Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n78e055afg

size. The APROM on N78E059A/N78E055A is 32k/16k-byte size. 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.

N78E059A/N78E055A 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 91. 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.

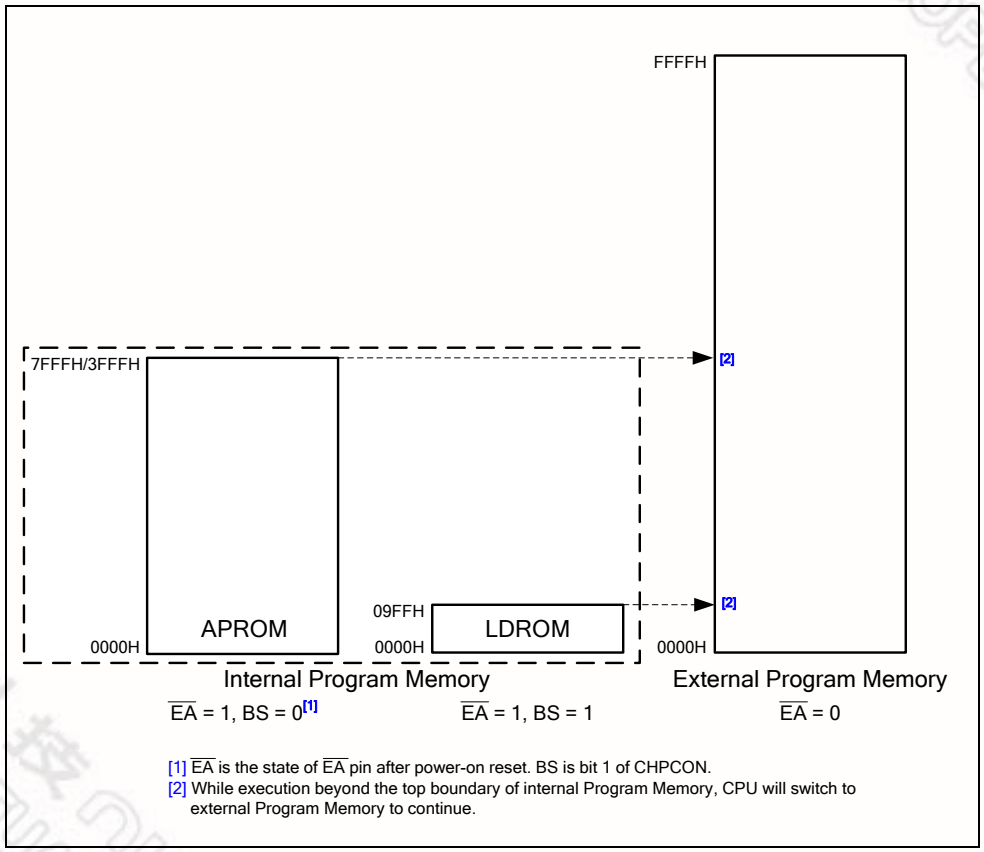


Figure 5–1. N78E059A/N78E055A Program Memory Structure

5.4 On-chip XRAM

N78E059A/N78E055A 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 29.

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. This will facilitate 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.

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.

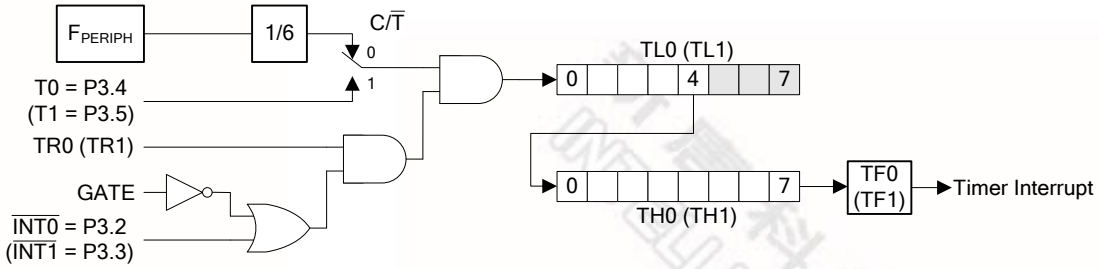


Figure 10–1. Timer/Counters 0 and 1 in Mode 0

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.

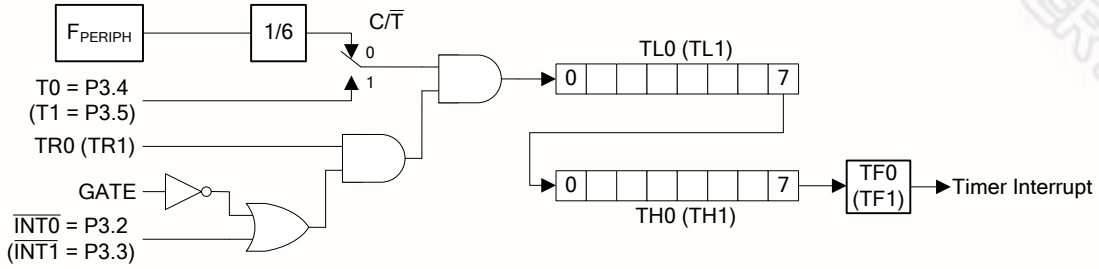


Figure 10–2. Timer/Counters 0 and 1 in Mode 1

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.

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.

Bit	Name	Description
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.
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. N78E059A/N78E055A SFR Descriptions 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.

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.

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.

15. PULSE WIDTH MODULATOR (PWM)

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

PWMCON0 – PWM Control 0

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

Address: DCH reset value: 0000 0000b

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

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.

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 N78E059A/N78E055A, the TA protected SFRs includes CHPCON (9FH), ISPTRG (A4H), PMC (ACH), and WDCON (AAH).

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. N78E059A/N78E055A (on PLCC-44, PQFP-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).

RETI instruction, but it would not inform the Interrupt controller that the interrupt service routine is completed. 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 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.

```
        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
```

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.**

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. N78E059A/N78E055A SFR Descriptions 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

N78E059A/N78E055A 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-

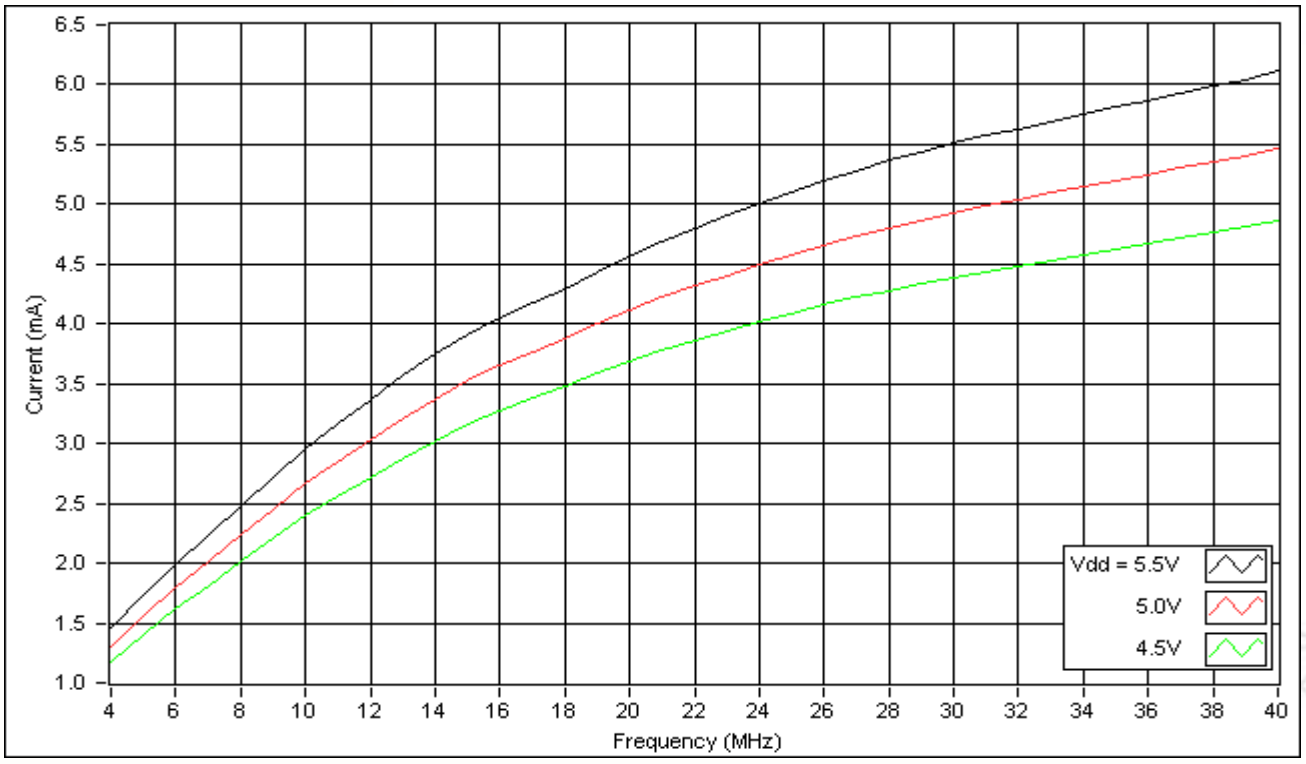


Figure 26–5. Idle Mode Current Under 12T Mode, External Clock (1)

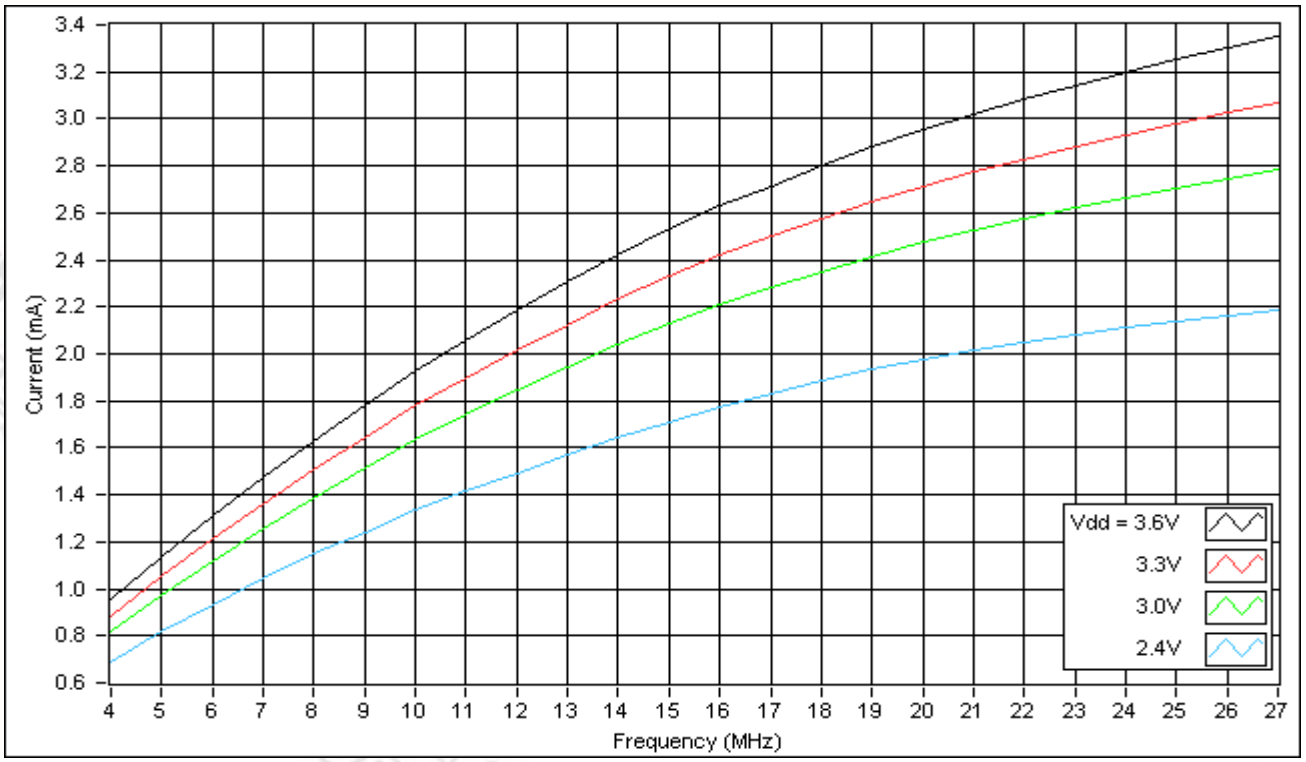


Figure 26–6. Idle Mode Current Under 12T Mode, External Clock (2)

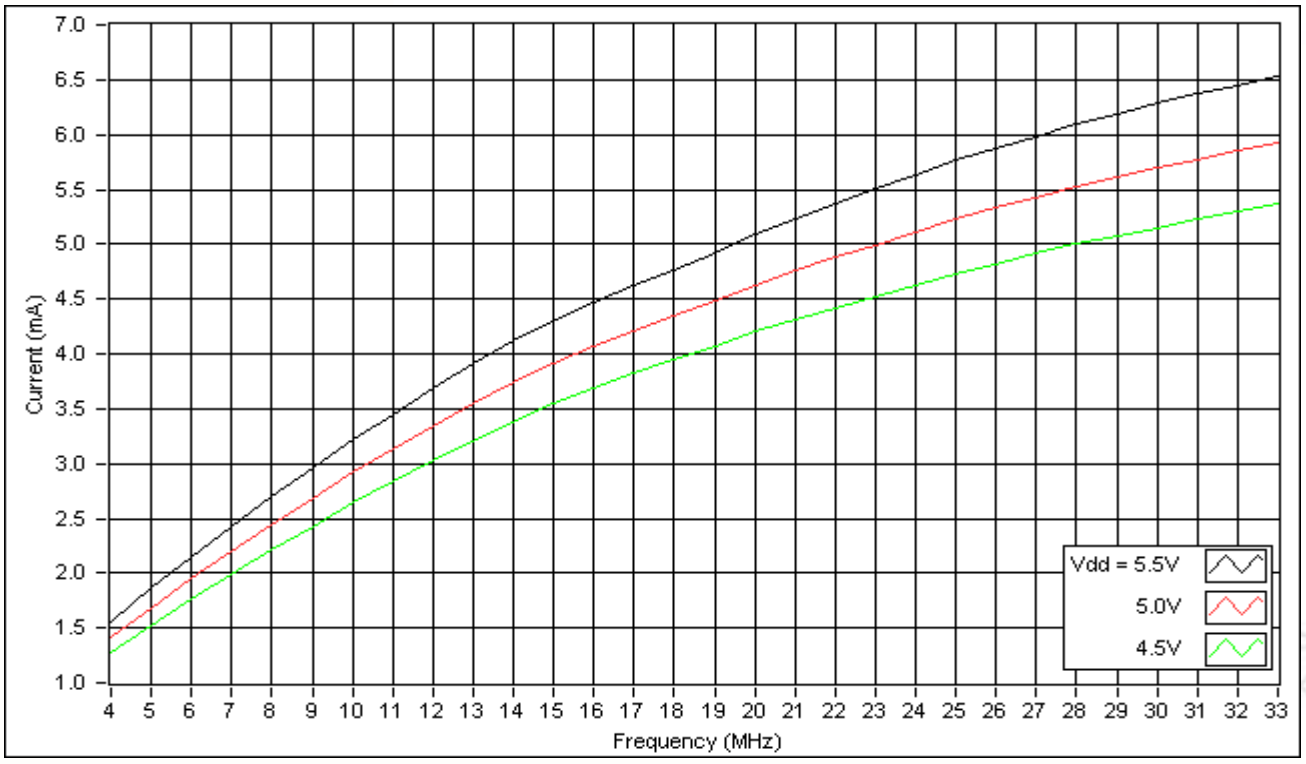


Figure 26–7. Idle Mode Current Under 6T Mode, External Clock (1)

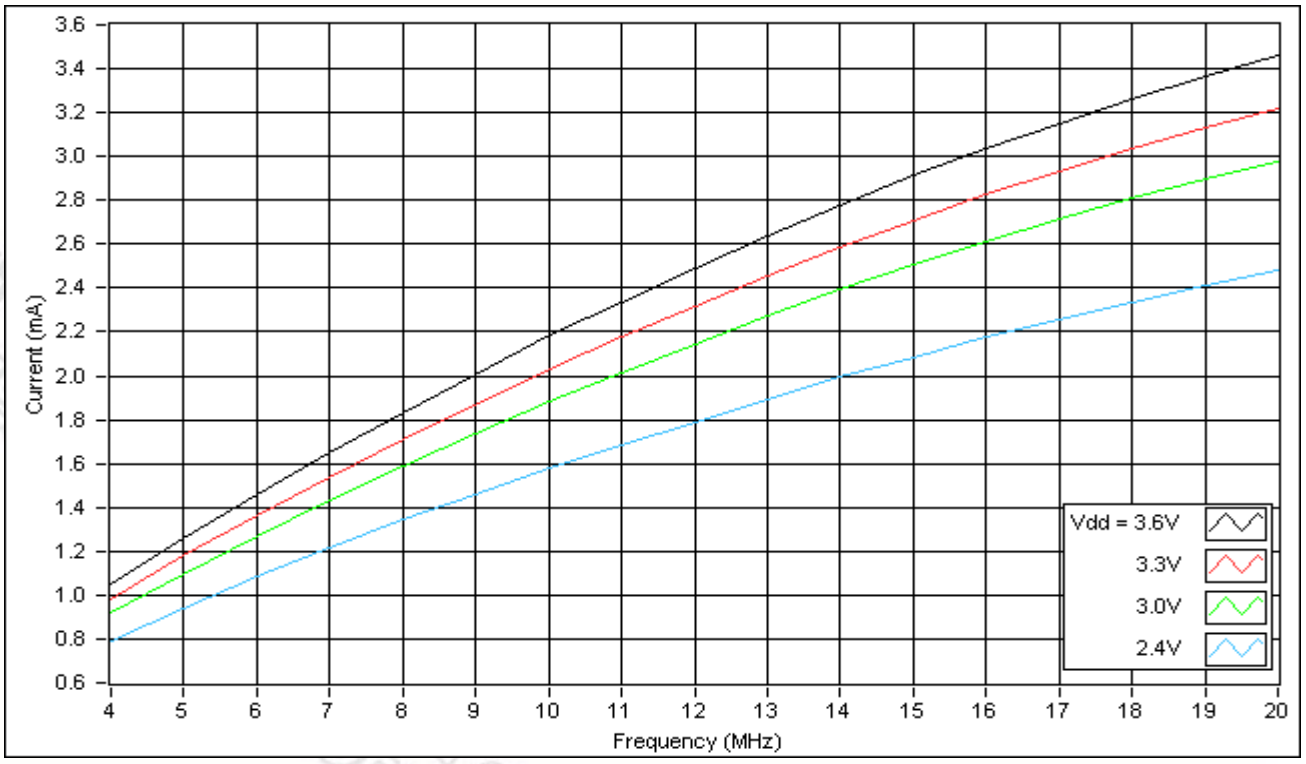


Figure 26–8. Idle Mode Current Under 6T Mode, External Clock (2)

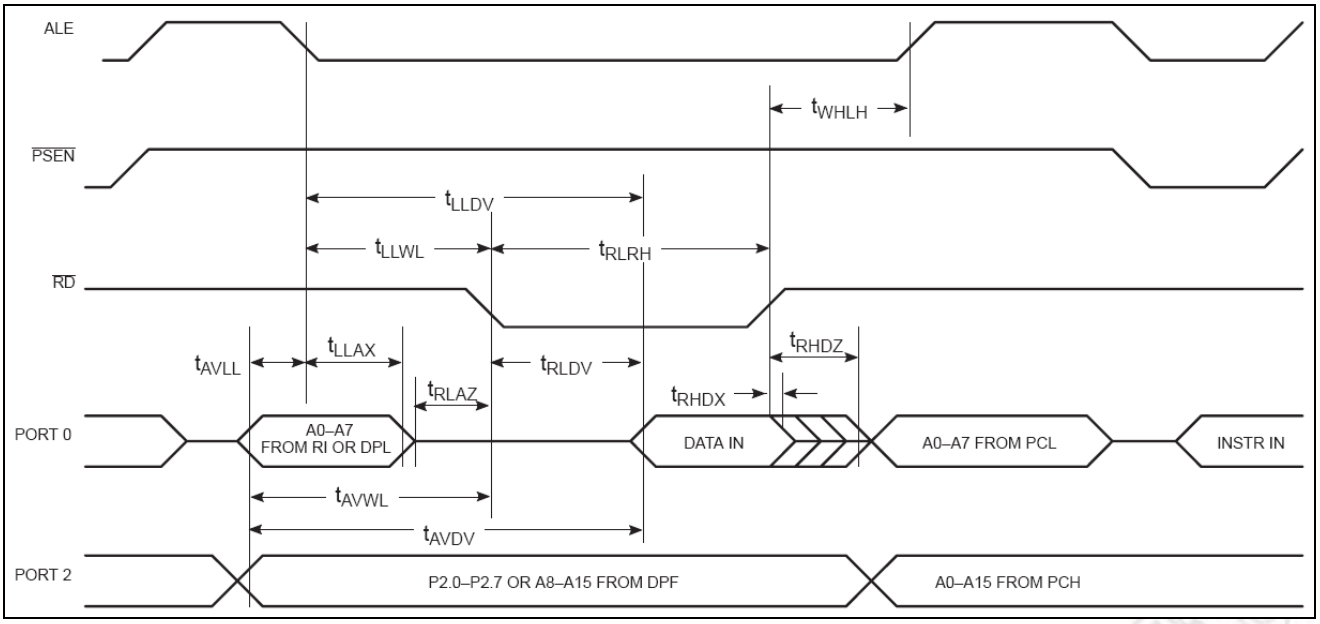


Figure 26–11. External Data Memory Read Cycle

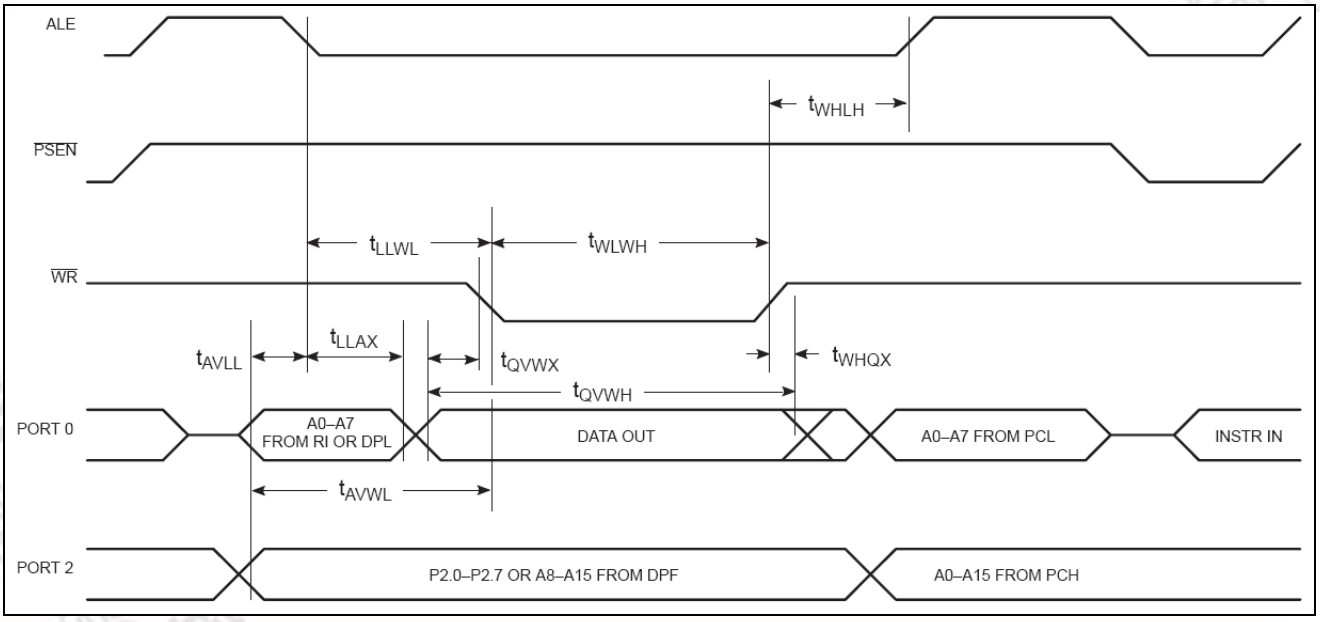
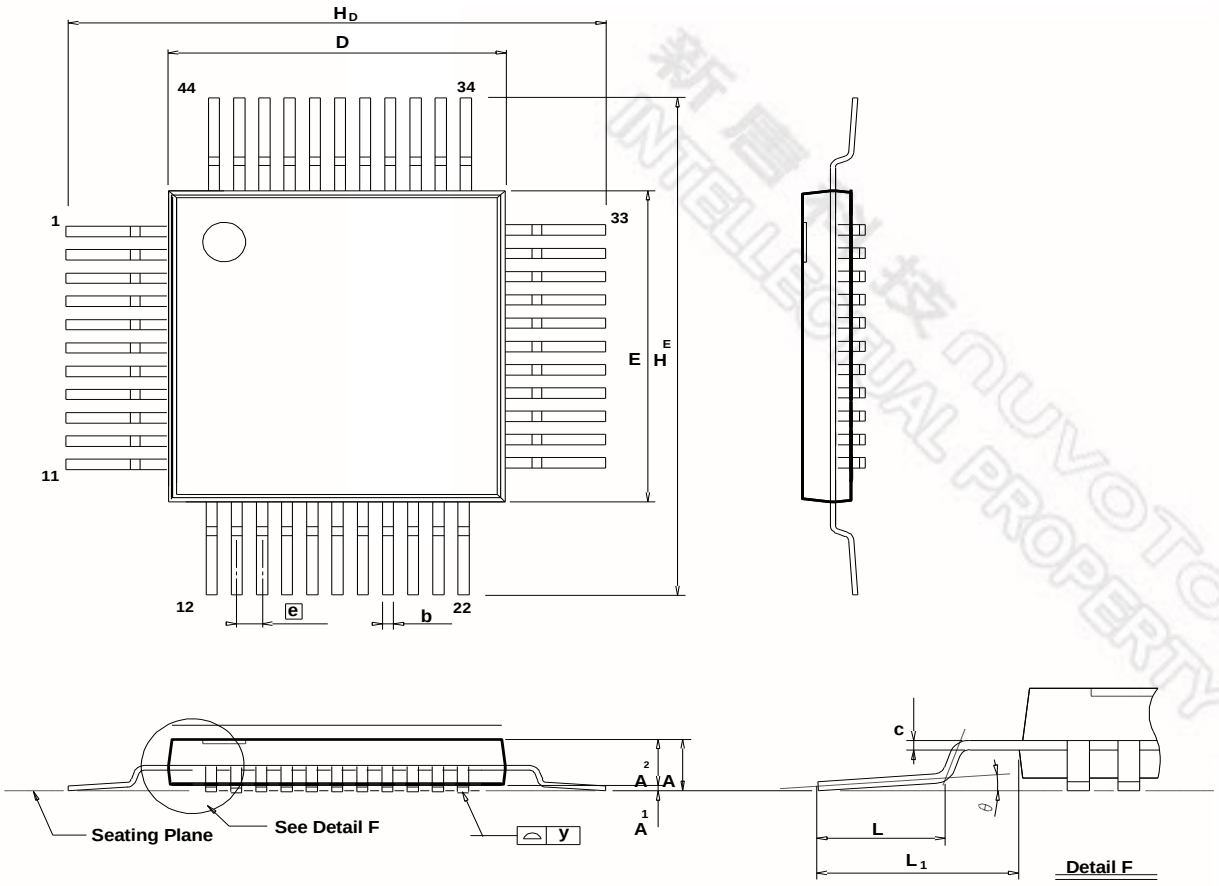


Figure 26–12. External Data Memory Write Cycle



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