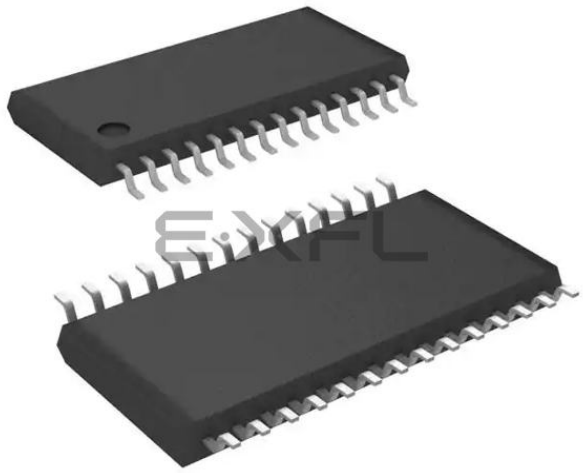

Welcome to E-XFL.COM

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	25
Program Memory Size	16KB (16K 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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	28-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n79e855awg

Table 5–1 Pin Description

Pin No.	Symbol	Alternate Function				Type	Description
SOP28 TSSOP28		1	2	3			
14	P2.3				MISO2	I/O	The SPI-2 ports are by software switched from SPI-1 port. ADC7: ADC channel input. IC2: Input Capture pin
15	P2.4				/SS2	I/O	
16	P2.5				SPICLK2	I/O	
27	P2.6	TXD2	ADC7			I/O	
28	P2.7	RXD2				I/O	
9	P3.0	XTAL2	CLKOUT			I/O	**PORT3:** Port 3 has 4-type I/O port. Its multifunction pins are for XTAL1, XTAL2 and CLKOUT, **CLKOUT:** Internal RC OSC/4 output pin. **XTAL2:** This is the output pin from the internal inverting amplifier. It emits the inverted signal of XTAL2. **XTAL1:** This is the output pin from the internal inverting amplifier. It emits the inverted signal of XTAL1.
8	P3.1	XTAL1				I/O	

[1] I/O type description – I: input, O: output, I/O: quasi bi-direction, D: open-drain, P: power pins, ST: Schmitt trigger.

新唐科技 nuvoTon INTELLECTUAL PROPERTY

6.5 On-chip XRAM

The N79E855/854 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 N79E855/854 provides the on-chip 256 bytes scratch pad RAM and Special Function Registers (SFRs) which be accessed by software. The SFRs be accessed only by direct addressing, while the on-chip RAM be accessed by either direct or indirect addressing.

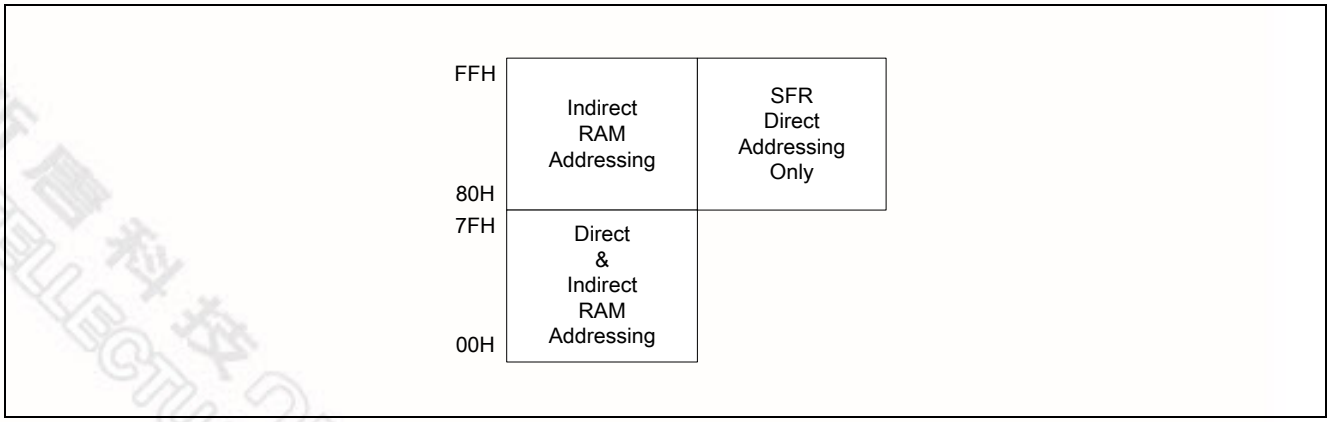

Figure 6-3 256 bytes RAM and SFR

Since the scratch-pad RAM is only 256 byte it can be used only when data contents are small. There are several other special purpose areas within the scratch-pad RAM, which are described as follows.

Table 7–2 N79E855/854 SFR Description and Reset Values

Symbol	Definition	Address	MSB							LSB	Reset Value[1]
EIP	Interrupt Priority 1	FFH	PT2	PSPI	PPWM	PWDI	-	-	PKB	PI2	0000 0000B
ADCCON0	ADC control register 0	F8H	(FF) ADC.1	(FE) ADC.0	(FD) ADCEX	(FC) ADCI	(FB) ADCS	(FA) AADR2	(F9) AADR1	(F8) AADR0	0000 0000B
EIPH	Interrupt High Priority 1	F7H	PT2H	PSPIH	PPWMH	PWDIH	-	-	PKBH	PI2H	0000 0000B
P0DIDS	Port 0 Digital Input Disable	F6H	P0DIDS[7:0]								0000 0000B
SPDR	Serial Peripheral Data Register	F5H	SPDR[7:0]								0000 0000B
SPSR	Serial Peripheral Status Register	F4H	SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-	0000 0000B
SPCR	Serial Peripheral Control Register	F3H	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0100B
B	B register	F0H	(F7) B.7	(F6) B.6	(F5) B.5	(F4) B.4	(F3) B.3	(F2) B.2	(F1) B.1	(F0) B.0	0000 0000B
C2H	Input Capture 2 High	EEH	C2H[7:0]								0000 0000B
C2L	Input Capture 2 Low	EDH	C2L[7:0]								0000 0000B
KBLS1	Keyboard level select 1	ECH	KBLS1[7:0]								0000 0000B
KBLS0	Keyboard level select 0	EBH	KBLS0[7:0]								0000 0000B
KBIF	KBI Interrupt Flag	EAH	KBIF[7:0]								0000 0000B
KBIE	Keyboard Interrupt Enable	E9H	KBIE[7:0]								0000 0000B
EIE	Interrupt enable 1	E8H	(EF) ET2	(EE) ESPI	(ED) EPWM	(EC) EWDI	(E7)	(E8) ECPTF	(E9) EKB	(E8) EI2C	0000 0000B
C1H	Input Capture 1 High	E7H	C1H[7:0]								0000 0000B
C1L	Input Capture 1 Low	E6H	C1L[7:0]								0000 0000B
C0H	Input Capture 0 High	E5H	C0H[7:0]								0000 0000B
C0L	Input Capture 0 Low	E4H	C0L[7:0]								0000 0000B
ADCH	ADC converter result	E2H	ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2	0000 0000B
ADCCON1	ADC control register1	E1H	ADCEN	-	-	-	-	-	RCCLK	ADC0SEL	0000 0000B
ACC	Accumulator	E0H	(E7) ACC.7	(E6) ACC.6	(E5) ACC.5	(E4) ACC.4	(E3) ACC.3	(E2) ACC.2	(E1) ACC.1	(E0) ACC.0	0000 0000B
PWMCON1	PWM control register 1	DFH	BKCH	BKPS	BPEN	BKEN	PWM3B	PWM2B	PWM1B	PWM0B	0000 0000B
PWM3L	PWM 3 low bits register	DEH	PWM3.7	PWM3.6	PWM3.5	PWM3.4	PWM3.3	PWM3.2	PWM3.1	PWM3.0	0000 0000B
PWM2L	PWM 2 low bits register	DDH	PWM2.7	PWM2.6	PWM2.5	PWM2.4	PWM2.3	PWM2.2	PWM2.1	PWM2.0	0000 0000B
PWMCON0	PWM control register 0	DCH	PWMRUN	LOAD	CF	CLRPWM	PWM3I	PWM2I	PWM1I	PWM0I	0000 0000B
PWM1L	PWM 1 low bits register	DBH	PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0	0000 0000B
PWM0L	PWM 0 low bits register	DAH	PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0	0000 0000B
PWMPL	PWM counter low register	D9H	PWMP0.7	PWMP0.6	PWMP0.5	PWMP0.4	PWMP0.3	PWMP0.2	PWMP0.1	PWMP0.0	0000 0000B
WDCON0[4] [3]	Watch-Dog control 0	D8H	(DF) WDTEN	(DE) WDCLR	(DD) WDTF	(DC) WIDPD	(DB) WDTRF	(DA) WPS2	(D9) WPS1	(D8) WPS0	Power-ON C000 0000B Watch reset C0UU 1UUUB Other reset C0UU UUUUB
PWMCON2	PWM control register 2	D7H	-	-	-	-	FP1	FP0	-	BKF	0000 0000B
PWM3H	PWM 3 high bits register	D6H	-	-	-	-	-	-	PWM3.9	PWM3.8	0000 0000B
PWM2H	PWM 2 high bits register	D5H	-	-	-	-	-	-	PWM2.9	PWM2.8	0000 0000B
PWM1H	PWM 1 high bits register	D3H	-	-	-	-	-	-	PWM1.9	PWM1.8	0000 0000B
PWM0H	PWM 0 high bits register	D2H	-	-	-	-	-	-	PWM0.9	PWM0.8	0000 0000B
PWMPH	PWM counter high register	D1H	-	-	-	-	-	-	PWMP0.9	PWMP0.8	0000 0000B
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

Table 7–2 N79E855/854 SFR Description and Reset Values

Symbol	Definition	Address	MSB							LSB	Reset Value[1]
SHBDA[4]	High-byte Data Flash Start Address	9CH	SHBDA[7:0], SHBDA Initial by CHBDA								Power ON CCCC CCCCB Other Reset UUUU UUUUB
SBUF	Serial buffer	99H	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0	0000 0000B
SCON	Serial control	98H	(9F) SM0/FE	(9E) SM1	(9D) SM2	(9C) REN	(9B) TB8	(9A) RB8	(99) TI	(98) RI	0000 0000B
P3M2	Port 3 output mode 2	97H	-	-	-	-	-	ENCLK	P3M2.1	P3M2.0	00000000B
P3M1	Port 3 output mode 1	96H	P3S	P2S	P1S	P0S	T1OE	T0OE	P3M1.1	P3M1.0	00000000B
DIVM	CPU Clock Divide Register	95H	DIVM[7:0]								0000 0000B
CAPCON2	Input capture control 2	94H	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000B
CAPCON1	Input capture control 1	93H	-	-	CAP2LS1[2:0]		CAP1LS1[2:0]		CAP1LS1[2:0]		0000 0000B
CAPCON0	Input capture control 0	92H	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000B
P1	Port 1	90H	(97) P17	(96) P16	-	(94) P14	(93) P13	(92) P12	(91) P11	(90) P10	1111 1111B
CKCON	Clock control	8EH	-	-	-	T1M	T0M	-	-	-	0000 0000B
TH1	Timer high 1	8DH	TH1[7:0]								0000 0000B
TH0	Timer high 0	8CH	TH0[7:0]								0000 0000B
TL1	Timer low 1	8BH	TL1[7:0]								0000 0000B
TL0	Timer low 0	8AH	TL0[7:0]								0000 0000B
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000 0000B
TCON	Timer control	88H	(8F) TF1	(8E) TR1	(8D) TF0	(8C) TR0	(8B) IE1	(8A) IT1	(89) IE0	(88) IT0	0000 0000B
PCON	Power control	87H	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	Power-on 0001 0000B Other reset 000u 0000B
DPH	Data pointer high	83H	DPH[7:0]								0000 0000B
DPL	Data pointer low	82H	DPL[7:0]								0000 0000B
SP	Stack pointer	81H	SP[7:0]								0000 0111B
P0	Port 0	80H	(87) P07	(86) P06	(85) P05	(84) P04	(83) P03	(82) P02	(81) P01	(80) P00	1111 1111B

Note: Bits marked in "-" should be kept in their own initial states. User should never change their values.

Note:

[1.] ( ) item means the bit address in bit-addressable SFRs.

[2.] BODEN, BOV and BORST are initialized by CONFIG2 at power-on reset, and keep unchanged at any other resets. If BODEN=1, BOF will be automatically set by hardware at power-on reset, and keeps unchanged at any other resets.

[3.] Initialized by power-on reset. WDTEN=/CWDTEN; BS=/CBS;

[4.] With TA-Protection. (Time Access Protection)

[5.] Notation “C” means the bit is defined by CONFIG-bits; “U” means the bit is unchanged after any reset except power-on reset.

[6.] Reset value symbol description. 0: logic 0, 1: logic 1, U: unchanged, X:, C: initial by CONFIG..

INTERNAL USE ONLY NUVOTON CONFIDENTIAL PROPRIETARY

DPL – Data Pointer Low Byte

7	6	5	4	3	2	1	0
DPL[7:0]							
R/W							

Address: 82H Reset value: 0000 0000B

Bit	Name	Description
7:0	DPL[7:0]	**Data Pointer Low Byte** This is the low byte of the standard 8051 16-bit data pointer. DPL combined with DPH serve as a 16-bit data pointer DPTR to address non-scratch-pad memory or Program Memory.

DPH – Data Pointer High Byte

7	6	5	4	3	2	1	0
DPH[7:0]							
R/W							

Address: 83H Reset value: 0000 0000B

Bit	Name	Description
7:0	DPH[7:0]	**Data Pointer High Byte** This is the high byte of the standard 8051 16-bit data pointer. DPH combined with DPL serve as a 16-bit data pointer DPTR to address non-scratch-pad memory or Program Memory.

PSW – Program Status Word (Bit-addressable)

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Address: D0H Reset value: 0000 0000B

Bit	Name	Description
7	CY	**Carry Flag** For a adding or subtracting operation, CY will be set when the previous operation resulted in a carry-out from or a borrow-in to the Most Significant bit, otherwise cleared. If the previous operation is MUL or DIV, CY is always 0. CY is affected by DA A instruction which indicates that if the original BCD sum is greater than 100. For a CJNE branch, CY will be set if the first unsigned integer value is less than the second one. Otherwise, CY will be cleared.
6	AC	**Auxiliary Carry** Set when the previous operation resulted in a carry-out from or a borrow-in to the 4th bit of the low order nibble, otherwise cleared.

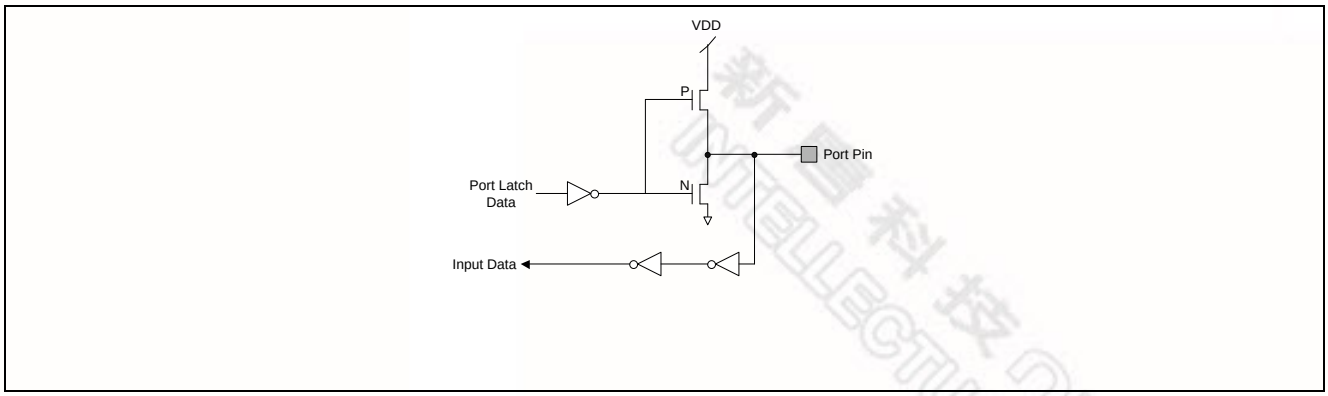

Figure 9-3 Push-Pull Output

9.4 Input Only Configuration

By setting this mode; the ports are only input mode. After setting this mode, the pin will be Hi-Impendence.

P0 – Port 0 (Bit-addressable)

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: 80H Reset value: 1111 1111B

Bit	Name	Description
7:0	P0[7:0]	**Port 0** Port 0 is an 8-bit quasi bidirectional I/O port.

P1 – Port 1 (Bit-addressable)

7	6	5	4	3	2	1	0
P17	P16	-	P14	P13	P12	P11	P10
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Address: 90H Reset value: 1111 1111B

Bit	Name	Description
7:0	P1[7:0]	**Port 1** These pins are in quasi-bidirectional mode except P1.2 and P1.3 pins. The P1.2 and P1.3 are dedicating open-drain pins for I^2C interface after reset.

Bit	Name	Description
6:4	T2DIV[2:0]	**Timer 2 Clock Divider** 000 = Timer 2 clock divider is 1/4. 001 = Timer 2 clock divider is 1/8. 010 = Timer 2 clock divider is 1/16. 011 = Timer 2 clock divider is 1/32. 100 = Timer 2 clock divider is 1/64. 101 = Timer 2 clock divider is 1/128. 110 = Timer 2 clock divider is 1/256. 111 = Timer 2 clock divider is 1/512.
3	CAPCR	**Capture Auto-clear** This bit enables auto-clear Timer 2 value in TH2 and TL2 when a determined input capture event occurs. 0 = Timer 2 continues counting when a capture event occurs. 1 = Timer 2 value is auto-cleared as 0000H when a capture event occurs.
2	COMPCR	**Compare Match Auto-clear** This bit enables auto-clear Timer 2 value in TH2 and TL2 when a compare match occurs. 0 = Timer 2 continues counting when a compare match occurs. 1 = Timer 2 value is auto-cleared as 0000H when a compare match occurs.
1:0	LDTS[1:0]	**Auto-reload Trigger Selection** These bits select the reload trigger event. 00 = Reload when Timer 2 overflows. 01 = Reload when input capture 0 event occurs. 10 = Reload when input capture 1 event occurs. 11 = Reload when input capture 2 event occurs.

RCOMP2L – Timer 2 Reload/Compare Low Byte

7	6	5	4	3	2	1	0
RCOMP2L[7:0]							
R/W							

Address: CAH Reset value: 0000 0000B

Bit	Name	Description
7:0	RCOMP2L[7:0]	**Timer 2 Reload/Compare Low Byte** This register stores the low byte of compare value when Timer 2 is configured in compare mode, It holds the low byte of the reload value when auto-reload mode.

RCOMP2H – Timer 2 Reload/Compare High Byte

7	6	5	4	3	2	1	0
RCOMP2H[7:0]							
R/W							

Address: CBH Reset value: 0000 0000B

Bit	Name	Description
7:0	RCOMP2H[7:0]	**Timer 2 Reload/Compare High Byte** This register stores the high byte of compare value when Timer 2 is configured in compare mode. And it holds the high byte of the reload value when auto-reload mode.

Bit	Name	Description
4	REN	**Receiving Enable** 0 = Serial port reception is disabled. 1 = Serial port reception is enabled in Mode 1,2, and 3. In Mode 0, clearing and then setting REN initiates one-byte reception. After reception is complete, this bit will not be cleared via hardware. The user should clear and set REN again via software to triggering the next byte reception.
3	TB8	**9th Transmitted Bit** This bit defines the state of the 9th transmission bit in serial port Mode 2 and 3. It is not used in Mode0 and 1.
2	RB8	**9th Received Bit** The bit identifies the logic level of the 9th received bit in Modes 2 and 3. In Mode 1, if SM2 0, RB8 is the logic level of the received stop bit. RB8 is not used in Mode 0.
1	TI	**Transmission Interrupt Flag** This flag is set via hardware when a byte of data has been transmitted by the UART after the 8th bit in Mode 0 or the last bit of data in other modes. When the UART interrupt is enabled, setting this bit causes the CPU to execute the UART interrupt service routine. This bit should be cleared manually via software.
0	RI	**Receiving Interrupt Flag** This flag is set via hardware when a 8-bit or 9-bit data has been received by the UART after the 8th bit in Mode 0, after sampling the stop bit in Mode 1, or after sampling the 9th bit in Mode 2 and 3. SM2 bit has restriction for exception. When the UART interrupt is enabled, setting this bit causes the CPU to execute to the UART interrupt service routine. This bit should be cleared manually via software.

Table 12–1 Serial Port Mode Description

Mode	SM0	SM1	Description	Frame Bits	Baud Rate
0	0	0	Synchronous	8	F_{SYS} divided by 12 or by 4[1]
1	0	1	Asynchronous	10	Timer 1 overflow rate divided by 32 or divided by 16[2]
2	1	0	Asynchronous	11	F_{SYS} divided by 64 or 32[2]
3	1	1	Asynchronous	11	Timer 1 overflow rate divided by 32 or divided by 16[2]

[1] While SM2 (SCON.5) is logic 1.
[2] While SMOD (PCON.7) is logic 1.

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 N79E855/854 SFR Description and Reset Values

Bit	Name	Description
7	SMOD	**Serial Port Double Baud Rate Enable** Setting this bit doubles the serial port baud rate in UART mode 2 and mode 1 or 3 only if Timer 1 overflow is used as the baud rate source. See Table 12–1 Serial Port Mode Description for details.

As shown, there is one bidirectional data line (RXD) and one shift clock line (TXD). The shift clock is used to shift data in or out of the serial port controller bit by bit for a serial communication. Data bits enter or exit LSB first. The band rate is equal to the shift clock frequency.

Transmission is initiated by any instruction writes to SBUF. The control block will then shift out the clock and begin to transfer data until all 8 bits are complete. Then the transmitted flag TI (SCON.1) will be set 1 to indicate one byte transmitting complete.

Reception is initiated by clearing and then setting REN (SCON.4) while RI (SCON..0) is 0. This condition tells the serial port controller that there is data to be shifted in. This process will continue until 8 bits have been received. Then the received flag RI will be set as 1. Note that REN will not be cleared via hardware. The user should first clear RI, clear REN and then set REN again via software to triggering the next byte reception.

12.2 Mode 1

Mode 1 supports asynchronous, full duplex serial communication. The asynchronous mode is commonly used for communication with PCs, modems or other similar interfaces. In Mode 1, 10 bits are transmitted (through TXD) or received (through RXD) including a start bit (logic 0), 8 data bits (LSB first) and a stop bit (logic 1). The baud rate is determined by the Timer 1. SMOD (PCON.7) setting 1 makes the baud rate double while Timer 1 is selected as the clock source. Figure 12–2 shows a simplified functional diagram of the serial port in Mode 1 and associated timings for transmitting and receiving.

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.

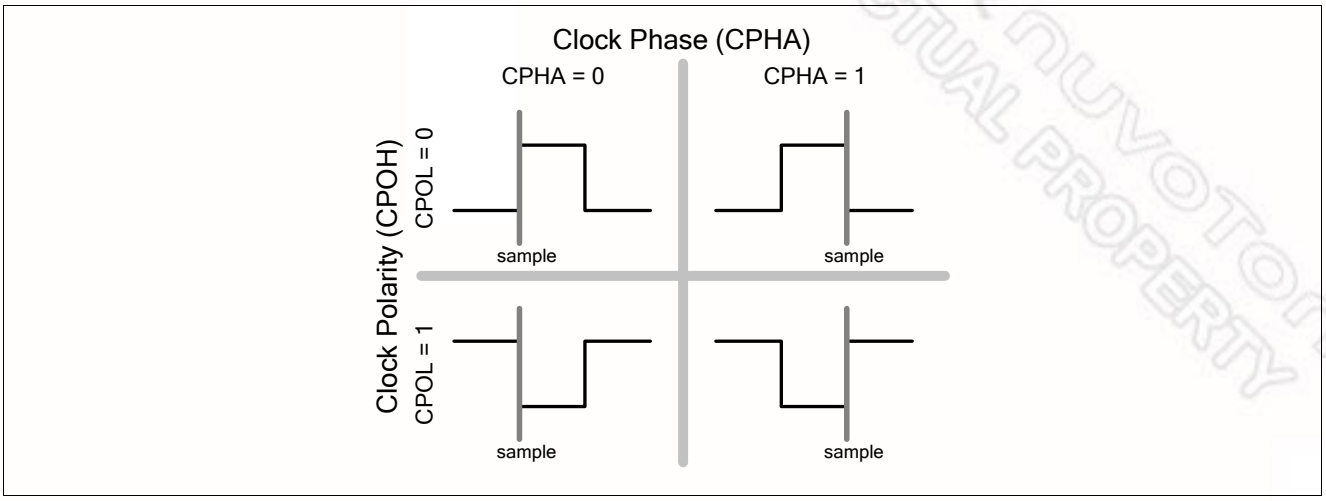

Figure 13–4 SPI Clock Format

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.

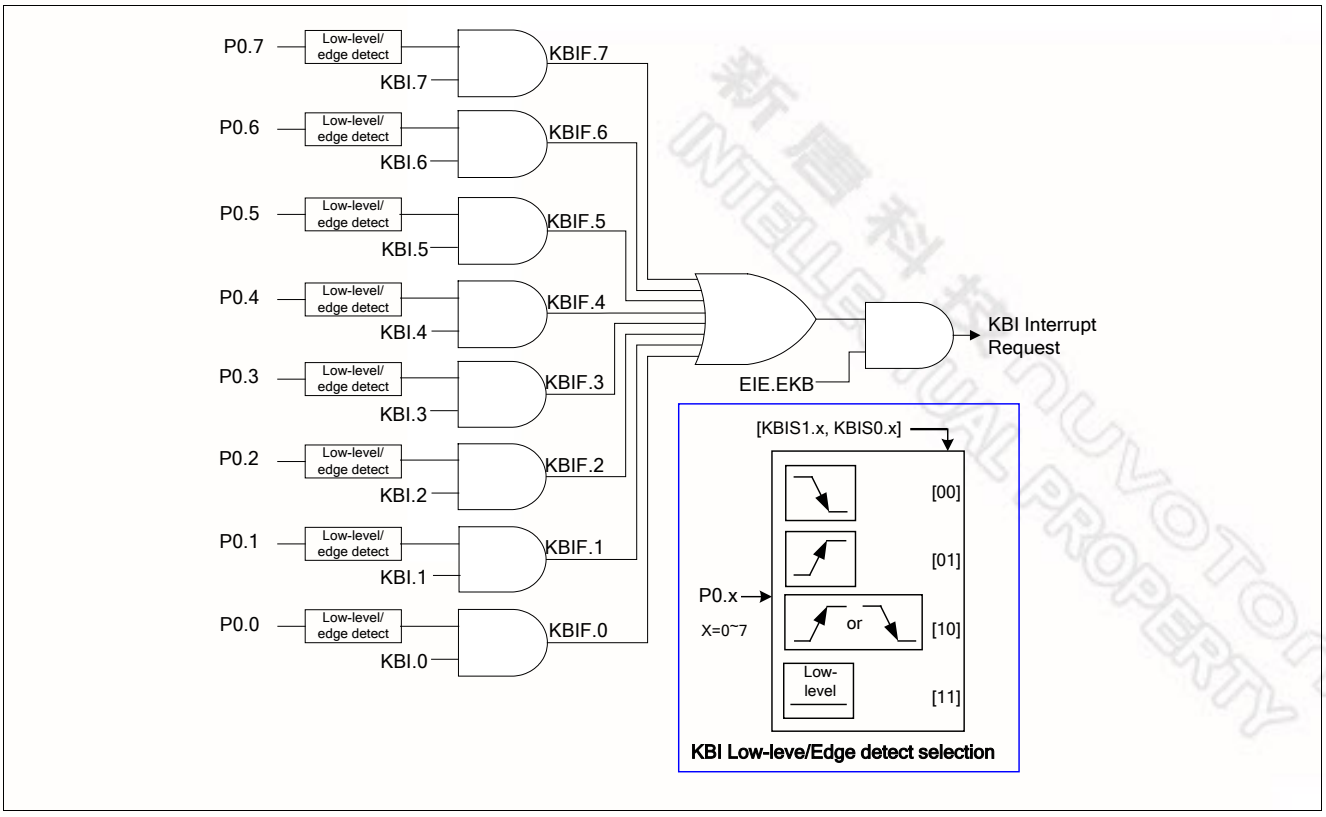

Figure 14-1 Keyboard Interrupt Detection

15 Analog-To-Digital Converter (ADC)

The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (V_{DAC}) 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 V_{DAC}, 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 V_{DAC} is compared to Vin again. If the input voltage is greater than V_{DAC}, 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.

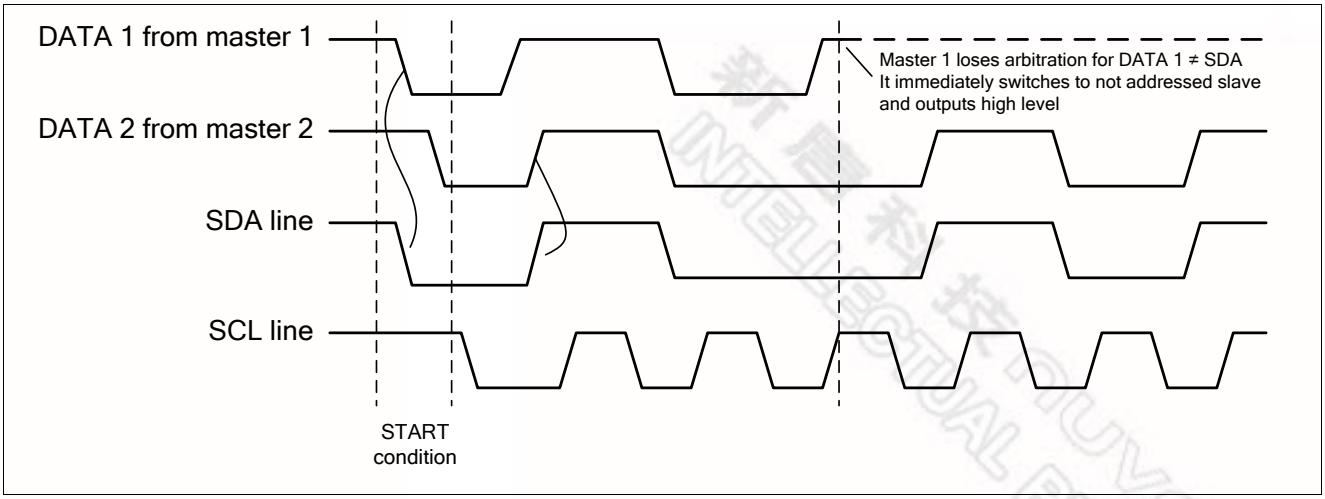

Figure 16–6 Arbitration Procedure of Two Masters

Since the control of I²C bus is decided solely by the address or master code and data sent by competing masters, there is no central master, nor any order of priority on the bus.

Slaves are not involved in the arbitration procedure.

16.3 Control Registers of I²C

There are five control registers to interface the I²C bus. They are I2CON, I2STA, I2DAT, I2ADDR, I2CLK, and I2TMR. These registers provide protocol control, status, data transmit and receive functions, clock rate configuration, and timeout notification. The following registers relate to I²C function.

I2CON – I²C Control

7	6	5	4	3	2	1	0
-	I2CEN	STA	STO	SI	AA	-	-
-	R/W	R/W	R/W	R/W	R/W	-	-

Address: C0H Reset value: 0000 0000B

Bit	Name	Description
7	-	**Reserved**
6	I2CEN	**I²C Bus Enable** 0 = I²C bus is disabled. 1 = I²C bus is enabled. Before enabling the I²C, Px.x and Px.x port latches should be set to logic 1. Once the I²C bus is enabled, SDA pin (Px.x) and SCL pin (Px.x) will be automatically switched to the open-drain mode. PxM2 and PxM1 registers will also be re-configured accordingly.

the data direction bit “read” (SLA+R). After the SLA+R byte is transmitted and an acknowledge bit has been returned, the SI flag is set again and I2STA is read as 40H. SI flag should then be cleared to receive data from the slave transmitter. If AA flag (I2CON.3) is set, the master receiver will acknowledge the slave transmitter. If AA is cleared, the master receiver will not acknowledge the slave and release the slave transmitter as a not addressed slave. After that, the master can generate a STOP condition or a repeated START condition to terminate the transmission or initial another one.

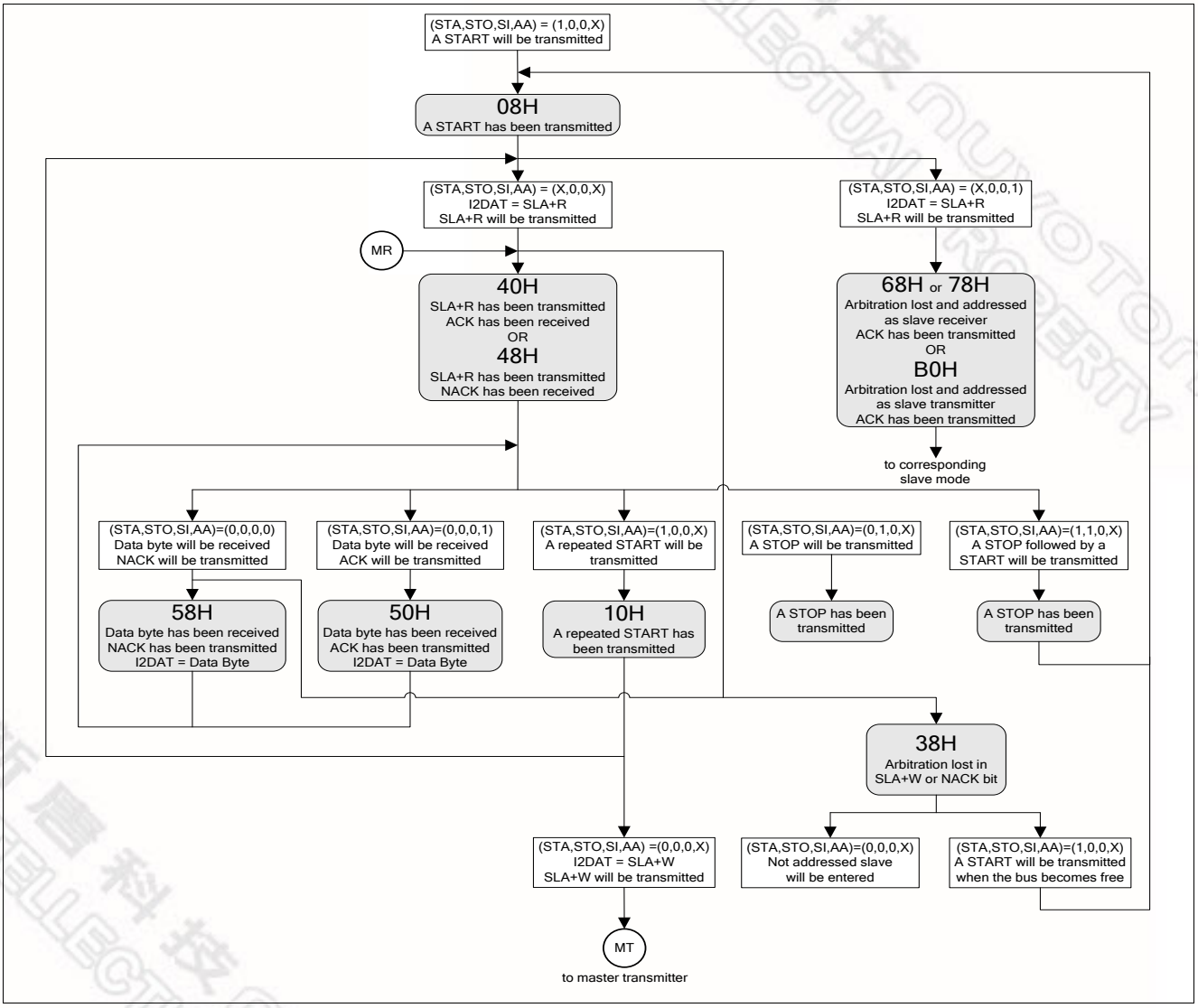

Figure 16–8 Flow and Status of Master Receiver Mode

16.4.3 Slave Receiver Mode

In Slave Receiver mode, several bytes of data are received form a master transmitter. Before a transmission is commenced, I2ADDR should be loaded with the address to which the device will respond when addressed by a master. I2CLK does not affect in slave mode. The AA bit should be set to enable acknowledging its own slave address or General

After the slave is addressed by SLA+W, it should clear its SI flag to transmit the data to the master transmitter. Normally the master receiver will return an acknowledge after every byte of data is transmitted by the slave. If the acknowledge is not received, it will transmit all "1" data if it continues the transaction. It becomes a not addressed slave. If the AA flag is cleared during a transaction, the slave transmit the last byte of data. The next transmitting data will be all "1" and the slave becomes not addressed.

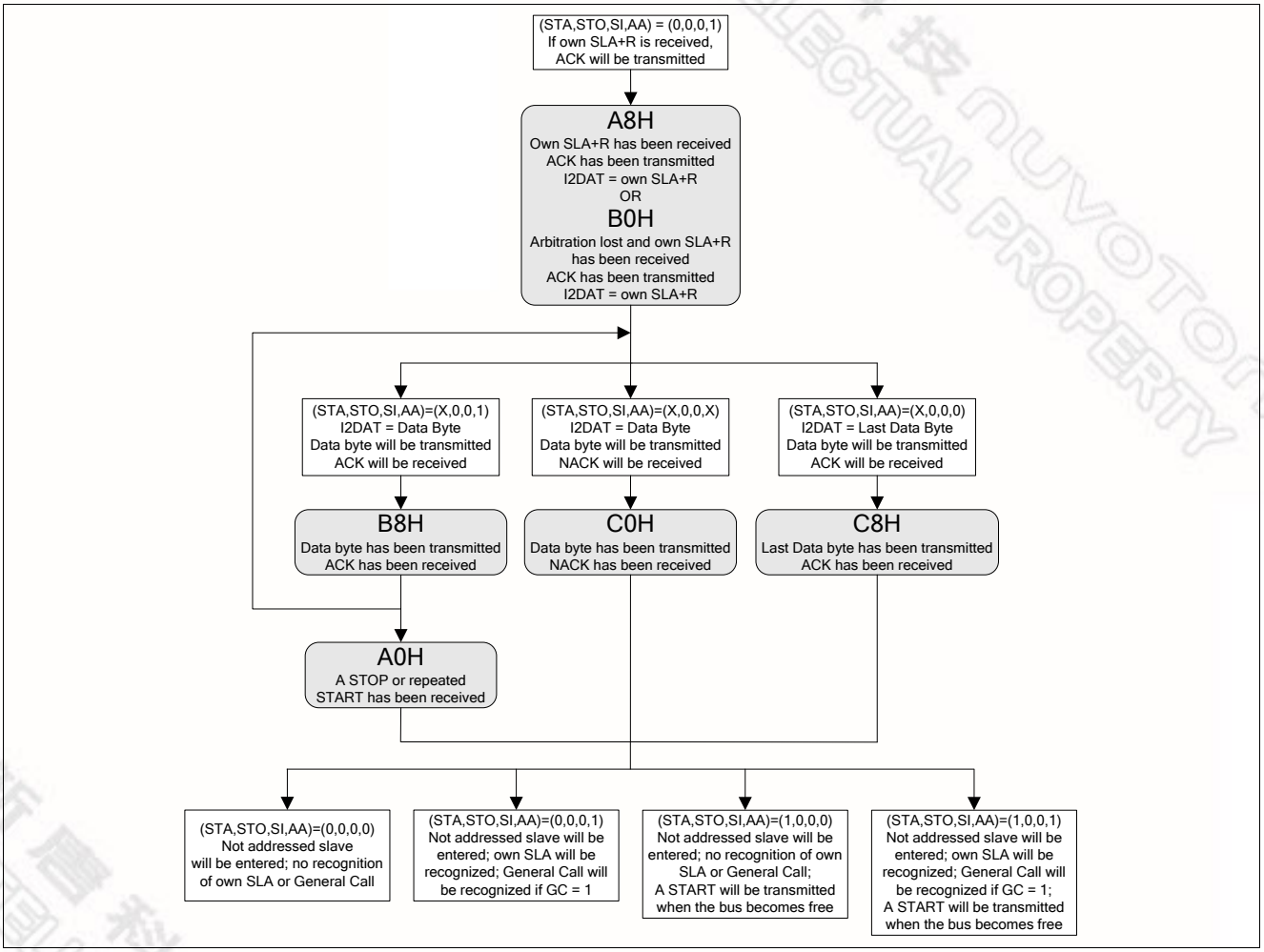

Figure 16–10 Flow and Status of Slave Transmitter Mode

16.4.5 General Call

The General Call is a special condition of slave receiver mode by sending all "0" data in slave address with data direction bit. The slave addressed by a General Call has different status codes in I2STA with normal slave receiver mode. The General Call may also be produced if arbitration is lost.

Bit	Name	Description
6	EADC	**Enable ADC Interrupt**
5	EBOD	**Enable BOD Interrupt**
4	ES	**Enable Serial Port (UART) Interrupt** 0 = Disable all UART interrupts. 1 = Enable interrupt generated by TI (SCON.1) or RI (SCON.0).
3	ET1	**Enable Timer 1 Interrupt** 0 = Disable Timer 1 interrupt 1 = Enable interrupt generated by TF1 (TCON.7).
2	EX1	**Enable External interrupt 1** 0 = Disable external interrupt 1. 1 = Enable interrupt generated by $\overline{INT1}$ pin (P1.4).
1	ET0	**Enable Timer 0 Interrupt** 0 = Disable Timer 0 interrupt 1 = Enable interrupt generated by TF0 (TCON.5).
0	EX0	**Enable External Interrupt 0** 0 = Disable external interrupt 0. 1 = Enable interrupt generated by $\overline{INT0}$ pin (P1.3).

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
7	ET2	0 = Disable Timer 2 Interrupt. 1 = Enable Timer 2 Interrupt.
6	ESPI	SPI interrupt enable: 0 = Disable SPI Interrupt. 1 = Enable SPI Interrupt.
5	EPWM	0 = Disable PWM Interrupt when external brake pin was braked. 1 = Enable PWM Interrupt when external brake pin was braked.

Bit	Name	Description
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
1	PKB	1 = Set interrupt priority of Keypad as higher priority level.
0	PI2	1 = Set interrupt priority of I^2C as higher priority level.

EIPH – Interrupt High Priority-1 Register

7	6	5	4	3	2	1	0
PT2H	PSPIH	PPWMH	PWDIH	-	-	PKBH	PI2H
R/W	R/W	R/W	R/W	-	-	R/W	R/W

Address: F7H Reset value: 0000 0000B

Bit	Name	Description
7	PT2H	1 = Set interrupt high priority of Timer 2 as the highest priority level.
6	PSPIH	1 = Set interrupt high priority of SPI as the highest priority level.

25 CONFIG Bits (CONFIG)

The N79E855/854 has several hardware configuration bytes, called CONFIG bits, which are used to configure the hardware options such as the security bits, clock system source, and so on. These hardware options can be re-configured through the Programmer/Writer or ISP modes. N79E855/854 have four CONFIG bits those are CONFIG0~3. Several functions which are defined by certain CONFIG bits are also available to be re-configured by certain SFR bits. Therefore, there is a need to LOAD such CONFIG bits into respective SFR bits. Such loading will occurs after resets. (Software reset will reload all CONFIG bits except CBS bit in CONFIG0.7) These SFR bits can be continuously controlled via user's software. Other resets will remain the values in these SFR bits unchanged.

Note: CONFIG bits marked as "-" should always keep unprogrammed.

25.1 CONFIG0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	-	LOCK	DFEN
R/W	-	-	-	-	-	R/W	R/W

Unprogrammed value: 1111 1111B

Bit	Name	Description
7	CBS	**CONFIG Boot Selection** This bit defines from which block MCU boots after all resets except software reset. 1 = MCU will boot from APROM after all resets except software reset. 0 = MCU will boot from LDROM after all resets except software reset.
6:2	-	**Reserved**
1	LOCK	**Chip Lock Enable** 1 = Chip is unlocked. All of APROM, LDROM, and Data Flash are not locked. Their contents can be read out through a parallel Programmer/Writer. 0 = Chip is locked. APROM, LDROM, and Data Flash are locked. Their contents read through parallel Programmer/Writer will become FFH. Note that CONFIG bytes are always unlocked and can be read. Hence, once the chip is locked, the CONFIG bytes cannot be erased or programmed individually. The only way to disable chip lock is to use the whole chip erase mode. However, all data within APROM, LDROM, Data Flash, and other CONFIG bits will be erased when this procedure is executed. If the chip is locked, it does not alter the ISP function.

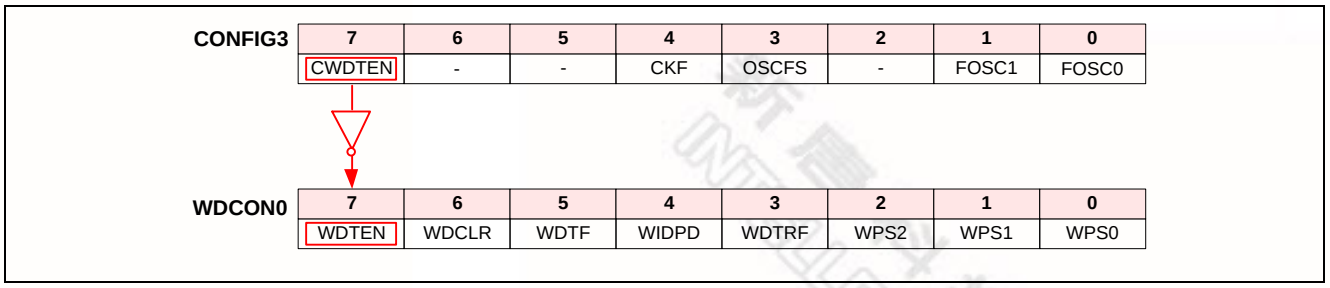

Figure 25–4 CONFIG3 Reset Reloading

27 In-Circuit Program (ICP)

The ICP (In-Circuit-Program) mode is another approach to access the Flash EPROM. There are only 3 pins needed to perform the ICP function. One is input /RST pin, which should be fed to GND in the ICP working period. One is clock input, shared with P1.7, which accepts serial clock from external device. Another is data I/O pin, shared with P1.6, that an external ICP program tool shifts in/out data via P1.6 synchronized with clock(P1.7) to access the Flash EPROM of N79E855/854.

Upon entry into ICP program mode, all pin will be set to quasi-bidirectional mode, and output to level "1". The N79E855/854 support programming of Flash EPROM (16K/8K bytes APROM EPROM), Data Flash memory (**128** bytes per page) and LDROM. User has the option to program the APROM, Data Flash and LDROM.

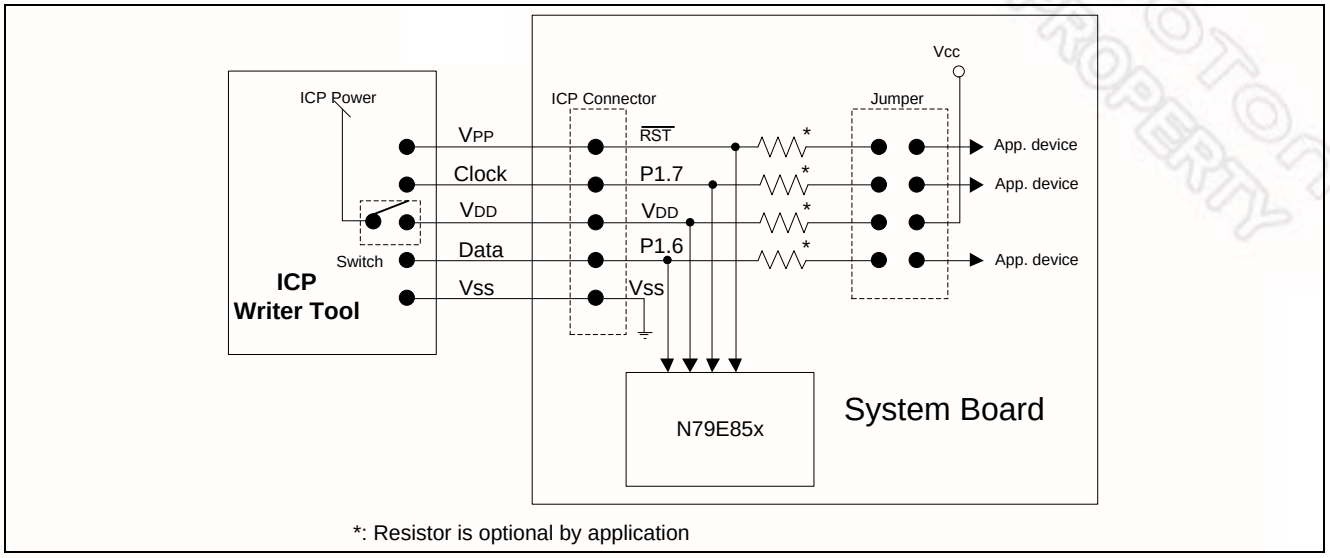

Figure 27–1 ICP Connection with N79E85xA

Note:

1. When using ICP to upgrade code, the /RST, P1.6 and P1.7 should be taken within design system board.
2. After program finished by ICP, to suggest system power should power off and remove ICP connector then power on.
3. It is recommended that user perform erase function and programming configure bits continuously without any interruption.