Nuvoton 8-bit 8051-based Microcontroller

N78E517A

Data Sheet

TABLE OF CONTENTS

1. DESCRIPTION

N78E517A is an 8-bit microcontroller, which has an in-system programmable Flash supported. The instruction set of N78E517A is fully compatible with the standard 8051. N78E517A contains a maximum 64k[1] bytes of main Flash APROM, in which the contents of the main program code can be updated by parallel Programmer/Writer or In System Programming (ISP) method which enables on-chip firmware updating. There is an additional 2.5k bytes called LDROM for ISP function. N78E517A has a configurable size of Data Flash[1] which is 64k-byte shared with APROM, accessed with ISP. N78E517A provides 256 bytes of SRAM, 1k bytes of auxiliary RAM (XRAM), four 8-bit bi-directional and bit-addressable I/O ports, an additional 8-bit bi-directional and bit-addressable port P4 for LQPF-48 package (PLCC-44, PQFP-44, and TQFP-44 just have low nibble 4 bits of P4 and DIP-40 does not have this additional P4), three 16-bit Timers/Counters, one UART, five PWM output channels, and one SPI. These peripherals equip with 11-source with 4-level priority interrupts capability. To facilitate programming and verification, the Flash inside the N78E517A allows the Program Memory to be programmed and read electronically. Once the code confirms, the user can lock the code for security.

N78E517A is built in a precise on-chip RC oscillator of 22.1184MHz/11.0592MHz selected by CONFIG setting, factory trimmed to ±1% at room temperature. N78E517A provides additional power monitoring detection such as power-on and Brown-out detection. It stabilizes the power-on/off sequence for a high reliability system design.

N78E517A microcontroller operation consumes a very low power. Two economic power modes to reduce power consumption, Idle mode and Power Down mode. Both of them are software selectable. The Idle mode turns off the CPU clock but allows continuing peripheral operation. The Power Down mode stops the whole system clock for minimum power consumption.

[1] Data Flash and APROM share 64k-byte space.

2. FEATURES

- Fully static design 8-bit CMOS microcontroller.
- Wide supply voltage of 2.4V to 5.5V and wide frequency from 4MHz to 40MHz.
- 12T mode compatible with the tradition 8051 timing.
- 6T mode supported for double performance.
- On-chip RC oscillator of 22.1184MHz/11.0592MHz, trimmed to ±1% at room temperature for the precise system clock.
- Maximum 64k bytes Flash APROM for the application program.
- 2.5k bytes Flash LDROM for ISP code.
- Configurable size of Data Flash, 64k-byte shared with APROM.
- In-System-Programmable (ISP) built in. ISP Erasing or programming supports wide operating voltage 3.0V to 5.5V.
- Flash 10,000 writing cycle endurance. Greater than 10 years data retention under 85℃.
- 256 bytes of on-chip RAM.
- 1k bytes of on-chip auxiliary RAM (XRAM).
- 64k bytes Program Memory address space and 64k bytes Data Memory address space.
- Maximum five 8-bit general purpose I/O ports pin-to-pin compatible with standard 8051, additional $\overline{INT2}$ and $\overline{INT3}$ on packages except DIP-40.
- Three 16-bit Timers/Counters.
- One dedicate timer for Power Down mode waking-up.
- One full-duplex UART port.
- Five pulse width modulated (PWM) output channels.
- One SPI communication port.
- 11-source, 4-priority-level interrupts capability.
- Programmable Watchdog Timer.
- Power-on reset.
- Brown-out detection interrupt and reset, 4-level selected.
- Supports software reset function.
- Built-in power management with Idle mode and Power Down mode.

- Code lock for data security.
- Much lower power consumption than other standard 8051 productions.
- Industrial temperature grade, -40℃to +85℃ range.
- Strong ESD, EFT immunity.
- Development Tool:
  - Parallel Programmer/Writer.
  - Nuvoton 8-bit Microcontroller ISP Writer.
- Package:

Part Number	APROM	Data Flash	Package
N78E517ADG	Shared 64k bytes		40-pin DIP
N78E517APG			44-pin PLCC
N78E517AFG			44-pin PQFP
N78E517ATG			44-pin TQFP
N78E517ALG			48-pin LQFP

3. BLOCK DIAGRAM

Figure 3–1 shows the functional block diagram of N78E517A. It gives the outline of the device. The user can find all the device's peripheral functions in the diagram.

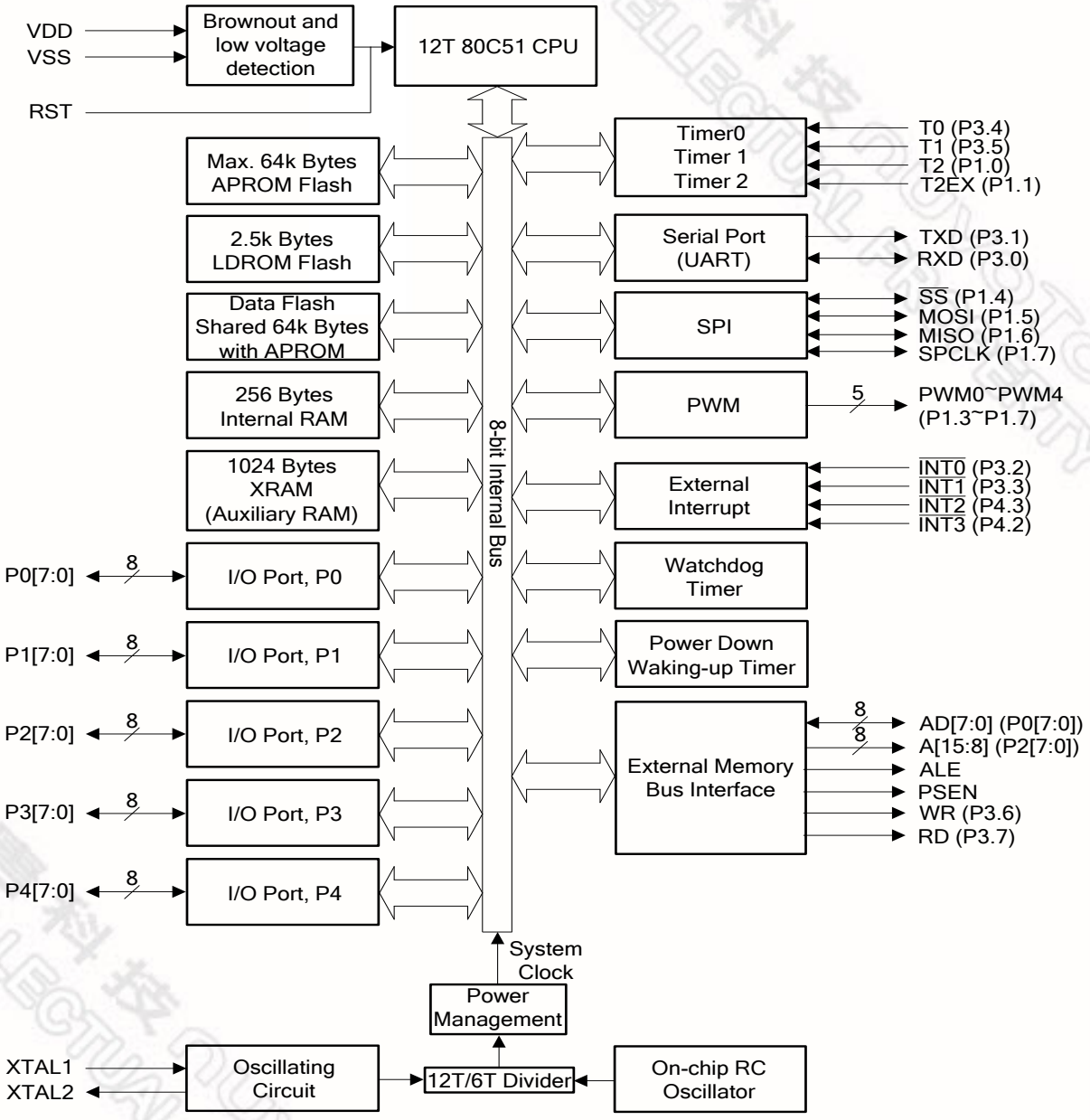


Figure 3–1. N78E517A Function Block Diagram

4. PIN CONFIGURATIONS

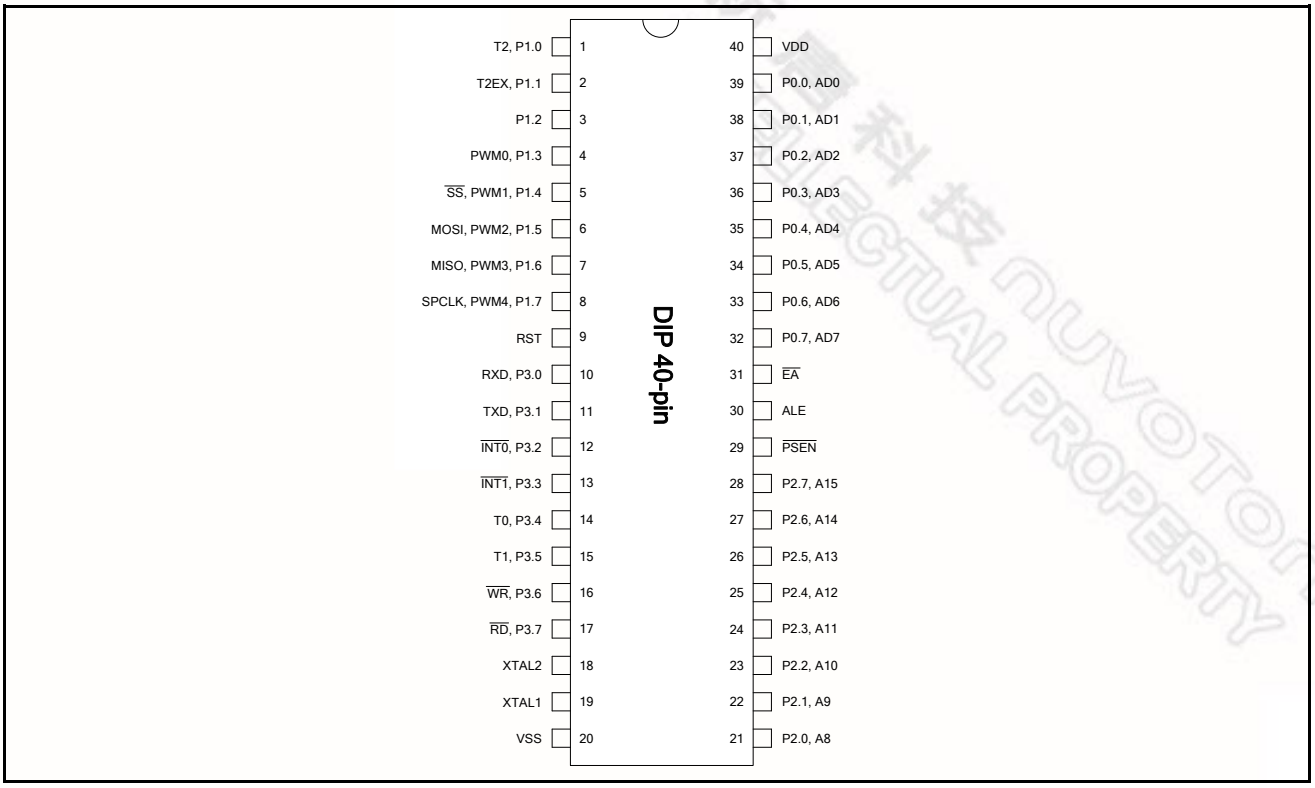


Figure 4–1. Pin Assignment of DIP 40-Pin

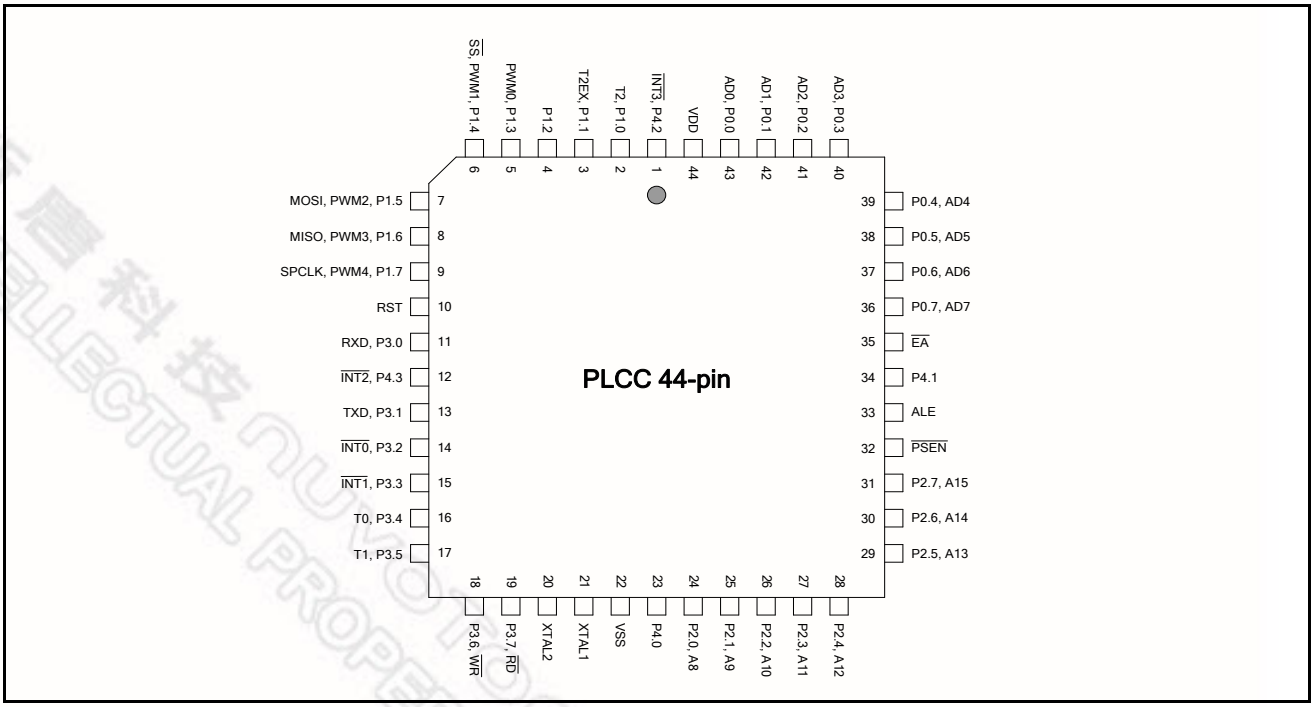


Figure 4–2. Pin Assignment of PLCC 44-Pin

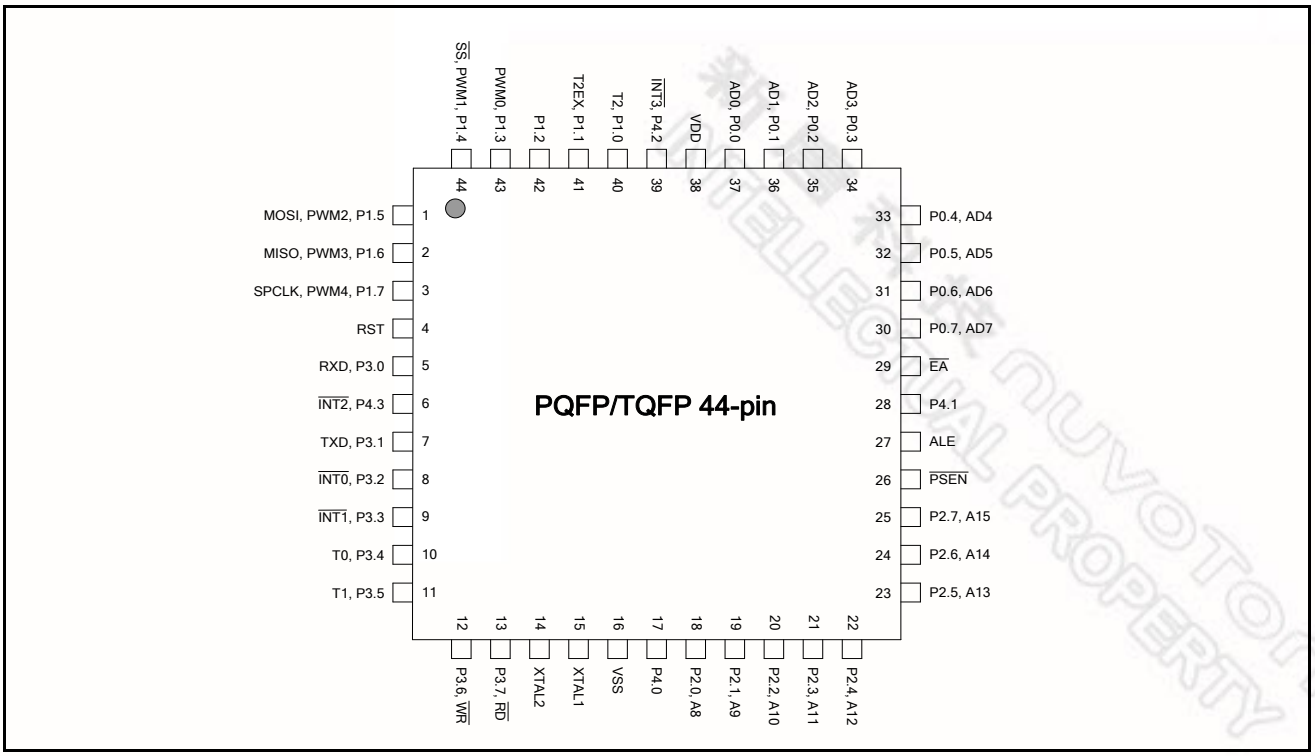


Figure 4–3. Pin Assignment of PQFP/TQFP 44-Pin

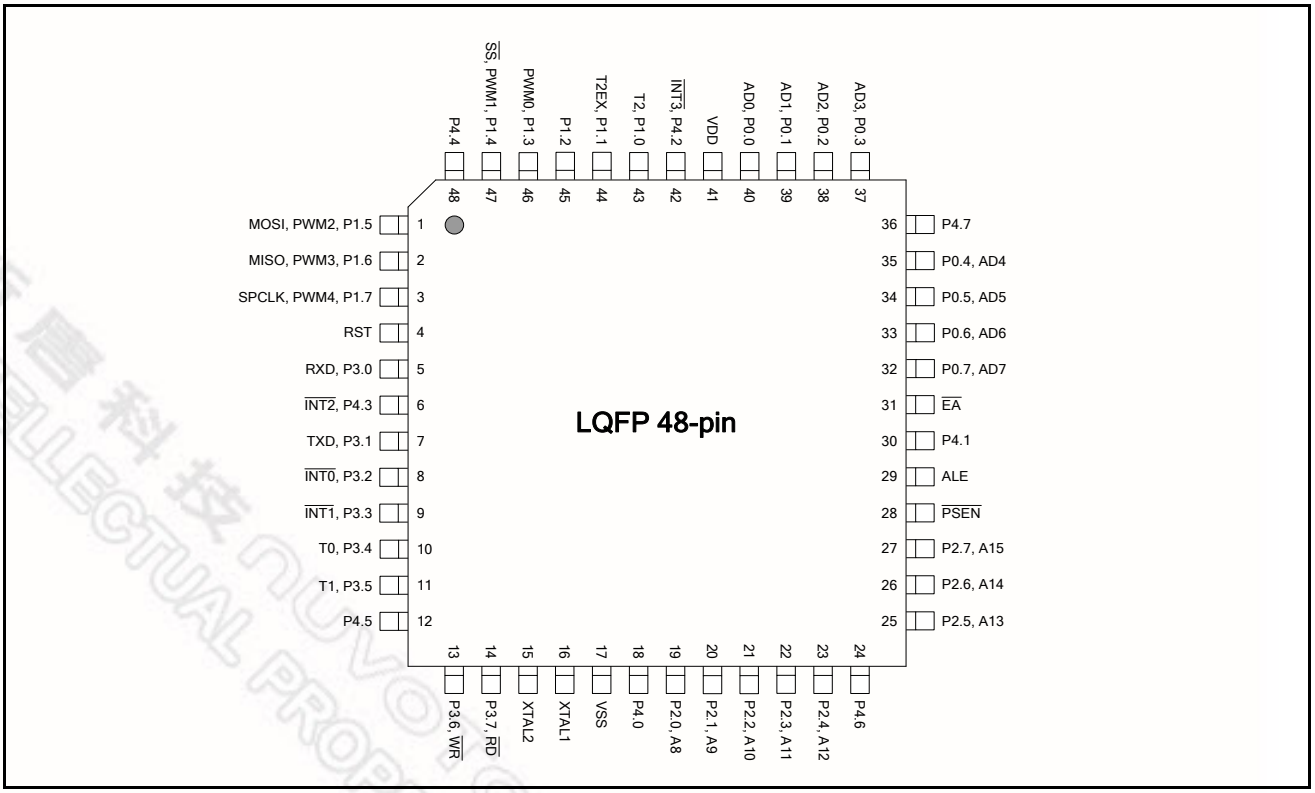


Figure 4–4. Pin Assignment of LQFP 48-Pin

Table 4–1. Pin Description

Pin number				Symbol	Alternate Function		Type[1]	Description
DIP	PLCC	PQFP TQFP	LQFP		1	2		
19	21	15	16	XTAL1			I (ST)	**CRYSTAL1:** This is the input pin to the internal inverting amplifier. The system clock is from external crystal or resonator when FOSC (CONFIG3.1) is logic 1 by default. ***A 0.1µF capacitor is recommended to be added on XTAL1 pin to gain the more precise frequency of the internal RC oscillator frequency if it is selected as the system clock source.***
18	20	14	15	XTAL2			O	**CRYSTAL2:** This is the output pin from the internal inverting amplifier. It emits the inverted signal of XTAL1. While on-chip RC oscillator is used, float XTAL2 pin always.
40	44	38	41	VDD			P	**POWER SUPPLY:** Supply voltage V_{DD} for operation.
20	22	16	17	VSS			P	**GROUND:** Ground potential.
31	35	29	31	$\overline{EA}$			I	**EXTERNAL ACCESS ENABLE:** To force $\overline{EA}$ low will make the CPU execute the external Program Memory. The address and data will be presented on the bus P0 and P2. If the $\overline{EA}$ pin is high, CPU will fetch internal code unless the Program Counter addresses the area out of the internal Program Memory. It will make CPU run external Program Memory continuously. $\overline{EA}$ possesses reset lock. After all reset, the $\overline{EA}$ state will be latched and any state change of this pin after reset will not switch between internal and external Program Memory execution. ***The user should take care of this pin from floating but connecting to V_{DD} directly if internal Program Memory is used.***
30	33	27	29	ALE			O	**ADDRESS LATCH ENABLE:** ALE is used to enable the address latch that separates the address from the data on Port 0. ALE runs at 1/6 of the Fosc[2]. An ALE pulse is omitted always. The user can turn ALE off by setting ALEOFF (AUXR.0) to reduce EMI. Setting ALEOFF will just make ALE activating only during external memory access through a MOVC or MOVX instruction. ALE will stay high in other conditions.
29	32	26	28	$\overline{PSEN}$			O	**PROGRAM STORE ENABLE:** $\overline{PSEN}$ strobes the external Program Memory. When internal Program Memory access is performed, there will be no $\overline{PSEN}$ strobe signal output from this pin.

Table 4–1. Pin Description

Pin number				Symbol	Alternate Function		Type[1]	Description
DIP	PLCC	PQFP TQFP	LQFP		1	2		
9	10	4	4	RST			I (ST)	**RESET:** RST pin is a Schmitt trigger input pin for hardware device reset. A high on this pin for two machine-cycles while the system clock is running will reset the device. RST pin has an internal pull-down resistor allowing power-on reset by simply connecting 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	

Table 4–1. Pin Description

Pin number				Symbol	Alternate Function		Type[1]	Description
DIP	PLCC	PQFP TQFP	LQFP		1	2		
28	31	25	27	P2.7		A15	I/O	
10	11	5	5	P3.0	RXD		I/O	**PORT3:** Port 3 is an 8-bit quasi bi-directional I/O port. Its multifunction pins are for RXD, TXD, $\overline{INT0}$, $\overline{INT1}$, T0, T1, $\overline{WR}$, and $\overline{RD}$.
11	13	7	7	P3.1	TXD		I/O	
12	14	8	8	P3.2	$\overline{INT0}$		I/O	
13	15	9	9	P3.3	$\overline{INT1}$		I/O	
14	16	10	10	P3.4	T0		I/O	
15	17	11	11	P3.5	T1		I/O	
16	18	12	13	P3.6	$\overline{WR}$		I/O	
17	19	13	14	P3.7	$\overline{RD}$		I/O	
-	23	17	18	P4.0			I/O	**PORT4**[3]**:** Port 4 is an 8-bit quasi bi-directional I/O port. It also possesses bit-addressable feature as P0-P3. P4.2 and P4.3 are alternative function pins of $\overline{INT3}$ and $\overline{INT2}$.
-	34	28	30	P4.1			I/O	
-	1	39	42	P4.2	$\overline{INT3}$		I/O	
-	12	6	6	P4.3	$\overline{INT2}$		I/O	
-	-	-	48	P4.4			I/O	
-	-	-	12	P4.5			I/O	
-	-	-	24	P4.6			I/O	
-	-	-	36	P4.7			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.
[2] While switching to 6T mode, ALE will run at 1/3 of Fosc.
[3] A full 8-bit P4 is just on LQPF-48 package. PLCC-44, PQFP-44, and TQFP-44 just have low nibble 4 bits of P4. DIP-40 does not have this additional P4.

The application circuit is shown below. The user is recommended follow the circuit enclosed by gray blocks to achieve the most stable and reliable operation of MCU especially in a noisy power environment for a healthy EMS immunity. If internal RC oscillator is used as the system clock, a 0.1μF capacitor should be added to gain a precise RC frequency.

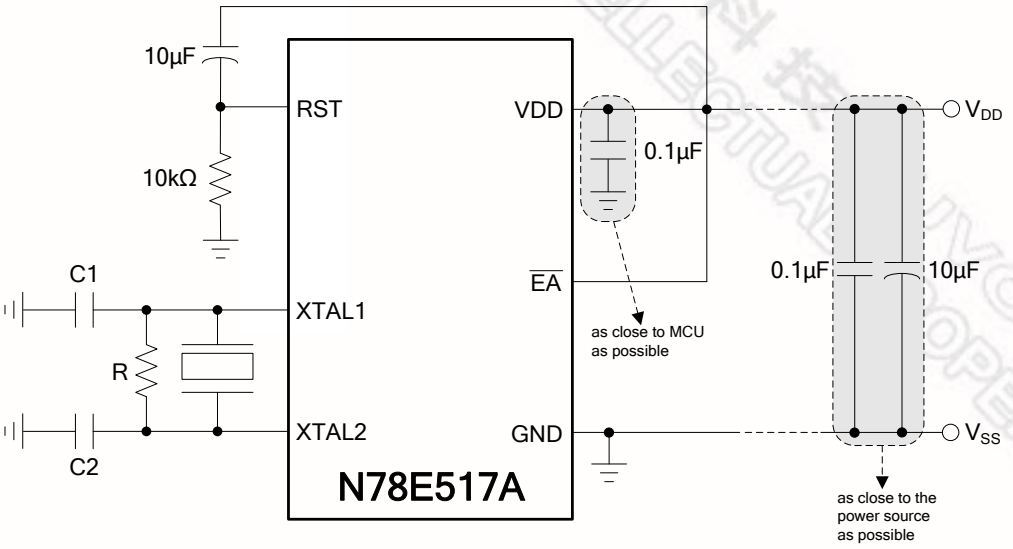


Figure 4–5. Application Circuit for Execution of Internal Program Code with External Crystal

Crystal Frequency	R	C1	C2
4MHz-33MHz	Without	Depend on crystal specifications	
33MHZ-40MHz	5kΩ-10kΩ		

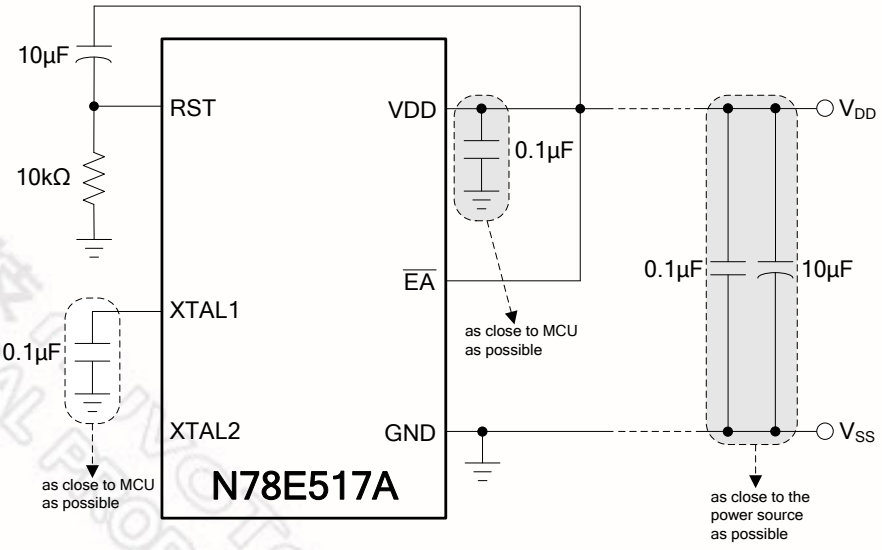


Figure 4–6. Application Circuit for Execution of Internal Program Code with Internal RC Oscillator

5. MEMORY ORGANIZATION

A standard 8051 based MCU divides the memory into two different sections, the Program Memory and the Data Memory. The Program Memory is used to store the instruction codes, whereas the Data Memory is used to store data or variations during the program execution.

Data Memory occupies a separate address space from Program Memory. In N78E517A, there are 256 bytes of internal scratch-pad RAM and up to 64k bytes of memory space for external Data Memory. The MCU generates the 16-bit or 8-bit addresses, read and write strobe signals ($\overline{RD}$ and $\overline{WR}$, respectively) during external Data Memory access. For many applications which need more internal RAM, N78E517A possesses on-chip 1k bytes of RAM (called XRAM) accessed by MOVX instruction.

The whole embedded flash is divided into 4 banks, APROM for storage of user's program code, Data Flash for parameter data storage, LDROM for ISP program and CONFIG bytes. Each bank is accumulated page by page and the page size is 256 bytes. The flash control unit supports Page Erase, Byte Program, and Byte Read modes. The external writer tools though specific I/O pins or the internal ISP (In System Programming) function can both performs these modes.

5.1 Internal Program Memory

Program Memory is the one, which stores the program codes to execute, as shown in Figure 5–1. While $\overline{EA}$ pin is pulled high and after any reset, the CPU begins execution from location 0000H where should be the starting point of the user's application code. To service the interrupts, the interrupt service locations (called interrupt vectors) should be located in the Program Memory. Each interrupt is assigned with a fixed location in the Program Memory. The interrupt causes the CPU to jump to that location with where it commences execution of the interrupt service routine (ISR). External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at an interval of 8 bytes: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. However longer service routines should use a JMP instruction to skip over subsequent interrupt locations if other interrupts are in use.

N78E517A provides two internal Program Memory bank APROM and LDROM. Although they both behave the same as the standard 8051 Program Memory, they play different rules according to their ROM size. The APROM on N78E517A can be 64k-byte size maximum. The user's main program code is normally put inside.

All instructions are fetched for execution from this area. The MOVC instruction can also read this flash memory region.

N78E517A supports the other individual Program Memory bank called LDROM besides APROM. The main function of LDROM is to store the ISP application program. User may develop the ISP in LDROM for updating APROM content. The program in APROM can also re-program LDROM. For ISP details and configuration bit setting related with APROM and LDROM, see Section 18. "IN SYSTEM PROGRAMMING (ISP)" on page 92. Note that because APROM and LDROM are hardware individual blocks, consequently if CPU reboots from LDROM, CPU will automatically re-vectors Program Counter 0000H to the LDROM start address. Therefore, CPU accounts the LDROM as an independent Program Memory and all interrupt vectors are independent from APROM.

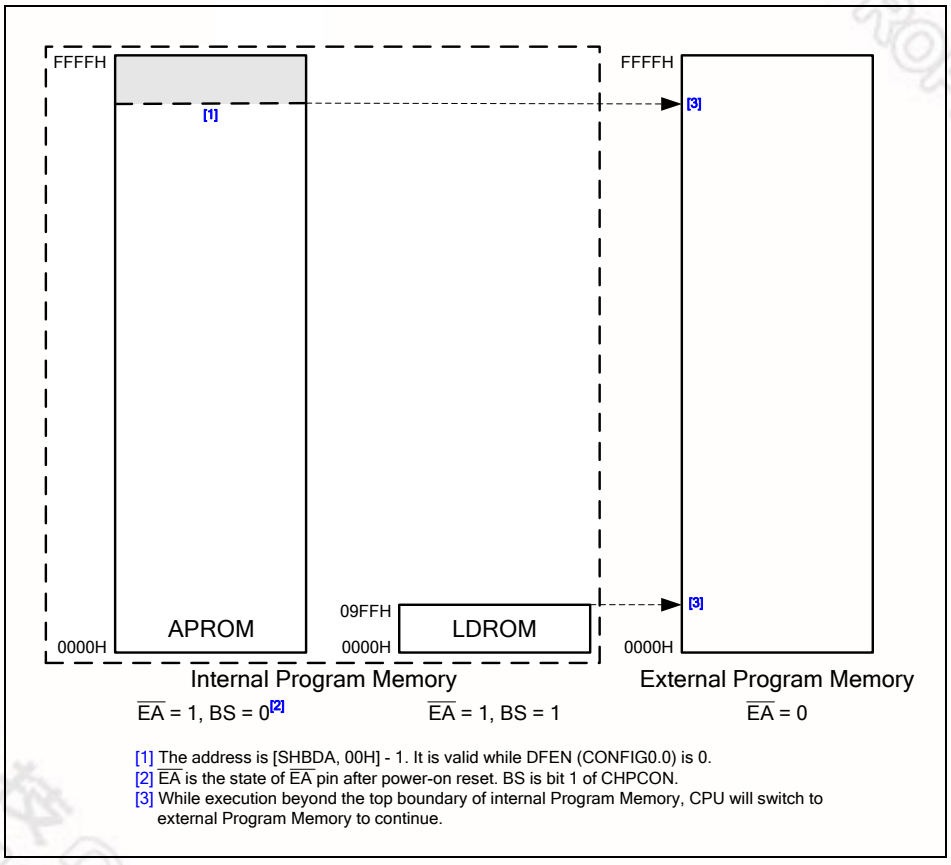


Figure 5–1. N78E517A Program Memory Structure

5.2 External Program Memory

N78E517A is a 16-bit address-width CPU. It can address 64k-byte program code. Besides the internal Program Memory, the external additional Program Memory is also can be used. The external program addressing will be executed under cases below,

1. The PC (Program Counter) value is beyond the boundary size address of APROM or LDROM while $\overline{EA}$ pin is pulled high during power on. The CPU will continue to fetch the external Program Memory.

2. While $\overline{EA}$ pin is pulled low during power on period, The CPU will run totally 64k-byte code externally.

While the external mode is running, the P0 and P2 will produce address and data signals to fetching external Program Memory. In this case, P0 and P2 cannot be general purpose I/O anymore. $\overline{PSEN}$ will also toggle out to strobe the external Program Memory. For the hardware circuit for external program execution, see Figure 5–2. Program Memory Interface.

For security $\overline{EA}$ pin state will be locked after power on. The user cannot switch the program running internally or externally by $\overline{EA}$ after power on. The other design for data security is MOVC lock enable (MOVCL, CONFIG0.2). While this bit is set 0, The external Program Memory code is inhibited to read internal APROM or LDROM contents through MOVC instruction.

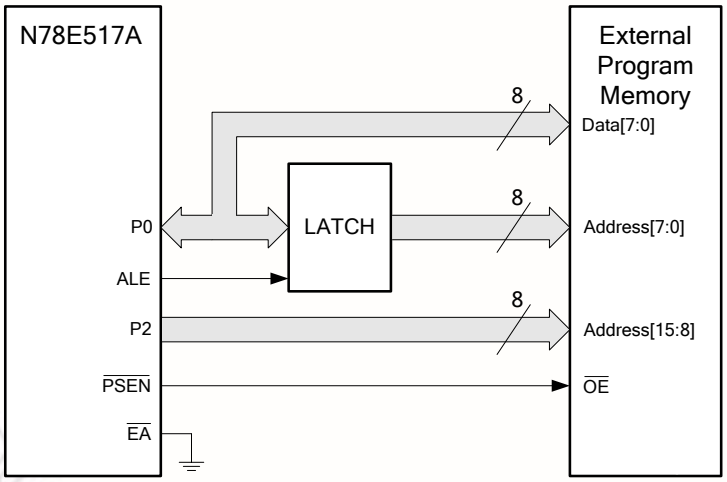


Figure 5–2. Program Memory Interface

5.3 Internal Data Memory

Figure 5–3 shows the internal and external Data Memory spaces available on N78E517A. 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 either byte-addressable or 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. It 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.

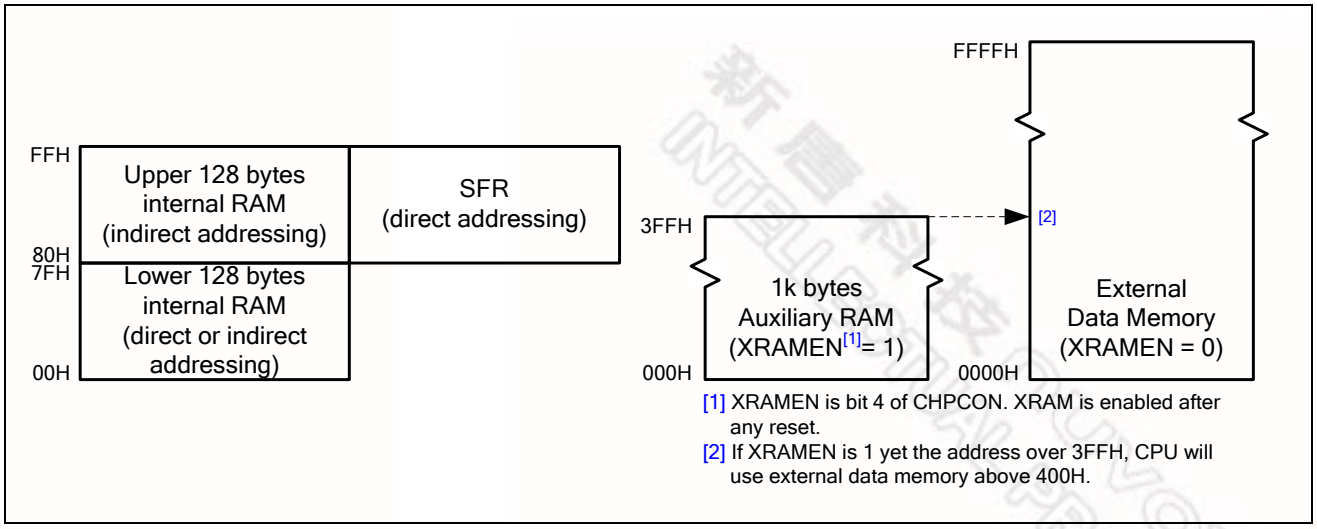


Figure 5–3. N78E517A Data Memory Structure

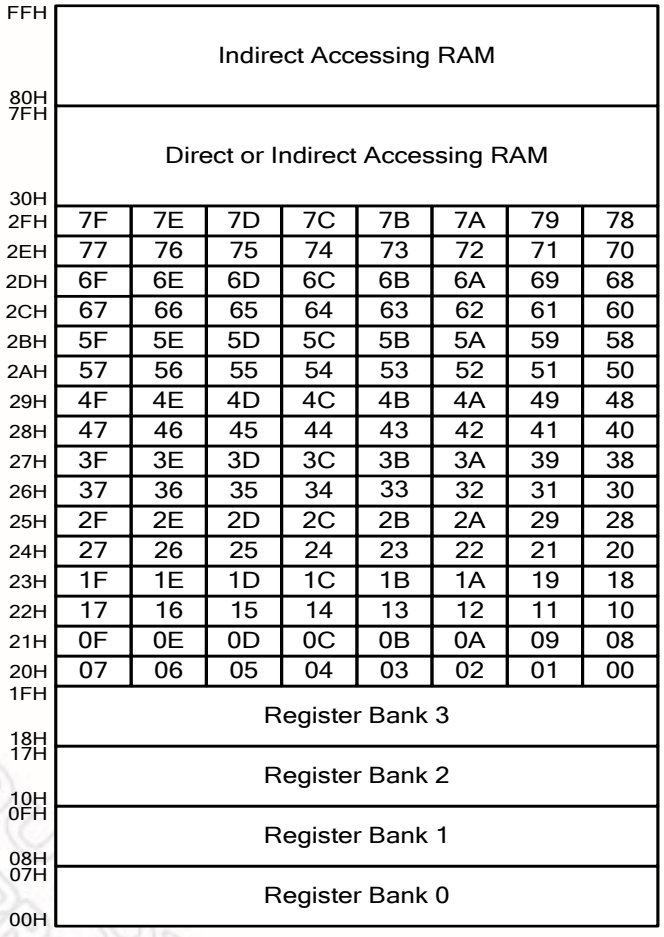


Figure 5–4. 256 bytes Internal RAM Addressing

5.4 On-chip XRAM

N78E517A provides additional on-chip auxiliary RAM called XRAM to enlarge RAM space. The 1024 bytes of XRAM (000H to 3FFH) are indirectly accessed by move external instruction MOVX. For details, see Section 8. "AUXILIARY RAM (XRAM)" on page 30.

5.5 External Data Memory

Access to external Data Memory can use either a 16-bit address (using 'MOVX @DPTR') or an 8-bit address (using 'MOVX @Ri', i = 0 or 1). For another 1k-byte XRAM exists, remember the bit XRAMEN (CHPCON.4) should be cleared as logic 0 in order to access the range of 000H to 3FFH address of the external Data Memory.

16-bit addresses are often used to access up to 64k bytes of external RAM. Whenever a 16-bit address is used, P0, P2, P3.7 and P3.6 serve as the low byte address/data, the high byte address, $\overline{RD}$ strobe and $\overline{WR}$ strobe signals respectively. Meanwhile the pins listed above cannot be used as general purpose I/O during external Data Memory access.

8-bit addresses are often used in conjunction with one or more other I/O lines to page the RAM. For example, if a 1k-byte external RAM is used, Port 0 serves as a multiplexed address/data bus to the RAM, and 2 pins of Port 2 are used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. In 8-bit addressing mode, P2 pins other than the two pins for RAM paging are free for general purpose I/O usage. It facilitates P2 application. Of course, the user may use any other I/O lines instead of P2 to page the RAM.

In all cases, the low byte of the address is time-multiplexed with the data byte on Port 0. ALE (Address Latch Enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated. During any access to external memory, the CPU writes 0FFH to the Port 0 latch (P0 in SFRs), thus obliterating whatever information the Port 0 SFR may have been holding.

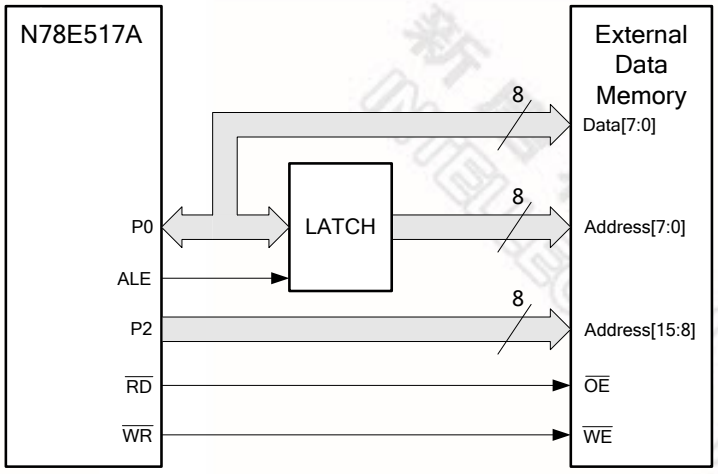


Figure 5–5. Data Memory Interface

5.6 On-chip Non-volatile Data Flash

N78E517A 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 92 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 software adjustable on N78E517A by updating the content of SHBDA. SHBDA[7:0] represents the high byte of 16-bit Data Flash start address and the low byte are 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 protect while a write to SHBDA is required. Be aware that if CHBDA is 00H, the Data Flash size will be 64k bytes and there will be no APROM. CPU will execute codes in the external Program Memory.

The CONFIG bit DFEN (CONFIG0.0) should be programmed as a 0 before access the Data Flash block. If DFEN remains its unprogrammed value 1, APROM will occupy whole 64k-byte block.

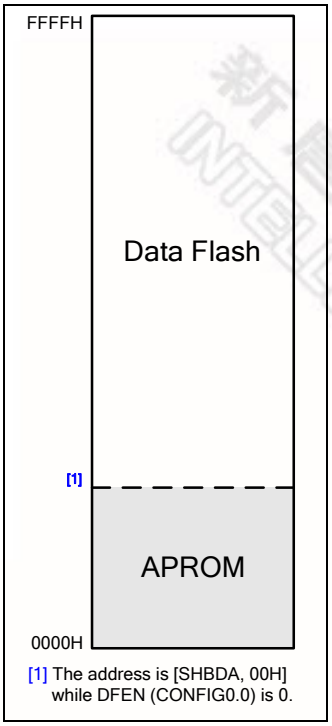


Figure 5–6. N78E517A Data Flash

CONFIG0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	MOVCL	LOCK	DFEN
r/w	-	-	-	-	r/w	r/w	r/w

unprogrammed value: 1111 1111b

Bit	Name	Description
0	DFEN	**Data Flash enable.** 1 = There is no Data Flash space. The APROM size is 64k-byte. 0 = Data Flash exists. The Data Flash and APROM share 64k bytes depending on SHBDA setting.

CONFIG1

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

unprogrammed value: 1111 1111b

Bit	Name	Description
7:0	CHBDA[7:0]	**CONFIG high byte of Data Flash starting address.** This byte is valid only when DFEN (CONFIG0.0) being 0 condition. It is used to determine the starting address of the Data Flash.

[1] Note that there will be no APROM if setting CHBDA 00H. CPU will execute codes in the external Program Memory.

SHBDA – SFR High Byte of Data Flash Starting Address (TA protected)

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

Address: 9CH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

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. SPECIAL FUNCTION REGISTER (SFR)

The N78E517A uses Special Function Registers (SFRs) to control and monitor peripherals and their modes. The SFRs reside in the register locations 80-FFH and are accessed by direct addressing only. Some of the SFRs are bit-addressable. It is very useful in cases where users would like to modify a particular bit directly without changing other bits. Those which are bit-addressable SFRs end their addresses as 0H or 8H. N78E517A contains all the SFRs presenting in the standard 8051. However some additional SFRs are built in. Therefore, some of unused bytes in the original 8051 have been given new functions. The SFRs is listed below.

Table 6–1. N78E517A Special Function Registers Mapping

F8	-	-	-	-	-	-	-	-	**FF**
F0	**B**	-	-	SPCR	SPSR	SPDR	-	-	**F7**
E8	-	-	-	-	-	-	-	-	**EF**
E0	**ACC**	-	-	-	-	-	-	-	**E7**
D8	**P4**	PWMP	PWM0	PWM1	PWMCON0	PWM2	PWM3	-	**DF**
D0	**PSW**	-	-	-	-	-	-	-	**D7**
C8	**T2CON**	T2MOD	RCAP2L	RCAP2H	TL2	TH2	PWMCON1	PWM4	**CF**
C0	**XICON**	-	-	-	-	-	-	TA	**C7**
B8	**IP**	-	IPH	EIPH	EIP	EIE	-	-	**BF**
B0	**P3**	-	-	-	-	-	-	-	**B7**
A8	**IE**	-	WDCON	PDCON	PMC	-	ISPFD	ISPCN	**AF**
A0	**P2**	XRAMAH	-	-	ISPTRG	-	ISPAL	ISPAH	**A7**
98	**SCON**	SBUF	-	-	SHBDA	-	-	CHPCON	**9F**
90	**P1**	-	-	-	-	-	RSR	-	**97**
88	**TCON**	TMOD	TL0	TL1	TH0	TH1	AUXR	-	**8F**
80	**P0**	SP	DPL	DPH	-	-	P0OR	PCON	**87**

In Bold	bit-addressable
-	reserved

Note that the reserved SFR addresses must be kept in their own initial states. Users should never change their values.

Table 6–2. N78E517A SFR Description 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) $\overline{INT2}$	(DA) $\overline{INT3}$	(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) $\overline{C/T2}$	(C8) $\overline{CP/RL2}$	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) $\overline{RD}$	(B6) $\overline{WR}$	(B5) T1	(B4) T0	(B3) $\overline{INT1}$	(B2) $\overline{INT0}$	(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

Table 6–2. N78E517A SFR Description and Reset Values

Symbol	Definition	Address	MSB							LSB[1]	Reset Value[2]
CHPCON[3]	Chip control	9FH	SWRST	ISPF	LDUEN	XRAMEN	-	-	BS	ISPEN	Software[6], 0001 00U0b Others, 0001 00X0b
SHBDA[3]	SFR high byte of Data Flash starting address	9CH									XXXX XXXXb[6]
SBUF	Serial buffer	99H									0000 0000b
SCON	Serial control	98H	(9F) SM0	(9E) SM1	(9D) SM2	(9C) REN	(9B) TB8	(9A) RB8	(99) TI	(98) RI	0000 0000b
RSR	Reset status register	96H	-	-	-	-	-	BORF	-	SWRF	Power-on, 0000 0000b Brown-out, 0000 010Ub Software, 0000 0U01b Others, 0000 0U0Ub
P1	Port 1	90H	(97) PWM4 SPCLK	(96) PWM3 MISO	(95) PWM2 MOSI	(94) PWM1 $\overline{SS}$	(93) PWM0	(92)	(91) T2EX	(90) T2	1111 1111b
AUXR	Auxiliary register	8EH	-	-	-	-	-	-	-	ALEOFF	0000 0000b
TH1	Timer 1 high byte	8DH									0000 0000b
TH0	Timer 0 high byte	8CH									0000 0000b
TL1	Timer 1 low byte	8BH									0000 0000b
TL0	Timer 0 low byte	8AH									0000 0000b
TMOD	Timer 0 and 1 mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000 0000b
TCON	Timer 0 and 1control	88H	(8F) TF1	(8E) TR1	(8D) TF0	(8C) TR0	(8B) IE1	(8A) IT1	(89) IE0	(88) IT0	0000 0000b
PCON	Power control	87H	SMOD	-	-	POF	GF1	GF0	PD	IDL	Power-on, 0001 0000b Others, 000U 0000b
P0OR	P0 option register	86H	-	-	-	-	-	-	-	P0UP	0000 0000b
DPH	Data pointer high byte	83H									0000 0000b
DPL	Data pointer low byte	82H									0000 0000b
SP	Stack pointer	81H									0000 0111b
P0	Port 0	80H	(87) A7	(86) A6	(85) A5	(84) A4	(83) A3	(82) A2	(81) A1	(80) A0	1111 1111b

[1] ( ) item means the bit address in bit-addressable SFRs.
[2] Reset value symbol description. 0: logic 0, 1: logic 1, U: unchanged, X: see **[4]** - **[6]**.
[3] These SFRs have TA protected writing.
[4] BOF has different power-on reset value according to CBODEN (CONFIG2.7) and CBORST (CONFIG2.4). See Table 21–1. BOF Reset Value.
[5] BOS is a read-only flag decided by V_{DD} level while Brown-out detection is enabled.
[6] These SFRs have bits which are initialized after specified reset by loading certain bits in CONFIG bytes. See Section 24. "CONFIG BYTES" on page 117 for details.

Note that bits marked in "-" must be kept in their own initial states. Users should never change their values.

7. GENERAL 80C51 SYSTEM CONTROL

A or ACC – Accumulator (bit-addressable)

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

Address: E0H reset value: 0000 0000b

Bit	Name	Description
7:0	ACC[7:0]	**Accumulator.** The A or ACC register is the standard 8051 accumulator for arithmetic operation.

B – B Register (bit-addressable)

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

Address: F0H reset value: 0000 0000b

Bit	Name	Description
7:0	B[7:0]	**B register.** The B register is the other accumulator of the standard 8051. It is used mainly for MUL and DIV operations.

SP – Stack Pointer

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

Address: 81H reset value: 0000 0111b

Bit	Name	Description
7:0	SP[7:0]	**Stack pointer.** The Stack Pointer stores the scratch-pad RAM address where the stack begins. It is incremented before data is stored during PUSH or CALL instructions. Note that the default value of SP is 07H. It causes the stack to begin at location 08H.

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, it is 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 4^{th} bit of the low order nibble; otherwise, it is cleared.
5	F0	**User flag 0.** The general purpose flag that can be set or cleared by the user.
4	RS1	**Register Bank selecting bits.** These two bits select one of four banks in which R0-R7 locate.
3	RS0	

RS1	RS0	Register Bank	RAM Address
0	0	0	00-07H
0	1	1	08-0FH
1	0	2	10-17H
1	1	3	18-1FH

Bit	Name	Description
2	OV	**Overflow flag.** OV is used for a signed character operands. For a ADD or ADDC instruction, OV will be set if there is a carry out of bit 6 but not out of bit 7, or a carry out of bit 7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands. For a SUBB, OV is set if a borrow is needed into bit6 but not into bit 7, or into bit7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number. For a MUL, if the product is greater than 255 (00FFH), OV will be set. Otherwise, it is cleared. For a DIV, it is normally 0. However, if B had originally contained 00H, the values returned in A and B will be undefined. Meanwhile, the OV will be set.
1	F1	**User flag 1.** The general purpose flag that can be set or cleared by the user via software.
0	P	**Parity flag.** Set to 1 to indicate an odd number of ones in the accumulator. Cleared for an even number of ones. It performs even parity check.

Table 7–1. Instructions that affect flag settings

Instruction	CY	OV	AC	Instruction	CY	OV	AC
ADD	X[1]	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						

[1] X indicates the modification depends on the result of the instruction.

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. N78E517A 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.

8. AUXILIARY RAM (XRAM)

N78E517A 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. N78E517A SFR Description 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
MOVX    A,@R0
```

```
MOV     DPTR,#0123H         ;write #5BH to XRAM with address @0123H.
MOV     A,#5BH
MOVX    @DPTR,A

MOV     DPTR,#0123H         ;read from XRAM with address @0123H.
MOVX    A,@DPTR
```

9. I/O PORT STRUCTURE AND OPERATION

N78E517A has maximum five 8-bit width, bit-addressable ports P0-P4. The configuration of P1-P4 is the quasi bi-directional I/O. This type rules as both input and output. When the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is pulled low, it is driven strongly and able to sink a large current. In the quasi bi-directional I/O structure, there are three pull-up transistors. Each of them serves different purposes. One of these pull-ups, called the “very weak” pull-up, is turned on whenever the port latch contains a logic 1. The “very weak” pull-up sources a very small current that will pull the pin high if it is left floating.

A second pull-up, called the “weak” pull-up, is turned on when the outside port pin itself is at a logic 1 level. This pull-up provides the primary source current for a quasi bi-directional pin that is outputting a 1. If a pin that has a logic 1 on it is pulled low by an external device, the “weak” pull-up turns off, and only the “very weak” pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current (larger than I_{TL}) to overcome the “weak” pull-up and make the voltage on the port pin below its input threshold (lower than V_{IL}).

The third pull-up is the “strong” pull-up. This pull-up is used to speed up low-to-high transitions on a quasi bi-directional port pin when the port latch changes from a logic 0 to a logic 1. When this occurs, the strong pull-up turns on for two-peripheral-clock time in order to pull the port pin high quickly. Then it turns off and “weak”: pull-up continues remaining the port pin high. The quasi bi-directional port structure is shown below.

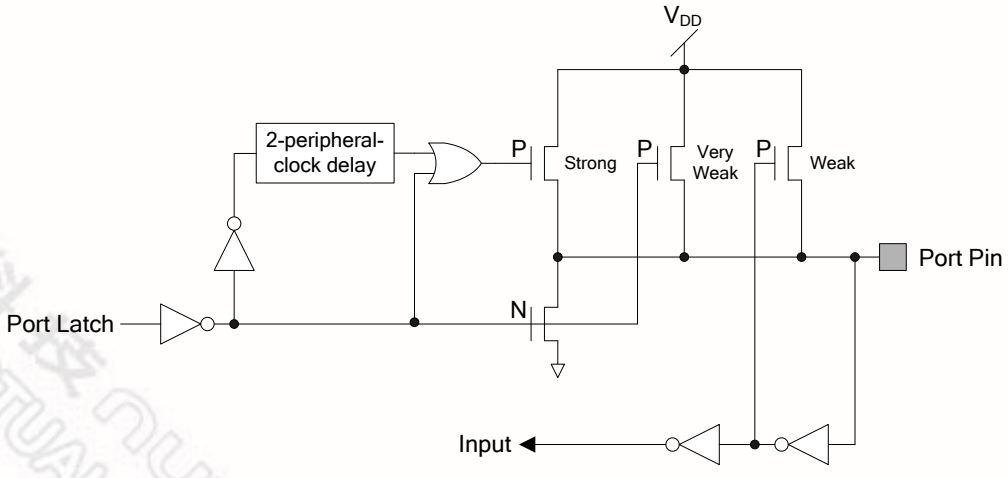


Figure 9–1. Quasi Bi-direction I/O Structure

The default configuration of P0 is open-drain structure. To serve as an I/O port the external pull-up resistor is always necessary. N78E517A also provide an internal P0 pull-up resistors for each pins. Via setting P0UP (P0OR.0) P0 will switch on its weak pull-up internally and behave the same as the quasi bi-directional I/O pins.

P0 and P2 also serve as address/data bus when external memory is running or is accessed by MOVC or MOVX instruction. In these cases, it has strong pull-up and pull-down. In this application, there is no need of any external pull-up resistor. While external mode execution, P0 and P2 cannot be used as general purpose I/O anymore.

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

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

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

P0 – Port 0 (bit-addressable)

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
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 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 external Program Memory execution, SFR P0 cannot be accessed.

P0OR – P0 Option Register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	P0UP
-	-	-	-	-	-	-	r/w

Address: 86H reset value: 0000 0000b

Bit	Name	Description
7:1	-	**Reserved.**
0	P0UP	**Port 0 pull-up enable.** 0 = Disable internal pull-up resistors of all 8-bits of Port 0. 1 = Enable internal pull-up resistors of all 8-bits of Port 0.

P1 – Port 1 (bit-addressable)

7	6	5	4	3	2	1	0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
r/w	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.** Port 1 is an 8-bit quasi bi-directional I/O port.

P2 – Port 2 (bit-addressable)

7	6	5	4	3	2	1	0
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: A0H reset value: 1111 1111b

Bit	Name	Description
7:0	P2[7:0]	**Port 2.** 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 external Program Memory execution, SFR P2 cannot be accessed.

P3 – Port 3 (bit-addressable)

7	6	5	4	3	2	1	0
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: B0H reset value: 1111 1111b

Bit	Name	Description
7:0	P3[7:0]	**Port 3.** Port 3 is an 8-bit quasi bi-directional I/O port.

P4 – Port 4 (bit-addressable)

7	6	5	4	3	2	1	0
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: D8H reset value: 1111 1111b

Bit	Name	Description
7:0	P4[7:0]	**Port 4.** Port 4 is an 8-bit quasi bi-directional I/O port. It also possesses bit-addressable feature as P0-P3. Note that a full 8-bit P4 is just on LQPF-48 package. PLCC-44, PQFP-44, and TQFP-44 just have low nibble 4 bits of P4. DIP-40 does not have this additional P4.

10. TIMERS/COUNTERS

N78E517A has three 16-bit programmable timers/counters.

10.1 Timer/Counters 0 and 1

Timer/Counter 0 and 1 on N78E517A are two 16-bit Timer/Counters. Each of them has two 8 bit registers which form the 16 bit counting register. For Timer/Counter 0 they are TH0, the upper 8 bits register, and TL0, the lower 8 bit register. Similarly Timer/Counter 1 has two 8 bit registers, TH1 and TL1. TCON and TMOD can configure modes of Timer/Counter 0 and 1.

The Timer or Counter function is selected by the C/$\overline{T}$ bit in TMOD. Each Timer/Counter has its own selection bit. TMOD.2 selects the function for Timer/Counter 0 and TMOD.6 selects the function for Timer/Counter 1

When configured as a "Timer", the timer counts clock cycles. The timer clock is 1/6 of the peripheral clock (F_{PERIPH}). In the "Counter" mode, the register increases on the falling edge of the external input pins T0 for Timer 0 and T1 for Timer 1. If the sampled value is high in one machine-cycle and low in the next, a valid 1 to 0 transition on the pin is recognized and the count register increases.

In addition, each Timer/Counter can be set to operate in any one of four possible modes. Bits M0 and M1 in TMOD do the mode selection.

TMOD – Timer 0 and 1 Mode

7	6	5	4	3	2	1	0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: 89H reset value: 0000 0000b

Bit	Name	Description
7	GATE	**Timer 1 gate control.** 0 = Timer 1 will clock when TR1 = 1 regardless of $\overline{INT1}$ logic level. 1 = Timer 1 will clock only when TR1 = 1 and $\overline{INT1}$ is logic 1.
6	C/$\overline{T}$	**Timer 1 Counter/Timer select.** 0 = Timer 1 is incremented by internal peripheral clocks. 1 = Timer 1 is incremented by the falling edge of the external pin T1.
5	M1	**Timer 1 mode select.**
4	M0	M1 M0 Timer 1 Mode 0 0 Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL1[4:0]) 0 1 Mode 1: 16-bit Timer/Counter 1 0 Mode 2: 8-bit Timer/Counter with auto-reload from TH1 1 1 Mode 3: Timer 1 halted

Bit	Name	Description
3	GATE	**Timer 0 gate control.** 0 = Timer 0 will clock when TR0 = 1 regardless of $\overline{INT0}$ logic level. 1 = Timer 0 will clock only when TR0 = 0 and $\overline{INT0}$ is logic 1.
2	C/$\overline{T}$	**Timer 0 Counter/Timer select.** 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 select.** 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
0	M0	

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.

TL0 – Timer 0 Low Byte

7	6	5	4	3	2	1	0
TL0[7:0]							
r/w							

Address: 8AH reset value: 0000 0000b

Bit	Name	Description
7:0	TL0[7:0]	**Timer 0 low byte.** The TL0 register is the low byte of the 16-bit Timer 0.

TH0 – Timer 0 High Byte

7	6	5	4	3	2	1	0
TH0[7:0]							
r/w							

Address: 8CH reset value: 0000 0000b

Bit	Name	Description
7:0	TH0[7:0]	**Timer 0 high byte.** The TH0 register is the high byte of the 16-bit Timer 0.

TL1 – Timer 1 Low Byte

7	6	5	4	3	2	1	0
TL1[7:0]							
r/w							

Address: 8BH reset value: 0000 0000b

Bit	Name	Description
7:0	TL1[7:0]	**Timer 1 low byte.** The TL1 register is the low byte of the 16-bit Timer 1.

TH1 – Timer 1 High Byte

7	6	5	4	3	2	1	0
TH1[7:0]							
r/w							

Address: 8DH reset value: 0000 0000b

Bit	Name	Description
7:0	TH1[7:0]	**Timer 1 high byte.** The TH1 register is the high byte of the 16-bit Timer 1.

10.1.1 Mode 0 (13-bit Timer)

In Mode 0, the Timer/Counter is a 13-bit counter. The 13-bit counter consists of THx and the five lower bits of TLx. The upper three bits of TLx are ignored. The Timer/Counter is enabled when TRx is set and either GATE is 0 or $\overline{INTx}$ is 1. Gate = 1 allows the Timer to calculate the pulse width on external input pin $\overline{INTx}$. When the 13-bit value moves from 1FFFH to 0000H, the Timer overflow flag TFx is set and an interrupt occurs if enabled. Note that the peripheral clock is $F_{OSC}/2$ in 12T mode and is F_{OSC} in 6T mode. See Section 20. “CLOCK SYSTEM” on page 103.

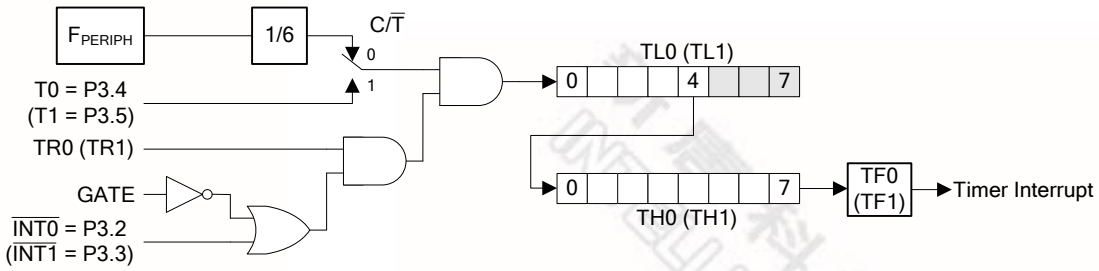


Figure 10–1. Timer/Counters 0 and 1 in Mode 0

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. Roll-over 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.

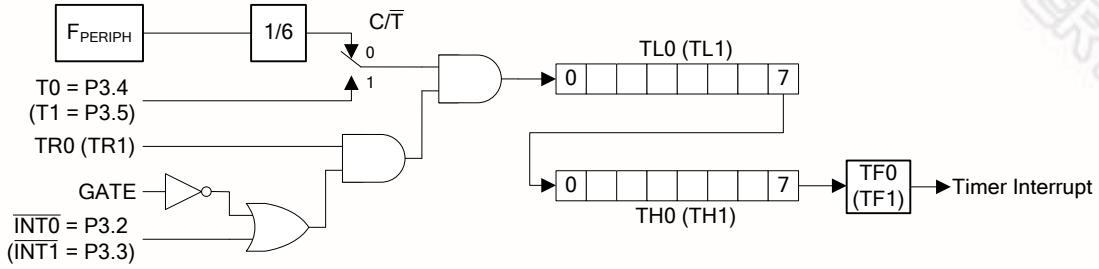


Figure 10–2. Timer/Counters 0 and 1 in Mode 1

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, 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.

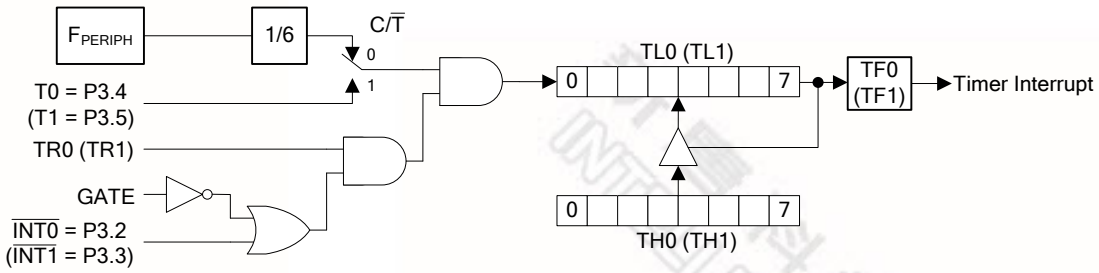


Figure 10–3. Timer/Counter 0 and 1 in Mode 2

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.

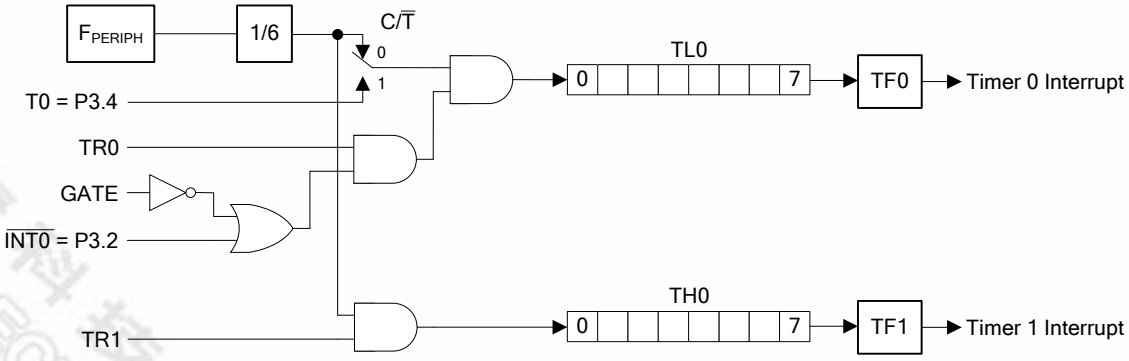


Figure 10–4. Timer/Counter 0 in Mode 3

10.2 Timer/Counter 2

Timer/Counter 2 is a 16-bit up counter, which is configured by the T2MOD and T2CON registers. The count stores in two 8-bit cascade registers TH2 and TL2. Timer/Counter 2 is additionally equipped with a capture or reload capability. It also can be configured as the baud rate generator for UART or a square wave generator. The features listed above could be achieved because of the addition Timer/Counter 2 capture registers RCAP2H and RCAP2L. As with the Timer 0 and Timer 1 counters, there exists considerable flexibility in selecting and controlling the clock and in defining the operating mode. The clock source for Timer/Counter 2 may be selected from either the external T2 pin ($C/\overline{T2}$ (T2CON.1) = 1) or the crystal oscillator ($C/\overline{T2}$ = 0). The clock is then enabled when TR2 (T2CON.2) is a 1, and disabled when TR2 is a 0. The following registers are related to Timer/Counters 2 function.

T2CON – Timer 2 Control (bit-addressable)

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: C8H reset value: 0000 0000b

Bit	Name	Description
7	TF2	**Timer 2 overflow flag.** This bit is set when Timer 2 overflows. If the Timer 2 interrupt and the global interrupt are enable, setting this bit will make CPU execute Timer 2 interrupt service routine. This bit is not automatically cleared via hardware and must be cleared via software. TF2 will not be set while Timer 2 is configured in the baud rate generator or clock-out mode.
6	EXF2	**Timer 2 external flag.** This bit is set via hardware when a 1-to-0 transition on the T2EX input pin occurs and EXEN2 is logic 1. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to execute the Timer 2 Interrupt service routine. This bit is not automatically cleared via hardware and must be cleared via software.
5	RCLK	**Receive clock flag.** This bit selects which Timer is used for the UART's receive clock in serial Mode 1 or 3. 0 = Timer 1 overflows is used for UART receive baud rate clock. 1 = Timer 2 overflows is used for UART receive baud rate clock.
4	TCLK	**Transmit clock flag.** This bit selects which Timer is used for the UART's transmit clock in serial Mode 1 or 3. 0 = Timer 1 overflows is used for UART transmit baud rate clock. 1 = Timer 2 overflows is used for UART transmit baud rate clock.

Bit	Name	Description
3	EXEN2	**Timer 2 external enable.** This bit enables 1-to-0 transitions on T2EX trigger. 0 = 1-to-0 transitions on T2EX is ignored. 1 = 1-to-0 transitions on T2EX will set EXF2 logic 1. If Timer 2 is configured in capture or auto-reload mode, the 1-to-0 transitions on T2EX will cause capture or reload event.
2	TR2	**Timer 2 run control.** 0 = Timer 2 is halted. Clearing this bit will halt Timer 2 and the current count will be preserved in TH2 and TL2. 1 = Timer 2 is enabled.
1	C/$\overline{T2}$	**Timer 2 Counter/Timer select.** 0 = Timer 2 is incremented by internal peripheral clocks. 1 = Timer 2 is incremented by the falling edge of the external pin T2. If Timer 2 would like to be set in clock-out mode, C/$\overline{T2}$ must be 0.
0	CP/$\overline{RL2}$	**Timer 2 Capture or Reload select.** This bit selects whether Timer 2 functions in capture or auto-reload mode. EXEN2 must be logic 1 for 1-to-0 transitions on T2EX to be recognized and used to trigger captures or reloads. If RCLK or TCLK is set, this bit is ignored and Timer 2 will function in auto-reload mode. 0 = Auto-reload on Timer 2 overflow or 1-to-0 transition on T2EX pin. 1 = Capture on 1-to-0 transition at T2EX pin.

T2MOD – Timer 2 Mode

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T2OE	-
-	-	-	-	-	-	r/w	-

Address: C9H reset value: 0000 0000b

Bit	Name	Description
7:2	-	**Reserved.**
1	T2OE	**Timer 2 clock-out enable.** 0 = Disable Timer 2 clock-out function. T2 pin functions either as a standard port pin or as a counter input for Timer 2. 1 = Enable Timer 2 clock-out function. Timer 2 will drive T2 pin with a clock output if C/$\overline{T2}$ is 0.
0	-	**Reserved.**

RCAP2L – Timer 2 Reload/Capture Low Byte

7	6	5	4	3	2	1	0
RCAP2L[7:0]							
r/w							

Address: CAH reset value: 0000 0000b

Bit	Name	Description
7:0	RCAP2L[7:0]	**Timer 2 reload/capture low byte.** This register captures and stores the low byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is in auto-reload mode, baud rate generator mode, or clock-out mode, it holds the low byte of the reload value.

RCAP2H – Timer 2 Reload/Capture High Byte

7	6	5	4	3	2	1	0
RCAP2H[7:0]							
r/w							

Address: CBH reset value: 0000 0000b

Bit	Name	Description
7:0	RCAP2H[7:0]	**Timer 2 reload/capture high byte.** This register captures and stores the high byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is in auto-reload mode, baud rate generator mode, or clock-out mode, it holds the high byte of the reload value.

TL2 – Timer 2 Low Byte

7	6	5	4	3	2	1	0
TL2[7:0]							
r/w							

Address: CCH reset value: 0000 0000b

Bit	Name	Description
7:0	TL2[7:0]	**Timer 2 low byte.** The TL2 register is the low byte of the 16-bit Timer 2.

TH2 – Timer 2 High Byte

7	6	5	4	3	2	1	0
TH2[7:0]							
r/w							

Address: CDH reset value: 0000 0000b

Bit	Name	Description
7:0	TH2[7:0]	**Timer 2 high byte.** The TH2 register is the high byte of the 16-bit Timer 2.

Timer/Counter 2 provides four operating mode which can be selected by control bits in T2CON and T2MOD as shown in table below. Note that the TH2 and TL2 are accessed separately. It is strongly recommended that the user stop Timer 2 temporally for a reading from or writing to TH2 and TL2. The free-running reading or writing may cause unpredictable situation.

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	0
16-bit auto-reload	0	0	0
Baud rate generator	1	X	0
Clock-out[2]	0	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.

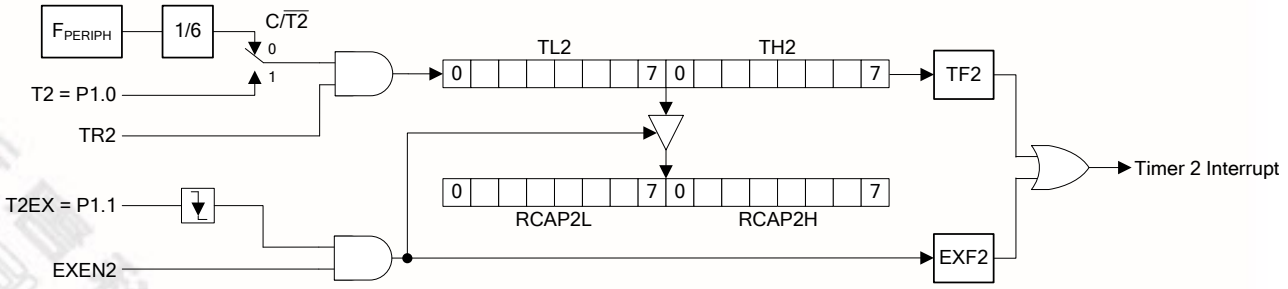


Figure 10–5. Timer/Counter 2 in Capture Mode

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.

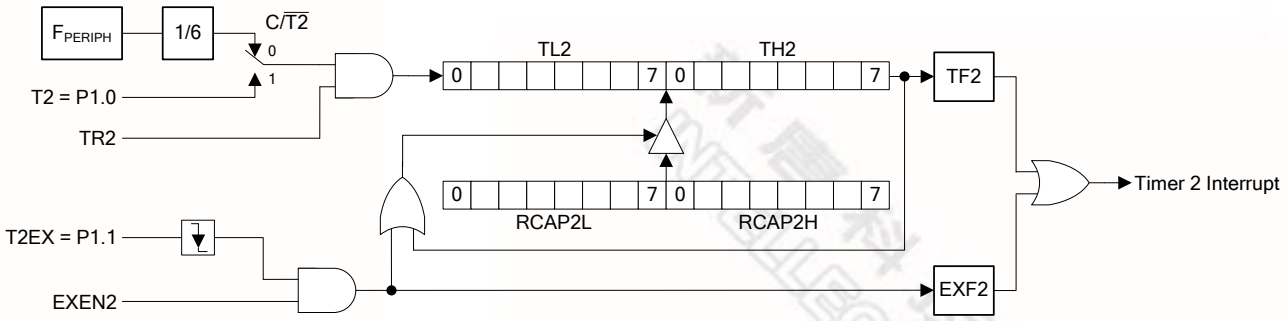


Figure 10–6. Timer/Counter 2 in Auto-reload Mode

10.2.3 Baud Rate Generator Mode

The Timer 2 can generate the baud rate for UART in its Mode 1 and 3. The baud rate generator mode is enabled by setting either the RCLK or TCLK bits in T2CON register. While in the baud rate generator mode, Timer/Counter 2 is a 16 bit counter with auto-reload when the count rolls over from FFFFH. However, rolling over is used to generate the shift clock for UART data rather than to set the TF2 bit. If EXEN2 bit is set, then a negative transition of the T2EX pin will set EXF2 bit in the T2CON register and cause an interrupt request. It simply provides a external interrupt. Note that TCLK and RCLK are selected individually, the serial port transmit rate can be different from the receive rate. For example the transmit clock can be generated from Timer 2 by setting TCLK and the receive clock from Timer 1 by clearing RCLK.

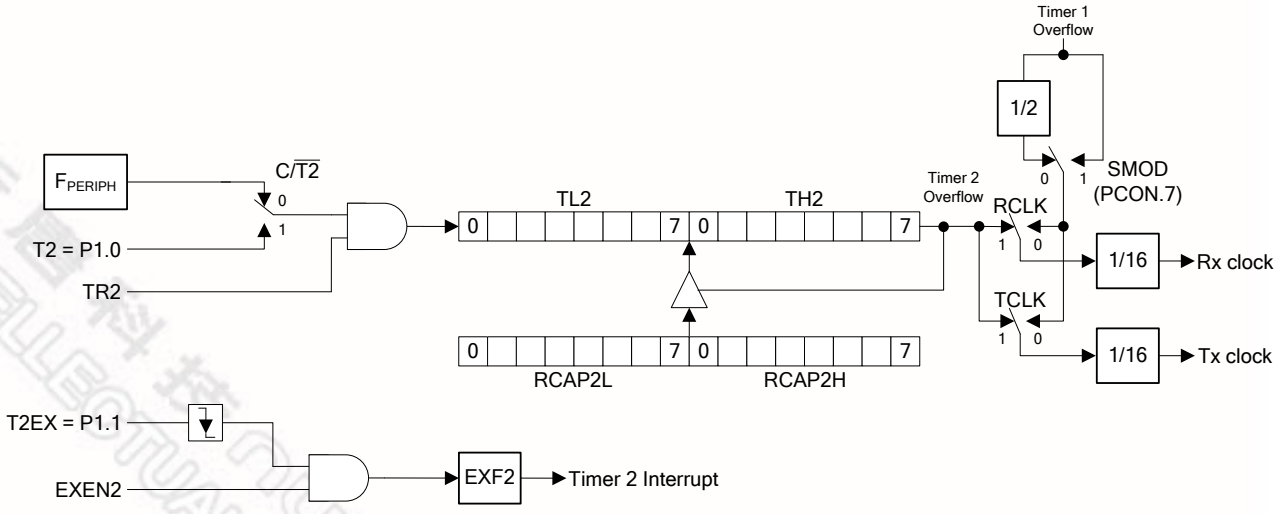


Figure 10–7. Timer/Counter 2 in Baud Rate Generator Mode

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 $256 \times RCAP2H + RCAP2L$.

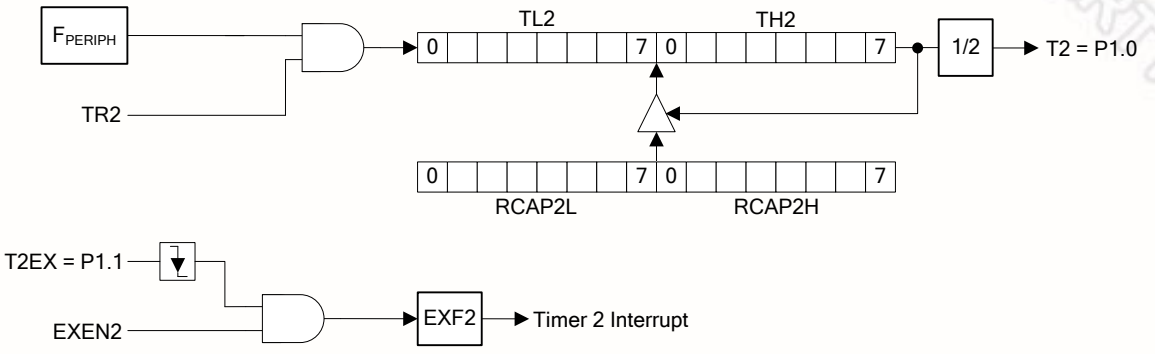


Figure 10–8. Timer/Counter 2 in Clock-out Mode

11. WATCHDOG TIMER

11.1 Functional Description of Watchdog Timer

N78E517A provides one Watchdog Timer to serve as a system monitor, which improve the reliability of the system. Watchdog Timer is useful for systems that are susceptible to noise, power glitches, or electrostatic discharge. The Watchdog Timer is basic a setting of dividers that divide the peripheral clock. The divider output is selectable and determines the time-out interval. When the time-out interval is fulfilled, a direct system reset will occur.

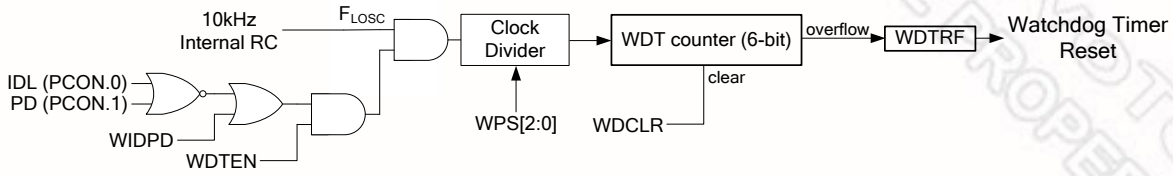


Figure 11–1. Watchdog Timer Block Diagram

The Watchdog Timer should first be reset 00H by using WDCLR(WDCON.6) to ensure that the timer starts from a known state. The WDCLR bit is used to reset the Watchdog Timer. This bit is self-cleared thus the user doesn't need to clear it. After writing a 1 to WDCLR, the hardware will automatically clear it. After WDTEN set as 1, the Watchdog Timer starts counting. The time-out interval is selected by the three bits WPS2, WPS1, and WPS0 (WDCON[2:0]). When the selected time-out occurs, the Watchdog Timer will reset the system directly. Once a reset due to Watchdog Timer occurs, the Watchdog Timer reset flag WDTRF (WDCON.3) will be set. This bit keeps unchanged after any reset other than a power-on reset. The user may clear WDTRF via software. In general, software should restart the counter to put it into a known state by setting WDCLR. The Watchdog Timer also provides an WIDPD bit (WDCON.4) to allow the Watchdog Timer continuing running after the system enters into Idle or Power Down operating mode.

WDT counter should be specially taken care. The hardware automatically clears WDT counter after entering into or being woken-up from Idle or Power Down mode. It prevents unconscious system reset.

CONFIG3

7	6	5	4	3	2	1	0
CWDTEN	EN6T	ROG	CKF	INTOSCFS	-	FOSC	-
r/w	r/w	r/w	r/w	r/w	-	r/w	-

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CWDTEN	**CONFIG Watchdog Timer enable.** 1 = Disable Watchdog Timer after all resets. 0 = Enable Watchdog Timer after all resets.

WDCON – Watchdog Timer Control (TA protected)

7	6	5	4	3	2	1	0
WDTEN[1]	WDCLR	-	WIDPD[2]	WDTRF[3]	WPS2[2]	WPS1[2]	WPS0[2]
r/w	w	-	r/w	r/w	r/w	r/w	r/w

Address: AAH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
7	WDTEN	**Watchdog Timer enable.** 0 = Disable Watchdog Timer. 1 = Enable Watchdog Timer. The WDT counter starts running.
6	WDCLR	**Watchdog Timer clear.** Setting this bit will reset the Watchdog Timer count to 00H. It puts the counter in a known state and prohibit the system from reset. Note that this bit is written-only and has no need to be cleared via software.
5	-	**Reserved.**
4	WIDPD	**Watchdog Timer running in Idle and Power Down mode.** This bit decides whether Watchdog Timer runs in Idle or Power Down mode. 0 = WDT counter is halted while CPU is in Idle or Power Down mode. 1 = WDT keeps running while CPU is in Idle or Power Down mode.
3	WDTRF	**Watchdog Timer reset flag.** When the CPU is reset by Watchdog Timer time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software.
2	WPS2	**Watchdog Timer clock pre-scalar select.** These bits determine the scale of the clock divider for WDT counter. The scale is from 1/1 through 1/256. See Table 11–1.
1	WPS1	
0	WPS0	

[1] WDTEN is initialized by the inversed value of CWDTEN (CONFIG3.7) after all resets.
[2] WIDPD and WPS[2:0] are cleared after power-on reset, and keep unchanged after any other resets.
[3] WDTRF will be cleared after power-on reset, be set after Watchdog Timer reset, and remains unchanged after any other resets.

The Watchdog 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 Watchdog time-out interval under different F_{WCK} and pre-scalars.

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.

12. POWER DOWN WAKING-UP TIMER

12.1 Functional Description of Power Down Waking-up Timer

N78E517A provides another free-running Timer, Power Down waking-up timer which serves as a event timer or a durational system supervisor in a monitoring system which generally operates in Idle or Power Down modes. This timer constructs basically by a set of dividers that divide the peripheral clock. The divider output is selectable and determines the time-out interval. When the time-out interval is fulfilled, it will wake the system up from Idle or Power Down mode and an interrupt event will occur.

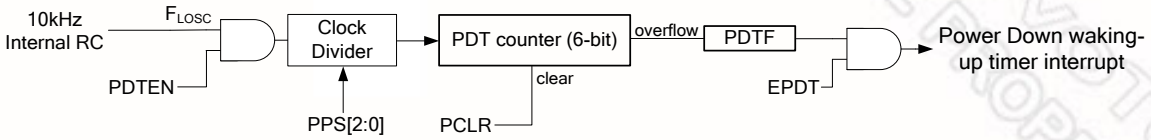


Figure 12–1. Power Down Waking-up Timer Block Diagram

The Power Down waking-up timer should first be reset 00H by using PDCLR(PDCON.6) to ensure that the timer starts from a known state. The PDCLR bit is used to restart the Power Down waking-up timer. This bit is self-cleared thus the user doesn't need to clear it. After writing a 1 to PDCLR, the hardware will automatically clear it. After PDTEN set as 1, the Power Down waking-up timer will start counting clock cycles. The time-out interval is selected by the three bits PPS2, PPS1, and PPS0 (PDCON[2:0]). When the selected time-out occurs, the Power Down waking-up timer will set the interrupt flag PDTF (PDCON.5). The Power Down waking-up timer interrupt enable bit locates at bit 1 in EIE. In general, software should restart the counter to put it into a known state by setting WDCLR.

PDCON – Power Down Waking-up Timer Control

7	6	5	4	3	2	1	0
PDTEN	PDCLR	PDTF	-	-	PPS2	PPS1	PPS0
r/w	w	r/w	-	-	r/w	r/w	r/w

Address: ABH reset value: 0000 0000b

Bit	Name	Description
7	PDTEN	**Power Down waking-up timer enable.** 0 = Disable Power Down waking-up timer. 1 = Enable Power Down waking-up timer. The PDT counter starts running.
6	PDCLR	**Power Down waking-up timer clear.** Setting this bit will reset the Power Down waking-up timer count to 00H. It put the counter in a known state. This bit is written-only and has no need to be cleared via software.
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.

Bit	Name	Description
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 pro-

grammable interval. N78E517A 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
```

13. SERIAL PORT

N78E517A includes one enhanced full duplex serial port. The serial port supports three modes of full duplex UART (Universal Asynchronous Receiver and Transmitter) in Mode 1, 2, and 3. Full duplex means it can transmit and receive simultaneously. The serial port is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. The serial port receive and transmit registers are both accessed at SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register. There are four operation modes in serial port. In all four modes, transmission initiates by any instruction that uses SBUF as a destination register. Note that before serial port function works, the port latch bits of P3.0 and P3.1 (for RXT and TXD pins) have to be set to 1.

SCON – Serial Port Control (bit-addressable)

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: 98H reset value: 0000 0000b

Bit	Name	Description
7	SM0	**Serial port mode select.** See Table 13–1. Serial Port Mode Description for details.
6	SM1	
5	SM2	**Multiprocessor communication mode enable.** The function of this bit is dependent on the serial port mode. Mode 0: This bit has no effect. Mode 1: This bit checks valid stop bit. 0 = Reception is always valid no matter the logic level of stop bit. 1 = Reception is ignored if the received stop bit is not logic 1. Mode 2 or 3: For multiprocessor communication. 0 = Reception is always valid no matter the logic level of the 9th bit. 1 = Reception is ignored if the received 9th bit is not logic 1.
4	REN	**Receive enable.** 0 = Disable serial port reception. 1 = Enable serial port reception in Mode 1,2, and 3. In Mode 0, reception is initialized by the condition REN = 1 and RI = 0.
3	TB8	**9th transmit 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 receive 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.

Bit	Name	Description
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 must 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 must be cleared manually via software.

Table 13–1. Serial Port Mode Description

Mode	SM0	SM1	Description	Data Bits	Baud Rate
0	0	0	Synchronous	8	F_{OSC} divided by 12 for 12T mode, by 6 for 6T mode
1	0	1	Asynchronous	10	Timer 1 overflow rate divided by 16 or divided by 32[1], or Timer 2 overflow rate divided by 16
2	1	0	Asynchronous	11	F_{OSC} divided by 32 or 64[1] for 12T mode, by 16 or 32[1] for 6T mode
3	1	1	Asynchronous	11	Timer 1 overflow rate divided by 16 or divided by 32[1], or Timer 2 overflow rate divided by 16

[1] While SMOD (PCON.7) is logic 0.

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. N78E517A 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 13–1. Serial Port Mode Description in details.

SBUF – Serial Data Buffer

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

Address: 99H reset value: 0000 0000b

Bit	Name	Description
7:0	SBUF[7:0]	**Serial data buffer.** This byte actually consists two separate registers. One is the receive resister, and the other is the transmit buffer. When data is moved to SBUF, it goes to the transmit buffer and is shifted for serial transmission. When data is moved from SBUF, it comes from the receive buffer. The transmission is initiated through moving a byte to SBUF.

13.1 Mode 0

Mode 0 provides synchronous communication with external devices. Serial data enters and exits through RXD pin. TXD outputs the shift clock. 8 bits are transmitted or received. Mode 0 therefore provides half-duplex communication because the transmitting or receiving data is via the same data line RXD. The baud rate is fixed at 1/12 the oscillator frequency in 12T Mode or 1/6 the oscillator frequency in 6T Mode. Note that whenever transmitting or receiving, the serial clock is always generated by the microcontroller. Thus any device on the serial port in Mode 0 must accept the microcontroller as the Master. Figure 13–1 shows a simplified functional diagram of the serial port in Mode 0 and associated timing. Note that the peripheral clock is $F_{OSC}/2$ in 12T mode and is F_{OSC} in 6T mode. See Section 20. “CLOCK SYSTEM” on page 103.

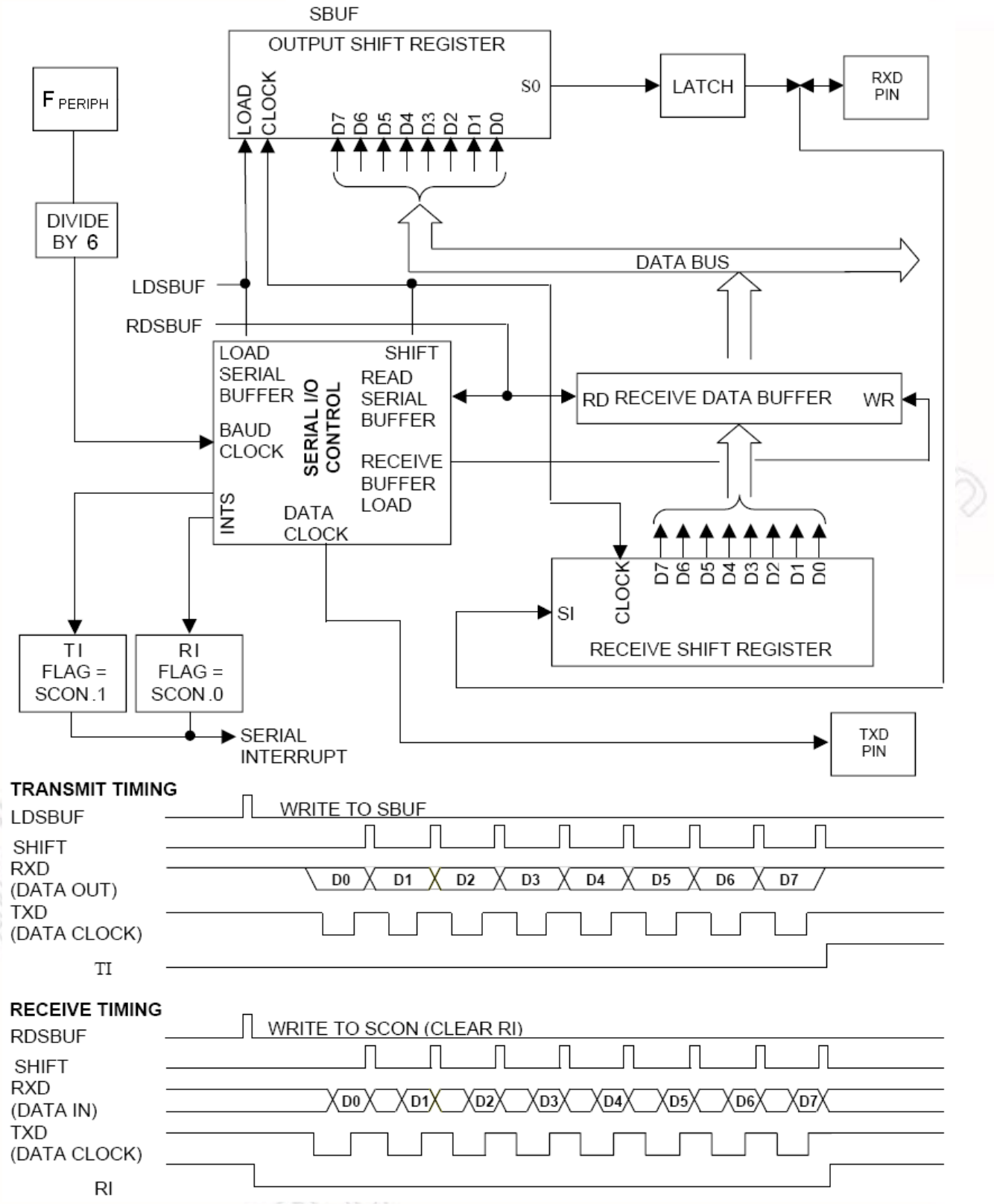


Figure 13–1. Serial Port Mode 0 Function Block and Timing Diagram

As shown there is one bi-direction 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 the condition REN (SCON.4) = 1 and RI (SCON.0) = 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. The user can clear RI to trigger the next byte reception.

13.2 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 or Timer 2 overflow rate according to RCLK and TCLK bits in T2CON. SMOD (PCON.7) setting 1 makes the baud rate double while Timer 1 is selected as the clock source. Figure 13–2 shows a simplified functional diagram of the serial port in Mode 1 and associated timings for transmit and receive.

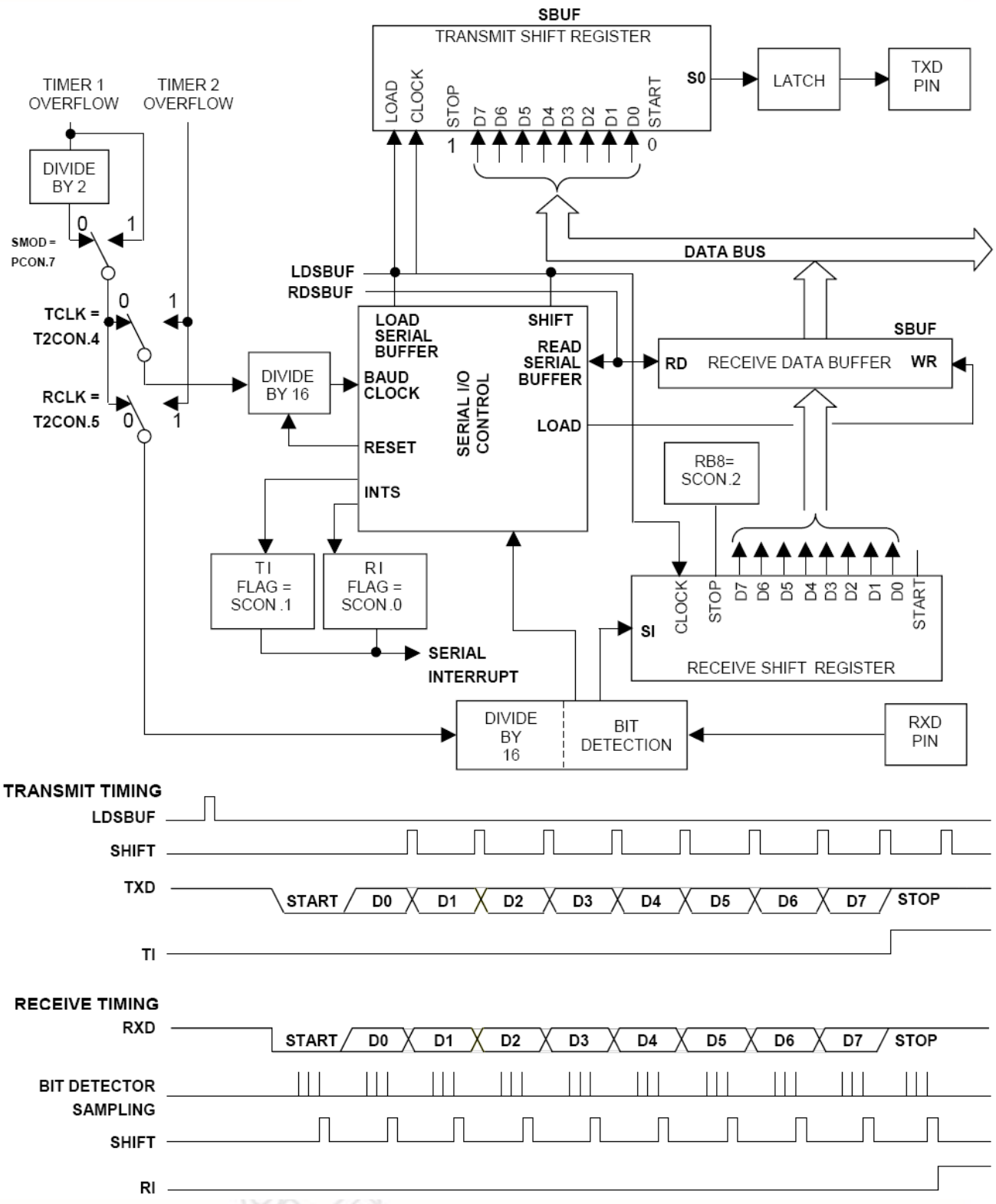


Figure 13–2. Serial Port Mode 1 Function Block and Timing Diagram

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

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

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

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

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

13.3 Mode 2

Mode 2 supports asynchronous, full duplex serial communication. Different from Mode1, there are 11 bits to be transmitted or received. They are a start bit (logic 0), 8 data bits (LSB first), a programmable 9th bit TB8 or RB8 bit and a stop bit (logic 1). The most common use of 9th bit is to put the parity bit in it. The baud rate is fixed as 1/32 or 1/64 the oscillator frequency depending on SMOD bit. (Condition above is under 12T mode. Under 6T mode, the baud rate will be 1/16 or 1/32 the oscillator frequency.) Figure 13–3 shows a simplified functional diagram of the serial port in Mode 2 and associated timings for transmit and receive.

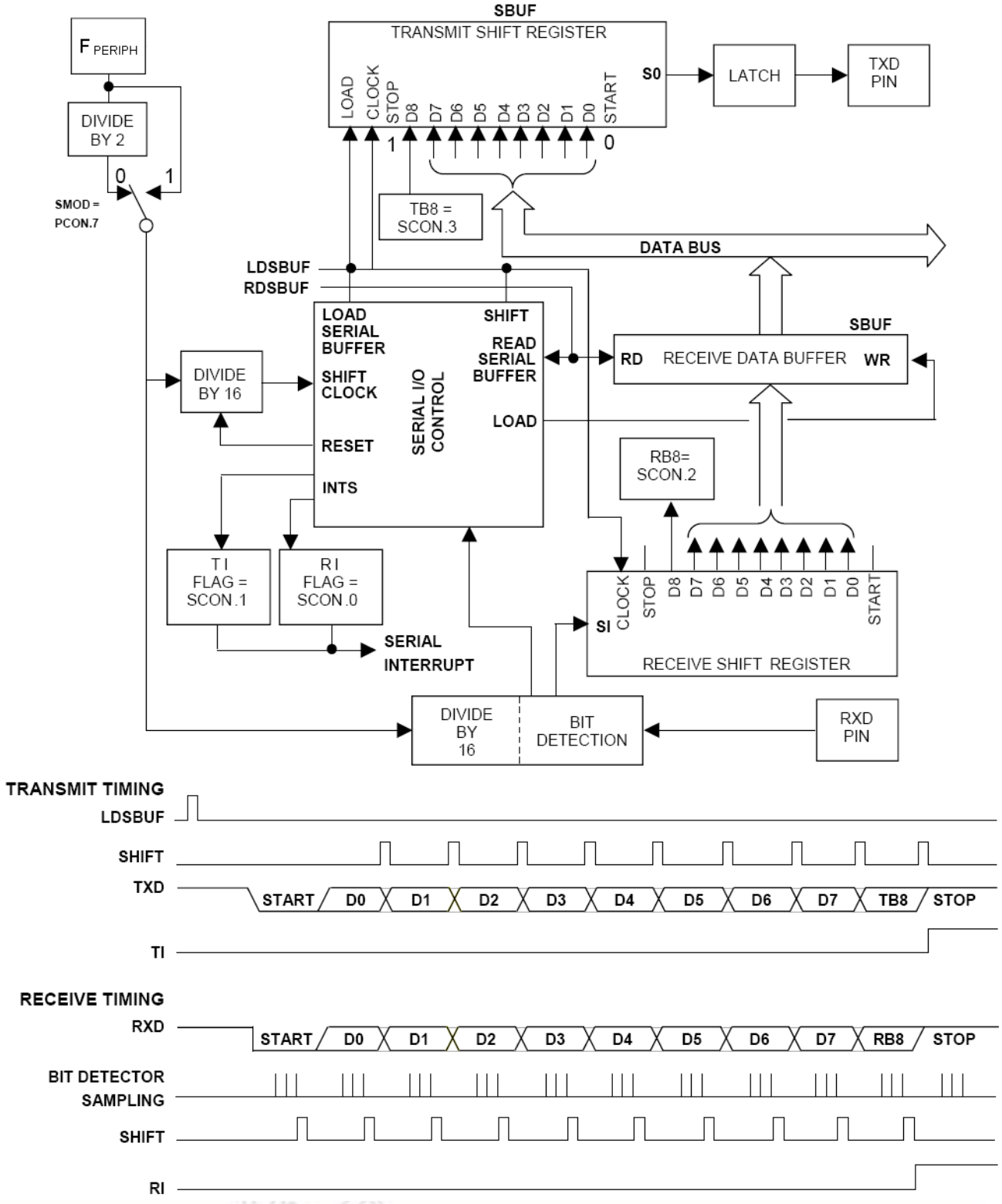


Figure 13–3. Serial Port Mode 2 Function Block and Timing Diagram

Transmission is initiated by any writing instructions to SBUF. Transmission takes place on TXD pin. First the start bit comes out, the 8-bit data and bit TB8 (SCON.3) follows to be shifted out and then ends with a stop bit. After the stop bit appears, TI will be set to indicate the transmission complete.

While REN is set, the reception is allowed at any time. A falling edge of a start bit on RXD will initiate the reception progress. Data will be sampled and shifted in at the selected baud rate. In the midst of the 9th bit, certain conditions must be met to load SBUF with the received data:

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

2. Either SM2(SCON.5) = 0, or the received 9th bit = 1 while SM2 = 1.

If these conditions are met, then the SBUF will be loaded with the received data, the RB8(SCON.2) with TB8 bit and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-0 transition on RXD pin in order to start next data reception.

13.4 Mode 3

Mode 3 has the same operation as Mode 2, except its baud rate clock source. As shown is Figure 13–4, Mode 3 uses Timer 1 or Timer 2 overflow as its baud rate clock.

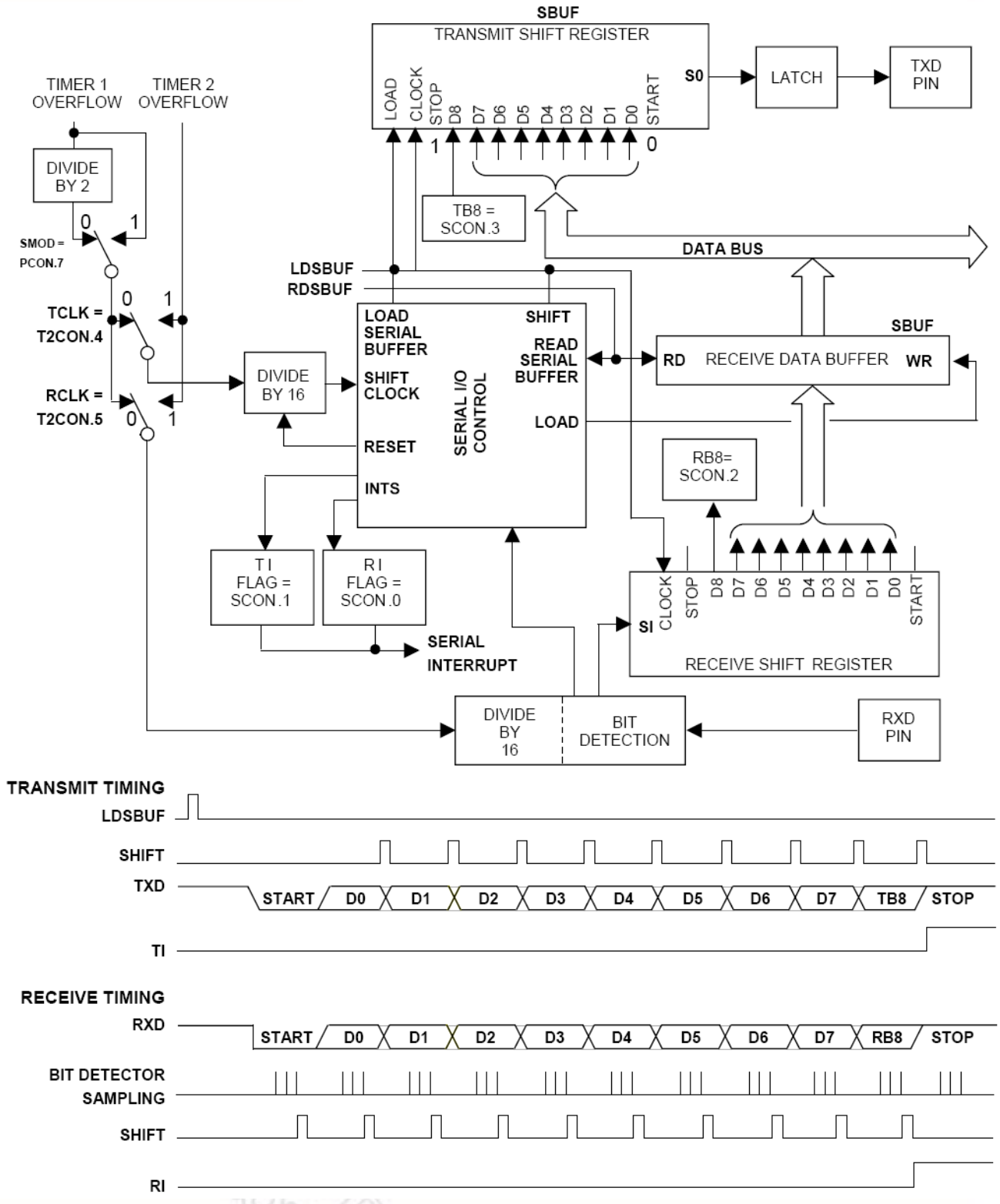


Figure 13–4. Serial Port Mode 3 Function Block and Timing Diagram

13.5 Baud Rate

Table 13–2. UART Baud Rate Formulas

UART mode	Baud rate clock source	EN6T (CONFIG3.6) value	
		1 (12T mode)	0 (6T mode)
0	Oscillator	$F_{OSC}/12$	$F_{OSC}/6$
2	Oscillator	$\frac{2^{SMOD}}{64}\times F_{OSC}$	$\frac{2^{SMOD}}{32}\times F_{OSC}$
1 or 3	Timer/Counter 1 overflow[1]	$\frac{2^{SMOD}}{32}\times\frac{F_{OSC}}{12\times(256-TH1)}$	$\frac{2^{SMOD}}{16}\times\frac{F_{OSC}}{12\times(256-TH1)}$
	Timer/Counter 2 overflow[2]	$\frac{F_{OSC}}{32\times(65536-(RCAP2H,RCAP2L))}$ [3]	$\frac{F_{OSC}}{16\times(65536-(RCAP2H,RCAP2L))}$

[1] Timer 1 is configured as a timer in auto-reload mode (Mode 2).
[2] Timer 2 is configured as a timer in baud rate generator mode.
[3] (RCAP2H,RCAP2L) in the formula means $256\times RCAP2H+RCAP2L$.

Note that in using Timer 1 as the baud rate generator, the interrupt should be disabled. In using Timer 2, the interrupt is automatically switched off. The Timer itself can be configured for either “Timer” or “Counter” operation. Timer 1 can be in any of its 3 running modes. In the most typical applications, it is configured for “Timer” operation, in the auto-reload mode (Mode2). If Timer 1 is used as the baud rate generator, the reloaded value is stored in TH1. Therefore the baud rate is determined by TH1 value. If Timer 2 is used, the user should configure it in baud rate generator mode (RCLK or TCLK in T2CON is logic 1) and give 16-bit reloaded value in RCAP2H and RCAP2L.

Table 13–3 lists various commonly used baud rates and how they can be obtained from Timer 1. In this mode, Timer 1 as an auto-reload Timer operates in 12T mode and SMOD (PCON.7) is 0. Table 13–4 is for Timer 2 as the baud rate generator. Timer 2 operates in baud rate generator mode in 12T mode. In 6T mode, the baud rate generated from both Timer 1 and Timer 2 overflows will be doubled.

Table 13–3. Timer 1 Generated Commonly Used Baud Rates

TH1 reload value	Oscillator Frequency (MHz)				
	11.0592	14.7456	18.432	22.1184	36.864
Baud Rate					
57600				FFh	
38400		FFh			
19200		FEh		FDh	FBh

TH1 reload value	Oscillator Frequency (MHz)				
	11.0592	14.7456	18.432	22.1184	36.864
Baud Rate					
9600	FDh	FCh	FBh	FAh	F6h
4800	FAh	F8h	F6h	F4h	ECh
2400	F4h	F0h	ECh	E8h	D8h
1200	E8h	E0h	D8h	D0h	B0h
300	A0h	80h	60h	40h	

Table 13–4. Timer 2 Generated Commonly Used Baud Rates

RCAP2H, RCAP2L reload value	Oscillator Frequency (MHz)				
	11.0592	14.7456	18.432	22.1184	36.864
Baud Rate					
115200	FFh, FDh	FFh, FCh	FFh, FBh	FFh, FAh	FFh, F6h
57600	FFh, FAh	FFh, F8h	FFh, F6h	FFh, F4h	FFh, ECh
38400	FFh, F7h	FFh, F4h	FFh, F1h	FFh, EEh	FFh, E2h
19200	FFh, EEh	FFh, E8h	FFh, E2h	FFh, DCh	FFh, C4h
9600	FFh, DCh	FFh, D0h	FFh, C4h	FFh, B8h	FFh, 88h
4800	FFh, B8h	FFh, A0h	FFh, 88h	FFh, 70h	FFh, 10h
2400	FFh, 70h	FFh, 40h	FFh, 10h	FEh, E0h	FEh, 20h
1200	FEh, E0h	FEh, 80h	FEh, 20h	FDh, C0h	FCh, 40h
300	FBh, 80h	FAh, 00h	F8h, 80h	F7h, 00h	F1h, 00h

13.6 Multiprocessor Communication

N78E517A multiprocessor communication feature of UART lets a Master device send a multiple frame serial message to a Slave device in a multi-slave configuration. It does this without interrupting other slave devices that may be on the same serial line. This feature can be used only in UART mode 2 or 3 mode. After 9 data bits are received. The 9th bit value is written to RB8 (SCON.2). The user can enable this function by setting SM2 (SCON.5) as a logic 1 so that when the stop bit is received, the serial interrupt will be generated only if RB8 is 1. When the SM2 bit is 1, serial data frames that are received with the 9th bit as 0 do not generate an interrupt. In this case, the 9th bit simply separates the address from the serial data.

When the Master device wants to transmit a block of data to one of several slaves on a serial line, it first sends out an address byte to identify the target slave. Note that in this case, an address byte differs from a data byte: In an address byte, the 9th bit is 1 and in a data byte, it is 0. The address byte interrupts all slaves so that each slave can examine the received byte and see if it is being addressed. The addressed slave then clears its SM2

bit and prepares to receive incoming data bytes. The SM2 bits of slaves that were not addressed remain set, and they continue operating normally while ignoring the incoming data bytes.

Follow these steps to configure multiprocessor communications.

1. Set all devices (Masters and Slaves) to UART mode 2 or 3.

2. Write the SM2 bit of all the Slave devices to 1.

3. The Master device's transmission protocol is:

   – First byte: the address, identifying the target slave device, (9^{th} bit = 1).

   – Next bytes: data, (9^{th} bit = 0).

4. When the target Slave receives the first byte, all of the Slaves are interrupted because the 9^{th} data bit is 1. The targeted Slave compares the address byte to its own address and then clears its SM2 bit in order to receive incoming data. The other slaves continue operating normally.

5. After all data bytes have been received, set SM2 back to 1 to wait for next address.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. For mode 1 reception, if SM2 is 1, the receive interrupt will not be issue unless a valid stop bit is received.

14. SERIAL PERIPHERAL INTERFACE (SPI)

14.1 Features

N78E517A exists a Serial Peripheral Interface (SPI) block to support high speed serial communication. SPI is a full-duplex, high speed, synchronous communication bus between MCUs or other peripheral devices such as serial EEPROM, LCD driver, or D/A converter. It provides either Master or Slave mode, high speed rate up to $F_{PERIPH}/16$ for Master mode and $F_{PERIPH}/4$ for Slave mode, transfer complete and write collision flag. For a multi-master system, SPI supports Master Mode Fault to protect a multi-master conflict.

14.2 Functional Description

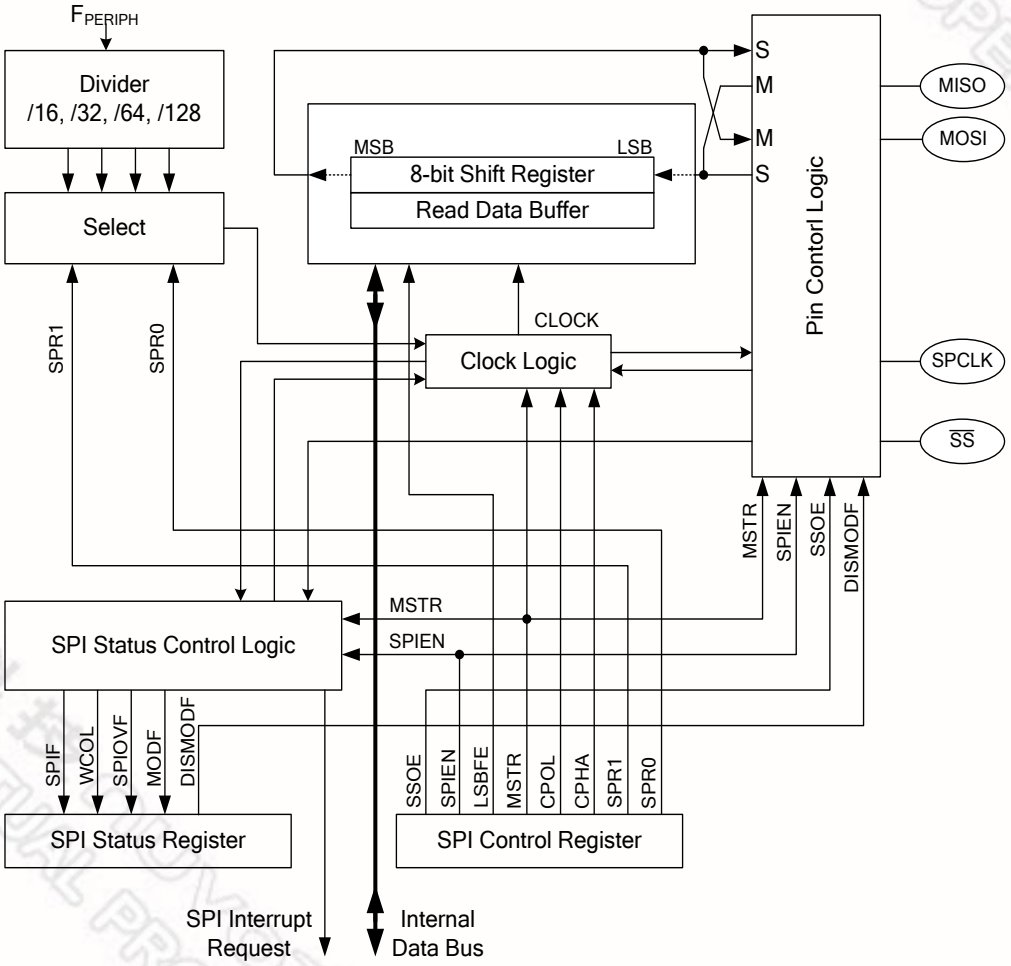


Figure 14–1. SPI Block Diagram

Figure 14–1 shows SPI block diagram. It provides an overview of SPI architecture in this device. The main blocks of SPI are the SPI control register logic, SPI status logic, clock rate control logic, and pin control logic. For a serial data transfer or receiving, The SPI block exists a shift register and a read data buffer. It is single buffered in the transmit direction and double buffered in the receiving direction. Transmit data cannot be written to the shifter until the previous transfer is complete. Receiving logic consists of parallel read data buffer so the shift register is free to accept a second data, as the first received data will be transferred to the read data buffer.

The four pins of SPI interface are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SPCLK), and Slave Select ($\overline{SS}$). The MOSI pin is used to transfer a 8-bit data in series from the Master to the Slave. Therefore, MOSI is an output pin for Master device and a input for Slave. Respectively, the MISO is used to receive a serial data from the Slave to the Master.

The SPCLK pin is the clock output in Master mode, but is the clock input in Slave mode. The shift clock is used to synchronize the data movement both in and out of the devices through their MOSI and MISO pins. The shift clock is driven by the Master mode device for eight clock cycles which exchanges one byte data on the serial lines. For the shift clock is always produced out of the Master device, the system should never exist more than one device in Master mode for avoiding device conflict.

Each Slave peripheral is selected by one Slave Select pin ($\overline{SS}$). The signal must stay low for any Slave access. When $\overline{SS}$ is driven high, the Slave device will be inactivated. If the system is multi-slave, there should be only one Slave device selected at the same time. In the Master mode MCU, the $\overline{SS}$ pin does not function and it can be configured as a general purpose I/O. However, $\overline{SS}$ can be used as Master Mode Fault detection (see Section 14.7 “Mode Fault Detection” on page 75) via software setting if multi-master environment exists. N78E517A also provides auto-activating function to toggle $\overline{SS}$ between each byte-transfer.

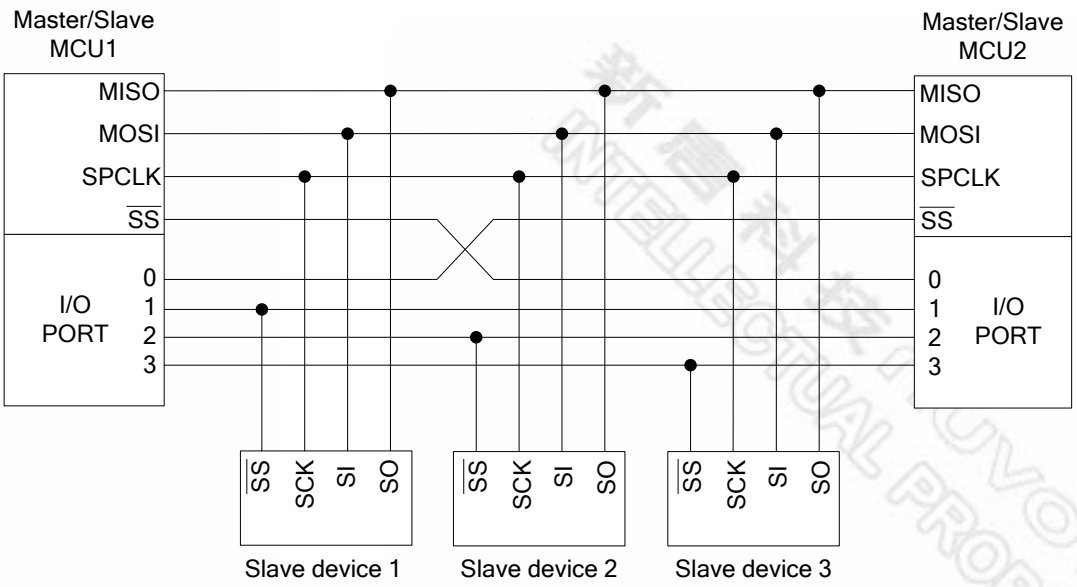


Figure 14–2. SPI Multi-master, Multi-slave Interconnection

Figure 14–2 shows a typical interconnection of SPI devices. The bus generally connects devices together through three signal wires, MOSI to MOSI, MISO to MISO, and SPCLK to SPCLK. The Master devices select the individual Slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins. MCU1 and MCU2 play either Master or Slave mode. The $\overline{SS}$ should be configured as Master Mode Fault detection to avoid multi-master conflict.

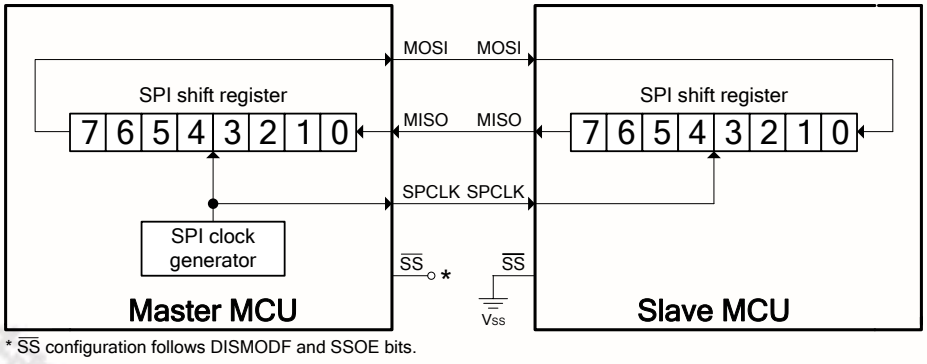


Figure 14–3. SPI Single-master, Single-slave Interconnection

Figure 14–3 shows the simplest SPI system interconnection, single-master and signal-slave. During a transfer, the Master shifts data out to the Slave via MOSI line. While simultaneously, the Master shifts data in from the Slave via MISO line. The two shift registers in the Master MCU and the Slave MCU can be considered as one 16-bit circular shift register. Therefore, while a transfer data pushed from Master into Slave, the data in Slave will also be pulled in Master device respectively. The transfer effectively exchanges the data which was in the SPI shift registers of the two MCUs.

By default, SPI data is transferred MSB first. If the LSBFE (SPCR.5) is set, SPI data shifts LSB first. This bit does not affect the position of the MSB and LSB in the data register. Note that all following description and figures are under the condition of LSBFE logic 0. MSB is transmitted and received first.

14.3 Control Registers of SPI

There are three SPI registers to support its operations, they are SPI control register (SPCR), SPI status register (SPSR), and SPI data register (SPDR). These registers provide control, status, data storage functions, and clock rate selection. The following registers relate to SPI function.

SPCR – Serial Peripheral Control Register

7	6	5	4	3	2	1	0
SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: F3H reset value: 0000 0000b

Bit	Name	Description
7	SSOE	**Slave select output enable.** This bit is used in combination with the DISMODF (SPSR.3) bit to determine the feature of $\overline{SS}$ pin as shown in Table 14–1. Slave Select Pin Configurations. This bit takes effect only under MSTR = 1 and DISMODF = 1 condition. 0 = $\overline{SS}$ functions as a general purpose I/O pin. 1 = $\overline{SS}$ automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device.
6	SPIEN	**SPI enable.** 0 = Disable SPI function. 1 = Enable SPI function.
5	LSBFE	**LSB first enable.** 0 = The SPI data is transferred MSB first. 1 = The SPI data is transferred LSB first.
4	MSTR	**Master mode enable.** This bit switches the SPI operating between Master and Slave modes. 0 = The SPI is configured as Slave mode. 1 = The SPI is configured as Master mode.
3	CPOL	**SPI clock polarity select.** CPOL bit determines the idle state level of the SPI clock. See Figure 14–4. SPI Clock Formats. 0 = The SPI clock is low in idle state. 1 = The SPI clock is high in idle state.
2	CPHA	**SPI clock phase select.** CPHA bit determines the data sampling edge of the SPI clock. See Figure 14–4. SPI Clock Formats. 0 = The data is sampled on the first edge of the SPI clock. 1 = The data is sampled on the second edge of the SPI clock.

Bit	Name	Description
1	SPR1	**SPI clock rate select.** These two bits select four grades of SPI clock divider. SPR1 SPR0 Divider SPI clock rate 0 0 16 1.25M bit/s 0 1 32 625k bit/s 1 0 64 312k bit/s 1 1 128 156k bit/s The clock rates above are illustrated under F_{PERIPH} = 20MHz condition.
0	SPR0	

Table 14–1. Slave Select Pin Configurations

DISMODF	SSOE	Master Mode (MSTR = 1)	Slave Mode (MSTR = 0)
0	x	$\overline{SS}$ input for Mode Fault	$\overline{SS}$ Input for Slave select
1	0	General purpose I/O	
1	1	Automatic $\overline{SS}$ output	

SPSR – Serial Peripheral Status Register

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-
r/w	r/w	r/w	r/w	r/w	-	-	-

Address: F4H reset value: 0000 0000b

Bit	Name	Description
7	SPIF	**SPI complete flag.** This bit is set to logic 1 via hardware while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. If ESPI (EIE .0) and EA are enabled, an SPI interrupt will be required. This bit must be cleared via software. Attempting to write to SPDR is inhibited if SPIF is set.
6	WCOL	**Write collision error flag.** This bit indicates a write collision event. Once a write collision event occurs, this bit will be set. It must be cleared via software.
5	SPIOVF	**SPI overrun error flag.** This bit indicates an overrun event. Once an overrun event occurs, this bit will be set. If ESPI and EA are enabled, an SPI interrupt will be required. This bit must be cleared via software.
4	MODF	**Mode Fault error flag.** This bit indicates a Mode Fault error event. If $\overline{SS}$ pin is configured as Mode Fault input (MSTR = 1 and DISMODF = 0) and $\overline{SS}$ is pulled low by external devices, a Mode Fault error occurs. Instantly MODF will be set as logic 1. If ESPI and EA are enabled, an SPI interrupt will be required. This bit must be cleared via software.

Bit	Name	Description
3	DISMODF	**Disable Mode Fault error detection.** This bit is used in combination with the SSOE (SPCR.7) bit to determine the feature of $\overline{SS}$ pin as shown in Table 14–1. Slave Select Pin Configurations. DISMODF affects only in Master mode (MSTR = 1). 0 = Mode Fault detection is not disabled. $\overline{SS}$ serves as input pin for Mode Fault detection disregard of SSOE. 1 = Mode Fault detection is disabled. The feature of $\overline{SS}$ follows SSOE bit.
2:0	-	**Reserved.**

SPDR – Serial Peripheral Data Register

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

Address: F5H reset value: 0000 0000b

Bit	Name	Description
7:0	SPDR[7:0]	**Serial peripheral data.** This byte is used of transmitting or receiving data on SPI bus. A write of this byte is a write to the shift register. A read of this byte is actually a read of the read data buffer. In Master mode, a write to this register initiates transmission and reception of a byte simultaneously.

14.4 Operating Modes

14.4.1 Master mode

The SPI can operate in Master mode while MSTR (SPCR.4) is set as 1. Only one Master SPI device can initiate transmissions. A transmission always begins by Master through writing to SPDR. The byte written to SPDR begins shifting out on MOSI pin under the control of SPCLK. Simultaneously, another byte shifts in from the Slave on the MISO pin. After 8-bit data transfer complete, SPIF (SPSR.7) will automatically set via hardware to indicate one byte data transfer complete. At the same time, the data received from the Slave is also transferred in SPDR. The user can clear SPIF and read data out of SPDR.

14.4.2 Slave Mode

When MSTR is 0, the SPI operates in Slave mode. The SPCLK pin becomes input and it will be clocked by another Master SPI device. The $\overline{SS}$ pin also becomes input. The Master device cannot exchange data with the Slave device until the $\overline{SS}$ pin of the Slave device is externally pulled low. Before data transmissions occurs, the $\overline{SS}$ of the Slave device must be pulled and remain low until the transmission is complete. If $\overline{SS}$ goes

high, the SPI is forced into idle state. If the $\overline{SS}$ is force to high at the middle of transmission, the transmission will be aborted and the rest bits of the receiving shifter buffer will be high and goes into idle state.

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

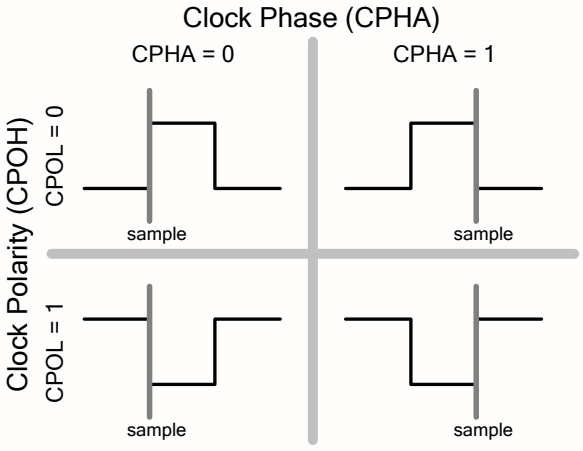


Figure 14–4. SPI Clock Formats

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) is set in both Master and Slave. 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.

Concerning the Slave mode, the $\overline{SS}$ signal needs to be taken care. As shown in Figure 14–4. SPI Clock Formats, when CPHA = 0, the first SPCLK edge is the sampling strobe of MSB (for an example of LSBFE = 0, MSB first). Therefore, the Slave must 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 must 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 must be disabled first.

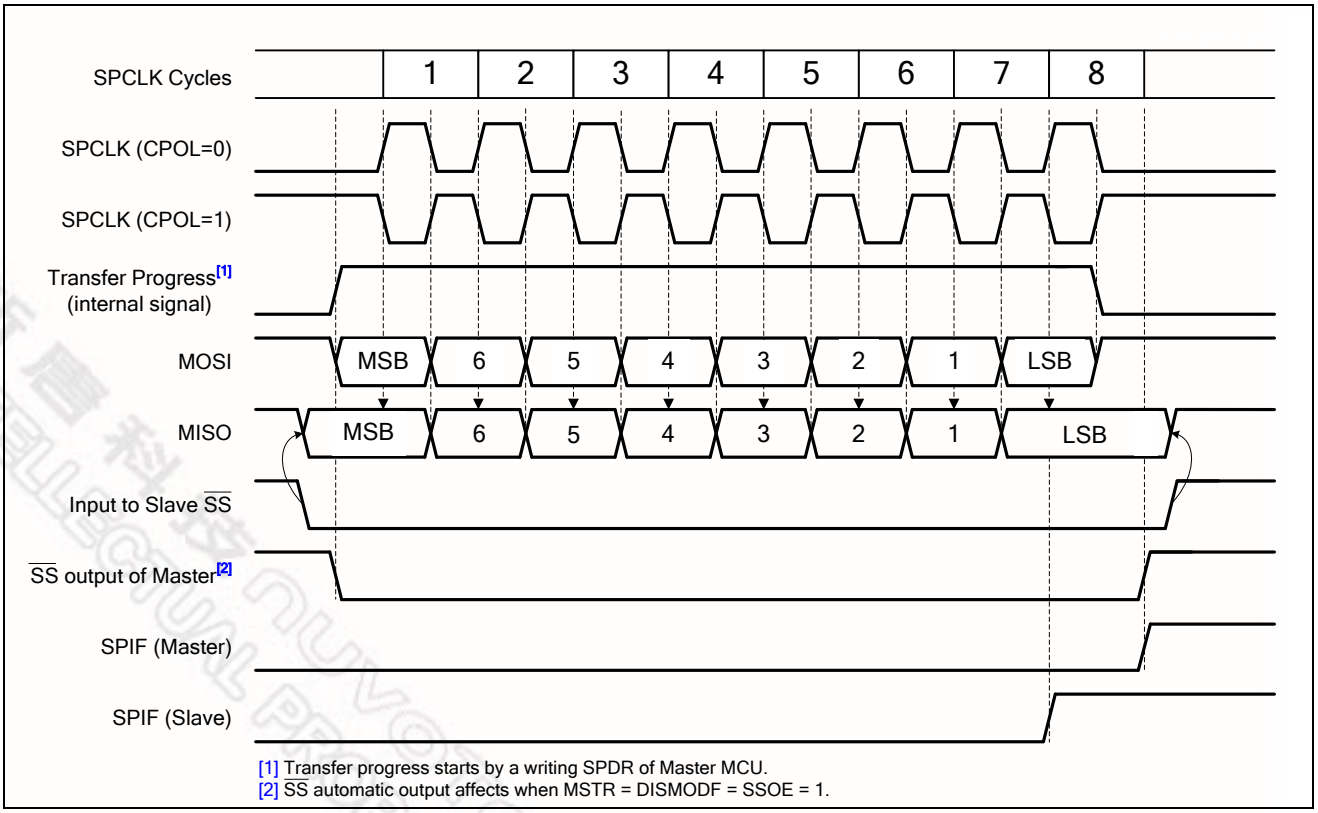


Figure 14–5. SPI Clock and Data Format with CPHA = 0

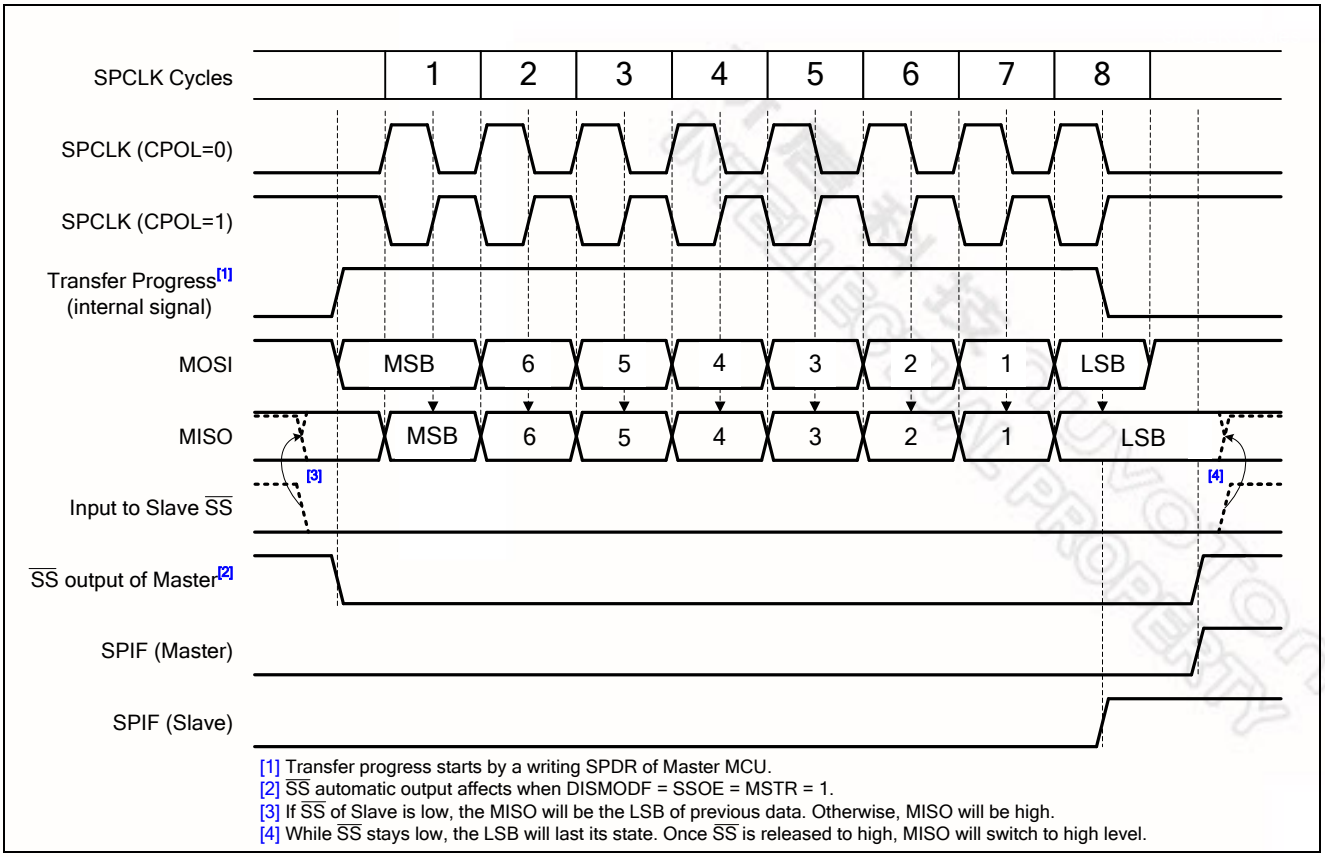


Figure 14–6. SPI Clock and Data Format with CPHA = 1

14.6 Slave Select Pin Configuration

N78E517A SPI gives a flexible $\overline{SS}$ pin feature for different system requirements. When the SPI operates as a Slave, $\overline{SS}$ pin always rules as Slave select input. When the Master mode is enabled, $\overline{SS}$ has three different functions according to DISMODF (SPSR.3) and SSOE (SPCR.7). By default, DISMODF is 0. It means that the Mode Fault detection activates. $\overline{SS}$ is configured as a input pin to check if the Mode Fault appears. On the contrary, if DISMODF is 1, Mode Fault is inactivated and the SSOE bit takes over to control the function of the $\overline{SS}$ pin. While SSOE is 1, it means the Slave select signal will generate automatically to select a Slave device. The $\overline{SS}$ as output pin of the Master usually connects with the $\overline{SS}$ input pin of the Slave device. The $\overline{SS}$ output automatically goes low for each transmission when selecting external Slave device and goes high during each idle state to de-select the Slave device. While SSOE is 0 and DISMODF is 1, $\overline{SS}$ is no more used by the SPI and reverts to be a general purpose I/O pin.

14.7 Mode Fault Detection

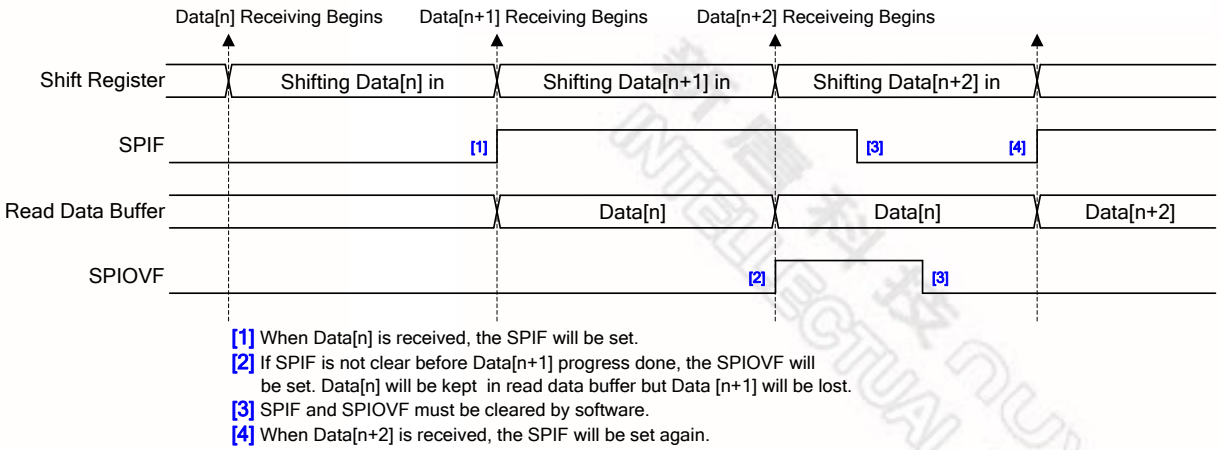
The Mode Fault detection is useful in a system where more than one SPI devices might become Masters at the same time. It may induce data contention. When the SPI device is configured as a Master and the $\overline{SS}$ input line is configured for Mode Fault input depending on Table 14–1. Slave Select Pin Configurations, a Mode Fault error occurs once the $\overline{SS}$ is pulled low by others. It indicates that some other SPI device is trying to address this Master as if it is a Slave. Instantly the MSTR and SPIEN control bits in the SPCR are cleared via hardware to disable SPI, Mode Fault flag MODF (SPSR.4) is set and an interrupt is generated if ESPI (EIE .0) and EA are enabled.

14.8 Write Collision Error

The SPI is signal buffered in the transfer direction and double buffered in the receiving direction. New data for transmission cannot be written to the shift register until the previous transaction is complete. Write collision occurs while an attempt was made to write data to the SPDR while a transfer was in progress. SPDR is not double buffered in the transmit direction. Any writing to SPDR cause data to be written directly into the SPI shift register. Once a write collision error is generated, WCOL (SPSR.6) will be set as a 1 via hardware to indicate a write collision. In this case, the current transferring data continues its transmission. However the new data that caused the collision will be lost. Although the SPI logic can detect write collisions in both Master and Slave modes, a write collision is normally a Slave error because a Slave has no indicator when a Master initiates a transfer. During the receive of Slave, a write to SPDR causes a write collision in Slave mode. WCOL flag needs to be cleared via software.

14.9 Overrun Error

For receiving data, the SPI is double buffered in the receiving direction. The received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial byte. However, the received data must be read from SPDR before the next data has been completely shifted in. As long as the first byte is read out of the read data buffer and SPIF is cleared before the next byte is ready to be transferred, no overrun error condition occurs. Otherwise the overrun error occurs. In this condition, the second byte data will not be successfully received into the read data register and the previous data will remains. If overrun occur, SPIOVF (SPSR.5) will be set via hardware. An SPIOVF setting will also require an interrupt if enabled. Figure 14–7. SPI Overrun Waveform shows the relationship between the data receiving and the overrun error.



[1] When Data[n] is received, the SPIF will be set.
[2] If SPIF is not clear before Data[n+1] progress done, the SPIOVF will be set. Data[n] will be kept in read data buffer but Data [n+1] will be lost.
[3] SPIF and SPIOVF must be cleared by software.
[4] When Data[n+2] is received, the SPIF will be set again.

Figure 14–7. SPI Overrun Waveform

14.10 SPI Interrupts

Three SPI status flags, SPIF, MODF, and SPIOVF, can generate an SPI event interrupt requests. All of them locate in SPSR. SPIF will be set after completion of data transfer with external device or a new data have been received and copied to SPDR. MODF becomes set to indicate a low level on $\overline{SS}$ causing the Mode Fault state. SPIOVF denotes a receiving overrun error. If SPI interrupt mask is enabled via setting ESPI (EIE.0) and EA is 1, CPU will executes the SPI interrupt service routine once any of these three flags is set. The user needs to check flags to determine what event caused the interrupt. These three flags are software cleared.

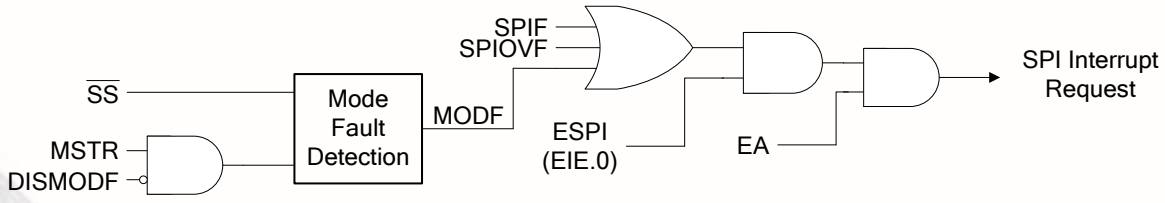


Figure 14–8. SPI Interrupt Request

15. PULSE WIDTH MODULATOR (PWM)

N78E517A 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.

Bit	Name	Description
0	PWM0EN	**PWM0 enable.** 0 = PWM0 is disabled and stops. 1 = PWM0 is enabled and runs.

PWMCON1 – PWM Control 1

7	6	5	4	3	2	1	0
-	-	-	-	-	PWM4OE	-	PWM4EN
-	-	-	-	-	r/w	-	r/w

Address: CEH reset value: 0000 0000b

Bit	Name	Description
7:3	-	**Reserved.**
2	PWM4OE	**PWM4 output enable.** 0 = P1.7 serves as general purpose I/O. 1 = P1.7 serves as output pin of PWM4 signal.
1	-	**Reserved.**
0	PWM4EN	**PWM0 enable.** 0 = PWM4 is disabled and stops. 1 = PWM4 is enabled and runs.

PWMP – PWM Period

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

Address: D9H reset value: 0000 0000b

Bit	Name	Description
7:0	PWMP[7:0]	**PWM period.** This byte controls the period of the PWM output of PWM0-PWM4 channels.

PWM0 – PWM0 Duty

7	6	5	4	3	2	1	0
PWM0[7:0]							
r/w							

Address: DAH reset value: 0000 0000b

Bit	Name	Description
7:0	PWM0[7:0]	**PWM0 duty.** This byte controls the duty of the PWM0 output.

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 = $\frac{PWMx}{255}$.

PWMP gives a repetition frequency range of 122Hz to 31.25kHz (F_{PERIPH} = 16MHz). By loading the PWMx registers with either 00H or FFH, the PWM channels will generate a constant low or high level output, respectively.

When a compare register PWMx is loaded with a new value, the associated output updated immediately. It does not have to wait until the end of the current counter period.

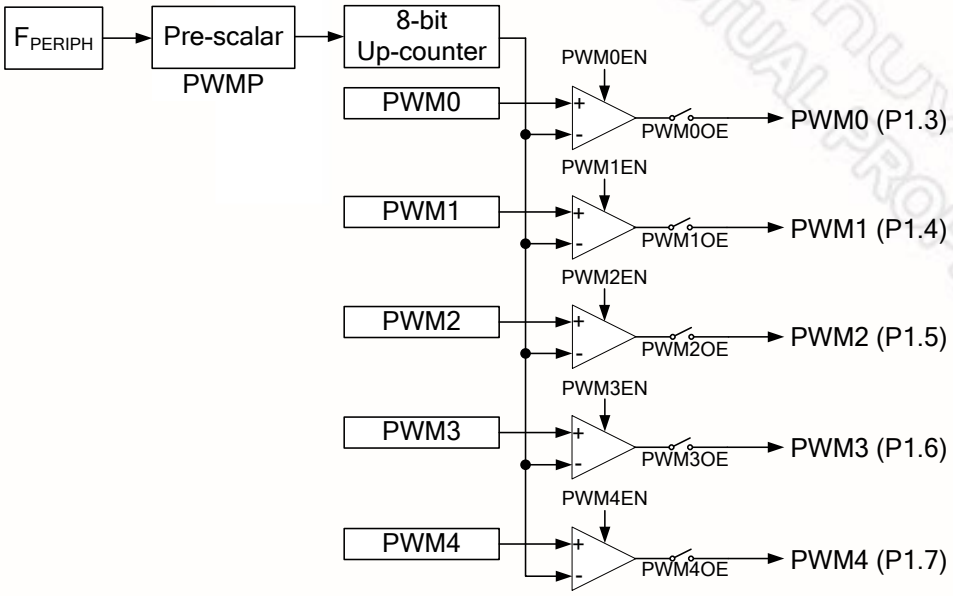


Figure 15–1. PWM Function Block

PWM demo code,

```
MOV     PWMP,#128               ;determine PWM period
MOV     PWM0,#0H                ;duty = 0%
MOV     PWM1,#40H               ;duty = 25%
MOV     PWM2,#80H               ;duty = 50%
MOV     PWM3,#0C0H              ;duty = 75%
MOV     PWM4,#0FFH              ;duty = 100%
ORL     PWMCON0,#00110011b      ;enable PWM0-3
ORL     PWMCON1,#00000001b      ;enable PWM4
ORL     PWMCON0,#11001100b      ;output enable PWM0-3
ORL     PWMCON1,#00000100b      ;output enable PWM4
```

16. TIMED ACCESS PROTECTION (TA)

N78E517A has several features like the Watchdog Timer, the ISP function, Boot select control, etc. are crucial to proper operation of the system. If leaving these control registers unprotected, errant code may write undetermined value into them, it results in incorrect operation and loss of control. In order to prevent this risk, the N78E517A has a protection scheme which limits the write access to critical SFRs. This protection scheme is done using a timed access. The following registers are related to TA process.

TA – Timed Access

7	6	5	4	3	2	1	0
TA[7:0]							
w							

Address: C7H reset value: 0000 0000b

Bit	Name	Description
7:0	TA[7:0]	**Timed access.** The timed access register controls the access to protected SFRs. To access protected bits, the user must first write AAH to the TA and immediately followed by a write of 55H to TA. After these two steps, a writing permission window is opened for three machine-cycles during which the user may write to protected SFRs.

In timed access method, the bits, which are protected, have a timed write enable window. A write is successful only if this window is active; otherwise, the write will be discarded. When the software writes AAH to TA, a counter is started. This counter waits for three machine-cycles looking for a write of 55H to TA. If the second write of 55H occurs within three machine-cycles of the first write of AAH, then the timed access window is opened. It remains open for three machine-cycles during which the user may write to the protected bits. After three machine-cycles, this window automatically closes. Once the window closes, the procedure must be repeated to access the other protected bits. Not that the TA protected SFRs are required timed access for writing. But the reading is not protected. The user may read TA protected SFR without giving AAH and 55H to TA. The suggestion code for opening the timed access window is shown below.

```
(CLR    EA)                 ;if any interrupt is enabled, disable temporally
 MOV    TA,#0AAH
 MOV    TA,#55H
(Instruction that writes a TA protected register)
(SETB   EA)                 ;resume interrupts enabled
```

The writings of AAH, 55H to TA register and the writing-protection register must occur within 3 machine-cycles of each other. Any enabled interrupt should be disabled during this procedure to avoid delay between these three writings. If there is no interrupt enabled, the `CLR EA` and `SETB EA` instructions can be left out. Once the timed access window closes, the procedure must be repeated to access the other protected bits.

Examples of timed assessing are shown to illustrate correct or incorrect writing processes.

Example 1,

```
MOV     TA,#0AAH            ;2 machine-cycles.
MOV     TA,#55H             ;2 machine-cycles.
ORL     CHPCON,#data        ;2 machine-cycles.
```

Example 2,

```
MOV     TA,#0AAH            ;2 machine-cycles.
MOV     TA,#55H             ;2 machine-cycles.
NOP                         ;1 machine-cycle.
NOP                         ;1 machine-cycle.
ANL     ISPTRG,#data        ;2 machine-cycles.
```

Example 3,

```
MOV     TA,#0AAH            ;2 machine-cycles.
NOP                         ;1 machine-cycle.
MOV     TA,#55H             ;2 machine-cycles.
MOV     WDCON,#data1        ;2 machine-cycles.
ORL     PMC,#data2          ;2 machine-cycles.
```

Example 4,

```
MOV     TA,#0AAH            ;2 machine-cycles.
NOP                         ;1 machine-cycle.
NOP                         ;1 machine-cycle.
MOV     TA,#55H             ;2 machine-cycles.
ANL     WDCON,#data         ;2 machine-cycles.
```

In the first examples, the writing to the protected bits is done before the three-machine-cycle window closes. In example 2, however, the writing to ISPTRG does not complete during the window opening, there will be no change of the value of ISPTRG. In example 3, the WDCON is successful written but the PMC access is out of the three-machine-cycle window. Therefore PMC value will not change either. In Example 4, the second write 55H to TA completes after three machine-cycles of the first write TA of AAH, therefore the timed access window in not opened at all, and the write to the protected bit fails.

In N78E517A, the TA protected SFRs includes CHPCON (9FH), ISPTRG (A4H), PMC (ACH), SHBDA (9CH), and WDCON (AAH).

17. INTERRUPT SYSTEM

The purpose of the interrupt is to make the software deal with unscheduled or asynchronous events. N78E517A 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. N78E517A Interrupt Vectors. When the interrupt occurs if enabled, the CPU will vector to the appropriate location, execute the code at this location, and stay in an interrupt service state until the ISR is done. 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. N78E517A 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	ET2	**Enable Timer 2 interrupt.** 0 = Disable all Timer 2 interrupts. 1 = Enable interrupt generated by TF2 (T2CON.7) or EXF2 (T2CON.6).
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 (P3.3).
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 (P3.2).

EIE – Extensive Interrupt Enable

7	6	5	4	3	2	1	0
-	-	-	-	-	EBOD	EPDT	ESPI
-	-	-	-	-	r/w	r/w	r/w

Address: BDH reset value: 0000 0000b

Bit	Name	Description
7:3	-	**Reserved.**
2	EBOD	**Enable Brown-out detection interrupt.** 0 = Disable Brown-out detection interrupt. 1 = Enable interrupt generated by BOF (PMC.3).
1	EPDT	**Enable Power Down waking-up timer interrupt.** 0 = Disable Power Down waking-up timer interrupt 1 = Enable interrupt generated by PDTF (PDCON.5).
0	ESPI	**Enable SPI interrupt.** 0 = Disable SPI interrupt. 1 = Enable interrupt generated by SPIF (SPSR.7), SPIOVF (SPSR.5), or MODF (SPSR.4).

IP – Interrupt Priority (bit-addressable)[1]

7	6	5	4	3	2	1	0
-	-	PT2	PS	PT1	PX1	PT0	PX0
-	-	r/w	r/w	r/w	r/w	r/w	r/w

Address: B8H reset value: 0000 0000b

Bit	Name	Description
7:6	-	**Reserved.**
5	PT2	**Timer 2 interrupt priority low bit.**

Bit	Name	Description
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.**

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

EIPH – Extensive Interrupt Priority High[1]

7	6	5	4	3	2	1	0
-	-	-	-	-	PBODH	PPDTH	PSPIH
-	-	-	-	-	r/w	r/w	r/w

Address: BBH reset value: 0000 0000b

Bit	Name	Description
7:3	-	**Reserved.**
2	PBODH	**Brown-out detection interrupt priority high bit.**
1	PPDTH	**Power Down waking-up timer interrupt priority high bit.**
0	PSPIH	**SPI interrupt priority high bit.**

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

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
3	IE1	**External interrupt 1 edge flag.** This flag is set via hardware when an edge/level of type defined by IT1 is detected. If IT1 = 1, this bit will remain set until cleared via software or at the beginning of the External Interrupt 1 service routine. If IT1 = 0, this flag is the inverse of the $\overline{INT1}$ input signal's logic level.
2	IT1	**External interrupt 1 type select.** 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 select.** 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.

XICON – External Interrupt Control (bit-addressable)

7	6	5	4	3	2	1	0
PX3[1]	EX3	IE3	IT3	PX2[1]	EX2	IE2	IT2
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: C0H reset value: 0000 0000b

Bit	Name	Description
7	PX3	**External interrupt 3 priority low bit.**
6	EX3	**Enable external interrupt 3.** 0 = Disable external interrupt 3. 1 = Enable interrupt generated by $\overline{INT3}$ pin (P4.2).
5	IE3	**External interrupt 3 edge flag.** This flag is set via hardware when an edge/level of type defined by IT3 is detected. If IT3 = 1, this bit will remain set until cleared via software or at the beginning of the External Interrupt 3 service routine. If IT3 = 0, this flag is the inverse of the $\overline{INT3}$ input signal's logic level.
4	IT3	**External interrupt 3 type select.** This bit selects whether the $\overline{INT3}$ pin will detect falling edge or low level triggered interrupts. 0 = $\overline{INT3}$ is low level triggered. 1 = $\overline{INT3}$ is falling edge triggered.
3	PX2	**External interrupt 2 priority low bit.**
2	EX2	**Enable external interrupt 2.** 0 = Disable external interrupt 2. 1 = Enable interrupt generated by $\overline{INT2}$ pin (P4.3).
1	IE2	**External interrupt 2 edge flag.** This flag is set via hardware when an edge/level of type defined by IT2 is detected. If IT2 = 1, this bit will remain set until cleared via software or at the beginning of the External Interrupt 2 service routine. If IT2 = 0, this flag is the inverse of the $\overline{INT2}$ input signal's logic level.
0	IT2	**External interrupt 2 type select.** This bit selects whether the $\overline{INT2}$ pin will detect falling edge or low level triggered interrupts. 0 = $\overline{INT2}$ is low level triggered. 1 = $\overline{INT2}$ is falling edge triggered.

[1] PX2 and PX3 are used in combination with the PX2H (IPH.6) and PX3H (IPH.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.

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can be either edge or level triggered depending on bits IT0 (TCON.0) and IT1 (TCON.2). The bits IE0 (TCON.1) and IE1 (TCON.3) are the flags which are checked to generate the interrupt. In the edge triggered mode, the $\overline{INT0}$ or $\overline{INT1}$ inputs are sampled in every machine-cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IE0 or IE1 will be set. Since the external interrupts are sampled every machine-cycle, they have to be held high or low for at least one complete machine-cycle. The IE0 and IE1 are automatically cleared when the interrupt service routine is called. If the level triggered mode is selected, then the requesting source has to

hold the pin low till the interrupt is serviced. The IE0 and IE1 will not be cleared by the hardware on entering the service routine. In the level triggered mode, IE0 and IE1 follows the inverse value of $\overline{INT0}$ and $\overline{INT1}$ pins. If interrupt pins continue to be held low even after the service routine is completed, the processor will acknowledge another interrupt request from the same source. N78E517A (on PLCC-44, PQFP-44, TQFP-44, and LQFP-48 packages) possessed other two external interrupts $\overline{INT2}$ and $\overline{INT3}$. Their setting and operation are just the same as interrupt 0 and 1. All configuring bits locate in XICON. The individual interrupt flag corresponding to external interrupt 2 to 3 will also be automatically cleared via hardware once its own interrupt service routine is executed.

The Timer 0 and 1 Interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and Timer 1 and automatically cleared by the hardware when the timer interrupt is serviced. TF2 or EXF2 flag generates the Timer 2 interrupt. These flags are set by overflow, capture, or reload events in the Timer 2 operation. The hardware will not clear these flags when a Timer 2 interrupt service routine executes. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

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

The Power Down waking-up timer can be used as a simple timer. The Power Down waking-up timer interrupt flag PDTF (PDCON.5) is set once an overflow occurs. If the interrupt is enabled by the enable bit EPDT (EIE.1), then an interrupt will occur.

Brown-out detection, if enabled, can cause Brown-out flag BOF (PMC.3) to be asserted if power voltage drop below Brown-out voltage level. The interrupt will occur if BORST (PMC.4) is 0 and EBOD (EIE.2) is 1.

SPI asserts interrupt flag SPIF (SPSR.7) on completion of data transfer with an external device. If SPI interrupt enable bit ESPI (EIE.0), a serial peripheral interrupt generates. SPIF flag is software clear. MODF (SPSR.4) and SPIOVF (SPSR.5) will also generate SPI interrupt. They share the same vector address with SPIF. When interrupt is generated, the user should tell which flag requires the interrupt.

All the bits that generate interrupts can be set or reset via hardware, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing its controlling bit in the IE or EIE. IE also has a global enable bit EA (IE.7) which can be cleared to disable all the interrupts at once. It is set to enable all individually enabled interrupt.

Note that every interrupts, if enabled, is generated by a setting as a logic 1 of its interrupt flag no matter by hardware or software. The user should take care of each interrupt flag in its own interrupt service routine (ISR).

Most of interrupt flags must be cleared by writing it as a logic 0 via software. Without clearing the flag, the ISR of corresponding interrupt source will execute again and again non-stopped.

17.1 Priority Level Structure

There are four priority levels for the interrupts, highest, high, low, and lowest. The interrupt sources can be individually set to one of four priority levels by setting their own priority bits. Table 17–2 lists four priority setting. Naturally, a low priority interrupt can itself be interrupted by a high priority interrupt, but not by another same level interrupt or lower level. A highest priority cannot be interrupted by any other interrupt source. In addition, there exists a pre-defined hierarchy among the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown on Table 17–3. It also summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, natural priority and the permission to wake up the CPU from Power Down mode. For details of waking CPU up from Power Down mode, please see Section 19.2 “Power Down Mode” on page 101.

Table 17–2. Interrupt Priority Level Setting

Interrupt Priority Control Bits		Interrupt Priority Level
IPH / EIPH	IP / EIP / XICON[7,3]	
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

Table 17–3. Characteristics of Each Interrupt Source

Source	Vector Address	Flag	Enable Bit	Natural Priority	Priority Control Bits	Power Down Waking up
External interrupt 0	0003H	IE0[1]	EX0	1	PX0, PX0H	Yes
Timer 0 overflow	000BH	TF0[2]	ET0	2	PT0, PT0H	No
External interrupt 1	0013H	IE1[1]	EX1	3	PX1, PX1H	Yes
Timer 1 overflow	001BH	TF1[2]	ET1	4	PT1, PT1H	No
Serial port (UART)	0023H	RI + TI	ES	5	PS, PSH	No
Timer 2 overflow / capture / reload	002BH	TF2[2] + EXF2	ET2	6	PT2, PT2H	No
External interrupt 2	0033H	IE2[1]	EX2	7	PX2, PX2H	Yes
External interrupt 3	003BH	IE3[1]	EX3	8	PX3, PX3H	Yes

Source	Vector Address	Flag	Enable Bit	Natural Priority	Priority Control Bits	Power Down Waking up
SPI interrupt	0043H	SPIF (SPSR.7) + MODF (SPSR.4) + SPIOVF (SPSR.5)	ESPI (EIE.0)	9	PSPI (EIP.0), PSPIH (EIPH.0)	No
Power Down waking-up timer interrupt	004BH	PDTF (PDCON.5)	EPDT (EIE.1)	10	PPDT (EIP.1), PPDTH (EIPH.1)	Yes
Brown-out interrupt	0053H	BOF (PMC.3)	EBOD (EIE.2)	11	PBOD (EIP.2), PBODH (EIPH.2)	Yes

[1] While the external interrupt pin is set as edge triggered (ITx = 1), its own flag Iex will be automatically cleared if the interrupt service routine (ISR) is executed. While as level triggered (Itx = 0), Iex follows the inverse of respective pin state. It is not controlled via software.

[2] TF0 and TF1 will be automatically cleared if the interrupt service routine (ISR) is executed. But be aware that TF2 will not.

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

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

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

3. The current instruction does not involve a write to any enable or priority setting bits 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. If an interrupt flag is active in one cycle but not responded to for the above conditions are not met, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. This means that the interrupt flag, which was once active but not serviced, is 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 action may or may not clear the flag, which caused the interrupt according to different interrupt source. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack RAM but does not save the Program Status Word (PSW). The PC is reloaded with the vector address of that interrupt which caused the LCALL. Execution continues from the vectored address till an RETI instruction is executed. On execution of the RETI instruction the processor pops the Stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what it was after the hardware LCALL. If the execution is to return to the interrupted program, the processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a simple RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the Interrupt controller that the interrupt service routine is com-

pleted. RET would leave the controller still thinking that the service routine is underway, making future interrupts impossible.

17.2 Interrupt Latency

The response time for each interrupt source depends on several factors, such as the nature of the interrupt and the instruction underway. In the case of external interrupts $\overline{INT0}$ and $\overline{INT1}$, they are sampled at every machine-cycle and then their corresponding interrupt flags IE0 or IE1 will be set or reset. The value are not actually polled by the circuit until the next machine-cycle. If a request is active and all three previous conditions are met, then the hardware generated LCALL is executed. This LCALL itself takes 2 machine-cycles to be completed. Thus there is a minimum time of 3 machine-cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last machine-cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the device is performing a write to IE, IP and then executes a MUL or DIV instruction. From the time an interrupt source is activated, the longest reaction time is 9 machine-cycles. This time includes 1 machine-cycle to detect the interrupt, 2 machine-cycles to complete the IE, EIE, IP, IPH, EIP, or EIPH access, 4 machine-cycles to complete the MUL or DIV instruction and 2 machine-cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system the interrupt response time will always be more than 3 machine-cycles and not more than 9 machine-cycles.

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. N78E517A 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 N78E517A. 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, N78E517A 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.

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. N78E517A 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. I 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 into external one. This bit should be cleared via software.
5	LDUEN	**Updating LDROM enable.** 0 = The LDROM is inhibited to be erased or programmed when APROM code runs. LDROM remains read-only. 1 = The LDROM is allowed to be fully accessed when APROM code runs.
0	ISPEN	**ISP enable.** 0 = Enable ISP function. 1 = Disable ISP function. To enable ISP function will start the internal 22.1184MHz RC oscillator for timing control. To clear ISPEN should always be the last instruction after ISP operation in order to stop internal RC for reducing power consumption.

ISPCN – ISP Control

7	6	5	4	3	2	1	0
ISPA.17	ISPA.16	FOEN	FCEN	FCTRL.3	FCTRL.2	FCTRL.1	FCTRL.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: AFH reset value: 0000 0000b

Bit	Name	Description
7:6	ISPA[17:16]	**ISP control.** This byte is for ISP controlling command to decide ISP destinations and actions. For details, see Table 18–1. ISP Modes and Command Codes.
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

ISPAH – ISP Address High Byte

7	6	5	4	3	2	1	0
ISPA[15:8]							
r/w							

Address: A7H reset value: 0000 0000b

Bit	Name	Description
7:0	ISPA[15:8]	**ISP address high byte.** ISPAH contains address ISPA[15:8] for ISP operations.

ISPAL – ISP Address Low Byte

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

Address: A6H reset value: 0000 0000b

Bit	Name	Description
7:0	ISPA[7:0]	**ISP address low byte.** ISPAL contains address ISPA[7:0] for ISP operations.

ISPFD – ISP Flash Data

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

Address: AEH reset value: 0000 0000b

Bit	Name	Description
7:0	ISPFD[7:0]	**ISP flash data.** This byte contains flash data which is read from or is going to be written to the flash memory. The user should write data into ISPFD for program mode before triggering ISP processing and read data from ISPFD for read/verify mode after ISP processing is finished.

ISPTRG – ISP Trigger (TA protected)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	ISPGO
-	-	-	-	-	-	-	w

Address: A4H reset value: 0000 0000b

Bit	Name	Description
0	ISPGO	**ISP go.** ISP begins by setting this bit as a logic 1. After this instruction, the CPU holds the Program Counter (PC) and the ISP hardware automation takes over to control the progress. After ISP action completed, the Program Counter continues to run the following instructions. The ISPGO bit will be automatically cleared and always read as logic 0.

18.2 ISP Commands

N78E517A provides a wide application to perform ISP to APROM, LDROM or on-chip Data Flash. The ISP action mode and the destination of the flash block are defined by ISP control register ISPCN.

Table 18–1. ISP Modes and Command Codes

ISP Mode	ISPCN				ISPAH, ISPAL ISPA[15:0]	ISPFD[7:0]
	ISPA.17, ISPA.16	FOEN	FCEN	FCTRL[3:0]		
Standby	X, X[1]	1	1	X	X	X
APROM and Data Flash Page Erase	0, 0	1	0	0010	Address in[2]	X
LDROM Page Erase	0, 1	1	0	0010	Address in[2]	X
APROM and Data Flash Program	0, 0	1	0	0001	Address in	Data in
LDROM Program	0, 1	1	0	0001	Address in	Data in
APROM and Data Flash Read	0, 0	0	0	0000	Address in	Data out
LDROM Read	0, 1	0	0	0000	Address in	Data out
All CONFIG bytes Erase	1, 1	1	0	0010	00XXH	X
CONFIG Program	1, 1	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG3: 0003H	Data in
CONFIG Read	1, 1	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG3: 0003H	Data out

[1] "x" means "don't care".
[2] Each page is 256-byte size. Therefore, the address for Page Erase should be 0000H, 0100H, 0200H, 0300H, etc., which is incremented by one of high byte address.

18.3 User Guide of ISP

ISP facilitates the updating flash contents in a convenient way; however, the user should follow some restricted laws in order that the ISP operates correctly. Without noticing warnings will possible cause undetermined results even serious damages of devices. Be attention of these notices. Furthermore, this paragraph will also support useful suggestions during ISP procedures.

(1) If no more ISP operation needs, the user must clear ISPEN (CHPCON.0) to zero. It will make the system void to trigger ISP unaware. Furthermore, ISP requires internal 22.1184MHZ RC oscillator running. If the external clock source is chosen, disabling ISP will stop internal 22.1184MHz RC for saving power consumption. Note that a write to ISPEN is TA protected.

(2) If the loader code, which controls the ISP procedure, locates in the external Program Memory or runs from the internal into the external, the ISP will not work anymore and set error indicator ISPF for data security.

(3) CONFIG bytes can be ISP fully accessed only when loader code executing in LDROM. New CONFIG bytes other than CBS bit activate after all resets. New CBS bit activates after resets other than software reset.

(4) When the LOCK bit (CONFIG0.1) is activated, ISP reading, writing, or erasing can still be valid.

(5) ISP erasing or programming works from V_{DD} 3.0V through 5.5V.

(6) APROM and LDROM can read itself through ISP method.

During ISP progress, interrupts (if enabled) should be disabled temporally by clearing EA bit for implement limitation.

Note that If the user would like to develop your own ISP program, remember always erase and program CONFIG bytes at the last step for data security.

18.4 ISP Demo Code

```
;*****************************************************************************
;      This code illustrates how to do APROM and CONFIG ISP from LDROM.
;      APROM are re-programmed by the code to output P1 as 55h and P2 as aah.
;      The CONFIG3 is also updated to 6T mode.
;      The user should put this code in LDROM and boot from LDROM.
;*****************************************************************************
PAGE_ERASE_AP              EQU          00100010b
BYTE_PROGRAM_AP            EQU          00100001b
BYTE_READ_AP               EQU          00000000b
BYTE_READ_CONFIG           EQU          11000000b
BYTE_PROGRAM_CONFIG        EQU          11100001b
ALL_ERASE_CONFIG           EQU          11100010b

       ORG    0000h

       CALL   Enable_ISP
       CLR    EA                        ;disable all interrupts
       CALL   Erase_AP                  ;erase AP data
       CALL   Erase_AP_Verify           ;verify Erase AP data
       CALL   Program_AP                ;programming AP data
       CALL   Program_AP_Verify         ;verify Programmed AP data
       CALL   Read_Config               ;read back CONFIG3
       CALL   Erase_Config              ;erase CONFIG bytes
       CALL   Program_Config            ;programming CONFIG3 with new data
       CALL   Program_Config_Verify     ;verify Programmed CONFIG3
       CALL   Disable_ISP
       MOV    TA,#0Aah                  ;TA protection
       MOV    TA,#55h                   ;
       ANL    CHPCON,#0FDh              ;BS = 0, reset to APROM
       MOV    TA,#0Aah
       MOV    TA,#55h
       ORL    CHPCON,#80h               ;software reset and reboot from APROM
```

```
        SJMP   $
;*********************************************************************
;             ISP Function
;*********************************************************************
Enable_ISP:
        MOV    TA,#0Aah                    ;CHPCON is TA protected
        MOV    TA,#55h
        ORL    CHPCON,#00000001b           ;ISPEN = 1, enable ISP mode
        RET
Disable_ISP:
        MOV    TA,#0Aah                    ;CHPCON is TA protected
        MOV    TA,#55h
        ANL    CHPCON,#11111110b           ;ISPEN = 0, disable ISP mode
        RET
Trigger_ISP:
        MOV    TA,#0Aah
        MOV    TA,#55h
        ORL    ISPTRG,#00000001b           ;write '1' to ISPGO to trigger ISP process
        RET
;*********************************************************************
;             ISP AP Function
;*********************************************************************
Erase_AP:
        MOV    ISPCN,#PAGE_ERASE_AP
        MOV    ISPAL,#00h
        MOV    R0,#00h
Erase_AP_Loop:
        MOV    ISPAH,R0
        CALL   Trigger_ISP
        INC    R0
        CJNE   R0,#0,Erase_AP_Loop
        RET
Erase_AP_Verify:
        MOV    ISPCN,#BYTE_READ_AP
        MOV    ISPAH,#00h
        MOV    ISPAL,#00h
Erase_AP_Verify_Loop:
        MOV    ISPFD,#00h                  ;clear ISPFD Data
        CALL   Trigger_ISP
        MOV    A,ISPFD
        CJNE   A,#0FFh,Erase_AP_Verify_Error
        INC    ISPAL
        MOV    A,ISPAL
        CJNE   A,#0,Erase_AP_Verify_Loop
        INC    ISPAH
        MOV    A,ISPAH
        CJNE   A,#0,Erase_AP_Verify_Loop
        RET
Erase_AP_Verify_Error:
        CALL   Disable_ISP
        mov    P0,#00h
        SJMP   $
Program_AP:
        MOV    ISPCN,#BYTE_PROGRAM_AP
        MOV    ISPAH,#00h
        MOV    ISPAL,#00h
        MOV    DPTR,#AP_code
Program_AP_Loop:
        MOV    A,#0
```

```
        MOVC   A,@A+DPTR
        MOV    ISPFD,A
        CALL   Trigger_ISP
        INC    DPTR
        INC    ISPAL
        MOV    A,ISPAL
        CJNE   A,#8,Program_AP_Loop
        RET
Program_AP_Verify:
        MOV    ISPCN,#BYTE_READ_AP
        MOV    ISPAH,#00h
        MOV    ISPAL,#00h
        MOV    DPTR,#AP_code
Program_AP_Verify_Loop:
        MOV    ISPFD,#00h                ;clear ISPFD Data
        CALL   Trigger_ISP
        MOV    A,#0
        MOVC   A,@A+DPTR
        MOV    B,A
        MOV    A,ISPFD
        CJNE   A,B,Program_AP_Verify_Error
        INC    DPTR
        INC    ISPAL
        MOV    A,ISPAL
        CJNE   A,#8,Program_AP_Verify_Loop
        RET
Program_AP_Verify_Error:
        CALL   Disable_ISP
        mov    P0,#00h
        SJMP   $
;*********************************************************************
;             ISP Config Function
;*********************************************************************
Erase_Config:
        MOV    ISPCN,#ALL_ERASE_CONFIG
        MOV    ISPAH,#00h
        CALL   Trigger_ISP
        RET
Read_Config:
        MOV    ISPCN,#BYTE_READ_CONFIG
        MOV    ISPAH,#00h
        MOV    ISPAL,#03h
        CALL   Trigger_ISP
        MOV    A,ISPFD
        RET
Program_Config:
        MOV    ISPCN,#BYTE_PROGRAM_CONFIG
        MOV    ISPAH,#00h
        MOV    ISPAL,#03h
        ANL    A,#10111111b
        MOV    ISPFD,A                   ;switch to 6T mode
        MOV    R0,A                      ;temp data
        CALL   Trigger_ISP
        RET
Program_Config_Verify:
        MOV    ISPCN,#BYTE_READ_CONFIG
        MOV    ISPAH,#00h
        MOV    ISPAL,#03h
        MOV    ISPFD,#00h                ;clear ISPFD Data
        CALL   Trigger_ISP
        MOV    B,R0
        MOV    A,ISPFD
```

```
        CJNE   A,B,Program_CONFIG_Verify_Error
        RET
Program_CONFIG_Verify_Error:
        CALL   Disable_ISP
        mov    P0,#00h
        SJMP   $
;*****************************************************************
;            APROM code
;*****************************************************************
AP_code :
        DB     75h, 90h, 55h              ;OPCODEs of "mov    P1,#55h"
        DB     75h,0A0h,0Aah              ;OPCODEs of "mov    P2,#0aah"
        DB     80h,0Feh                   ;OPCODEs of "sjmp   $"

        END
```

19. POWER SAVING MODES

N78E517A has several features that help the user to control the power consumption of the device. The power saved features have the Power Down mode and the Idle mode of operation. For a stable current consumption, states of P0 pins should be taken care of. P0 should be set as 0 if floating or external pull-downs exist. Or P0 should be set as 1 if external pull-ups exist or internal pull-ups are enabled by P0UP (P0OR.0).

In system power saving modes, the Watchdog Timer should be specially taken care. The hardware will clear WDT counter automatically after entering into 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	-	-	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. N78E517A 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 that If IDL bit and PD bit are set simultaneously, the MCU will enter into 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.

19.1 Idle Mode

Idle mode suspends CPU processing by holding the Program Counter. No program code are fetched and run in Idle mode. It 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 their contents during Idle mode. The port pins hold the logical states they had at the time Idle was activated. Generally, it saves considerable power of typical half of the full operating power.

Since the clock provided for peripheral function logic circuit like timer or serial port still remain in Idle mode, the CPU can be released from the Idle mode using any of the interrupt sources if enabled. The user can put the device into Idle mode by writing 1 to the bit IDL (PCON.0). The instruction that sets the IDL bit is the last instruction that will be executed before the device goes into Idle mode.

The Idle mode can be terminated in two ways. First, any interrupt if enabled will cause an exit. It will automatically clear the IDL bit, terminate the Idle mode, and the interrupt service routine (ISR) will be executed. After using the RETI instruction to jump out of the ISR, execution of the program will be the one following the instruction which put the CPU into Idle mode. The second way to terminate the Idle mode is with any reset other than software reset. Remember that if Watchdog reset is used to exit Idle mode, the WIDPD (WDCON.4) needs to be set 1 to let Watchdog Timer keep running in Idle mode.

19.2 Power Down Mode

Power Down mode is the lowest power state that N78E517A can enter. It remain the power consumption as a ”μA” level by stopping the system clock no matter internal RC clock or external crystal. Both of CPU and peripheral functions like Timers or UART are frozen. Flash memory stops. All activity is completely stopped and the power consumption is reduced to the lowest possible value. The device can be put into Power Down mode by writing 1 to bit PD (PCON.1). The instruction that does this action will be the last instruction to be executed before the device goes into Power Down mode. In the Power Down mode, RAM maintains its content. The port pins output the values held by their respective.

There are two ways to exit N78E517A from the Power Down mode. First is with all resets except software reset. Brown-out reset will also wake up CPU from Power Down mode. Be sure that Brown-out detection is enabled before the system enters into Power Down. But for a principle of least power consumption, it is uncommon to enable Brown-out detection in Power Down mode. It is not a recommended application. Of course the RST pin reset and power-on reset will remove the Power Down status. After RST pin reset or power-on reset. The CPU is initialized and start executing program code from the beginning.

N78E517A can be woken up from the Power Down mode by forcing an external interrupt pin activated, providing the corresponding interrupt enabled and the global enable EA bit (IE.7) is set. If these conditions are met, then the trigger on the external pin will asynchronously restart the system clock. Then device executes the interrupt service routine (ISR) for the corresponding external interrupt. After the ISR is completed, the program execution returns to the instruction after the one which put the device into Power Down mode and continues.

The Power Down waking-up timer interrupt is also allowed to wake up Power Down. It is usually applied as a long period timer to monitoring a static behavior. For detail application, please see Section 12.2 “Applications of Power Down Waking-up Timer” on page 51. Brown-out interrupt is another source to wake up CPU from

Power Down. As mentioned before the user will endure the large current of Brown-out detection circuit. It is not a typical application.

20. CLOCK SYSTEM

N78E517A provides three options of the system clock source. It is configured by FOSC (CONFIG3.1),which switches the system clock from crystal/resonator/external clock from XTAL1 pin or on-chip RC oscillator. N78E517A embeds an on-chip RC oscillator of 22.1184MHz/11.0592MHz selected by CONFIG setting, factory trimmed to ± 1% at room temperature. If the external clock source is from the crystal, the frequency supports from 4MHz to 40MHz.

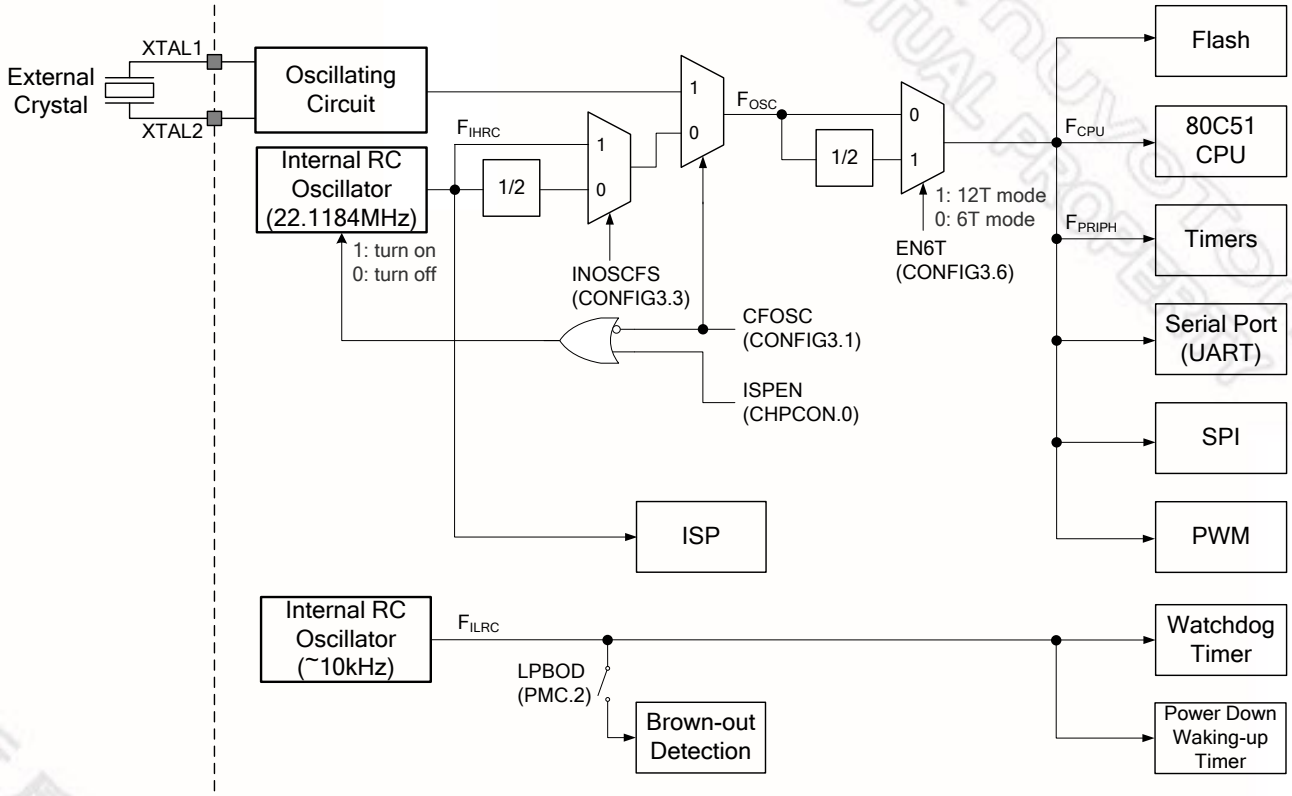


Figure 20–1. Clock System Block Diagram

20.1 12T/6T mode

The clock for the entire circuit and peripherals is normally divided by 2 before being used by the CPU core and peripherals. In 6T mode, this divider is bypassed. This facility provides the same performance when operating with a 24MHz oscillator in 12T mode as with a 12MHz oscillator in 6T mode, for example. The user may choose a divided-by-2 frequency oscillator in 6T mode to reach the same performance as in the original 12T mode. Therefore, it reduces EMI and power consumption if 6T mode is used.

CONFIG3

7	6	5	4	3	2	1	0
CWDTEN	EN6T	ROG	CKF	INTOSCFS	-	FOSC	-
r/w	r/w	r/w	r/w	r/w	-	r/w	-

unprogrammed value: 1111 1111b

Bit	Name	Description
6	EN6T	**Enable 6T mode.** This bit switches MCU between 12T and 6T mode. See Figure 20–1. Clock System Block Diagram for definitions in details. 1 = MCU runs at 12T mode. Each machine-cycle is equal to 12 clocks of system oscillator. The operating mode is the same as a standard 8051 MCU. (F_{CPU} and F_{PERIPH} is a half of F_{OSC}.) 0 = MCU runs at 6T mode. Each machine-cycle is equal to 6 clocks of system oscillator. This mode doubles the whole chip operation compared with the standard 8051. (F_{CPU} and F_{PERIPH} is equal to F_{OSC}.)
5	ROG	**Reducing oscillator gain.** 1 = Use normal gain for crystal oscillating. The crystal frequency can be up to 40MHz. 0 = Use reduced gain for crystal oscillating. The crystal frequency should be lower than 24MHz. In reduced gain mode, it will also help to decrease EMI.
4	CKF	**Clock filter enable.** 1 = Enable clock filter. It increases noise immunity and EMC capacity. 0 = Disable clock filter.
3	INTOSCFS	**Internal RC oscillator frequency select.** 1 = Select 22.1184MHz as the system clock if internal RC oscillator mode is used. It bypasses the divided-by-2 path of internal oscillator to select 22.1184MHz output as the system clock source. 0 = Select 11.0592MHz as the system clock if internal RC oscillator mode is used. The internal RC divided-by-2 path is selected. The internal oscillator is equivalent to 11.0592MHz output used as the system clock.
2	-	**Reserved.**
1	FOSC	**Oscillator selection bit.** This bit selects the source of the system clock. 1 = Crystal, resonator, or external clock input. 0 = Internal RC oscillator.
0	-	**Reserved.**

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. N78E517A SFR Description and Reset Values

Bit	Name	Description
0	ISPEN	**ISP enable.** 0 = Enable ISP function. 1 = Disable ISP function. To enable ISP function will start the internal 22.1184MHz RC oscillator for timing control. To clear ISPEN should always be the last instruction after ISP operation in order to stop internal RC for reducing power consumption.

PMC – Power Monitoring Control (TA protected)

7	6	5	4	3	2	1	0
BODEN	-	-	BORST	BOF	LPBOD	-	BOS
r/w	-	-	r/w	r/w	r/w	-	r

Address: ACH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
3	LPBOD	**Low power Brown-out detection enable.** This bit switches the Brown-out detection into a power saving mode. This bit is only effective while BODEN = 1. 0 = Disable Brown-out power saving mode. Brown-out detection operates in normal mode if enabled. The detection is always on. 1 = Enable Brown-out power saving mode. Brown-out detection operates in power saving mode if enabled. Enable this bit will switch on internal 10kHz RC to be a timer for about 12.8ms interval of detection. The discrete detection will save much power but the hysteresis feature disappears.

20.2 External Clock Source

The system clock source can be from external XTAL1 pin. When XTAL1 pin is driven by an external clock source, XTAL2 should be left floating. XTAL1 and XTAL2 are the input and output, respectively, of an internal inverting amplifier. A crystal or resonator can be used by connecting between XTAL1 and XTAL2 pins. The crystal or resonator frequency from 4MHz up to 40MHz is allowed. While an external crystal or resonator is used, ROG (CONFIG3.5) is for half gain selection of the inverting amplifier. When the system clock is lower than 24MHz and ROG is configured as a 0, the system EMI can be reduced. CKF (CONFIG3.4) is the control bit of clock filter circuit of XTAL1 input pin.

20.3 On-chip RC Oscillator

The on-chip RC oscillator is enabled while FOSC (CONFIG3.1) is 0. Setting INTOSCFS (CONFIG3.3) logic 0 will switch to a divided-by-2 path. Note that a 0.1μF capacitor is recommended to be added on XTAL1 pin to gain the more precise frequency of the internal RC oscillator frequency if it is selected as the system clock source.

21. POWER MONITORING

In order to prevent incorrect execution during power up and power drop, N78E517A provides three power monitor functions, power-on detection, Brown-out detection, and low power detection.

21.1 Power-on Detection

The power-on detection function is designed for detecting power up after power voltage reaches to a level about 2.0V where the system can work. After power-on detected, the POF (PCON.4) will be set 1 to indicate a cold reset, a power-on reset complete. The POF flag can be cleared via software.

21.2 Brown-out Detection

The other power monitoring function, Brown-out detection circuit is for monitoring the V_{DD} level during execution. There are four programmable Brown-out trigger levels available for wide voltage applications. The four nominal levels are 2.2V, 2.7V, 3.8V, and 4.5V selected via setting CBOV[1:0] in CONFIG2. When V_{DD} drops to the selected Brown-out trigger level (V_{BOD}), the Brown-out detection logic will either reset the CPU or request a Brown-out interrupt. The user may determine Brown-out reset or interrupt enable according to different application systems.

The Brown-out detection will request the interrupt while V_{DD} drops below V_{BOD} while BORST (PMC.4) is 0. In this case, BOF (PMC.3) will set as a 1. After the user cleared this flag whereas V_{DD} remains below V_{BOD}, BOF will not set again. BOF just acknowledge the user a power drop occurs. The BOF will set 1 after V_{DD} goes higher than V_{BOD} to indicate a power resuming. The Brown-out circuit provides an useful status indicator BOS (PMC.0), which is helpful to tell a Brown-out event or power resuming event occurrence. If BORST bit is set, this will enable Brown-out reset function. After a Brown-out reset, BORF (RSR.2) will set 1 via hardware. It will not be altered by reset other than power-on. Software can clear this bit. V_{BOD} has a hysteresis of 20-200mV.

The Brown-out detection circuit also provides a low power Brown-out detection mode for power saving. When LPBOD is set 1, the Brown-out detection repeatedly senses the power voltage about every 12.8ms. For the interval counting, the internal 10kHz RC oscillator will turn on in Brown-out low power mode. Note that the hysteresis feature will disappear in low power Brown-out detection mode.

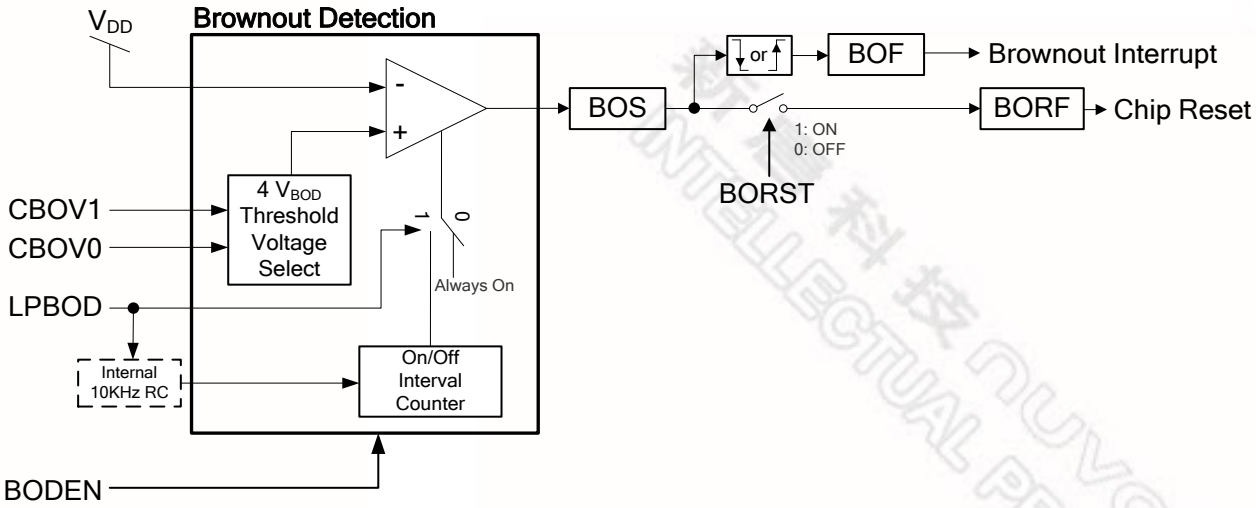
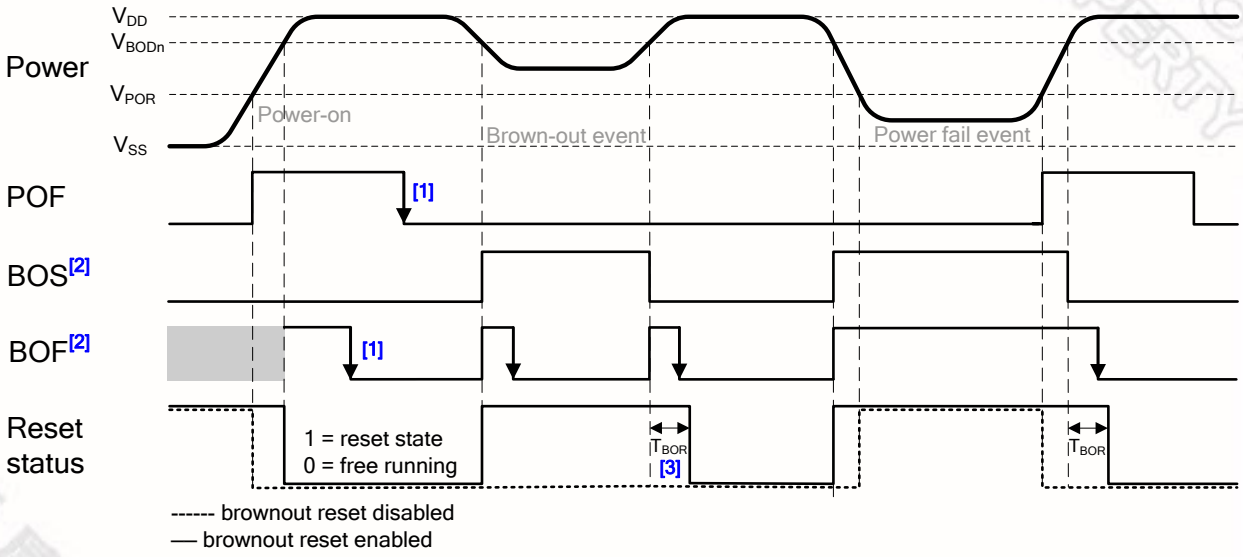


Figure 21–1. Brown-out Detection Block Diagram



[1] POF and BOF are both cleared by software.
[2] Brown-out reset is disabled. While the whole brown-out detection circuit disabled, BOS and BOF will keep 0.
[3] T_{BOR} ~= 8ms

Figure 21–2. Power Monitoring Timing Diagram

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	CBOV1	CBOV0	CBORST	-	-	-	-
r/w	r/w	r/w	r/w	-	-	-	-

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CBODEN	**CONFIG Brown-out detect enable.** 1 = Enable Brown-out detection. 0 = Disable Brown-out detection.
6 5	CBOV1 CBOV0	**CONFIG Brown-out voltage select.** These two bits select one of four Brown-out voltage level. CBOV1 CBOV0 Brown-out Voltage 1 1 2.2V 1 0 2.7V 0 1 3.8V 0 0 4.5V
4	CBORST	**CONFIG Brown-out reset enable.** This bit decides if a Brown-out reset is caused after a Brown-out event. 1 = Enable Brown-out reset when V_{DD} drops below V_{BOD}. 0 = Disable Brown-out reset when V_{DD} drops below V_{BOD}.

PMC – Power Monitoring Control (TA protected)

7	6	5	4	3	2	1	0
BODEN[1]	-	-	BORST[1]	BOF	LPBOD	-	BOS
r/w	-	-	r/w	r/w	r/w	-	r

Address: ACH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
7	BODEN	**Brown-out detect enable.** 0 = Disable Brown-out detection. 1 = Enable Brown-out detection.
6:5	-	**Reserved.**
4	BORST	**Brown-out reset enable.** This bit decides if a Brown-out reset is caused after a Brown-out event. 0 = Disable Brown-out reset when V_{DD} drops below V_{BOD}. 1 = Enable Brown-out reset when V_{DD} drops below V_{BOD}.
3	BOF	**Brown-out flag.** This flag will be set as a logic 1 via hardware after a V_{DD} dropping below or rising above V_{BOD} event occurs. If both EBOD (EIE.2) and EA (IE.7) are set, a Brown-out interrupt requirement will be generated. This bit must be cleared via software.
3	LPBOD	**Low power Brown-out detection enable.** This bit switches the Brown-out detection into a power saving mode. This bit is only effective while BODEN = 1. 0 = Disable Brown-out power saving mode. Brown-out detection operates in normal mode if enabled. The detection is always on. 1 = Enable Brown-out power saving mode. Brown-out detection operates in power saving mode if enabled. Enable this bit will switch on internal 10kHz RC to be a timer for about 12.8ms interval of detection. The discrete detection will save much power but the hysteresis feature disappears.

Bit	Name	Description
1	-	**Reserved.**
0	BOS	**Brown-out status.** This bit indicates the V_{DD} voltage level comparing with V_{BOD} while Brown-out circuit is enabled. It is helpful to tell a Brown-out event or power resuming event occurrence. This bit is read-only and keeps 0 if Brown-out detection is not enabled. 0 = V_{DD} voltage level is higher than V_{BOD}. 1 = V_{DD} voltage level is lower than V_{BOD}.

[1] BODEN and BORST will be directly loaded from CONFIG2 bit 7 and bit 4 after all resets.

Table 21–1. BOF Reset Value

Reset source	CBODEN (CONFIG2.7)	CBORST (CONFIG2.4)	V_{DD} stable level	BOF
Brown-out reset	1	1	> V_{BOD} always	1
Other resets	1	1	> V_{BOD} always	1
	1	0	> V_{BOD}	1
	1	0	< V_{BOD}	0
	0	X	X	0

Note that if BOF is 1 after chip reset, it is strongly recommended to initialize the user's program by clearing BOF.

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. N78E517A SFR Description and Reset Values

Bit	Name	Description
4	POF	**Power-on reset flag.** This bit will be set as 1 after a power-on reset. It indicates a cold reset, a power-on reset complete. This bit remains its value after any other resets. This flag is recommended to be cleared via software.

22. RESET CONDITIONS

N78E517A has several options to place device in reset condition. It also offers the software flags to indicate the source, which causes a reset. In general, most SFRs go to their reset value irrespective of the reset condition, but there are several reset source indicating flags whose state depends on the source of reset. The user can read back these flags to determine the cause of reset using software. There are 5 ways of putting the device into reset state. They are power-on reset, RST pin reset, software reset, Watchdog Timer reset, and Brown-out reset.

RSR – Reset Status Register

7	6	5	4	3	2	1	0
-	-	-	-	-	BORF	-	SWRF
-	-	-	-	-	r/w	-	r/w

Address: 96H reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
7:3	-	**Reserved.**
2	BORF	**Brown-out reset flag.** When the MCU is reset by Brown-out reset, this bit will be set via hardware. This flag is recommended to be cleared via software.
1	-	**Reserved.**
0	SWRF	**Software reset flag.** When the MCU is reset via software reset, this bit will be set via hardware. This flag is recommended to be cleared via software.

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. N78E517A SFR Description and Reset Values

Bit	Name	Description
4	POF	**Power-on reset flag.** This bit will be set as 1 after a power-on reset. It indicates a cold reset, a power-on reset complete. This bit remains its value after any other resets. This flag is recommended to be cleared via software.

WDCON – Watchdog Timer Control (TA protected)

7	6	5	4	3	2	1	0
WDTEN	WDCLR	-	WIDPD	WDTRF	WPS2	WPS1	WPS0
r/w	w	-	r/w	r/w	r/w	r/w	r/w

Address: AAH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
3	WDTRF	**Watchdog Timer reset flag.** When the CPU is reset by Watchdog Timer time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software.

22.1 Power-on Reset

N78E517A incorporate an internal voltage reference. During a power-on process of rising power supply voltage V_{DD}, this voltage reference will hold the CPU in power-on reset mode when V_{DD} is lower than the voltage reference threshold. This design makes CPU not access program flash while the V_{DD} is not adequate performing the flash reading. If a undetermined operating code is read from the program flash and executed, this will put CPU and even the whole system in to a erroneous state. After a while, V_{DD} rises above the reference threshold where the system can work, the selected oscillator will start and then program code will be executed from 0000H. At the same time, a power-on flag POF (PCON.4) will be set 1 to indicate a cold reset, a power-on reset complete. Note that the contents of internal RAM will be undetermined after a power-on. The user is recommended to give initial values for the RAM block.

The POF is recommended to be cleared to 0 via software in order to check if a cold reset or warm reset performed after the next reset occurs. If a cold reset caused by power off and on, POF will be set 1 again. If the reset is a warm reset caused by other reset sources, POF will remain 0. The user may take a different course to check other reset flags and deal with the warm reset event.

22.2 Brown-out Reset

Brown-out detection circuit is for monitoring the V_{DD} level during execution. When V_{DD} drops to the selected Brown-out trigger level (V_{BOD}), the Brown-out detection logic will reset the CPU if BORST (PMC.4) setting 1. After a Brown-out reset, BORF (RSR.2) will set 1 via hardware. It will not be altered by any reset other than a power-on reset. Software can clear this bit.

22.3 RST Pin Reset

The hardware reset input is RST pin which is the input with a Schmitt trigger. A hardware reset is accomplished by holding the RST pin high for at least two machine-cycles to ensure detection of a valid hardware reset sig-

nal. The reset circuitry then synchronously applies the internal reset signal. Thus the reset is a synchronous operation and requires the clock to be running to cause an external reset.

Once the device is in reset condition, it will remain so as long as RST pin is 1. After the RST high is removed, the CPU will exit the reset state and begin code executing from address 0000H. There is no flag associated with the RST pin reset condition. However since the other reset sources have flags, the external reset can be considered as the default reset if those reset flags are cleared.

If a RST pin reset applies while CPU is in Power Down mode, the way to trigger a hardware reset is slightly different. Since the Power Down mode stops system clock, the reset signal will asynchronously cause the system clock resuming. After the system clock is stable, CPU will enter into the reset state.

22.4 Watchdog Timer Reset

The Watchdog Timer is a free running timer with programmable time-out intervals. The user can clear the Watchdog Timer at any time, causing it to restart the count. When the selected time-out occurs, the Watchdog Timer will reset the system directly. The reset condition is maintained via hardware for two machine-cycles. After the reset is removed, the device will begin execution from 0000H.

Once a reset due to Watchdog Timer occurs the Watchdog Timer reset flag WDTRF (WDCON.3) will be set. This bit keeps unchanged after any reset other than a power-on reset. The user may clear WDTRF via software.

22.5 Software Reset

N78E517A is enhanced with a software reset. It allows the program code to reset the whole system in software approach. It is quite useful in the end of an ISP progress. For example, if an LDROM updating APROM ISP finishes and the code in APROM is correctly updated, a software reset can be asserted to reboot CPU from the APROM in order to check the result of the updated APROM program code immediately. Writing 1 to SWRST (CHPCON.7) will trigger a software reset. Note that this bit is timed access protection. See demo code below. After a software reset the SWRF (RSR.0) will be automatically set via hardware. This bit will be preserved its value after all resets except power-on reset. SWRF can also be cleared via software.

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. N78E517A SFR Description and Reset Values

Bit	Name	Description
7	SWRST	**Software reset.** To set this bit as a logic 1 will cause a software reset. It will automatically be cleared via hardware after reset in finished.

The software demo code are listed below.

```
MOV TA,#0Aah              ;TA protection.
MOV TA,#55h               ;
ANL CHPCON,#0FDh          ;BS = 0, reset to APROM.
MOV TA,#0Aah
MOV TA,#55h
ORL CHPCON,#80h           ;Software reset
```

22.6 Boot Select

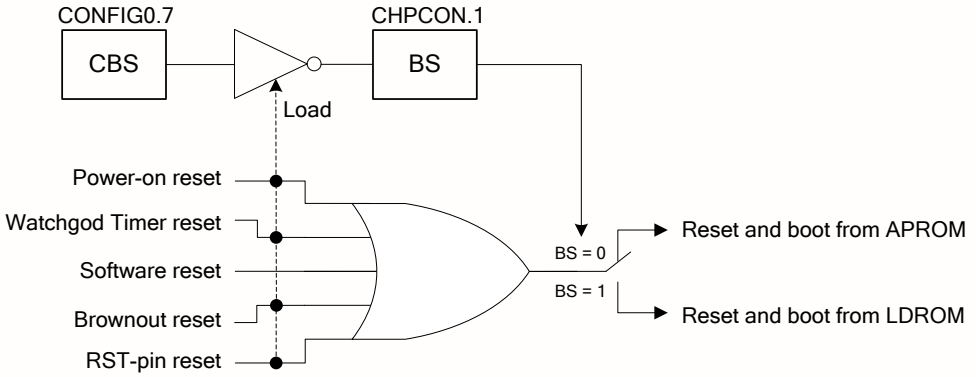


Figure 22–1. Boot Selecting Diagram

N78E517A provides users a flexible boot selection for variant application. The SFR bit BS in CHPCON.1 determines CPU booting from APROM or LDROM after any source of reset. If reset occurs and BS is 0, CPU will reboot from APPROM. Else, the CPU will reboot from LDROM.

CONFIG0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	MOVCL	LOCK	DFEN
r/w	-	-	-	-	r/w	r/w	r/w

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CBS	**CONFIG boot select.** 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.

CHPCON – Chip Control (TA protected)

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

Address: 9FH reset value: see Table 6–2. N78E517A SFR Description and Reset Values

Bit	Name	Description
1	BS	**Boot select.** 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.

[1] Note that this bit is initialized by being loaded from the inverted value of CBS bit in CONFIG0.7 at all resets except software reset. It keeps unchanged after software reset.

Note that after the CPU is released from all reset state, the hardware will always check the BS bit instead of the CBS bit to determine from APROM or LDROM that the device reboots.

22.7 Reset State

The reset state does not affect the on-chip RAM. The data in the RAM will be preserved during the reset. Note that the RAM contents may be lost if the V_{DD} falls below approximately 1.2V. It is the minimum voltage level required for RAM data retention. Therefore, after the power-on reset the RAM contents will be indeterminate. During a power fail condition. If the power falls below the data retention minimum voltage, the RAM contents will also lose.

After a reset, most of SFRs go to their initial values except bits which are affected by different reset events. See the notes of Table 6–2. N78E517A SFR Description and Reset Values. The Program Counter is forced to 0000H and held as long as the reset condition is applied. Note that the Stack Pointer is also reset to 07H,

therefore the stack contents may be effectively lost during the reset event even though the RAM contents are not altered.

After a reset, interrupts and Timers are disabled. The I/O port SFRs have FFH written into them which puts the port pins in a high state.

23. AUXILIARY FEATURES

ALE is used to enable the address latch that separates the address from the data on Port 0. ALE runs at 1/6 of the Fosc in 12T mode. An ALE pulse is omitted always. The user can turn ALE signal off via setting ALEOFF to reduce EMI. ALEOFF enable will just make ALE activating during external memory access through a MOVC or MOVX instruction. ALE will stay high in other conditions.

AUXR – Auxiliary Register

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

Address: 8EH reset value: 0000 0000b

Bit	Name	Description
7:1	-	**Reserved.**
0	ALEOFF	**ALE output off.** 0 = ALE is emitted always. 1 = ALE is off normally and active only during external memory access through a MOVX or MOVC instruction.

24. CONFIG BYTES

N78E517A has several hardware configuration bytes, called CONFIG bytes, those are used to configure the hardware options such as the security bits, system clock source, and so on. These hardware options can be re-configured through the Programmer/Writer or ISP modes. N78E517A has four CONFIG bytes 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 bytes except CBS bit in CONFIG0.) These SFR bits can be continuously controlled via user's software.

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

CONFIG0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	MOVCL	LOCK	DFEN
r/w	-	-	-	-	r/w	r/w	r/w

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CBS	**CONFIG boot select.** 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.
6:3	-	**Reserved.**
2	MOVCL	**MOVC lock enable.** This bit determines MOVC instruction is inhibited or not when reading internal Program Memory by executing on the external Program Memory. This mechanism is for data security. 1 = MOVC has no restriction. 0 = MOVC is restricted. The external Program Memory code is inhibited to read internal APROM or LDROM contents through MOVC instruction.
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.
0	DFEN	**Data Flash enable.** 1 = There is no Data Flash space. The APROM size is 64k-byte. 0 = Data Flash exists. The Data Flash and APROM share 64k bytes depending on SHBDA setting.

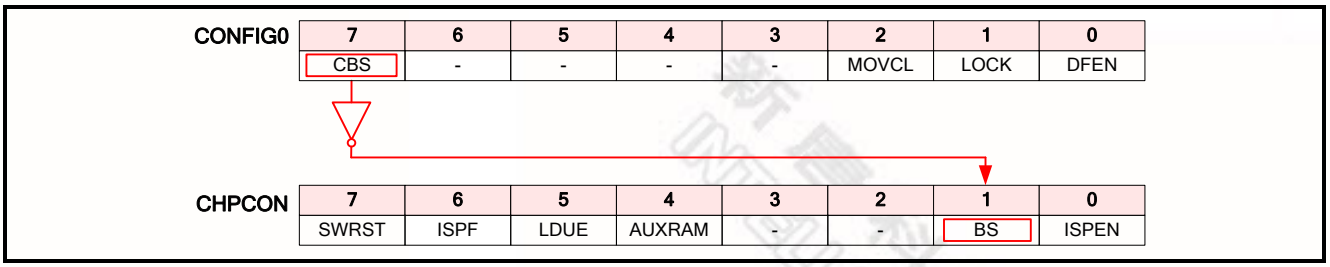


Figure 24–1. CONFIG0 Reset Reloading Except Software Reset

CONFIG1

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

unprogrammed value: 1111 1111b

Bit	Name	Description
7:0	CHBDA[7:0]	**CONFIG high byte of Data Flash starting address.** This byte is valid only when DFEN (CONFIG0.0) being 0 condition. It is used to determine the starting address of the Data Flash.

[1] Note that there will be no APROM if setting CHBDA 00H. CPU will execute codes in the external Program Memory.

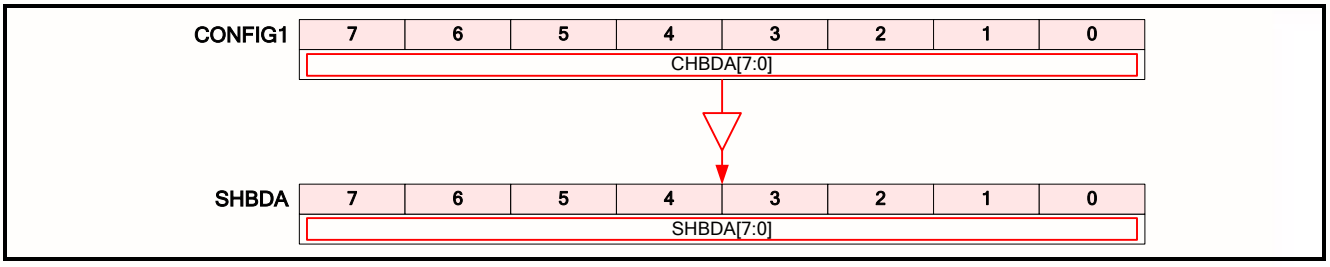


Figure 24–2. CONFIG1 Reset Reloading

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	CBOV1	CBOV0	CBORST	-	-	-	-
r/w	r/w	r/w	r/w	-	-	-	-

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CBODEN	**CONFIG Brown-out detect enable.** 1 = Enable Brown-out detection. 0 = Disable Brown-out detection.
6	CBOV1	**CONFIG Brown-out voltage select.**

Bit	Name	Description
5	CBOV0	These two bits select one of four Brown-out voltage level. CBOV1 CBOV0 Brown-out Voltage 1 1 2.2V 1 0 2.7V 0 1 3.8V 0 0 4.5V
4	CBORST	**CONFIG Brown-out reset enable.** This bit decides if a Brown-out reset is caused after a Brown-out event. 1 = Enable Brown-out reset when V_{DD} drops below V_{BOD}. 0 = Disable Brown-out reset when V_{DD} drops below V_{BOD}.
3:0	-	**Reserved.**

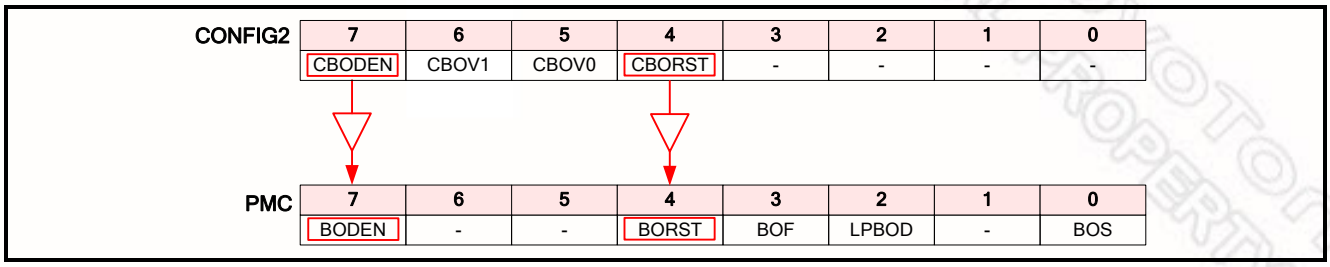


Figure 24–3. CONFIG2 Reset Reloading

CONFIG3

7	6	5	4	3	2	1	0
CWDTEN	EN6T	ROG	CKF	INTOSCFS	-	FOSC	-
r/w	r/w	r/w	r/w	r/w	-	r/w	-

unprogrammed value: 1111 1111b

Bit	Name	Description
7	CWDTEN	**CONFIG Watchdog Timer enable.** 1 = Disable Watchdog Timer after all resets. 0 = Enable Watchdog Timer after all resets.
6	EN6T	**Enable 6T mode.** This bit switches MCU between 12T and 6T mode. See Figure 20–1. Clock System Block Diagram for definitions in details. 1 = MCU runs at 12T mode. Each machine-cycle is equal to 12 clocks of system oscillator. The operating mode is the same as a standard 8051 MCU. (F_{CPU} and F_{PERIPH} is a half of F_{OSC}.) 0 = MCU runs at 6T mode. Each machine-cycle is equal to 6 clocks of system oscillator. This mode doubles the whole chip operation compared with the standard 8051. (F_{CPU} and F_{PERIPH} is equal to F_{OSC}.)
5	ROG	**Reducing oscillator gain.** 1 = Use normal gain for crystal oscillating. The frequency can be up to 40MHz. 0 = Use reduced gain for crystal oscillating. The frequency should be lower than 24MHz. In reduced gain mode, it will also help to decrease EMI.

Bit	Name	Description
4	CKF	**Clock filter enable.** 1 = Enable clock filter. It increases noise immunity and EMC capacity. 0 = Disable clock filter. ***Note that the clock filter should be always disabled if the crystal frequency is above 24MHz.***
3	INTOSCFS	**Internal RC oscillator frequency select.** 1 = Select 22.1184MHz as the system clock if internal RC oscillator mode is used. It bypasses the divided-by-2 path of internal oscillator to select 22.1184MHz output as the system clock source. 0 = Select 11.0592MHz as the system clock if internal RC oscillator mode is used. The internal RC divided-by-2 path is selected. The internal oscillator is equivalent to 11.0592MHz output used as the system clock.
2	-	**Reserved.**
1	FOSC	**Oscillator selection bit.** This bit selects the source of the system clock. 1 = Crystal, resonator, or external clock input. 0 = Internal RC oscillator.
0	-	**Reserved.**

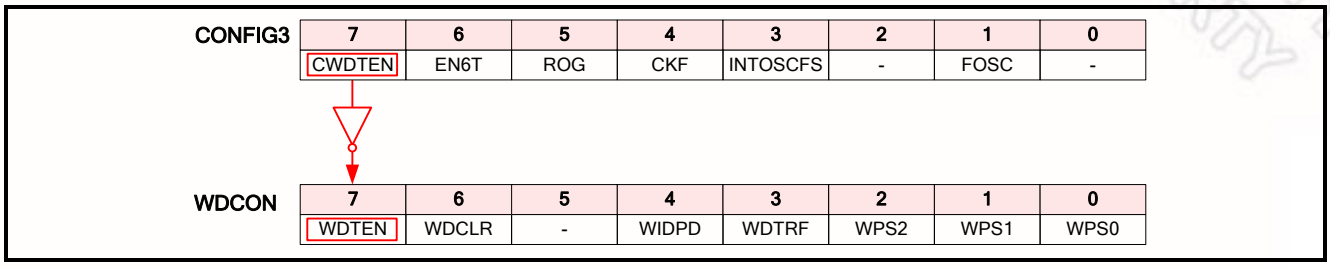


Figure 24–4. CONFIG3 Reset Reloading

25. INSTRUCTION SET

N78E517A executes all the instructions of the standard 8051 family. All instructions are coded within an 8-bit field called an OPCODE. This single byte must be fetched from Program Memory. The OPCODE is decoded by the CPU. It determines what action the microcontroller will take and whether more operation data is needed from memory. If no other data is needed, then only one byte was required. Thus the instruction is called a one byte instruction. In some cases, more data is needed. These will be two or three byte instructions.

Table 25–1 lists all instructions in details. Note of the instruction set and addressing modes are shown below.

Rn (n = 0-7)	Register R0-R7 of the currently selected Register Bank.
direct	8-bit internal data location's address. It could be an internal data RAM location (0-127) or a SFR (e.g., I/O port, control register, status register, etc. (128-255)).
@Ri (i = 0, 1)	8-bit internal data RAM location (0-255) addressed indirectly through register R0 or R1.
#data	8-bit constant included in the instruction.
#data16	16-bit constant included in the instruction.
addr16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of Program Memory as the first byte of the following instruction.
rel	Signed (2's complement) 8-bit offset byte. Used by SJMP and all conditional branches. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	Direct addressed bit in internal data RAM or SFR.

Table 25–1. Instruction Set for N78E517A

Instruction	OPCODE	Bytes	Clock Cycles in 12T Mode	Clock Cycles in 6T Mode
NOP	00	1	12	6
ADD A, Rn	28-2F	1	12	6
ADD A, @Ri	26, 27	1	12	6
ADD A, direct	25	2	12	6
ADD A, #data	24	2	12	6
ADDC A, Rn	38-3F	1	12	6
ADDC A, @Ri	36, 37	1	12	6
ADDC A, direct	35	2	12	6
ADDC A, #data	34	2	12	6
SUBB A, Rn	98-9F	1	12	6
SUBB A, @Ri	96, 97	1	12	6
SUBB A, direct	95	2	12	6
SUBB A, #data	94	2	12	6

Table 25–1. Instruction Set for N78E517A

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

Table 25–1. Instruction Set for N78E517A

Instruction		OPCODE	Bytes	Clock Cycles in 12T Mode	Clock Cycles in 6T Mode
MOV	A, #data	74	2	12	6
MOV	Rn, A	F8-FF	1	12	6
MOV	Rn, direct	A8-AF	2	24	12
MOV	Rn, #data	78-7F	2	12	6
MOV	@Ri, A	F6, F7	1	12	6
MOV	@Ri, direct	A6, A7	2	24	12
MOV	@Ri, #data	76, 77	2	12	6
MOV	direct, A	F5	2	12	6
MOV	direct, Rn	88-8F	2	24	12
MOV	direct, @Ri	86, 87	2	24	12
MOV	direct, direct	85	3	24	12
MOV	direct, #data	75	3	24	12
MOV	DPTR, #data16	90	3	24	12
MOVC	A, @A+DPTR	93	1	24	12
MOVC	A, @A+PC	83	1	24	12
MOVX	A, @Ri	E2, E3	1	24	12
MOVX	A, @DPTR	E0	1	24	12
MOVX	@Ri, A	F2, F3	1	24	12
MOVX	@DPTR, A	F0	1	24	12
PUSH	direct	C0	2	24	12
POP	direct	D0	2	24	12
XCH	A, Rn	C8-CF	1	12	6
XCH	A, @Ri	C6, C7	1	12	6
XCH	A, direct	C5	2	12	6
XCHD	A, @Ri	D6, D7	1	12	6
CLR	C	C3	1	12	6
CLR	bit	C2	2	12	6
SETB	C	D3	1	12	6
SETB	bit	D2	2	12	6
CPL	C	B3	1	12	6
CPL	bit	B2	2	12	6
ANL	C, bit	82	2	24	12
ANL	C, /bit	B0	2	24	12
ORL	C, bit	72	2	24	12
ORL	C, /bit	A0	2	24	12
MOV	C, bit	A2	2	12	6
MOV	bit, C	92	2	24	12
ACALL	addr11	11, 31, 51, 71, 91, B1, D1, F1[1]	2	24	12
LCALL	addr16	12	3	24	12

Table 25–1. Instruction Set for N78E517A

Instruction		OPCODE	Bytes	Clock Cycles in 12T Mode	Clock Cycles in 6T Mode
RET		22	1	24	12
RETI		32	1	24	12
AJMP	addr11	01, 21, 41, 61, 81, A1, C1, E1[2]	2	24	12
LJMP	addr16	02	3	24	12
JMP	@A+DPTR	73	1	24	12
SJMP	rel	80	2	24	12
JZ	rel	60	2	24	12
JNZ	rel	70	2	24	12
JC	rel	40	2	24	12
JNC	rel	50	2	24	12
JB	bit, rel	20	3	24	12
JNB	bit, rel	30	3	24	12
JBC	bit, rel	10	3	24	12
CJNE	A, direct, rel	B5	3	24	12
CJNE	A, #data, rel	B4	3	24	12
CJNE	@Ri, #data, rel	B6, B7	3	24	12
CJNE	Rn, #data, rel	B8-BF	3	24	12
DJNZ	Rn, rel	D8-DF	2	24	12
DJNZ	direct, rel	D5	3	24	12

[1] The most three significant bits in the 11-bit address [A10:A8] decide the ACALL hex code. The code will be [A10,A9,A8,1,0,0,0,1].

[2] The most three significant bits in the 11-bit address [A10:A8] decide the AJMP hex code. The code will be [A10,A9,A8,0,0,0,0,1].

26. ELECTRICAL CHARACTERISTICS

26.1 Absolute Maximum Ratings

Parameter	Rating	Unit
Operating temperature under bias	-40 to +85	°C
Storage temperature range	-55 to +150	°C
Voltage on VDD pin to V_{SS}	-0.3 to +6.5	V
Voltage on any other pin to V_{SS}	-0.3 to (V_{DD}+0.3)	V

Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. It is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

26.2 DC Electrical Characteristics

Temperature = -40℃ to +85℃; V_{SS} = 0V;

V_{DD} = 4.5V to 5.5V @ F = 0 to 40MHz (12T mode), F = 0 to 33MHz (6T mode)

V_{DD} = 2.4V to 5.5V @ F = 0 to 27MHz (12T mode), F = 0 to 20MHz (6T mode)

V_{DD} = 3.0V to 5.5V for ISP erasing or programming.

Table 26–1. DC Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
V_{IL}	Input low voltage				$0.2V_{DD}$ – 0.1	V
V_{IH}	Input high voltage (Ports 0-4, $\overline{EA}$)		$0.2V_{DD}$ + 0.9			V
V_{IH1}	Input high voltage (RST, XTAL1)		$0.7V_{DD}$			V
V_{OL}	Output low voltage[1]	V_{DD} = 4.5V, I_{OL} = 8.2mA V_{DD} = 3.0V, I_{OL} = 5.8mA V_{DD} = 2.4V, I_{OL} = 4.4mA			0.4	V
V_{OH}	Output high voltage (Ports 1-4 and Port 0 with internal pull-up enabled)	V_{DD} = 4.5V, I_{OH} = -300µA V_{DD} = 3.0V, I_{OH} = -75µA V_{DD} = 2.4V, I_{OH} = -35µA	2.4 2.4 2.0			V
V_{OH1}	Output high voltage (Ports 0 and 2 in external bus mode, ALE, $\overline{PSEN}$)	V_{DD} = 4.5V, I_{OH} = -9mA V_{DD} = 3.0V, I_{OH} = -2.4mA V_{DD} = 2.4V, I_{OH} = -1.3mA	2.4 2.4 2.0			V

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
I_{IL}	Logical 0 input current (Ports 1-4 and Port 0 with internal pull-up enabled)	V_{DD} = 5.5V, V_{IN} = 0.4V V_{DD} = 3.6V, V_{IN} = 0.4V			-50 -20	µA
I_{TL}	Logical 1-to-0 transition current[2] (Ports 1-4 and Port 0 with internal pull-up enabled)	V_{DD} = 5.5V V_{DD} = 3.6V		-570 -240	-650 -290	µA
I_{LI}	Input leakage current (Port 0)	0 < V_{IN} < V_{DD}			±10	µA
I_{DD}	Supply current[3]	V_{DD} = 5.0V, external clock, 12T			0.21F + 3.5	mA
		V_{DD} = 3.3V, external clock, 12T			0.15F + 2.9	mA
		V_{DD} = 5.0V, external clock, 6T			0.35F + 3.3	mA
		V_{DD} = 3.3V, external clock, 6T			0.32F + 2.3	mA
		V_{DD} = 5.0V, internal 22.1184MHz, 12T			4.8	mA
		V_{DD} = 3.3V, internal 11.0592MHz, 12T			3.4	mA
		V_{DD} = 5.0V, internal 22.1184MHz, 6T			7.0	mA
		V_{DD} = 3.3V, internal 11.0592MHz, 6T			4.5	mA
I_{ID}	Idle mode current	V_{DD} = 5.0V, external clock, 12T			0.11F + 2.0	mA
		V_{DD} = 3.3V, external clock, 12T			0.09F + 0.9	mA
		V_{DD} = 5.0V, external clock, 6T			0.15F + 1.7	mA
		V_{DD} = 3.3V, external clock, 6T			0.14F + 0.7	mA
		V_{DD} = 5.0V, internal 22.1184MHz, 12T			1.5	mA
		V_{DD} = 3.3V, internal 11.0592MHz, 12T			1.2	mA
		V_{DD} = 5.0V, internal 22.1184MHz, 6T			2.0	mA
		V_{DD} = 3.3V, internal 11.0592MHz, 6T			1.4	mA
I_{PD}	Power Down mode current			2	35	µA
R_{RST}	RST pin internal pull-down resistor	2.4 < V_{DD} < 5.5V	45		800	kΩ
V_{BOD0}	Brown-out threshold 2.2V		2.05	2.2	2.3	V
V_{BOD1}	Brown-out threshold 2.7V		2.6	2.7	2.85	V
V_{BOD2}	Brown-out threshold 3.8V		3.65	3.8	4.0	V
V_{BOD3}	Brown-out threshold 4.5V		4.35	4.5	4.75	V
V_{BODHYS}	Brown-out hysteresis		20		200	mV
V_{POR}	Power-on reset threshold			2.0		V

[1] Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin : 20mA
Maximum I_{OL} per 8-bit port : 40mA
Maximum total I_{OL} for all outputs: 100mA

[2] Pins of Ports 1-4 and Port 0 with internal pull-up enabled will source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V_{IN} is approximately 1.5V to 2.5V.

[3] It is measured while MCU keeps in running SJMP $ loop continuously. P0 is externally or internally pulled-up.

Figures below shows supply and Idle mode current under 12T/6T with internal program memory mode.

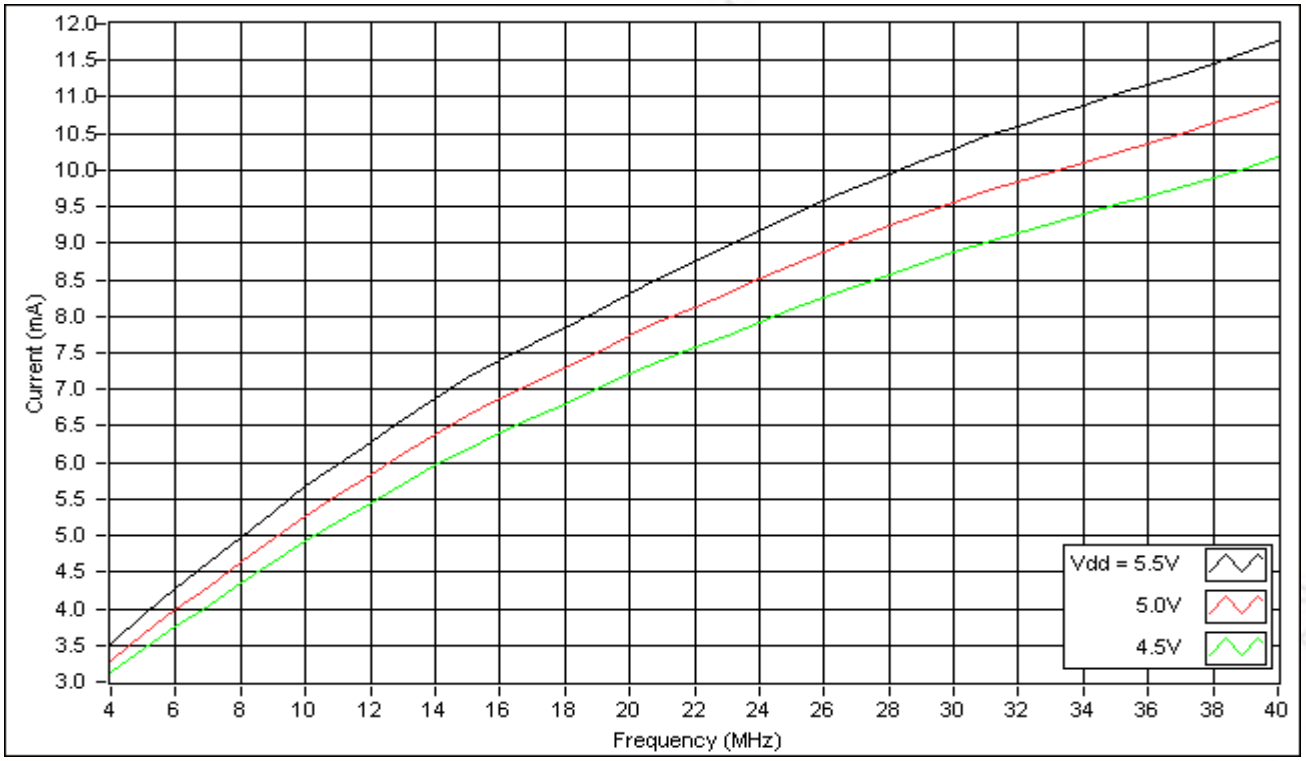


Figure 26–1. Supply Current Under 12T Mode, External Clock (1)

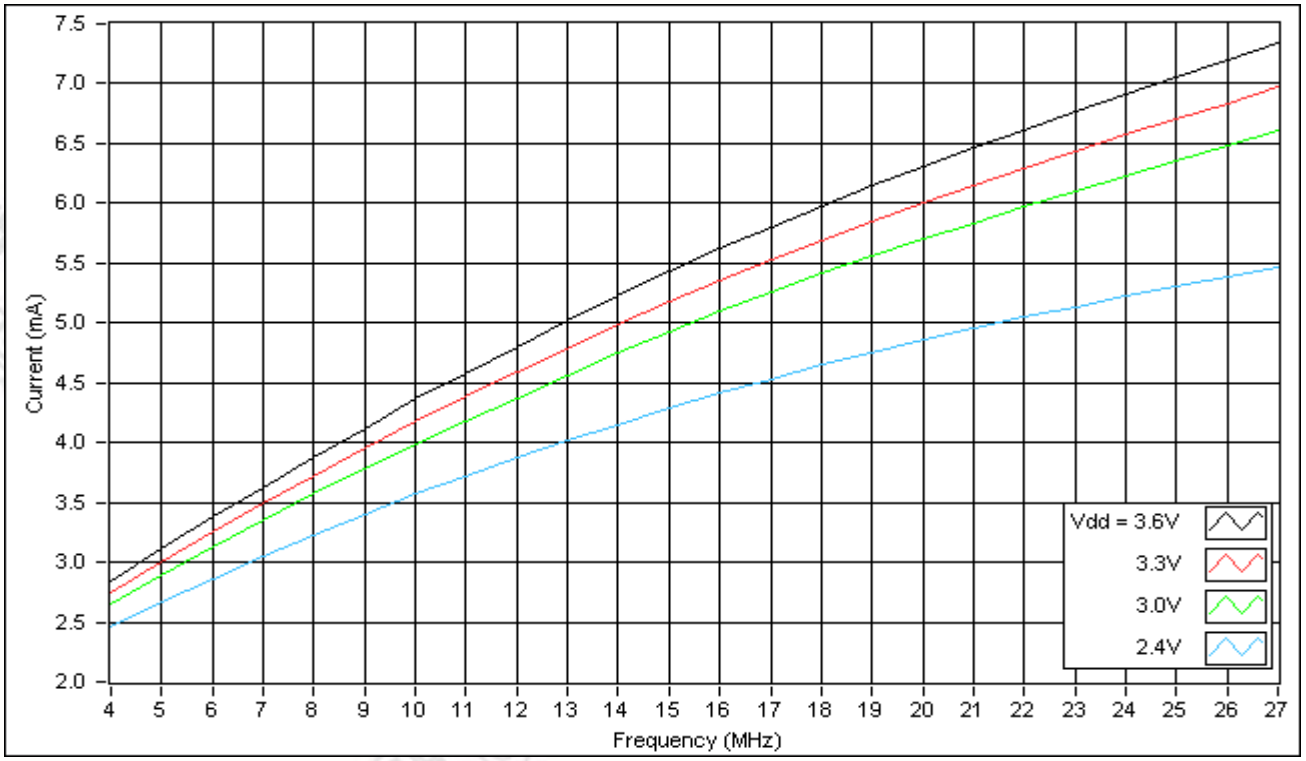


Figure 26–2. Supply Current Under 12T Mode, External Clock (2)

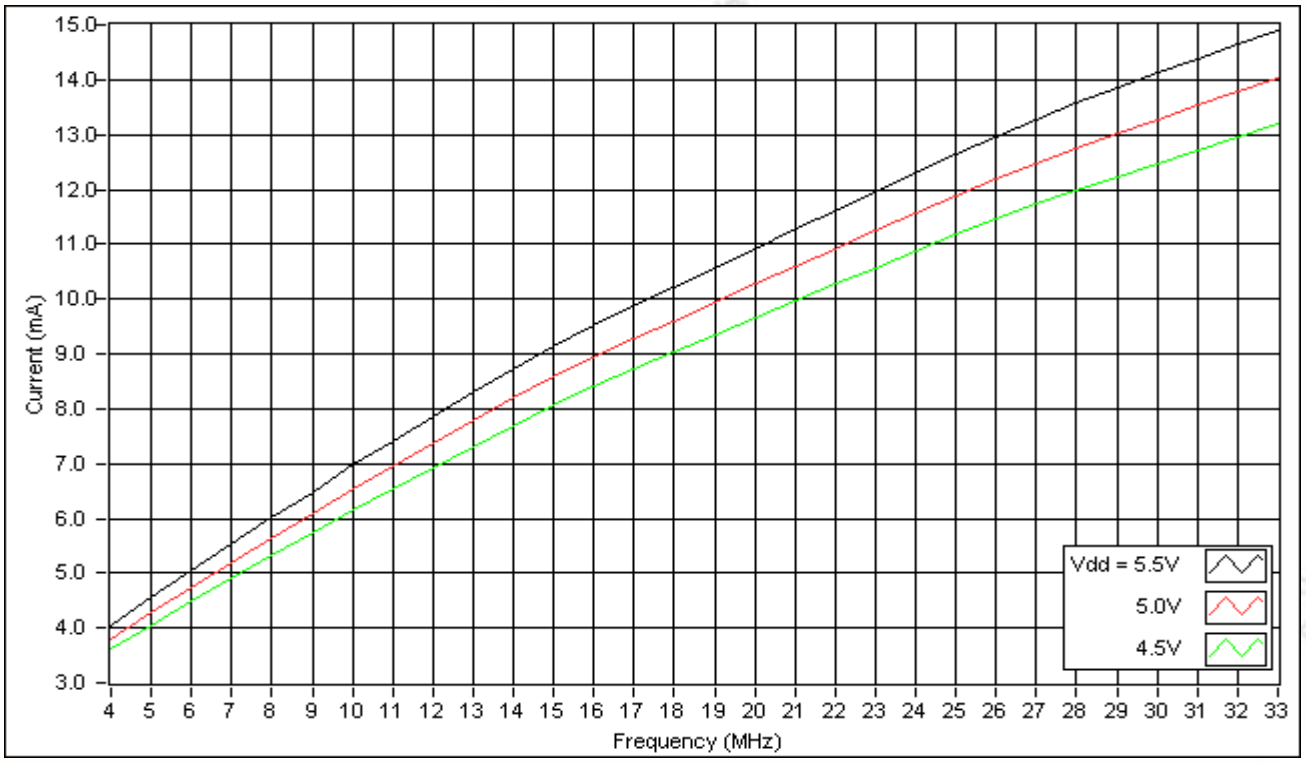


Figure 26–3. Supply Current Under 6T Mode, External Clock (1)

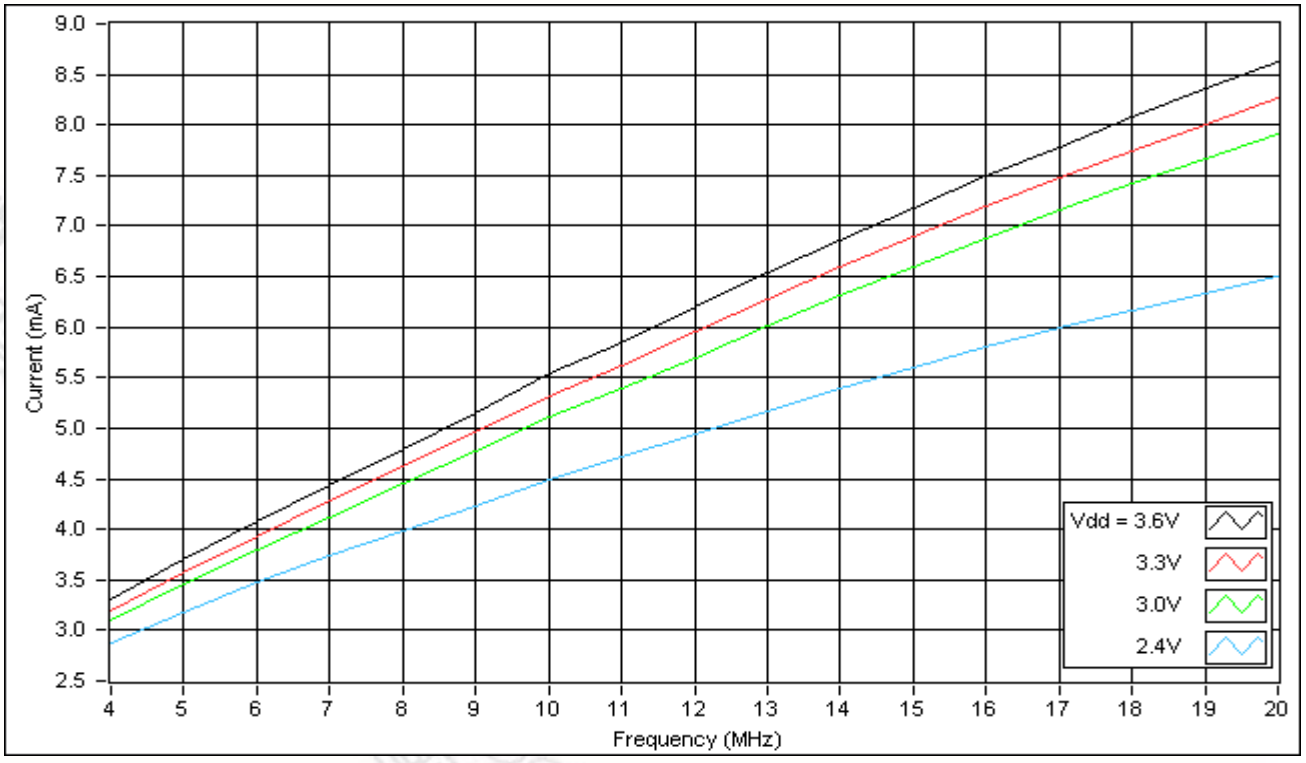


Figure 26–4. Supply Current Under 6T Mode, External Clock (2)

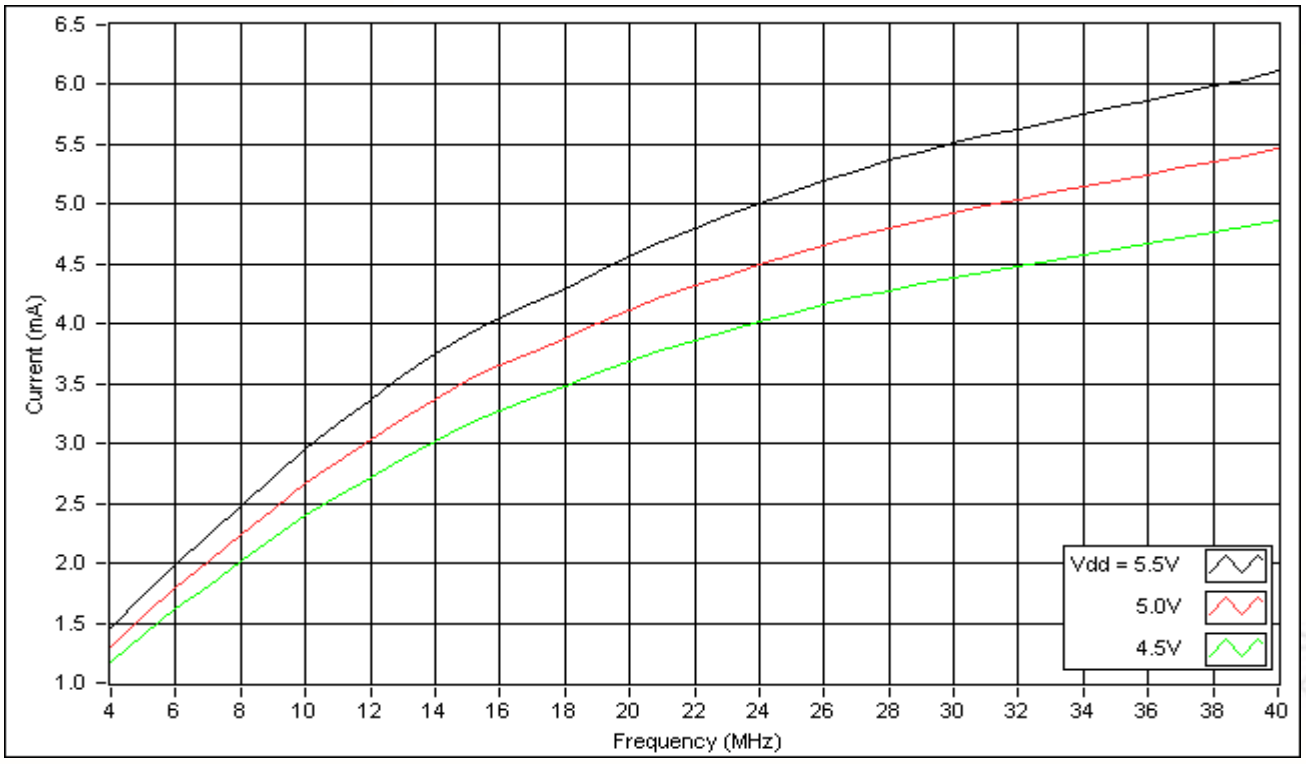


Figure 26–5. Idle Mode Current Under 12T Mode, External Clock (1)

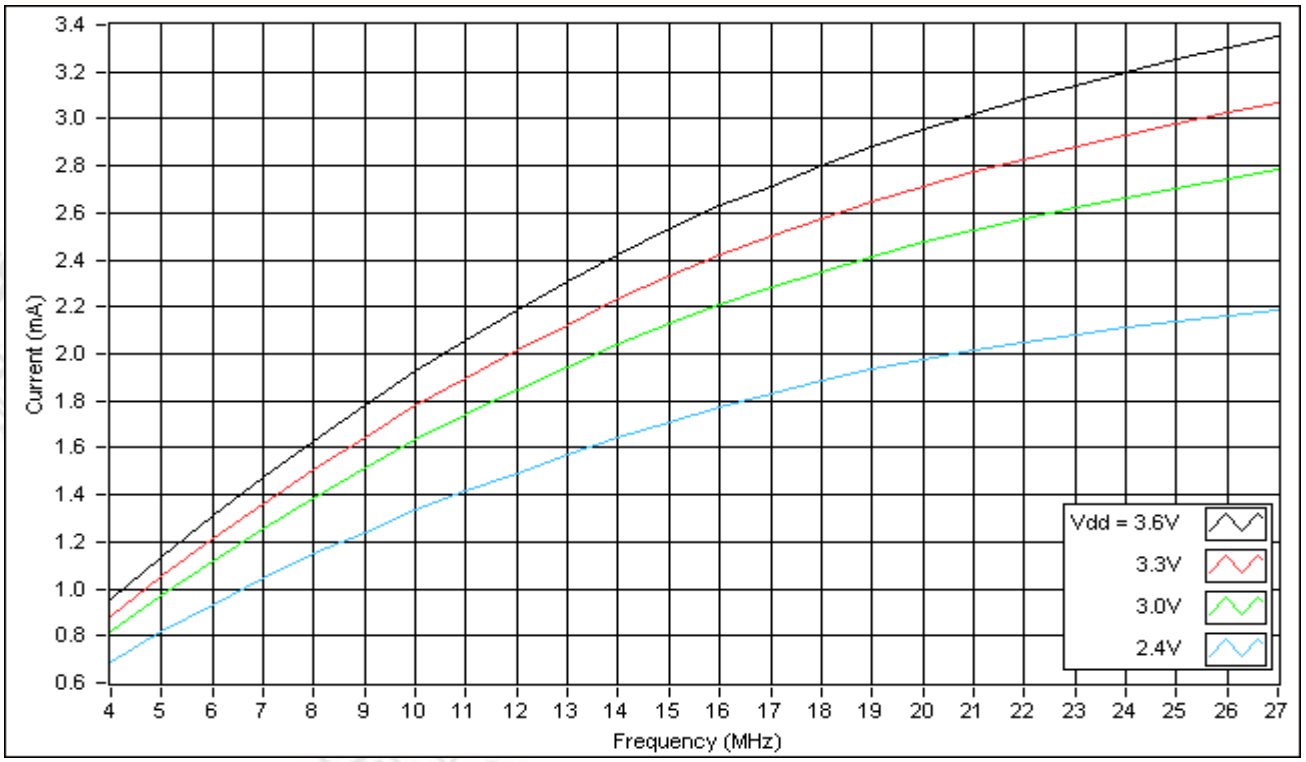


Figure 26–6. Idle Mode Current Under 12T Mode, External Clock (2)

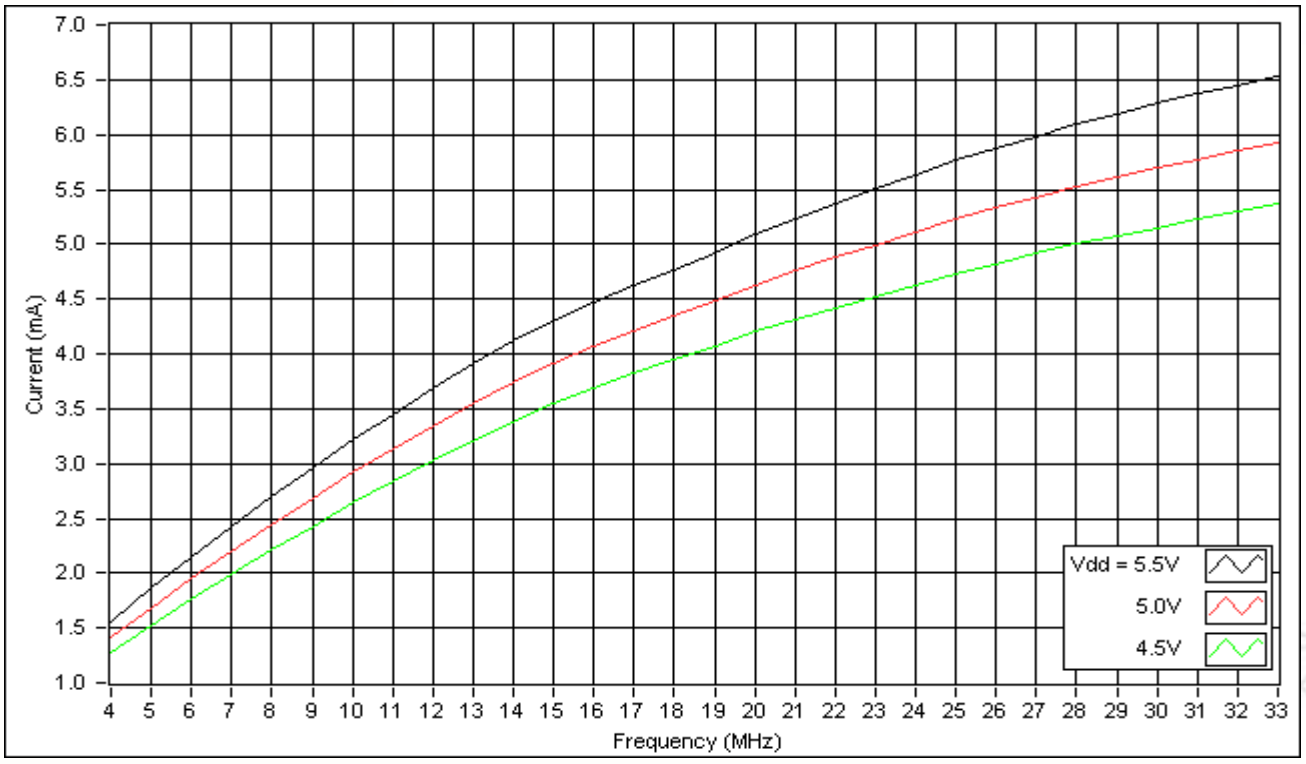


Figure 26–7. Idle Mode Current Under 6T Mode, External Clock (1)

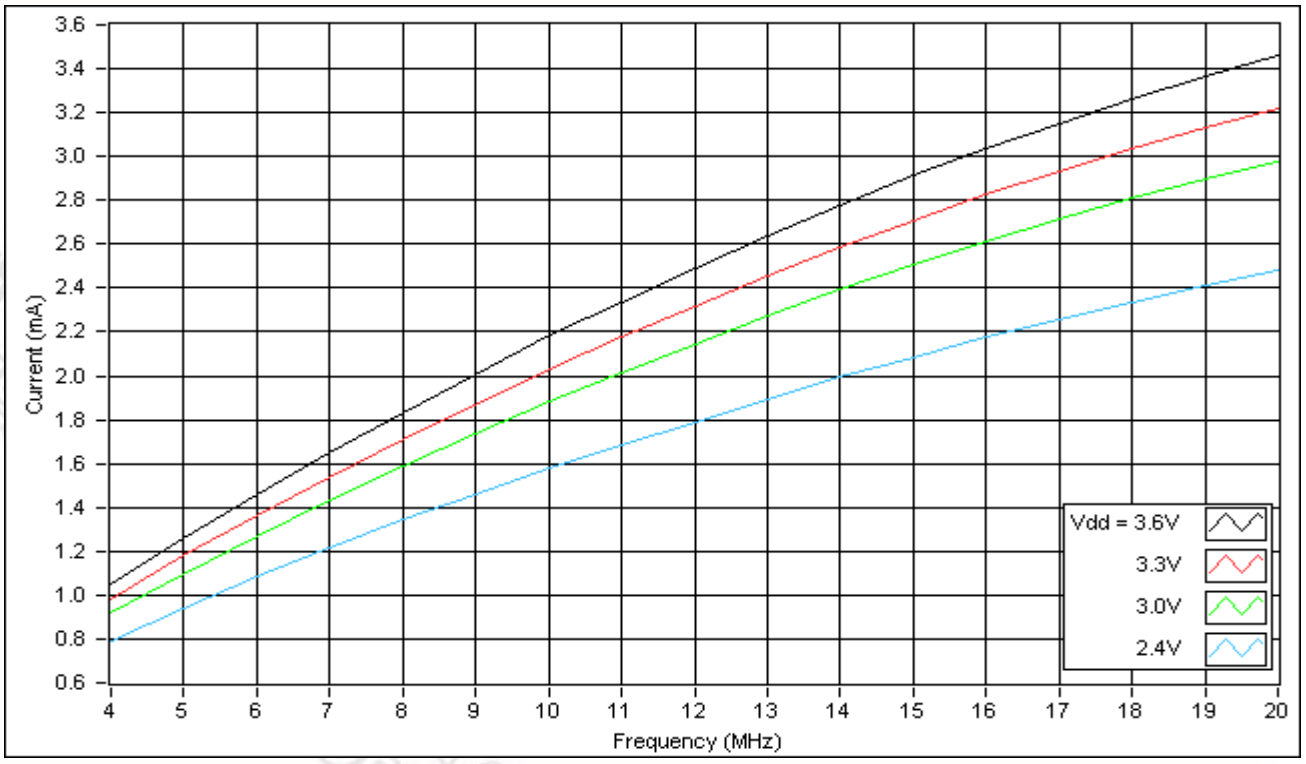


Figure 26–8. Idle Mode Current Under 6T Mode, External Clock (2)

26.3 AC Electrical Characteristics

Table 26–2. AC Characteristics

Symbol	Parameter	12T mode		6T mode		Unit
		Min.	Max.	Min.	Max.	
External Clock						
1/ t_{CLCL}	External clock input frequency	0	40	0	33	MHz
	Crystal/resonator frequency	4	40	4	33	
t_{CHCX}	High time	12		15		ns
t_{CLCX}	Low time	12		15		ns
t_{CLCH}	Rise time		8		5	ns
t_{CHCL}	Fall time		8		5	ns
Program Memory						
t_{LHLL}	ALE pulse width	2 t_{CLCL} -15		t_{CLCL} -15		ns
t_{AVLL}	Address valid to ALE low	t_{CLCL} -15		0.5 t_{CLCL} -15		ns
t_{LLAX}	Address hold after ALE low	t_{CLCL} -15		0.5 t_{CLCL} -15		ns
t_{LLIV}	ALE low to valid instruction in		4 t_{CLCL} -45		2 t_{CLCL} -45	ns
t_{LLPL}	ALE low to $\overline{PSEN}$ low	t_{CLCL} -15		0.5 t_{CLCL} -15		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	3 t_{CLCL} -15		1.5 t_{CLCL} -15		ns
t_{PLIV}	$\overline{PSEN}$ low to valid instruction in		3 t_{CLCL} -50		1.5 t_{CLCL} -50	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input instruction float after $\overline{PSEN}$		t_{CLCL} -15		0.5 t_{CLCL} -15	ns
t_{AVIV}	Address to valid instruction in		5 t_{CLCL} -60		2.5 t_{CLCL} -60	ns
t_{PLAZ}	$\overline{PSEN}$ low to address float		10		10	ns
Data Memory						
t_{RLRH}	$\overline{RD}$ pulse width	6 t_{CLCL} -30		3 t_{CLCL} -30		ns
t_{WLWH}	$\overline{WR}$ pulse width	6 t_{CLCL} -30		3 t_{CLCL} -30		ns
t_{RLDV}	$\overline{RD}$ low to valid data in		5 t_{CLCL} -50		2.5 t_{CLCL} -50	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		2 t_{CLCL} -12		t_{CLCL} -12	ns
t_{LLDV}	ALE low to valid data in		8 t_{CLCL} -50		4 t_{CLCL} -50	ns
t_{AVDV}	Address to valid data in		9 t_{CLCL} -75		4.5 t_{CLCL} -75	ns
t_{LLWL}	ALE low to $\overline{RD}$ or $\overline{WR}$ low	3 t_{CLCL} -15	3 t_{CLCL} +15	1.5 t_{CLCL} -15	1.5 t_{CLCL} +15	ns

Symbol	Parameter	12T mode		6T mode		Unit
		Min.	Max.	Min.	Max.	
t_{AVWL}	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	4 t_{CLCL} -30		2 t_{CLCL} -30		ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	t_{CLCL} -20		0.5 t_{CLCL} -20		ns
t_{WHQX}	Data hold after $\overline{WR}$	t_{CLCL} -15		0.5 t_{CLCL} -15		ns
t_{RLAZ}	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ high to ALE high	t_{CLCL} -15	t_{CLCL} +15	0.5 t_{CLCL} -15	0.5 t_{CLCL} +15	ns

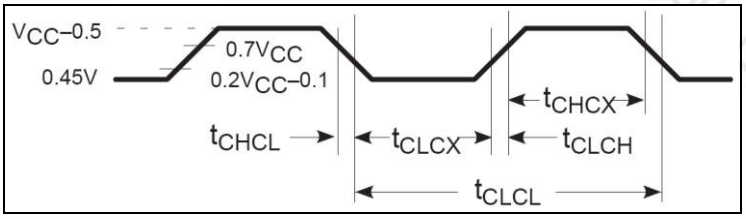


Figure 26–9. External Clock Input Timing

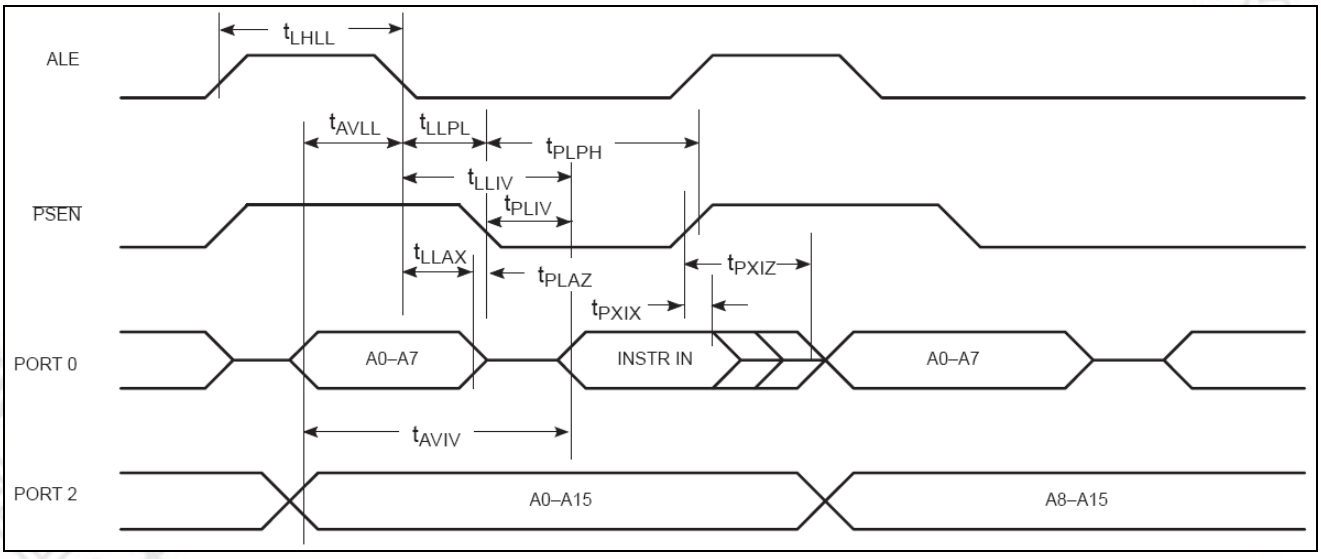


Figure 26–10. External Program Memory Read Cycle

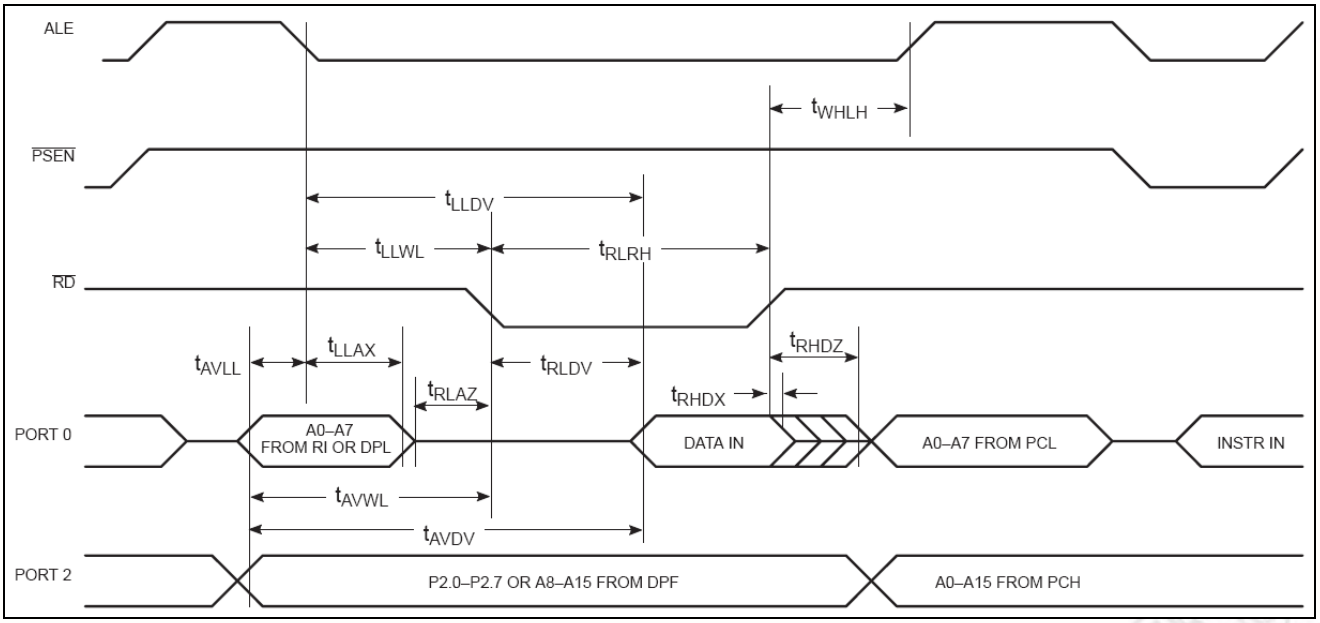


Figure 26–11. External Data Memory Read Cycle

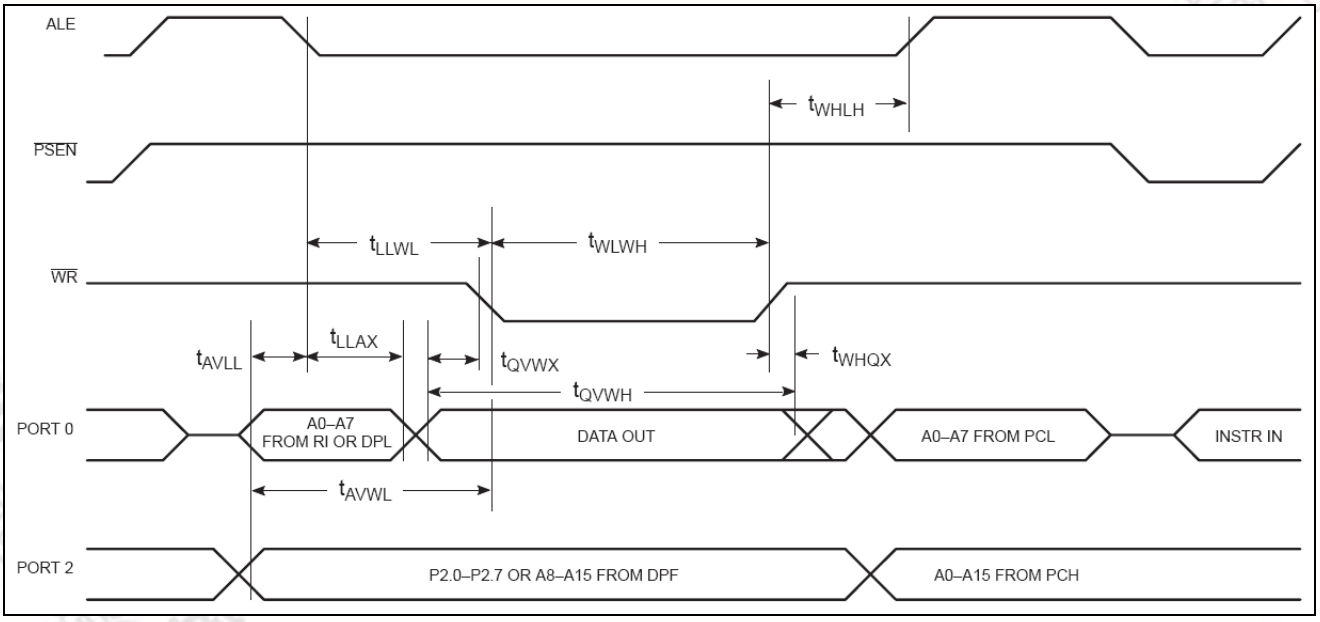


Figure 26–12. External Data Memory Write Cycle

Table 26–3. Characteristics of On-chip RC Oscillators

Symbol	Parameter	Condition	Frequency Deviation	Min.	Typ.	Max.	Unit
F_{IHRC}	System 22.1184MHz RC oscillator frequency[1][2]	T_A = 25℃	1%	21.8972	22.1184	22.3396	MHz
		T_A = -40℃ to +85℃	3%	21.4548	22.1184	22.7820	MHz
F_{ILRC}	WDT and PDT 10kHz RC oscillator frequency		30%	7	10	13	kHz

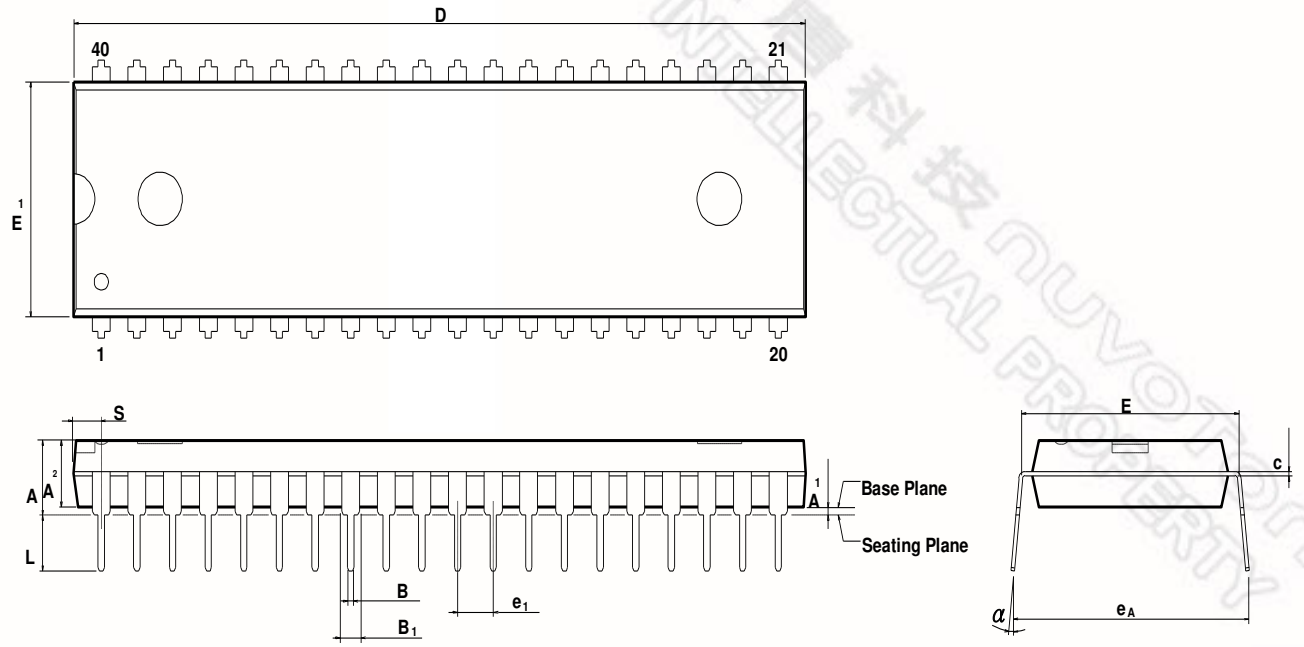
[1] Internal 11.0592MHz is not listed for the same frequency deviation due to directly divided by 2 from 22.1184MHz source.

[2] A 0.1μF capacitor is recommended to be added on XTAL1 pin to gain the more precise frequency of the internal RC oscillator frequency if it is selected as the system clock source.

Table 26–4. Characteristics of Brown-out Detection

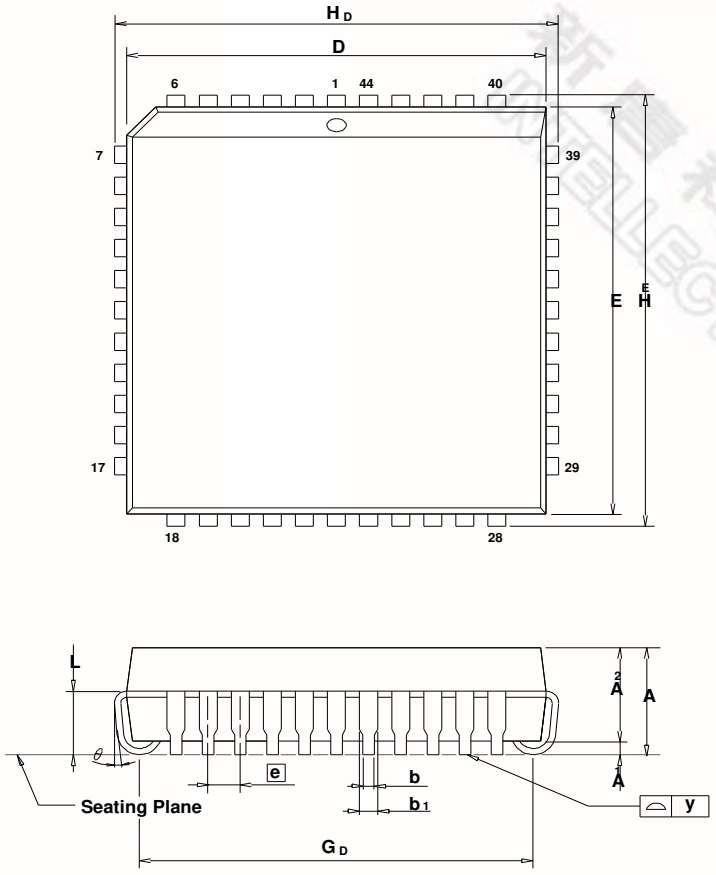
Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
T_{BOD}	Brown-out detect pulse width	$V_{DD} < V_{BOD}$	600	-	-	μs
T_{BODRD}	Brown-out release delay period	$V_{DD} > V_{BOD}$	5.6	8	10.4	ms

27. PACKAGES



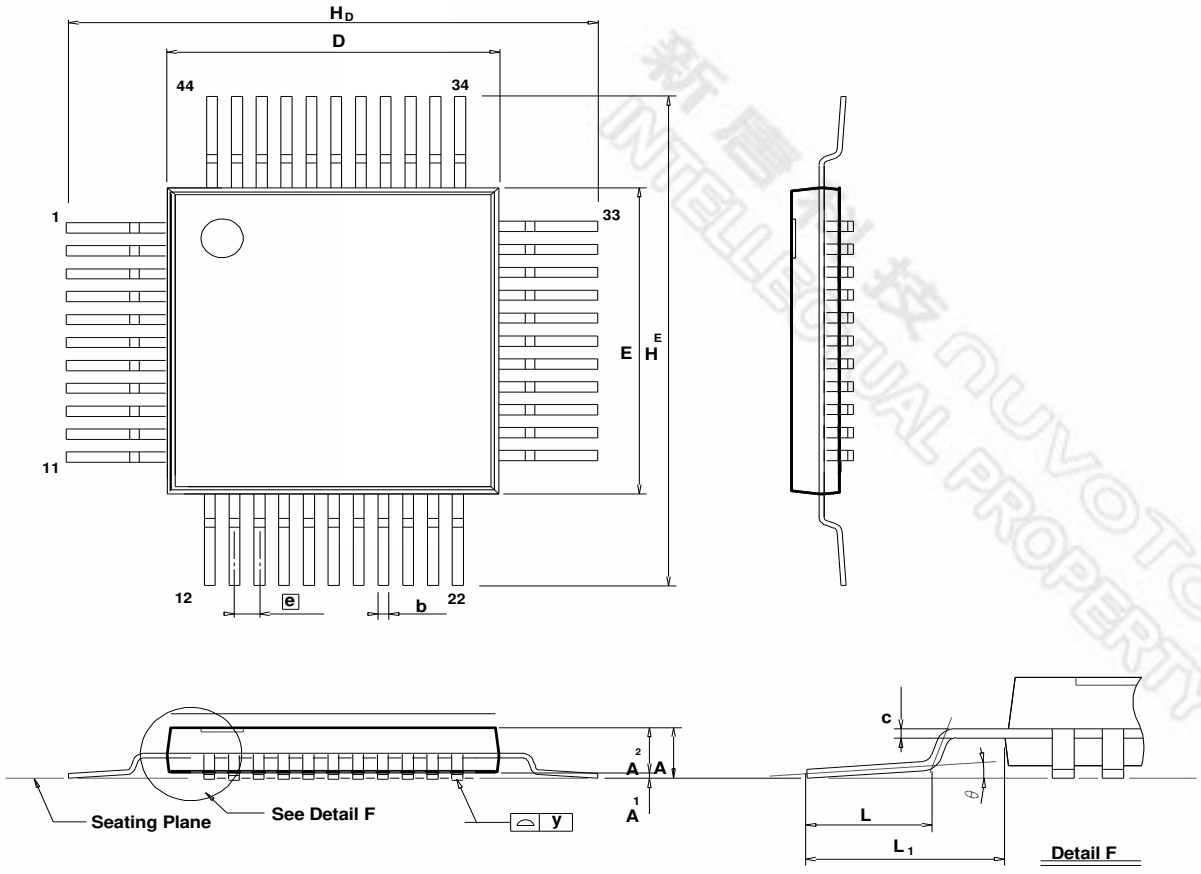
Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.210	—	—	5.33
A_1	0.010	—	—	0.25	—	—
A_2	0.150	0.155	0.160	3.81	3.94	4.06
B	0.016	0.018	0.022	0.41	0.46	0.56
B_1	0.048	0.050	0.054	1.22	1.27	1.37
c	0.008	0.010	0.014	0.20	0.25	0.36
D	—	2.055	2.070	—	52.20	52.58
E	0.590	0.600	0.610	14.99	15.24	15.49
E_1	0.540	0.545	0.550	13.72	13.84	13.97
e_1	0.090	0.100	0.110	2.29	2.54	2.79
L	0.120	0.130	0.140	3.05	3.30	3.56
α	0	—	15	0	—	15
e_A	0.630	0.650	0.670	16.00	16.51	17.02
S	—	—	0.090	—	—	2.29

Figure 27–1. DIP-40 Package Dimention



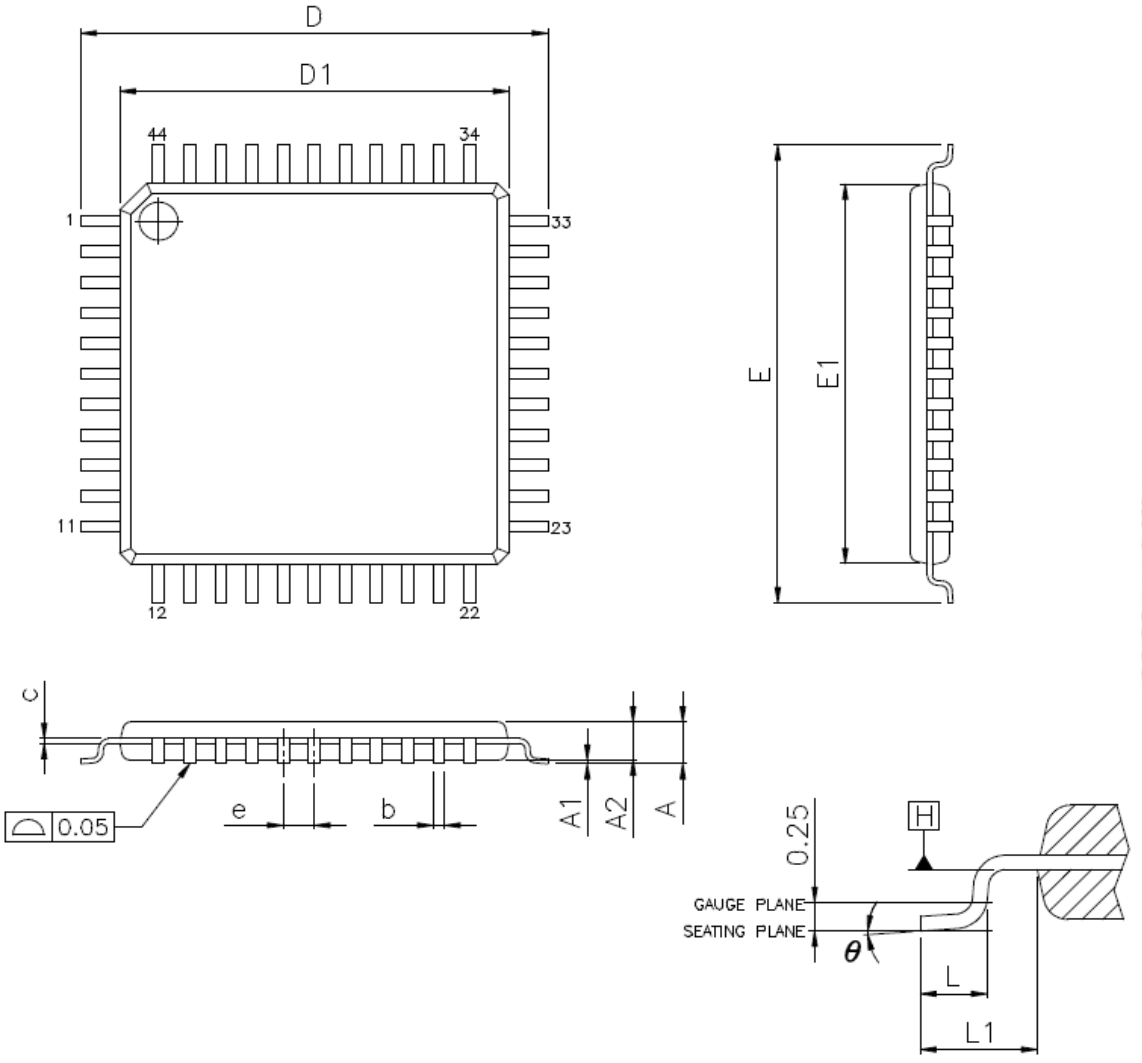
Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.185	—	—	4.70
A_1	0.020	—	—	0.51	—	—
A_2	0.145	0.150	0.155	3.68	3.81	3.94
b_1	0.026	0.028	0.032	0.66	0.71	0.81
b	0.016	0.018	0.022	0.41	0.46	0.56
c	0.008	0.010	0.014	0.20	0.25	0.36
D	0.648	0.653	0.658	16.46	16.59	16.71
E	0.648	0.653	0.658	16.46	16.59	16.71
e	0.050 BSC			1.27 BSC		
G_D	0.590	0.610	0.630	14.99	15.49	16.00
G_E	0.590	0.610	0.630	14.99	15.49	16.00
H_D	0.680	0.690	0.700	17.27	17.53	17.78
H_E	0.680	0.690	0.700	17.27	17.53	17.78
L	0.090	0.100	0.110	2.29	2.54	2.79
y	—	—	0.004	—	—	0.10

Figure 27–2. PLCC-44 Package Dimention



Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	-	—	—	-
A_1	0.002	0.01	0.02	0.05	0.25	0.5
A_2	0.075	0.081	0.087	1.90	2.05	2.20
b	0.01	0.014	0.018	0.25	0.35	0.45
c	0.004	0.006	0.010	0.10	0.15	0.25
D	0.390	0.394	0.398	9.9	10.00	10.1
E	0.390	0.394	0.398	9.9	10.00	10.1
e	—	.0315	—	—	0.80	—
H_D	0.510	0.520	0.530	12.95	13.20	13.45
H_E	0.510	0.520	0.530	12.95	13.20	13.45
L	0.025	0.031	0.037	0.65	0.8	0.95
L_1	—	0.063	—	—	1.60	—
y	—	—	0.004	—	—	0.10
θ	0°	—	10°	0°	—	10°

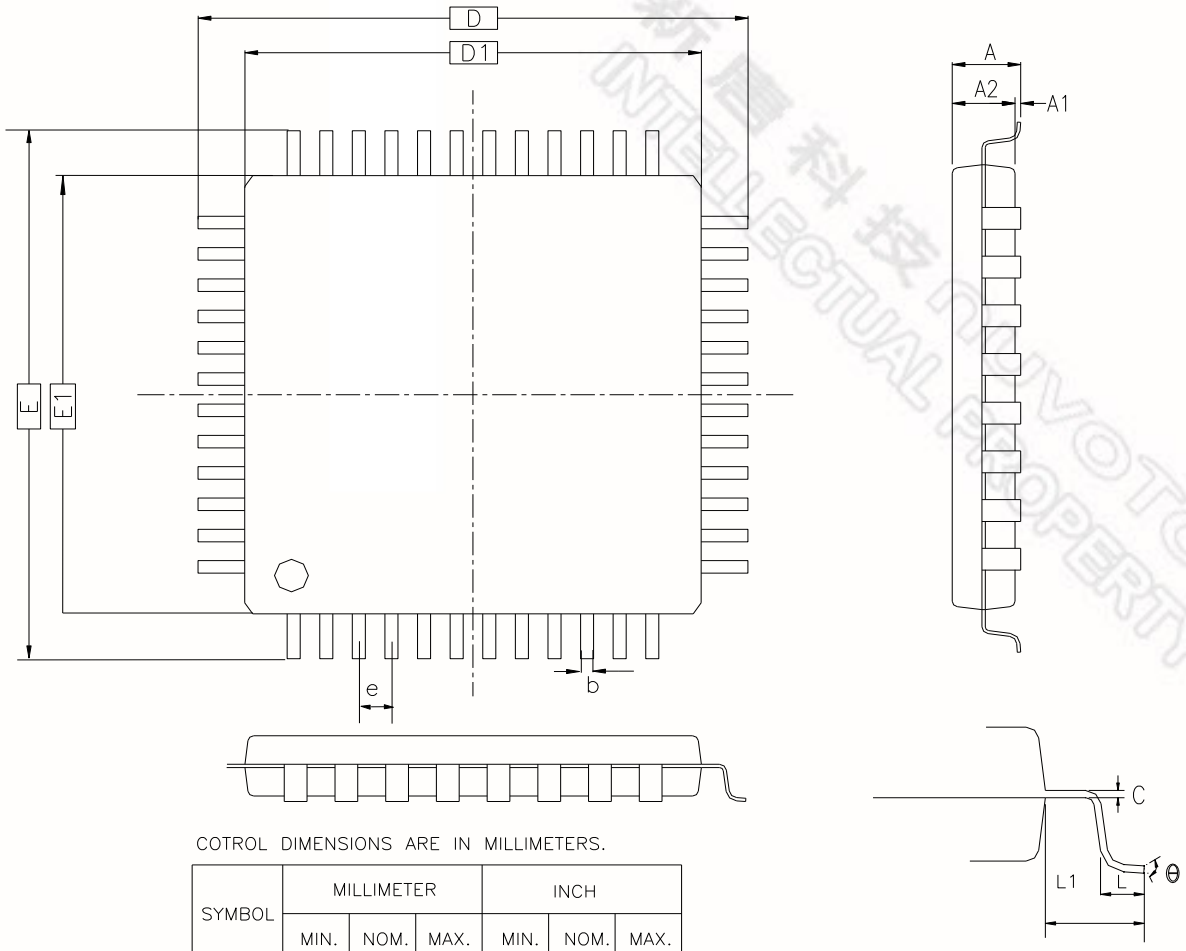
Figure 27–3. PQFP-44 Package Dimention



VARIATIONS (ALL DIMENSUIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	1.00	1.05
D	11.75	12.00	12.25
D1	9.90	10.00	10.10
E	11.75	12.00	12.25
E1	9.90	10.00	10.10
b	0.30	0.37	0.45
c	0.09	-	0.20
L	0.45	0.60	0.75
L1	1.00 REF		
e	0.80 BSC		
θ	0°	3.5°	7°

Figure 27–4. TQFP-44 Package Dimention



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	0.10	0.15	0.002	0.004	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D1	6.90	7.00	7.10	0.272	0.276	0.260
E1	6.90	7.00	7.10	0.272	0.276	0.260
e	0.35	0.50	0.65	0.014	0.020	0.260
D	8.9	9.00	9.10	0.350	0.354	0.358
E	8.9	9.00	9.10	0.350	0.354	0.358
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	—	1.00	—	—	0.039	—
C	0.09	—	0.20	0.0035	—	0.0079
θ	0°	—	7°	0°	—	7°
b	0.17	0.22	0.27	0.007	0.0087	0.011

Figure 27–5. LQFP-48 Package Dimention

28. DOCUMENT REVISION HISTORY

Version	Date	Page	Description
V1.0	2010/8/13		Initial release.
V1.1	2010/9/20	126	Increase the maximum value of Power Down mode current.
V1.2	2010/12/1	81 96 88	1. Add restriction of disabling interrupts during TA protected writing. 2. Add restriction of disabling interrupts during ISP. 3. Add more description of software clearing interrupt flags.
V2.0	2011/3/11	30	Change XRAM default state “enabled” after all resets to fit general applications.
V2.1	2011/9/9		1. TQFP-44 package available. 2. Correct some typo errors.
V2.2	2012/9/4	138	Correct TQFP-44 package dimension.

More product details and update information, please visit our website:
www.nuvoton.com

Headquarter-Taiwan

Nuvoton Technology Corp.
No. 4, Creation Rd. III, Hsinchu Science Park,
300 Taiwan
TEL: 886-3-5770066

Worldwide Sales Offices

Taipei Sales Office
8F, No. 480, Rueiguang Rd., Neihu Chiu, Taipei,
114 Taiwan
TEL: 886-2-26588066

Nuvoton Technology Corp. America
2727 N. First Street, San Jose, CA 95134, U.S.A.
TEL: 1-408-544-1718

Nuvoton Electronics Tech. (Shenzhen) Limited
Unit 1501, New World Center, 6009 Yitian Road,
Futian, Shenzhen, P.R.China 518026
TEL: 86-755-83515350

Nuvoton Electronics Tech. (H.K.) Limited
Unit 9-11, 22F, Millennium City *2,* 378 Kwun Tong Road,
Kowloon, Hong Kong
TEL: 852-27513100

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.