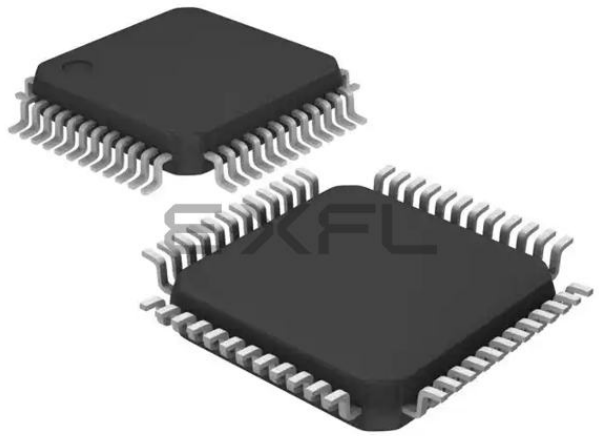

Welcome to E-XFL.COM

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	40MHz
Connectivity	EBI/EMI, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	40
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n78e055alg

Table 4–1. Pin Description

Pin number				Symbol	Alternate Function		Type[1]	Description
DIP	PLCC	PQFP	LQFP		1	2		
								an external capacitor to V_{DD}.
39	43	37	40	P0.0		AD0	D, I/O	**PORT0:** Port 0 is an 8-bit open-drain port by default. Via setting P0UP (P0OR.0), P0 will switch as weakly pulled up internally. P0 has an alternative function as AD[7:0] while external memory accessing. During the external memory access, P0 will output high will be internal strong pulled-up rather than weak pull-up in order to drive out high byte address for external devices.
38	42	36	39	P0.1		AD1	D, I/O	
37	41	35	38	P0.2		AD2	D, I/O	
36	40	34	37	P0.3		AD3	D, I/O	
35	39	33	35	P0.4		AD4	D, I/O	
34	38	32	34	P0.5		AD5	D, I/O	
33	37	31	33	P0.6		AD6	D, I/O	
32	36	30	32	P0.7		AD7	D, I/O	
1	2	40	43	P1.0	T2		I/O	**PORT1:** Port 1 is an 8-bit quasi bi-directional I/O port. Its multifunction pins are for T2, T2EX, PWM0~PWM4, $\overline{SS}$, MOSI, MISO, and SPCLK.
2	3	41	44	P1.1	T2EX		I/O	
3	4	42	45	P1.2			I/O	
4	5	43	46	P1.3	PWM0		I/O	
5	6	44	47	P1.4	PWM1	$\overline{SS}$	I/O	
6	7	1	1	P1.5	PWM2	MOSI	I/O	
7	8	2	2	P1.6	PWM3	MISO	I/O	
8	9	3	3	P1.7	PWM4	SPCLK	I/O	
21	24	18	19	P2.0		A8	I/O	**PORT2:** Port 2 is an 8-bit quasi bi-directional I/O port. It has an alternative function as A[15:8] while external memory accessing. During the external memory access, P2 will output high will be internal strong pulled-up rather than weak pull-up in order to drive out high byte address for external devices.
22	25	19	20	P2.1		A9	I/O	
23	26	20	21	P2.2		A10	I/O	
24	27	21	22	P2.3		A11	I/O	
25	28	22	23	P2.4		A12	I/O	
26	29	23	25	P2.5		A13	I/O	
27	30	24	26	P2.6		A14	I/O	
28	31	25	27	P2.7		A15	I/O	

5.3 Internal Data Memory

Figure 5–3 shows the internal and external Data Memory spaces available on N78E059A/N78E055A. Internal Data Memory can be divided into three blocks. They are the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 bytes of SFR space. Internal Data Memory addresses are always 8-bit wide, which implies an address space of only 256 bytes. Direct addressing higher than 7FH will access the special function registers (SFRs) space and indirect addressing higher than 7FH will access the upper 128 bytes of RAM. Although the SFR space and the upper 128 bytes of RAM share the same logic address, 80H through FFH, actually they are physically separate entities. Direct addressing to distinguish with the higher 128 bytes of RAM can only access these SFRs. Sixteen addresses in SFR space are both byte and bit-addressable. The bit-addressable SFRs are those whose addresses end in 0H or 8H.

The lower 128 bytes of internal RAM are present in all 8051 devices. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call these registers as R0 through R7. Two bits RS0 and RS1 in the Program Status Word (PSW[3:4]) select which Register Bank is used. This benefits more efficiency of code space, since register instructions are shorter than instructions that use direct addressing. The next 16 bytes above the Register Banks (byte-address 20H through 2FH) form a block of bit-addressable memory space (bit-address 00H through 7FH). The 8051 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All bytes in the lower 128-byte space can be accessed by either direct or indirect addressing. Indirect addressing can only access the upper 128.

Another application implemented with the whole block of internal 256-byte RAM is for the stack. This area is selected by the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a JMP, 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.

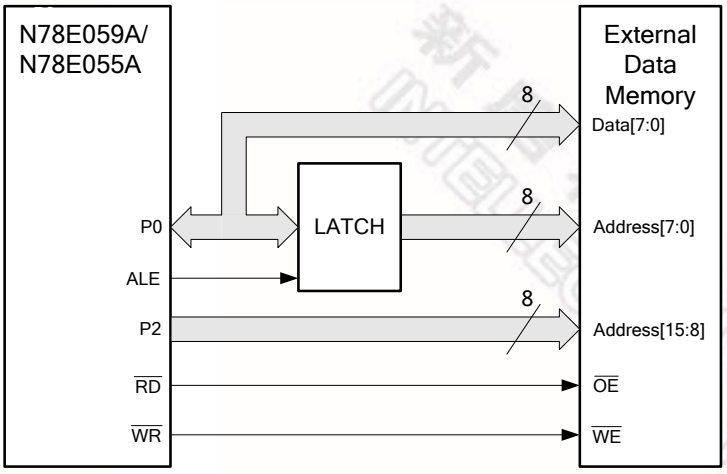

Figure 5–5. Data Memory Interface

5.6 On-chip Non-volatile Data Flash

N78E059A/N78E055A additionally has Data Flash. The Data Flash 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. Be aware of Data Flash writing endurance of 10,000 cycles. By the software path, the Data Flash can be accessed only through ISP mode. Note that the erasing or writing of Data Flash should not operates under V_{DD} 3.0V for ISP limitation. For Data Flash accessing with ISP, please see Section 18. "IN SYSTEM PROGRAMMING (ISP)" on page 91 for details. For the design for security, ISP is invalid while external Program Memory executes. The Data Flash, therefore, cannot be accessed with external memory code. Of course the Data Flash can be accessed via hardware with parallel Programmer/Writer.

The Data Flash size is fixed as 4k-byte size on N78E059A/N78E055A.

Table 6–2. N78E059A/N78E055A SFR Descriptions and Reset Values

Symbol	Definition	Address	MSB							LSB[1]	Reset Value[2]
SPDR	SPI data	F5H									0000 0000b
SPSR	SPI status	F4H	SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-	0000 0000b
SPCR	SPI control	F3H	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0000b
B	B register	F0H	(F7)	(F6)	(F5)	(F4)	(F3)	(F2)	(F1)	(F0)	0000 0000b
ACC	Accumulator	E0H	(E7)	(E6)	(E5)	(E4)	(E3)	(E2)	(E1)	(E0)	0000 0000b
PWM3	PWM3 duty	DEH									0000 0000b
PWM2	PWM2 duty	DDH									0000 0000b
PWMCON0	PWM control 0	DCH	PWM3OE	PWM2OE	PWM3EN	PWM2EN	PWM1OE	PWM0OE	PWM1EN	PWM0EN	0000 0000b
PWM1	PWM1 duty	DBH									0000 0000b
PWM0	PWM0 duty	DAH									0000 0000b
PWMP	PWM period	D9H									0000 0000b
P4	Port 4	D8H	(DF)	(DE)	(DD)	(DC)	(DB) $\overline{\text{INT2}}$	(DA) $\overline{\text{INT3}}$	(D9)	(D8)	1111 1111b
PSW	Program status word	D0H	(D7) CY	(D6) AC	(D5) F0	(D4) RS1	(D3) RS0	(D2) OV	(D1) F1	(D0) P	0000 0000b
PWM4	PWM4 duty	CFH									0000 0000b
PWMCON1	PWM control 1	CEH	-	-	-	-	-	PWM4OE	-	PWM4EN	0000 0000b
TH2	Timer 2 high byte	CDH									0000 0000b
TL2	Timer 2 low byte	CCH									0000 0000b
RCAP2H	Timer 2 reload/capture high byte	CBH									0000 0000b
RCAP2L	Timer 2 reload/capture low byte	CAH									0000 0000b
T2MOD	Timer 2 mode	C9H	-	-	-	-	-	-	T2OE	-	0000 0000b
T2CON	Timer 2 control	C8H	(CF) TF2	(CE) EXF2	(CD) RCLK	(CC) TCLK	(CB) EXEN2	(CA) TR2	(C9) $\overline{\text{C/T2}}$	(C8) $\overline{\text{CP/RL2}}$	0000 0000b
TA	Timed access protection	C7H									0000 0000b
XICON	External interrupt control	C0H	(C7) PX3	(C6) EX3	(C4) IE3	(C4) IT3	(C3) PX2	(C2) EX2	(C1) IE2	(C0) 1IT2	0000 0000b
EIE	Extensive interrupt enable	BDH	-	-	-	-	-	EBOD	EPDT	ESPI	0000 0000b
EIP	Extensive interrupt priority	BCH	-	-	-	-	-	PBOV	PPDT	PSPI	0000 0000b
EIPH	Extensive interrupt priority high	BBH	-	-	-	-	-	PBODH	PPDTH	PSPIH	0000 0000b
IPH	Interrupt priority high	BAH	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b
IP	Interrupt priority	B8H	(BF) -	(BE) -	(BD) PT2	(BC) PS	(BB) PT1	(BA) PX1	(B9) PT0	(B8) PX0	0000 0000b
P3	Port 3	B0H	(B7) $\overline{\text{RD}}$	(B6) $\overline{\text{WR}}$	(B5) T1	(B4) T0	(B3) $\overline{\text{INT1}}$	(B2) $\overline{\text{INT0}}$	(B1) TXD	(B0) RXD	1111 1111b
ISPCN	ISP flash control	AFH	ISPA17	ISPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0000 0000b
ISPFD	ISP flash data	AEH									0000 0000b
PMC[3]	Power monitoring control	ACH	BODEN	-	-	BORST	BOF[4]	LPBOD	-	BOS[5]	Power-on[6], XXXX X00Xb Brown-out, XXXX 100Xb Others, XXXX 000Xb
PDCON	Power Down waking-up timer control	ABH	PDTEN	PDTCK	PDTF	-	-	PPS2	PPS1	PPS0	0000 0000b
WDCON[3]	Watchdog Timer control	AAH	WDTEN	WDCLR	-	WIDPD	WDTRF	WPS2	WPS1	WPS0	Power-on[6], X000 0000b Watchdog, X00U 1UUUb Others, X00U UUUUb
IE	Interrupt enable	A8H	(AF) EA	(AE) -	(AD) ET2	(AC) ES	(AB) ET1	(AA) EX1	(A9) ET0	(A8) EX0	0000 0000b
ISPAH	ISP address high byte	A7H									0000 0000b
ISPAL	ISP address low byte	A6H									0000 0000b
ISPTRG[3]	ISP trigger	A4H	-	-	-	-	-	-	-	ISPGO	0000 0000b
XRAMAH	Auxiliary RAM address high byte	A1H	-	-	-	-	-	-	XRAMAH.1	XRAMAH.0	0000 0000b
P2	Port 2	A0H	(A7) A15	(A6) A14	(A5) A13	(A4) A12	(A3) A11	(A2) A10	(A1) A9	(A0) A8	1111 1111b

PCON – Power Control

7	6	5	4	3	2	1	0
SMOD	-	-	POF	GF1	GF0	PD	IDL
r/w	-	-	r/w	r/w	r/w	r/w	r/w

Address: 87H reset value: see Table 6–2. N78E059A/N78E055A SFR Descriptions 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.

8. AUXILIARY RAM (XRAM)

N78E059A/N78E055A provides additional on-chip 1k-byte RAM called XRAM to enlarge the RAM space. It occupies the address space from 000H through 3FFH. The XRAM is enabled after all resets. The 1024 bytes of XRAM are indirectly accessed by move external instruction MOVX @DPTR or MOVX @Ri along with XRAMAH. (If XRAM is enabled, MOVX @Ri cannot be used to access external RAM anymore.) This block of XRAM shares the same logic address of 000H through 3FFH with the external RAM. A DPTR value given larger than 03FFH will map to the external RAM no matter of the value of bit XRAMEN (CHPCON.4). If the user would like to access contents within 000H to 3FFH address of the off-chip external XRAM, the XRAMEN bit should be cleared as logic 0. (Note that CHPCON is a TA writing protected SFR.) When the XRAM is accessed, the address fetching signal will not emit via P0, P2, $\overline{WR}$, and $\overline{RD}$. Note that the stack pointer cannot locate in any part of XRAM.

CHPCON – Chip Control (TA protected)

7	6	5	4	3	2	1	0
SWRST	ISPF	LDUEN	XRAMEN	-	-	BS	ISPEN
w	r/w	r/w	r/w	-	-	r/w	r/w

Address: 9FH reset value: see Table 6–2. N78E059A/N78E055A SFR Descriptions and Reset Values

Bit	Name	Description
4	XRAMEN	**XRAM enable.** 0 = Disable on-chip XRAM. 1 = Enable on-chip XRAM. (The default value after all resets.)

XRAMAH – XRAM Address High Byte

7	6	5	4	3	2	1	0
-	-	-	-	-	-	XRAMAH.1	XRAMAH.0
-	-	-	-	-	-	r/w	r/w

Address: A1H reset value: 0000 0000b

Bit	Name	Description
7:2	-	**Reserved.**
1:0	XRAMAH[1:0]	**XRAM address high byte.** To set the XRAM high byte address. This setting works along with MOV @Ri instructions. The demo codes are listed below.

XRAM demo code:

```
MOV     XRAMAH,#01H         ;write #5AH to XRAM with address @0123H.
MOV     R0,#23H
MOV     A,#5AH
MOVX    @R0,A

MOV     XRAMAH,#01H         ;read from XRAM with address @0123H.
MOV     R0,#23H
```

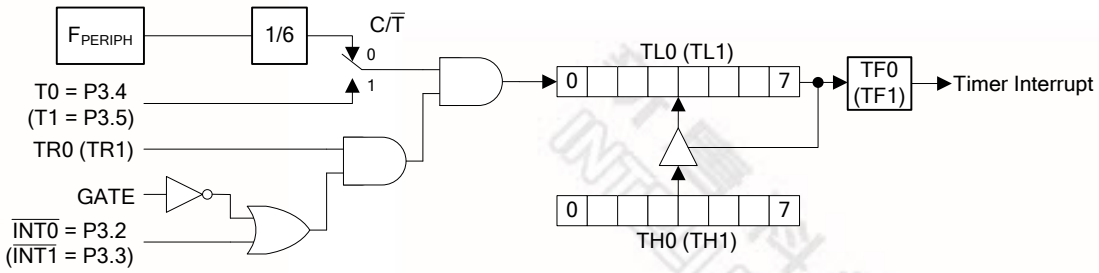

Figure 10–3. Timer/Counter 0 and 1 in Mode 2

10.1.4 Mode 3 (Two Separate 8-bit Timers)

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. TL0 uses the Timer/Counter 0 control bits C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. The TL0 also can be used as a 1-to-0 transition counter on pin T0 as determined by C/$\overline{T}$ (TMOD.2). TH0 is forced as a clock cycle counter and takes over the usage of TR1 and TF1 from Timer/Counter 1. Mode 3 is used in case which an extra 8 bit timer is needed. If Timer/Counter 0 is configured in Mode 3, Timer/Counter 1 can be turned on or off by switching it out of or into its own Mode 3. It can still be used in Modes 0, 1 and 2 although its flexibility is restricted. It no longer has control over its overflow flag TF1 and the enable bit TR1. However Timer 1 can still be used as a Timer/Counter and retains the use of GATE and $\overline{INT1}$ pin. It can be used as a baud rate generator for the serial port or other application not requiring an interrupt.

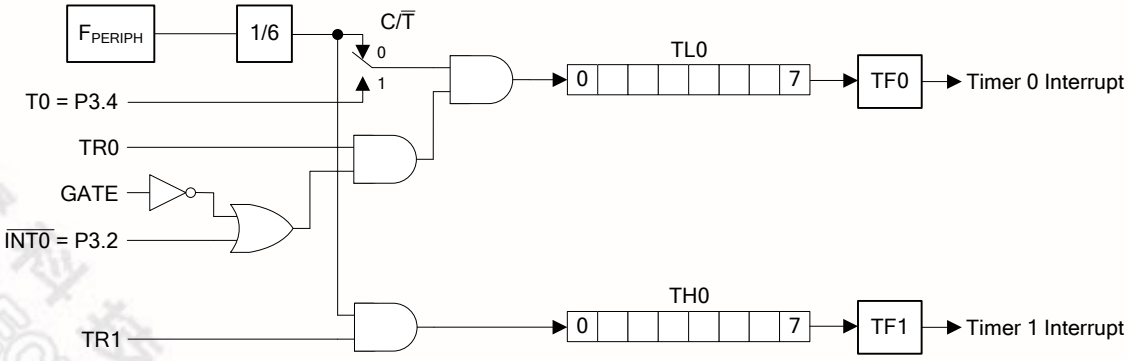

Figure 10–4. Timer/Counter 0 in Mode 3

Table 10–1. Timer 2 Operating Modes

Timer 2 Mode	RCLK (T2CON.5) or TCLK (T2CON.4)	CP/$\overline{RL2}$ (T2CON.0)	T2OE (T2MOD.1)
16-bit capture[1]	0	1	X
16-bit auto-reload	0	0	0
Baud rate generator	1	X	0
Clock-out[2]	X	0	1

[1] The capture is valid while EXEN2 (T2CON.3) is a 1. Or Timer/Counter 2 behaves just like a 16-bit timer/counter.

[2] C/$\overline{T2}$ (T2CON.1) must be 0.

10.2.1 Capture Mode

The capture mode is enabled by setting the CP/$\overline{RL2}$ bit in the T2CON register to 1. In the capture mode, Timer/Counter 2 serves as a 16 bit up counter. When the counter rolls over from FFFFH to 0000H, the TF2 bit is set, which will generate an Timer 2 interrupt request. If the EXEN2 bit is set, then a negative transition of T2EX pin (alternative function of P1.1) will cause the value in the TL2 and TH2 register to be captured by the RCAP2L and RCAP2H registers. The TH2 and TL2 keeps on counting while this capture event occurs. This capture action also causes the EXF2 (T2CON.6) bit set, which will also generate an Timer 2 interrupt. If Timer 2 interrupt enabled, both TF2 and EXF2 flags will generate interrupt vectoring to the same location. The user should check which one triggers the Timer 2 interrupt in the interrupt service routine.

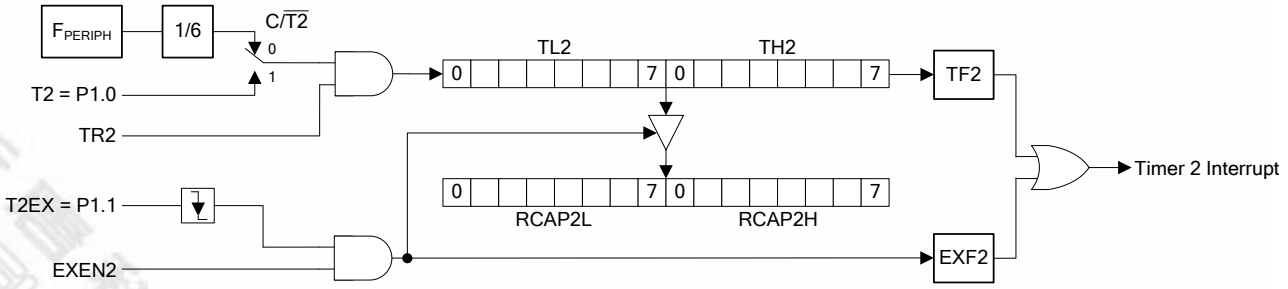

Figure 10–5. Timer/Counter 2 in Capture Mode

10.2.2 Auto-reload Mode

The auto-reload mode is enabled by clearing the CP/$\overline{RL2}$ bit in the T2CON register. In this mode, Timer/Counter 2 is a 16 bit up counter. When the counter rolls over from FFFFH, TF2 (T2CON.7) is set as 1 and a reload is generated that causes the contents of the RCAP2L and RCAP2H registers to be reloaded into the TL2 and TH2 registers respectively. If the EXEN2 bit is set, then a negative transition on T2EX pin will also cause a reload. This action also sets the EXF2 bit in T2CON.

10.2.4 Clock-out Mode

Timer 2 is equipped with a clock-out feature, which outputs a 50% duty cycle clock on P1.0. It can be invoked as a programmable clock generator. To configure Timer 2 with clock-out mode, software must initiate it by setting bit T2OE (TMOD.1) = 1, $C/\overline{T2}$ = 0 and $CP/\overline{RL2}$ = 0. Setting bit TR2 will start the clock output. This mode is similar to the baud rate generator mode which does not generate an interrupt while Timer 2 overflow. Similar with the baud rate generator mode, T2EX can also be configured as a simple external interrupt.

The clock-out frequency follows the equation $\frac{F_{OSC}}{2 \times 2^{EN6T} \times (65536 - (RCAP2H,RCAP2L))}$.

In this formula, EN6T is bit 6 of CONFIG3. While EN6T = 0, the clock system runs under 6T mode and the clock-out frequency will be double of that in 12T mode. (RCAP2H,RCAP2L) in the formula means $256 \times RCAP2H + RCAP2L$.

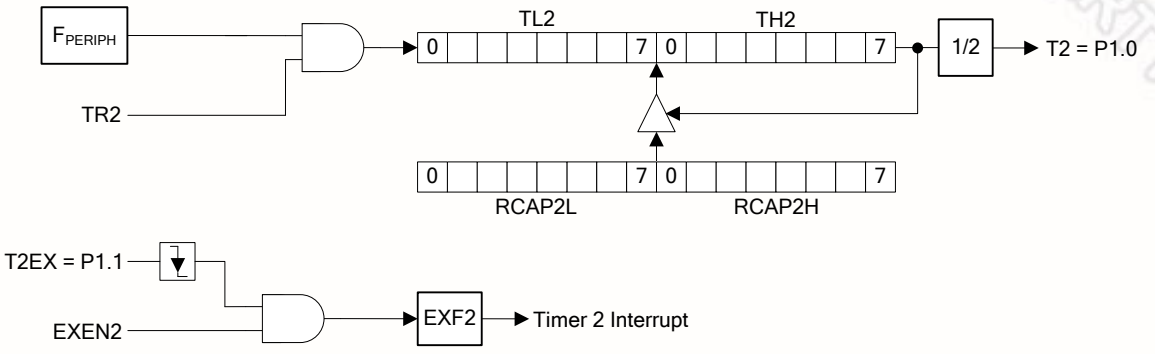

Figure 10–8. Timer/Counter 2 in Clock-out Mode

Table 11–1. Watchdog Timer-Out Interval under different pre-scalars

WPS2	WPS1	WPS0	Clock Divider Scale	Typical Watchdog Time-out Interval (F_{ILRC} ~= 10kHz)
0	0	0	1/1	6.40ms
0	0	1	1/2	12.80ms
0	1	0	1/8	51.20ms
0	1	1	1/16	102.40ms
1	0	0	1/32	204.80ms
1	0	1	1/64	409.60ms
1	1	0	1/128	819.20ms
1	1	1	1/256	1.638s

11.2 Applications of Watchdog Timer

The main application of the Watchdog Timer 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 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 WDCON requires a timed access writing.

Bit	Name	Description
5	PDTF	**Power Down waking-up timer Interrupt Flag.** This bit will be set via hardware when PDT counter overflows. This bit must be cleared via software.
4:3	-	**Reserved.**
2	PPS2	**Power Down waking-up timer clock pre-scalar select.** These bits determine the scale of the clock divider for PDT counter. The scale is from 1/1 through 1/1024. See Table 12–1.
1	PPS1	
0	PPS0	

The Power Down waking-up time-out interval is determined by the formula $\frac{1}{F_{LOSC} \times clockdividerscalar} \times 64$ where F_{ILRC} is the frequency of internal 10kHz RC. The following table shows an example of the Power Down waking-up time-out interval under different pre-scalars.

Table 12–1. Power Down Waking-up Timer-Out Interval under different pre-scalars

PPS2	PPS1	PPS0	Clock Divider Scale	Typical Power Down Waking-up Time-out Interval (F_{ILRC} ~= 10kHz)
0	0	0	1/1	6.40ms
0	0	1	1/4	25.60ms
0	1	0	1/8	51.20ms
0	1	1	1/32	204.80ms
1	0	0	1/64	409.60ms
1	0	1	1/256	1.638s
1	1	0	1/512	3.277s
1	1	1	1/1024	6.554s

12.2 Applications of Power Down Waking-up Timer

The main application of the Power Down waking-up timer is a simple timer. The PDTF flag will be set while the Power Down waking-up timer completes the selected time interval. The software polls the PDTF flag to detect a time-out and the PDCLR allows software to restart the timer. The Power Down waking-up timer can also be used as a very long timer. Every time the time-out occurs, an interrupt will occur if the individual interrupt EPDT (EIE.1) 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. N78E059A/N78E055A is equipped with this useful function. It provides a very low power internal RC 10kHz. Along with the low power consumption application, the Power Down Waking-up 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 Power Down waking-up timer waking up CPU from Power Down.

```
       ORG    0000H
       LJMP   START

       ORG    004BH
       LJMP   PDT_ISR

       ORG    0100H
PDT_ISR:
       ORL    PDCON,#01000000B          ;Clear Power Down Waking-up timer counter
       ANL    PDCON,#11011111B          ;Clear Power Down Waking-up timer interrupt flag
       RETI

START:
       ORL    PDCON,#00000111B          ;Choose interval length
       ORL    EIE,#00000010B            ;Enable Power Down Waking-up timer interrupt
       SETB   EA
       ORL    PDCON,#10000000B          ;Enable Power Down Waking-up timer to run

;*********************************************************************
;Enter into Power Down mode
;*********************************************************************
LOOP:
       ORL    PCON,#02H
       LJMP   LOOP
```

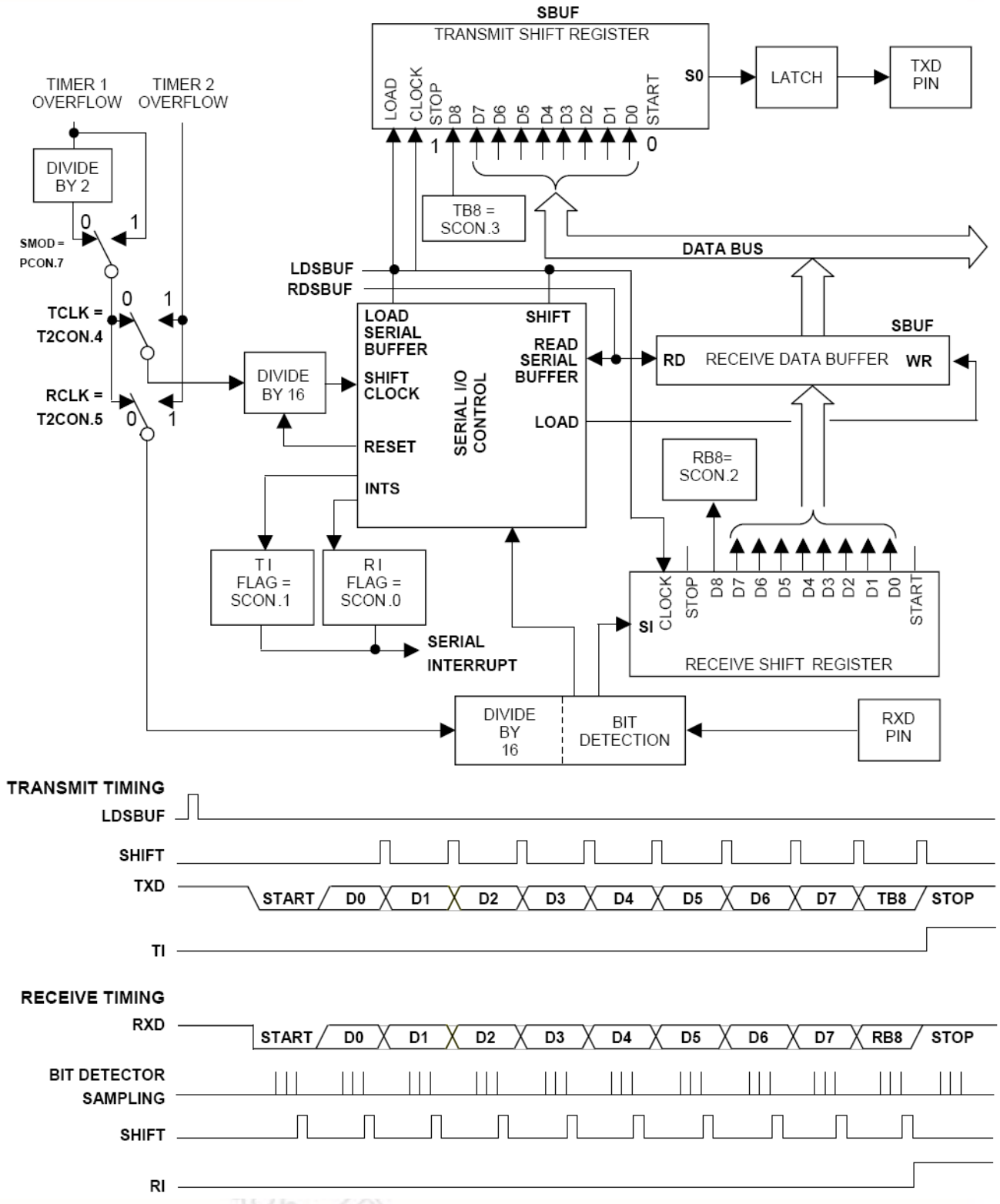

Figure 13–4. Serial Port Mode 3 Function Block and Timing Diagram

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 must read SPDR out and the first SPIF must be cleared before a second transfer of data from the Master device comes in the read data buffer.

14.5 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 14–4. SPI Clock Formats shows that CPOL and CPHA compose four different clock formats. The CPOL bit denotes the SPCLK line level in SPI 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. To Communicate in different data formats with one another will result undetermined result.

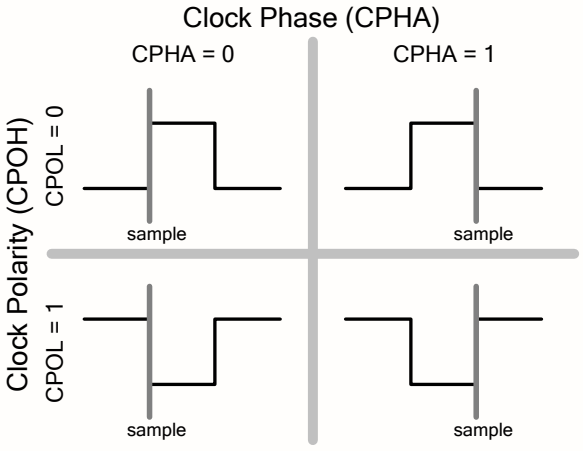

Figure 14–4. SPI Clock Formats

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.0) is set 1 and global interrupt is enabled (EA = 1), the interrupt service routine (ISR) of SPI will be executed.

15. PULSE WIDTH MODULATOR (PWM)

N78E059A/N78E055A provides five pulse width modulated (PWM) output channels to generate pulses of programmable length and interval. Five PWM channels, PWM0~4, shares the same pins with P1.3~P1.7. The PWM period is defined by an 8-bit pre-scalar PWMP, which supplies the clock of the PWM counter. The pre-scalar is common for all PWM channels. The duty of each PWM channel is determined by the value of five registers, PWM0, PWM1, PWM2, PWM3, and PWM4. If the contents of these registers are equal to or less than the 8-bit counter value, the output will be 0. Else the output will be 1 if these registers value are larger than the counter. Set PWMxEN (in PWMCON0[0,1,4,5] and PWMCON1.0) will enable to run or disable to stop each PWM channel respectively. In addition, the PWMxOM (in PWMCON0[2,3,6,7] and PWMCON1.2) must set 1 to output the internal PWM signal to port pins. Without setting PWMxOM, the pins which share with alternative PWM function will be normal general purpose I/O of P1.3~P1.7 even though PWM is enabled. The following registers relate to PWM function.

PWMCON0 – PWM Control 0

7	6	5	4	3	2	1	0
PWM3OE	PWM2OE	PWM3EN	PWM2EN	PWM1OE	PWM0OE	PWM1EN	PWM0EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: DCH reset value: 0000 0000b

Bit	Name	Description
7	PWM3OE	**PWM3 output enable.** 0 = P1.6 serves as general purpose I/O. 1 = P1.6 serves as output pin of PWM3 signal.
6	PWM2OE	**PWM2 output enable.** 0 = P1.5 serves as general purpose I/O. 1 = P1.5 serves as output pin of PWM2 signal.
5	PWM3EN	**PWM3 enable.** 0 = PWM3 is disabled and stops. 1 = PWM3 is enabled and runs.
4	PWM2EN	**PWM2 enable.** 0 = PWM2 is disabled and stops. 1 = PWM2 is enabled and runs.
3	PWM1OE	**PWM1 output enable.** 0 = P1.4 serves as general purpose I/O. 1 = P1.4 serves as output pin of PWM1 signal.
2	PWM0OE	**PWM0 output enable.** 0 = P1.3 serves as general purpose I/O. 1 = P1.3 serves as output pin of PWM0 signal.
1	PWM1EN	**PWM1 enable.** 0 = PWM1 is disabled and stops. 1 = PWM1 is enabled and runs.

PWM1 – PWM1 Duty

7	6	5	4	3	2	1	0
PWM1[7:0]							
r/w							

Address: DBH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM1[7:0]	**PWM1 duty.** This byte controls the duty of the PWM1 output.

PWM2 – PWM2 Duty

7	6	5	4	3	2	1	0
PWM2[7:0]							
r/w							

Address: DDH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM2[7:0]	**PWM2 duty.** This byte controls the duty of the PWM2 output.

PWM3 – PWM3 Duty

7	6	5	4	3	2	1	0
PWM3[7:0]							
r/w							

Address: DEH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM3[7:0]	**PWM3 duty.** This byte controls the duty of the PWM3 output.

PWM4 – PWM4 Duty

7	6	5	4	3	2	1	0
PWM4[7:0]							
r/w							

Address: CFH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM4[7:0]	**PWM4 duty.** This byte controls the duty of the PWM4 output.

The repetition frequency of PWM, F_{PWM} is given by,

$F_{PWM} = \frac{F_{PERIPH}}{(PWMP + 1) \times 255}$, pre-scalar division factor = PWM + 1.

PWM high duty of PWMx = $\frac{PWMx}{255}$.

17. INTERRUPT SYSTEM

The purpose of the interrupt is to make the software deal with unscheduled or asynchronous events. N78E059A/N78E055A has a four-priority-level interrupt structure with 11 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. When an interrupt occurs, the CPU is expected to service the interrupt. This service is specified as an Interrupt Service Routine (ISR). The ISR resides at a predetermined address as shown in Table 17–1. N78E059A/N78E055A Interrupt Vectors. When the interrupt occurs if enabled, the CPU will vector to the appropriate location. It will execute the code at this location, staying in an interrupt service state until done with the ISR. Once an ISR has begun, it can be interrupted only by a higher priority interrupt. The ISR is terminated by a return from interrupt instruction RETI. This instruction will force the CPU return to the instruction that would have been next when the interrupt occurred.

Table 17–1. N78E059A/N78E055A Interrupt Vectors

Source	Vector Address	Vector Number	Source	Vector Address	Vector Number
External Interrupt 0	0003H	0	Timer 0 Overflow	000BH	1
External Interrupt 1	0013H	2	Timer 1 Overflow	001BH	3
Serial Port Interrupt	0023H	4	Timer 2 Overflow/Capture/Reload	002BH	5
External Interrupt 2	0033H	6	External Interrupt 3	003BH	7
SPI Interrupt	0043H	8	Power Down Waking-up Timer Interrupt	004BH	9
Brown-out Detection Interrupt	0053H	10			

The SFRs associated with these interrupts are listed below.

IE – Interrupt Enable (bit-addressable)

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET1	EX1	ET0	EX0
r/w	-	r/w	r/w	r/w	r/w	r/w	r/w

Address: A8H reset value: 0000 0000b

Bit	Name	Description
7	EA	**Enable all interrupt.** This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings. 0 = Disable all interrupt sources. 1 = Enable each interrupt depending on its individual mask setting. Individual interrupts will occur if enabled.
6	-	**Reserved.**

Bit	Name	Description
5	PT2	**Timer 2 interrupt priority low bit.**
4	PS	**Serial port (UART) interrupt priority low bit.**
3	PT1	**Timer 1 interrupt priority low bit.**
2	PX1	**External interrupt 1 priority low bit.**
1	PT0	**Timer 0 interrupt priority low bit.**
0	PX0	**External interrupt 0 priority low bit.**

[1] IP is used in combination with the IPH to determine the priority of each interrupt source. See Table 17–2. Interrupt Priority Level Setting for correct interrupt priority configuration.

IPH – Interrupt Priority High

7	6	5	4	3	2	1	0
PX3H[2]	PX2H[2]	PT2H[3]	PSH[3]	PT1H[3]	PX1H[3]	PT0H[3]	PX0H[3]
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: BAH reset value: 0000 0000b

Bit	Name	Description
7	PX3H	**External interrupt 3 priority high bit.**
6	PX2H	**External interrupt 3 priority high bit.**
5	PT2H	**Timer 2 interrupt priority high bit.**
4	PSH	**Serial port (UART) interrupt priority high bit.**
3	PT1H	**Timer 1 interrupt priority high bit.**
2	PX1H	**External interrupt 1 priority high bit.**
1	PT0H	**Timer 0 interrupt priority high bit.**
0	PX0H	**External interrupt 0 priority high bit.**

[2] PX2H and PX3H are used in combination with the PX2 (XICON.3) and PX3 (XICON.7) respectively to determine the priority of external interrupt 2 and 3. See Table 17–2. Interrupt Priority Level Setting for correct interrupt priority configuration.

[3] These bits is used in combination with the IP respectively to determine the priority of each interrupt source. See Table 17–2. Interrupt Priority Level Setting for correct interrupt priority configuration.

EIP – Extensive Interrupt Priority[4]

7	6	5	4	3	2	1	0
-	-	-	-	-	PBOD	PPDT	PSPI
-	-	-	-	-	r/w	r/w	r/w

Address: BCH reset value: 0000 0000b

Bit	Name	Description
7:3	-	**Reserved.**
2	PBOD	**Brown-out detection interrupt priority low bit.**
1	PPDT	**Power Down waking-up timer interrupt priority low bit.**
0	PSPI	**SPI interrupt priority low bit.**

18. IN SYSTEM PROGRAMMING (ISP)

The internal Program Memory and on-chip Data Flash support both hardware programming and in system programming (ISP). Hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter into the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. ISP method makes it easy and possible. N78E059A/N78E055A supports ISP mode allowing a device to be reprogrammed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

ISP is performed without removing the microcontroller from the system. The most common method to perform ISP is via UART along with the firmware in LDROM. General speaking, PC transfers the new APROM code through serial port. Then LDROM firmware receives it and re-programs into APROM through ISP commands. Nuvoton provides ISP firmware, USB ISP writer and PC application program for N78E059A/N78E055A. It makes users quite easy perform ISP through Nuvoton standard ISP tool. Please explore Nuvoton 8-bit Microcontroller website: Nuvoton 80C51 Microcontroller Development Tool.

18.1 ISP Procedure

Unlike RAM's real-time operation, to update flash data often takes long time. Furthermore, it is a quite complex timing procedure to erase, program, or read flash data. Fortunately, N78E059A/N78E055A carried out the flash operation with convenient mechanism to help the user update the flash content. After ISP enabled by setting ISPEN (CHPCON.0 with TA protected), the user can easily fill the 16-bit target address in ISPAH and ISPAL, data in ISPFD and command in ISPCN. Then the ISP is ready to begin by setting a triggering bit ISPGO (ISPTRG.0). Note that ISPTRG is also TA protected. At this moment, the CPU holds the Program Counter and the built-in ISP automation takes over to control the internal charge-pump for high voltage and the detail signal timing. After ISP action completed, the Program Counter continues to run the following instructions. The ISPGO bit will be automatically cleared. The user may repeat steps above for next ISP action if necessary. Through this progress, the user can easily erase, program, and verify the embedded flash by just taking care of the pure software.

The following registers relate to ISP processing.

Table 25–1. Instruction Set for N78E059A/N78E055A

Instruction		OPCODE	Bytes	Clock Cycles in 12T Mode	Clock Cycles in 6T Mode
INC	A	04	1	12	6
INC	Rn	08~0F	1	12	6
INC	@Ri	06, 07	1	12	6
INC	direct	05	2	12	6
INC	DPTR	A3	1	24	12
DEC	A	14	1	12	6
DEC	Rn	18~1F	1	12	6
DEC	@Ri	16, 17	1	12	6
DEC	direct	15	2	12	6
MUL	AB	A4	1	48	24
DIV	AB	84	1	48	24
DA	A	D4	1	12	6
ANL	A, Rn	58~5F	1	12	6
ANL	A, @Ri	56, 57	1	12	6
ANL	A, direct	55	2	12	6
ANL	A, #data	54	2	12	6
ANL	direct, A	52	2	12	6
ANL	direct, #data	53	3	24	12
ORL	A, Rn	48~4F	1	12	6
ORL	A, @Ri	46, 47	1	12	6
ORL	A, direct	45	2	12	6
ORL	A, #data	44	2	12	6
ORL	direct, A	42	2	12	6
ORL	direct, #data	43	3	24	12
XRL	A, Rn	68~6F	1	12	6
XRL	A, @Ri	66, 67	1	12	6
XRL	A, direct	65	2	12	6
XRL	A, #data	64	2	12	6
XRL	direct, A	62	2	12	6
XRL	direct, #data	63	3	24	12
CLR	A	E4	1	12	6
CPL	A	F4	1	12	6
RL	A	23	1	12	6
RLC	A	33	1	12	6
RR	A	03	1	12	6
RRC	A	13	1	12	6
SWAP	A	C4	1	12	6
MOV	A, Rn	E8~EF	1	12	6
MOV	A, @Ri	E6, E7	1	12	6
MOV	A, direct	E5	2	12	6