mode when nothing needs to be served to save power consumption. After a while the CPU will be woken up to check if anything needs to be served at an interval of programmed period implemented by Timer 0, 1 or 2. However, the current consumption of Idle mode still keeps at a “mA” level. To further reducing the current consumption to “μA” level, the CPU should stay in Power-down mode when nothing needs to be served, and has the ability of waking up at a programmable interval. The N79E715 is equipped with this useful function. It provides a very low power LIRC. Along with the low power consumption application, the Watchdog Timer needs to count under Idle and Power-down mode and wake CPU up from Idle or Power-down mode. The demo code to accomplish this feature is shown below.

The demo code of Watchdog Timer wakes CPU up from Power Down.

```
    ORG     0000H
    LJMP    START

    ORG     0053H
    LJMP    WDT_ISR

    ORG     0100H
WDT_ISR:
    CLR     EA
    MOV     TA,#0AAH
    MOV     TA,#55H
    ORL     WDCON0,#01000000B       ;clear Watchdog Timer counter
    INC     ACC
    MOV     P0,ACC
    SETB    EA

    CLR     EA
    MOV     TA,#0AAH
    MOV     TA,#55H
    ANL     WDCON0,#11011111B       ;clear Watchdog Timer interrupt flag
    SETB    EA
    RETI
```

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.

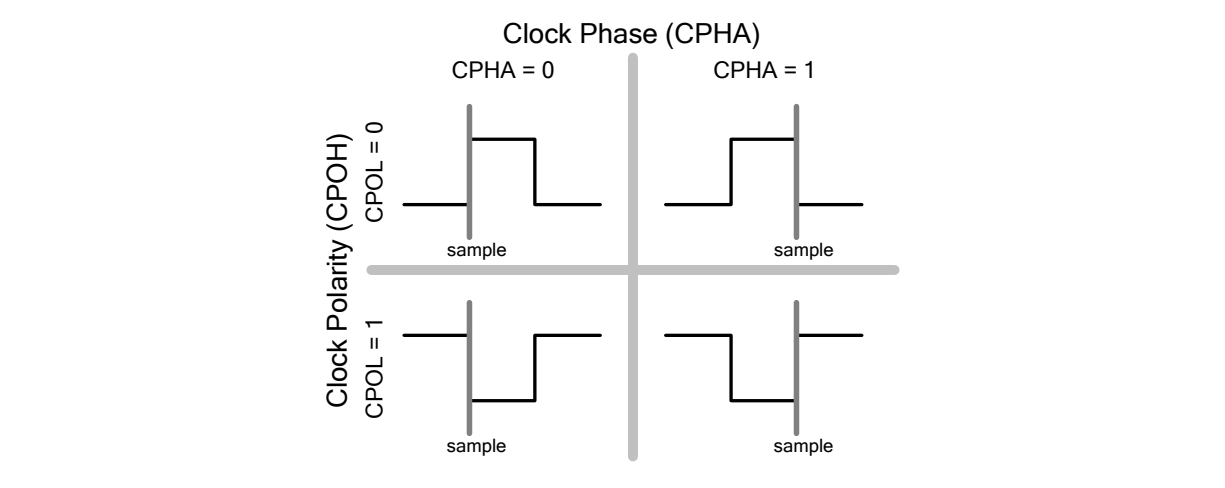


Figure 13-4 SPI Clock Format

In SPI, a Master device always initiates the transfer. If SPI is selected as Master mode (MSTR = 1) and enabled (SPIEN = 1), writing to the SPI data register (SPDR) by the Master device starts the SPI clock and data transfer. After shifting one byte out and receiving one byte in, the SPI clock stops and SPIF (SPSR.7) in both Master and Slave are set. If SPI interrupt enable bit ESPI (EIE.6) is set 1 and global interrupt is enabled (EA = 1), the interrupt service routine (ISR) of SPI will be executed.

Concerning the Slave mode, the $\overline{SS}$ signal needs to be taken care. As shown in Figure 13-4 SPI Clock Format, when CPHA = 0, the first SPCLK edge is the sampling strobe of MSB (for an example of LSBFE = 0, MSB first). Therefore, the Slave should shift its MSB data before the first SPCLK edge. The falling edge of $\overline{SS}$ is used for preparing the MSB on MISO line. The $\overline{SS}$ pin therefore should toggle high and then low between each successive serial byte. Furthermore, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error occurs.

When CPHA = 1, the sampling edge thus locates on the second edge of SPCLK clock. The Slave uses the first SPCLK clock to shift MSB out rather than the $\overline{SS}$ falling edge. Therefore, the $\overline{SS}$ line can remain low between successive transfers. This format may be preferred in systems having single fixed Master and single fixed Slave. The $\overline{SS}$ line of the unique Slave device can be tied to V_{SS} as long as only CPHA = 1 clock mode is used.

Note: The SPI should be configured before it is enabled (SPIEN = 1), or a change of LSBFE, MSTR, CPOL, CPHA and SPR[1:0] will abort a transmission in progress and force the SPI system into idle state. Prior to any configuration bit changed, SPIEN should be disabled first.

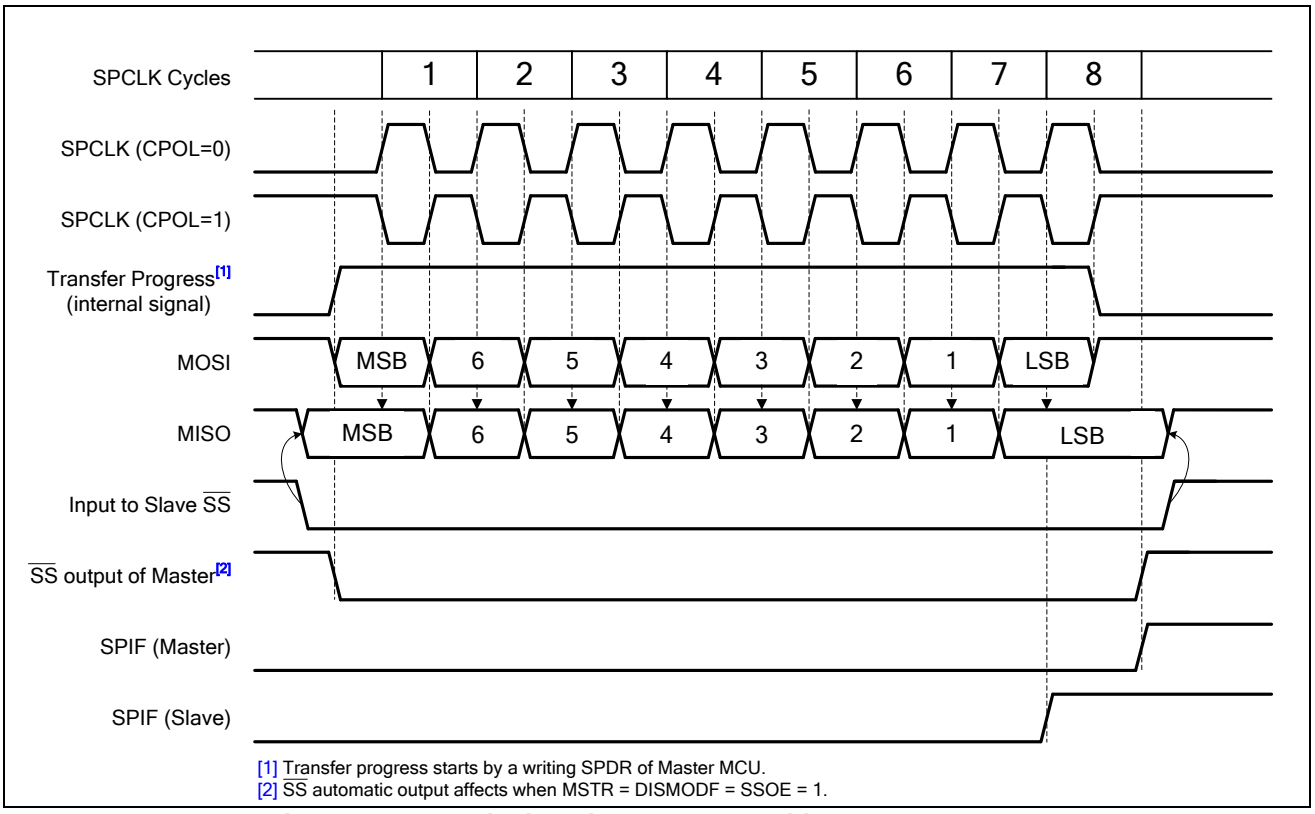


Figure 13-5 SPI Clock and Data Format with CPHA = 0

and the data direction bit “write” (SLA+W). The SI bit should then be cleared to commence SLA+W transaction.

After the SLA+W byte has been transmitted and an acknowledge (ACK) has been returned by the addressed slave device, the SI flag is set again and I2STA is read as 18H. The appropriate action to be taken follows the user defined communication protocol by sending data continuously. After all data is transmitted, the master can send a STOP condition by setting STO (I2CON.4) and then clearing SI to terminate the transmission. A repeated START condition can also be generated without sending STOP condition to immediately initial another transmission.

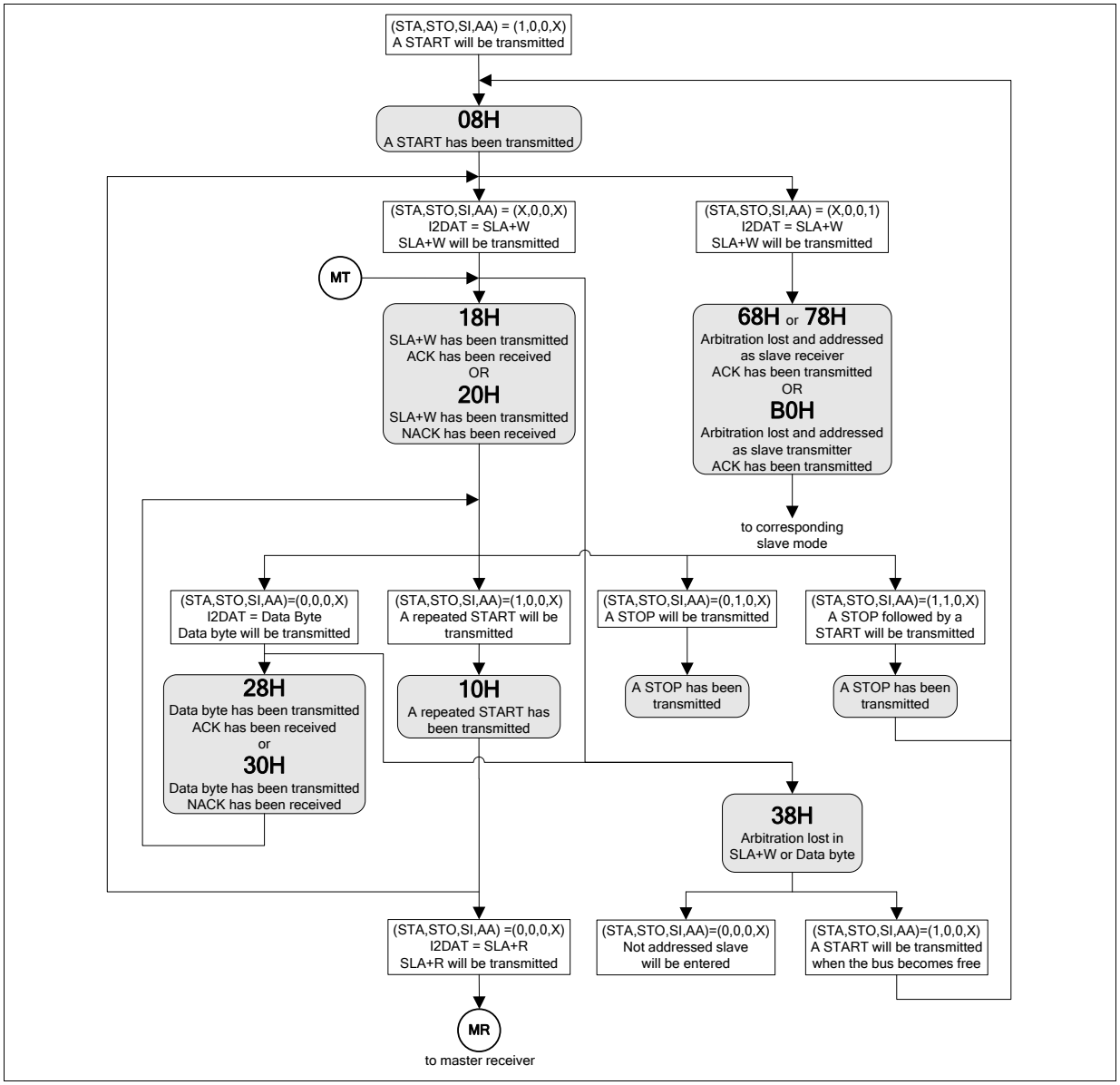


Figure 16-7 Flow and Status of Master Transmitter Mode

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. These address of vector for the different sources are as follows.

Bit	Name	Description
2	IT1	**External Interrupt 1 Type Selection** This bit selects whether the $\overline{INT1}$ pin will detect falling edge or low level triggered interrupts. 0 = $\overline{INT1}$ is low level triggered. 1 = $\overline{INT1}$ is falling edge triggered.
1	IE0	**External Interrupt 0 Edge Flag** This flag is set via hardware when an edge/level of type defined by IT0 is detected. If IT0 = 1, this bit will remain set until cleared via software or at the beginning of the External Interrupt 0 service routine. If IT0 = 0, this flag is the inverse of the $\overline{INT0}$ input signal's logic level.
0	IT0	**External Interrupt 0 Type Selection** This bit selects whether the $\overline{INT0}$ pin will detect falling edge or low level triggered interrupts. 0 = $\overline{INT0}$ is low level triggered. 1 = $\overline{INT0}$ is falling edge triggered.

CHPCON – Chip Control (TA Protected)

7	6	5	4	3	2	1	0
SWRST	ISPF	LDUEN	-	-	-	BS	ISPEN
W	R/W	R/W	-	-	-	R/W	R/W

Address: 9FH Reset value: see Table 7–2 N79E715 SFR Description and Reset Values

Bit	Name	Description
6	ISPF	**ISP Fault Flag** The hardware will set this bit when any of the following condition is met: 1. The accessing area is illegal, such as, (a) Erasing or programming APROM itself when APROM code runs. (b) Erasing or programming LDROM when APROM code runs but LDUEN is 0. (c) Erasing, programming, or reading CONFIG bytes when APROM code runs. (d) Erasing or programming LDROM itself when LDROM code runs. (e) Accessing oversize. 2. The ISP operating runs from internal Program Memory to external one. This bit should be cleared via software.
5	LDUEN	**Updating LDROM Enable** 0 = LDROM is inhibited to be erased or programmed when APROM code runs. LDROM remains read-only. 1 = LDROM is allowed to be fully accessed when APROM code runs.
4:2	-	**Reserved**
1	BS	**Boot Selection** There are different meanings of writing to or reading from this bit. Writing It defines from which block MCU boots after all resets. 0 = The next rebooting will be from APROM. 1 = The next rebooting will be from LDROM. Reading It indicates from which block MCU booted after previous reset. 0 = The previous rebooting is from APROM. 1 = The previous rebooting is from LDROM.
0	ISPEN	**ISP Enable**

21 Power Management

The N79E715 has several features that help the user to control the power consumption of the device. The power saved features have Power-down mode and Idle mode operations. For a stable current consumption, user should watch out for the states of P0 pins.

In system power saving modes, user should specifically watch out for the Watchdog Timer. The hardware will clear WDT counter automatically after entering or being woken-up from Idle or Power-down mode. It prevents unconscious system reset.

PCON – Power Control

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Address: 87H Reset value: see Table 7–2 N79E715 SFR Description and Reset Values

Bit	Name	Description
1	PD	**Power-down Mode** Setting this bit puts MCU into Power-down mode. Under this mode, both CPU and peripheral clocks stop and Program Counter (PC) suspends. It provides the lowest power consumption. After CPU is woken up from Power Down, this bit will be automatically cleared via hardware and the program continue executing the interrupt service routine (ISR) of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction which follows the instruction that put the system into Power-down mode. **Note:** If IDL bit and PD bit are set simultaneously, the MCU will enter Power-down mode. Then it does not go to Idle mode after exiting Power Down.
0	IDL	**Idle Mode** Setting this bit puts MCU into Idle mode. Under this mode, the CPU clock stops and Program Counter (PC) suspends. After CPU is woken up from Idle, this bit will be automatically cleared via hardware and the program continue executing the ISR of the very interrupt source that woke the system up before. After return from the ISR, the device continues execution at the instruction which follows the instruction that put the system into Idle mode.

21.1 Idle Mode

Idle mode suspends CPU processing by holding the Program Counter. No program code are fetched and run in Idle mode. This forces the CPU state to be frozen. The Program Counter (PC), the Stack Pointer (SP), the Program Status Word (PSW), the Accumulator (ACC), and the other registers hold

User may refer to the following website for ICP Program Tool. Entry the web site, please select "Nuvoton ISP-ICP Programmer".

http://www.nuvoton.com/hq/products/microcontrollers/8bit-8051-mcus/Software

Figure 27–2 Nuvoton ISP-ICP Programmer