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	Obsolete
Core Processor	8032
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	46
Program Memory Size	80KB (80K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3212cv-24u6

2.8.3 Register addressing

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of four banks is selected at execution time by the two bank select bits in the PSW.

Example:

```
mov PSW, #0001000B ; select Bank0
mov A, #30H
mov R1, A
```

2.8.4 Register-specific addressing

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point it. The opcode itself does that.

2.8.5 Immediate constants addressing

The value of a constant can follow the opcode in Program memory.

Example:

```
mov A, #10H.
```

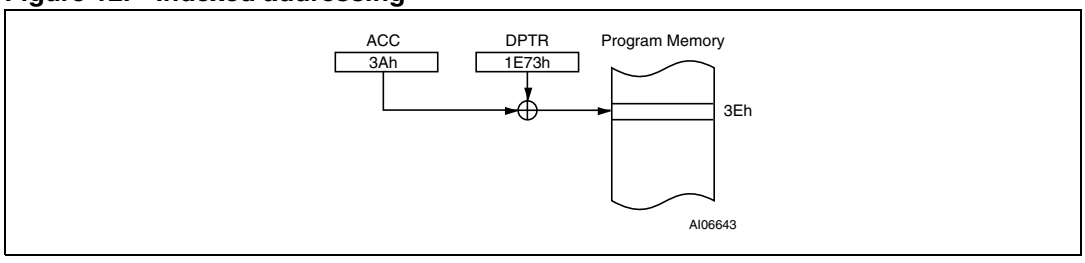
2.8.6 Indexed addressing

Only Program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program memory is formed by adding the Accumulator data to the base pointer.

Example:

```
movc A, @A+DPTR
```

Figure 12. Indexed addressing



The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a “Destination out of range” message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically. DPTR is set up with the address of a jump table. In a 5-way branch, for ex-ample, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV DPTR,#JUMP TABLE
MOV A,INDEX_NUMBER
RL A
JMP @A+DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP TABLE:
AJMP CASE 0
AJMP CASE 1
AJMP CASE 2
AJMP CASE 3
AJMP CASE 4
```

Table 13 shows a single “CALL addr” instruction, but there are two of them, LCALL and ACALL, which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case, the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done.

If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 13. Unconditional Jump instructions

Mnemonic	Operation
JMP addr	Jump to addr
JMP @A+DPTR	Jump to A+DPTR
CALL addr	Call Subroutine at addr
RET	Return from subroutine
RETI	Return from interrupt
NOP	No operation

Table 14 shows the list of conditional jumps available to the UPSD321xx device user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of -128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero Bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
MOV COUNTER,#10
LOOP: (begin loop)
  •
  •
  •
(end loop)
DJNZ COUNTER, LOOP
(continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in *Table 9*. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of *Table 9* Shifting a BCD Number One Digits to the Right, the two bytes were data in R1 and the constant 2Ah. The initial data in R1 was 2Eh.

Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2Ah.

Another application of this instruction is in “greater than, less than” comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry Bit is set (1). If the first is greater than or equal to the second, then the Carry Bit is cleared.

Table 15. SFR memory map (continued)

88	TCON(1)	TMOD	TL0	TL1	TH0	TH1			8F
80	P0(1)	SP	DPL	DPH				PCON	87

1. Register can be bit addressing

Table 16. List of all SFRs

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
80	P0									FF	Port 0
81	SP									07	Stack Ptr
82	DPL									00	Data Ptr Low
83	DPH									00	Data Ptr High
87	PCON	SMOD	SMOD1	LVREN	ADSFINT	RCLK1	TCLK1	PD	IDLE	00	Power Ctrl
88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00	Timer / Cntr Control
89	TMOD	Gate	C/T	M1	M0	Gate	C/T	M1	M0	00	Timer / Cntr mode Control
8A	TL0									00	Timer 0 Low
8B	TL1									00	Timer 1 Low
8C	TH0									00	Timer 0 High
8D	TH1									00	Timer 1 High
90	P1									FF	Port 1
91	P1SFS	P1S7	P1S6	P1S5	P1S4					00	Port 1 Select Register
93	P3SFS	P3S7	P3S6							00	Port 3 Select Register
94	P4SFS	P4S7	P4S6	P4S5	P4S4	P4S3	P4S2	P4S1	P4S0	00	Port 4 Select Register
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register
98	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	Serial Control Register
99	SBUF									00	Serial Buffer
9A	SCON2	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	2nd UART Ctrl Register
9B	SBUF2									00	2nd UART Serial Buffer
A0	P2									FF	Port 2

Table 16. List of all SFRs (continued)

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A1	PWMCON	PWML	PWMP	PWME	CFG4	CFG3	CFG2	CFG1	CFG0	00	PWM Control Polarity
A2	PWM0									00	PWM0 Output Duty Cycle
A3	PWM1									00	PWM1 Output Duty Cycle
A4	PWM2									00	PWM2 Output Duty Cycle
A5	PWM3									00	PWM3 Output Duty Cycle
A6	WDRST									00	Watch Dog Reset
A7	IEA				ES2			EI^{2}C		00	Interrupt Enable (2nd)
A8	IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
A9											
AA	PWM4P									00	PWM 4 Period
AB	PWM4W									00	PWM 4 Pulse Width
AE	WDKEY									00	Watch Dog Key Register
B0	P3									FF	Port 3
B1	PSCL0L									00	Prescaler 0 Low (8-bit)
B2	PSCL0H									00	Prescaler 0 High (8-bit)
B3	PSCL1L									00	Prescaler 1 Low (8-bit)
B4	PSCL1H									00	Prescaler 1 High (8-bit)
B7	IPA				PS2			PI2C		00	Interrupt Priority (2nd)
B8	IP			PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority
C0	P4									FF	New Port 4
C8	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00	Timer 2 Control

12 Standard serial interface (UART)

The UPSD321xx devices provides two standard 8032 UART serial ports. The first port is connected to pin P3.0 (RX) and P3.1 (TX). The second port is connected to pin P1.2 (RX) and P1.3(TX). The operation of the two serial ports are the same and are controlled by the SCON and SCON2 registers.

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still has not been read by the time reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF (or SBUF2 for the second serial port). Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes: Mode 0, 1, 2 or 3.

Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 the f_{OSC}.

Mode 1

10 bits are transmitted (through TxD) or received (through RxD): a Start bit (0), 8 data bits (LSB first), and a Stop bit (1). On receive, the Stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2

11 bits are transmitted (through TxD) or received (through RxD): Start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a Stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of ‘0’ or ‘1’. Or, for example, the Parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the Stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3

11 bits are transmitted (through TxD) or received (through RxD): a Start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a Stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

12.1 Multiprocessor communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a Stop bit. The port can be programmed such that when the Stop bit is received, the serial port interrupt will

Figure 35. Four-channel 8-bit PWM block diagram

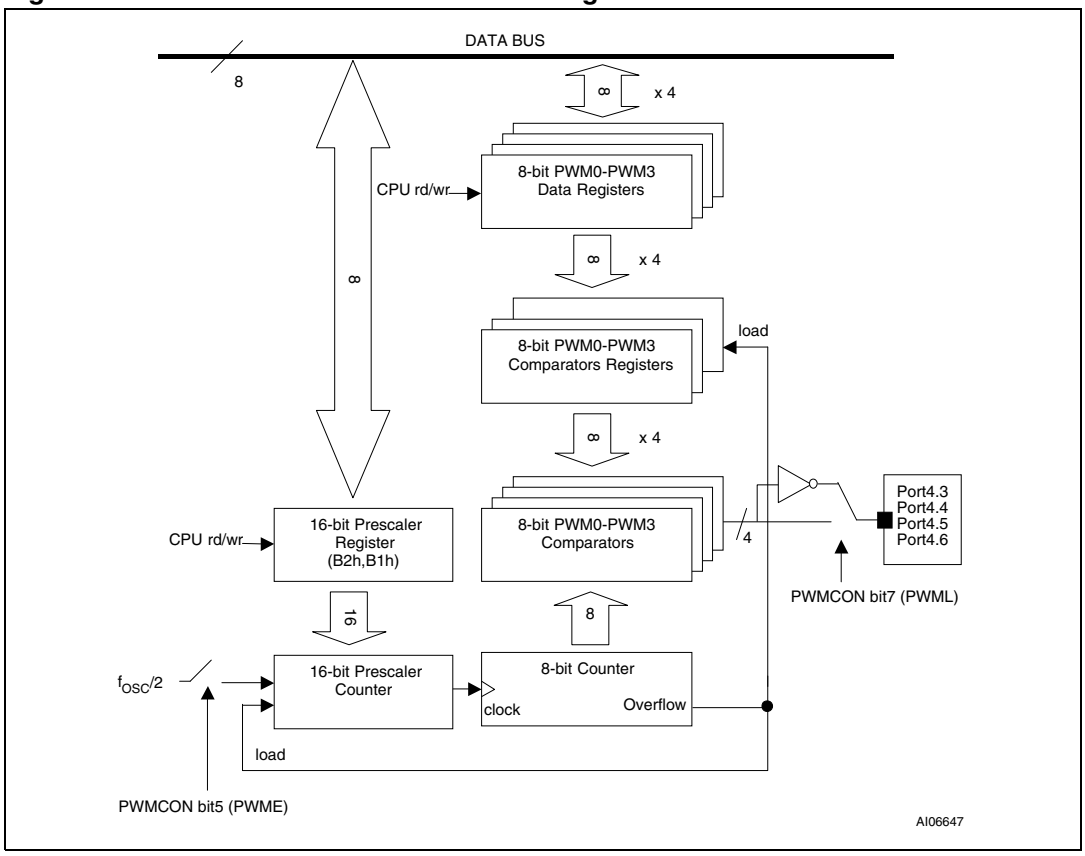


Table 49. PWM SFR memory map

SFR addr	Reg name	Bit register name								Reset value	Comments
		7	6	5	4	3	2	1	0		
A1	PWMCON	PWML	PWMP	PWME	CFG4	CFG3	CFG2	CFG1	CFG0	00	PWM Control Polarity
A2	PWM0									00	PWM0 Output Duty Cycle
A3	PWM1									00	PWM1 Output Duty Cycle
A4	PWM2									00	PWM2 Output Duty Cycle
A5	PWM3									00	PWM3 Output Duty Cycle
AA	PWM4P									00	PWM 4 Period
AB	PWM4W									00	PWM 4 Pulse Width
B1	PSCL0L									00	Prescaler 0 Low (8-bit)

Table 79. Transceiver AC characteristics

Symb	Parameter	Test Conditions(1)	Min	Max	Unit
tDRATE	Low Speed Data Rate	Ave. bit rate (1.5Mb/s ± 1.5%)	1.4775	1.5225	Mbit/s
tDJR1	Receiver Data Jitter Tolerance	To next transition, *Figure 42*(5)	–75	75	ns
tDJR2	Differential Input Sensitivity	For paired transition, *Figure 42*(5)	–45	45	ns
tDEOP	Differential to EOP Transition Skew	*Figure 43*(5)	–40	100	ns
tEOPR1	EOP Width at Receiver	Rejects as EOP(5,6)	165	—	ns
tEOPR2	EOP Width at Receiver	Accepts as EOP(5)	675	—	ns
tEOPT	Source EOP Width	—	–1.25	1.50	µs
tUDJ1	Differential Driver Jitter	To next transition, *Figure 44*	–95	95	ns
tUDJ2	Differential Driver Jitter	To paired transition, *Figure 44*	–150	150	ns
tR	USB Data Transition Rise Time	Notes 2, 3, 4	75	300	ns
tF	USB Data Transition Fall Time	Notes 2, 3, 4	75	300	ns
tRFM	Rise/Fall Time Matching	t_R / t_F	80	120	%
V_{CRS}	Output Signal Crossover Voltage	—	1.3	2.0	V

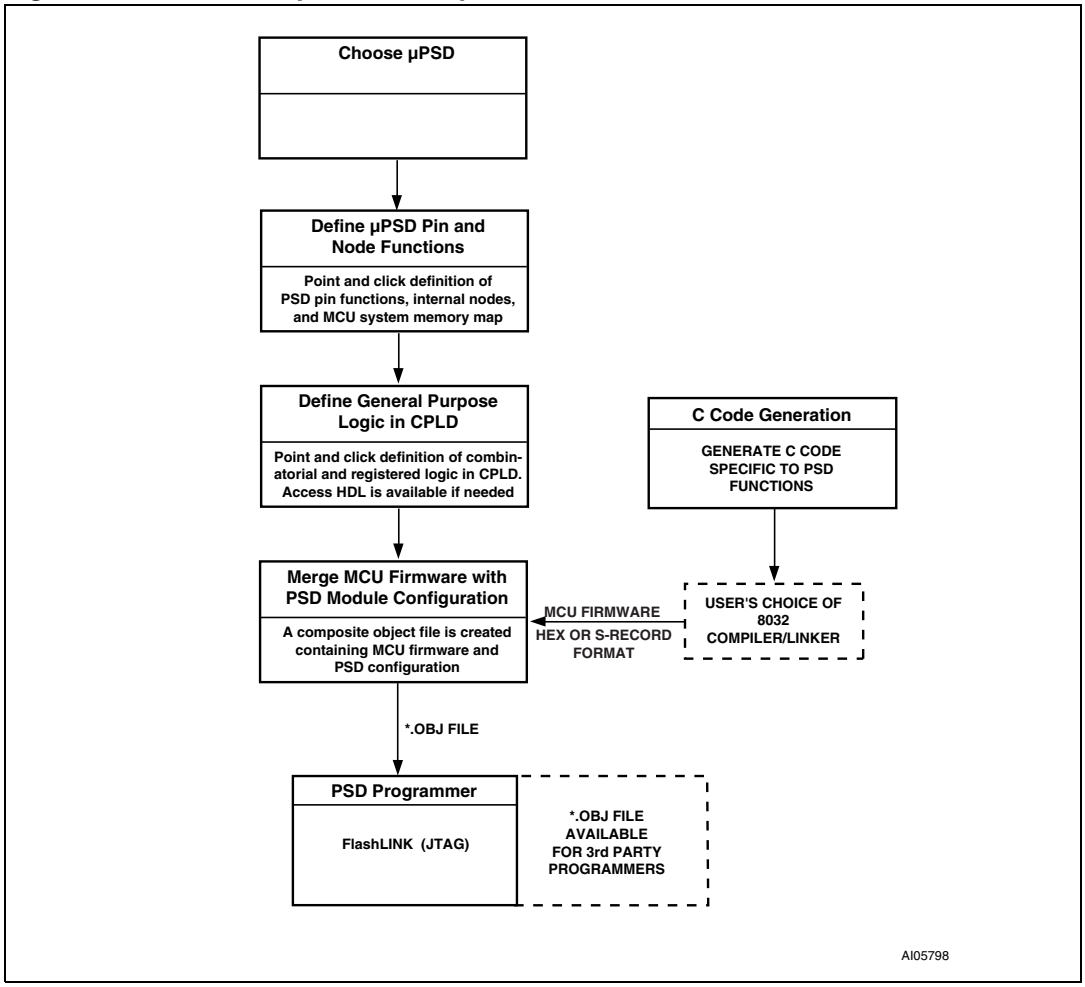
1. V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_A = 0 to 70°C.
2. Level guaranteed for range of V_{CC} = 4.5 V to 5.5 V.
3. With RPU, external idle resistor, 7.5κ±2%, D- to V_{CC}.
4. C_L of 50 pF (75 ns) to 350 pF (300 ns).
5. Measured at crossover point of differential data signals.
6. USB specification indicates 330 ns.

18 Development system

UPSD321xx devices are supported by PSDsoft, a Windows-based software development tool (Windows-95, Windows-98, Windows-NT). A PSD module design is quickly and easily produced in a point and click environment. The designer does not need to enter Hardware Description Language (HDL) equations, unless desired, to define PSD module pin functions and memory map information. The general design flow is shown in *Figure 46*. PSDsoft is available from our web site (the address is given on the back page of this data sheet) or other distribution channels.

PSDsoft directly supports a low cost device programmer from ST: FlashLINK (JTAG). The programmer may be purchased through your local distributor/representative. UPSD321xx devices are also supported by third party device programmers. See our web site for the current list.

Figure 46. PSDsoft express development tool



19 PSD module register description and address offset

Table 81 shows the offset addresses to the PSD module registers relative to the CSIOP base address. The CSIOP space is the 256 bytes of address that is allocated by the user to the internal PSD module registers. *Table 81* provides brief descriptions of the registers in CSIOP space. The following section gives a more detailed description.

Table 81. Register address offset

Register Name	Port A	Port B	Port C	Port D	Other[1]	Description
Data In	00	01	10	11		Reads Port pin as input, MCU I/O Input mode
Control	02	03				Selects mode between MCU I/O or Address Out
Data Out	04	05	12	13		Stores data for output to Port pins, MCU I/O Output mode
Direction	06	07	14	15		Configures Port pin as input or output
Drive Select	08	09	16	17		Configures Port pins as either CMOS or Open Drain on some pins, while selecting high slew rate on other pins.
Input Macrocell	0A	0B	18			Reads Input Macrocells
Enable Out	0C	0D	1A	1B		Reads the status of the output enable to the I/O Port driver
Output Macrocells AB	20	20				READ – reads output of macrocells AB WRITE – loads macrocell flip-flops
Output Macrocells BC		21	21			READ – reads output of macrocells BC WRITE – loads macrocell flip-flops
Mask Macrocells AB	22	22				Blocks writing to the Output Macrocells AB
Mask Macrocells BC		23	23			Blocks writing to the Output Macrocells BC
Primary Flash Protection					C0	Read-only – Primary Flash Sector Protection
Secondary Flash memory Protection					C2	Read-only – PSD module Security and Secondary Flash memory Sector Protection
PMMR0					B0	Power Management Register 0
PMMR2					B4	Power Management Register 2
Page					E0	Page Register
VM					E2	Places PSD module memory areas in Program and/or Data space on an individual basis.

1. Other registers that are not part of the I/O ports.

21.10.2 Memory Select configuration in Program and Data spaces

The MCU Core has separate address spaces for Program memory and Data memory. Any of the memories within the PSD module can reside in either space or both spaces. This is controlled through manipulation of the VM Register that resides in the CSIOP space.

The VM Register is set using PSDsoft Express to have an initial value. It can subsequently be changed by the MCU so that memory mapping can be changed on-the-fly.

For example, you may wish to have SRAM and primary Flash memory in the Data space at Boot-up, and secondary Flash memory in the Program space at Boot-up, and later swap the primary and secondary Flash memories. This is easily done with the VM Register by using PSDsoft Express Configuration to configure it for Boot-up and having the MCU change it when desired. *Table 86* describes the VM Register.

Figure 49. Priority level of memory and I/O components in the PSD module

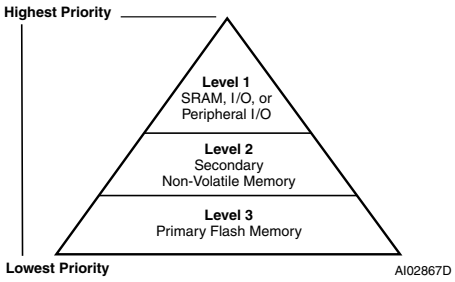


Table 86. VM register

Bit 7 PIO_EN	Bit 6	Bit 5	Bit 4 Primary FL_Data	Bit 3 Secondary Data	Bit 2 Primary FL_Code	Bit 1 Secondary Code	Bit 0 SRAM_Code
0 = disable PIO mode	not used	not used	0 = $\overline{RD}$ can't access Flash memory	0 = $\overline{RD}$ can't access Secondary Flash memory	0 = $\overline{PSEN}$ can't access Flash memory	0 = $\overline{PSEN}$ can't access Secondary Flash memory	0 = $\overline{PSEN}$ can't access SRAM
1= enable PIO mode	not used	not used	1 = $\overline{RD}$ access Flash memory	1 = $\overline{RD}$ access Secondary Flash memory	1 = $\overline{PSEN}$ access Flash memory	1 = $\overline{PSEN}$ access Secondary Flash memory	1 = $\overline{PSEN}$ access SRAM

21.10.3 Separate Space mode

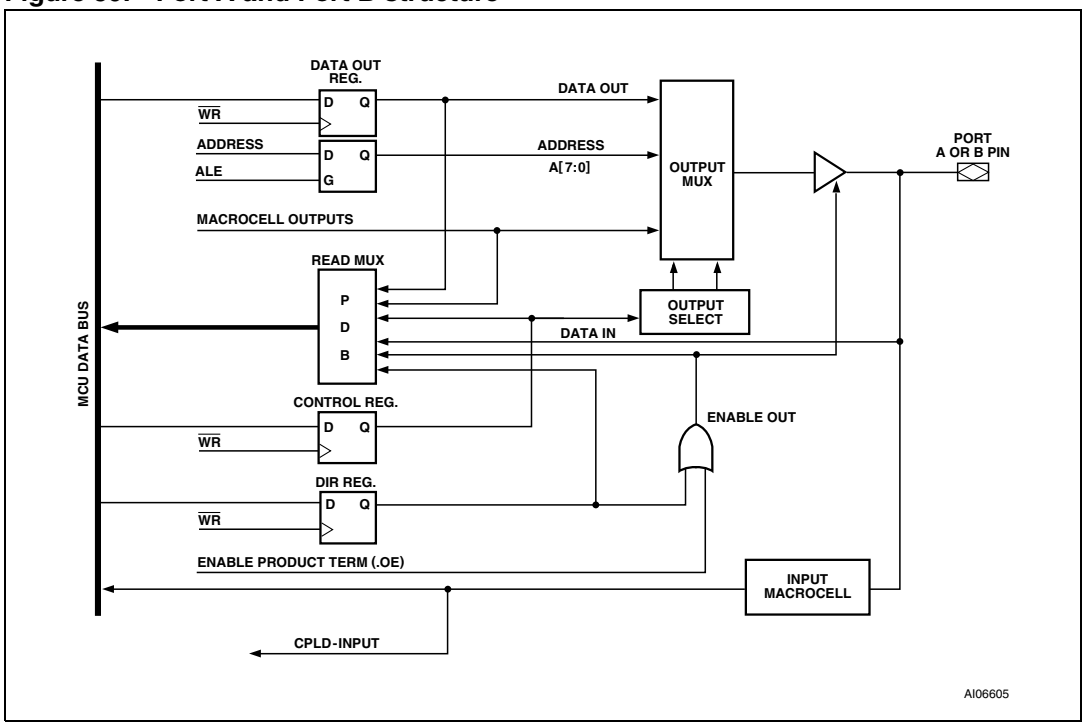
Program space is separated from Data space. For example, Program Select Enable ($\overline{PSEN}$) is used to access the program code from the primary Flash memory, while READ Strobe ($\overline{RD}$) is used to access data from the secondary Flash memory, SRAM and I/O Port blocks. This configuration requires the VM Register to be set to 0Ch (see *Figure 50*).

23.10 Ports A and B – functionality and structure

Ports A and B have similar functionality and structure, as shown in *Figure 59*. The two ports can be configured to perform one or more of the following functions:

- MCU I/O mode
- CPLD Output – Macrocells McellAB7-McellAB0 can be connected to Port A or Port B. McellBC7-McellBC0 can be connected to Port B or Port C.
- CPLD Input – Via the Input Macrocells (IMC).
- Latched Address output – Provide latched address output as per *Table 92*.
- Open Drain/Slew Rate – pins PA3-PA0 and PB3-PB0 can be configured to fast slew rate, pins PA7-PA4 and PB7-PB4 can be configured to Open Drain mode.
- Peripheral mode – Port A only (80-pin package)

Figure 59. Port A and Port B structure



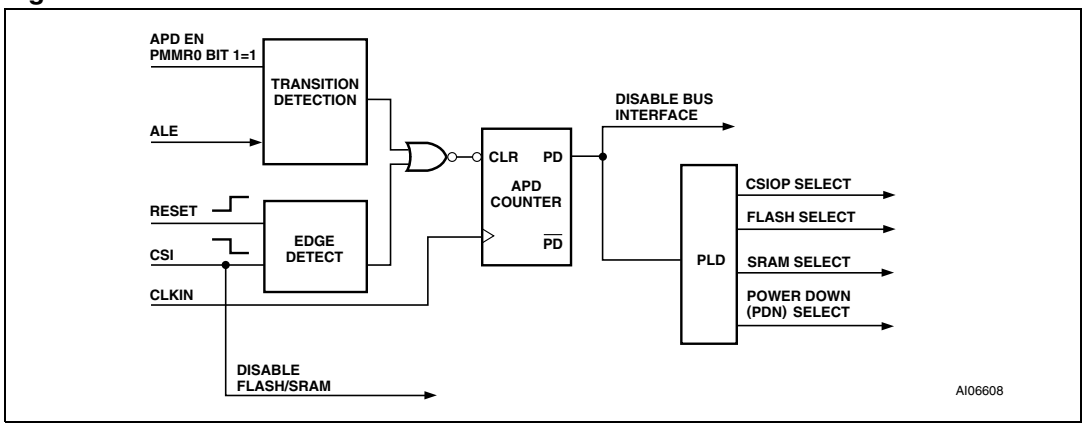
23.11 Port C – functionality and structure

Port C can be configured to perform one or more of the following functions (see *Figure 60*):

- MCU I/O mode
- CPLD Output – McellBC7-McellBC0 outputs can be connected to Port B or Port C.
- CPLD Input – via the Input Macrocells (IMC)
- In-System Programming (ISP) – JTAG pins (TMS, TCK, TDI, TDO) are dedicated pins for device programming. (See *Section 26: Programming in-circuit using the JTAG serial interface*, for more information on JTAG programming.)
- Open Drain – Port C pins can be configured in Open Drain mode

Port C does not support Address Out mode, and therefore no Control Register is required.

Figure 63. APD unit



The PSD module has a Turbo Bit in PMMR0. This bit can be set to turn the Turbo mode off (the default is with Turbo mode turned on). While Turbo mode is off, the PLDs can achieve standby current when no PLD inputs are changing (zero DC current). Even when inputs do change, significant power can be saved at lower frequencies (AC current), compared to when Turbo mode is on. When the Turbo mode is on, there is a significant DC current component and the AC component is higher.

Automatic Power-down (APD) Unit and Power-down mode

The APD Unit, shown in *Figure 63*, puts the PSD module into Power-down mode by monitoring the activity of Address Strobe (ALE). If the APD Unit is enabled, as soon as activity on Address Strobe (ALE) stops, a four-bit counter starts counting. If Address Strobe (ALE/AS, PD0) remains inactive for fifteen clock periods of CLKIN (PD1), Power-down (PDN) goes High, and the PSD module enters Power-down mode, as discussed next.

Power-down mode

By default, if you enable the APD Unit, Power-down mode is automatically enabled. The device enters Power-down mode if Address Strobe (ALE) remains inactive for fifteen periods of CLKIN (PD1).

The following should be kept in mind when the PSD module is in Power-down mode:

- If Address Strobe (ALE) starts pulsing again, the PSD module returns to normal Operating mode. The PSD module also returns to normal Operating mode if either PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) is Low or the $\overline{\text{RESET}}$ input is High.
- The MCU address/data bus is blocked from all memory and PLDs.
- Various signals can be blocked (prior to Power-down mode) from entering the PLDs by setting the appropriate bits in the PMMR registers. The blocked signals include MCU control signals and the common CLKIN (PD1).

Note: Blocking CLKIN (PD1) from the PLDs does not block CLKIN (PD1) from the APD Unit.

- All memories enter Standby mode and are drawing standby current. However, the PLD and I/O ports blocks do *not* go into Standby mode because you don't want to have to wait for the logic and I/O to "wake-up" before their outputs can change. See *Table 99* for Power-down mode effects on PSD module ports.
- Typical standby current is of the order of microamperes. These standby current values assume that there are no transitions on any PLD input.

26 Programming in-circuit using the JTAG serial interface

The JTAG Serial Interface pins (TMS, TCK, TDI, and TDO) are dedicated pins on Port C (see *Table 104*). All memory blocks (primary and secondary Flash memory), PLD logic, and PSD module Configuration Register Bits may be programmed through the JTAG Serial Interface block. A blank device can be mounted on a printed circuit board and programmed using JTAG.

The standard JTAG signals (IEEE 1149.1) are TMS, TCK, TDI, and TDO. Two additional signals, $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$, are optional JTAG extensions used to speed up Program and Erase cycles.

By default, on a blank device (as shipped from the factory or after erasure), four pins on Port C are the basic JTAG signals TMS, TCK, TDI, and TDO.

26.1 Standard JTAG Signals

At power-up, the standard JTAG pins are inputs, waiting for a JTAG serial command from an external JTAG controller device (such as FlashLINK or Automated Test Equipment). When the enabling command is received, TDO becomes an output and the JTAG channel is fully functional. The same command that enables the JTAG channel may optionally enable the two additional JTAG signals, $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$.

The $\overline{\text{RESET}}$ input to the uPS3200 should be active during JTAG programming. The active $\overline{\text{RESET}}$ puts the MCU module into RESET mode while the PSD module is being programmed. See Application Note AN1153 for more details on JTAG In-System Programming (ISP).

UPSD321xx devices support JTAG In-System-Configuration (ISC) commands, but not Boundary Scan. The PSDsoft Express software tool and FlashLINK JTAG programming cable implement the JTAG In-System-Configuration (ISC) commands.

Table 104. JTAG port signals

Port C Pin	JTAG Signals	Description
PC0	TMS	Mode Select
PC1	TCK	Clock
PC3	TSTAT	Status (optional)
PC4	TERR	Error Flag (optional)
PC5	TDI	Serial Data In
PC6	TDO	Serial Data Out

26.2 JTAG extensions

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ are two JTAG extension signals enabled by an “ISC_ENABLE” command received over the four standard JTAG signals (TMS, TCK, TDI, and TDO). They are used to speed Program and Erase cycles by indicating status on uPDS signals instead of having to

Table 115. DC characteristics (5 V devices)

Symbol	Parameter	Test conditions (in addition to those in *Table 110*)	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage (Ports 1, 2, 3, 4[Bits 7,6,5,4,3,1,0], XTAL1, $\overline{RESET}$)	4.5 V < V_{CC} < 5.5 V	0.7 V_{CC}		V_{CC} + 0.5	V
V_{IH1}	Input high voltage (Ports A, B, C, D, 4[Bit 2], USB+, USB–)	4.5 V < V_{CC} < 5.5 V	2.0		V_{CC} + 0.5	V
V_{IL}	Input low voltage (Ports 1, 2, 3, 4[Bits 7,6,5,4,3,1,0], XTAL1, $\overline{RESET}$)	4.5 V < V_{CC} < 5.5 V	V_{SS}– 0.5		0.3 V_{CC}	V
V_{IL1}	Input low voltage (Ports A, B, C, D, 4[Bit 2])	4.5 V < V_{CC} < 5.5 V	–0.5		0.8	V
	Input low voltage (USB+, USB–)	4.5 V < V_{CC} < 5.5 V	V_{SS}– 0.5		0.8	V
V_{OL}	Output low voltage (Ports A,B,C,D)	I_{OL} = 20 µA V_{CC} = 4.5 V		0.01	0.1	V
		I_{OL} = 8 mA V_{CC} = 4.5 V		0.25	0.45	V
V_{OL1}	Output low voltage (Ports 1,2,3,4, $\overline{WR}$, $\overline{RD}$)	I_{OL} = 1.6 mA			0.45	V
V_{OL2}	Output low voltage (Port 0, ALE, PSEN)	I_{OL} = 3.2 mA			0.45	V
V_{OH}	Output high voltage (Ports A,B,C,D)	I_{OH} = –20 µA V_{CC} = 4.5 V	4.4	4.49		V
		I_{OH} = –2 mA V_{CC} = 4.5 V	2.4	3.9		V
V_{OH2}	Output high voltage (Port 0 in ext. Bus mode, ALE, PSEN)	I_{OH} = –800 µA	2.4			V
		I_{OH} = –80 µA	4.05			V
V_{LVR}	Low Voltage $\overline{RESET}$	0.1 V hysteresis	3.75	4.0	4.25	V
V_{OP}	XTAL open bias voltage (XTAL1, XTAL2)	I_{OL} = 3.2 mA	2.0		3.0	V
V_{LKO}	V_{CC}(min) for Flash Erase and Program		2.5		4.2	V
I_{IL}	Logic '0' input current (Ports 1,2,3,4)	V_{IN} = 0.45 V (0 V for Port 4[pin 2])	–10		–50	µA
I_{TL}	Logic 1-to-0 transition current (Ports 1,2,3,4)	V_{IN} = 3.5 V (2.5 V for Port 4[pin 2])	–65		–650	µA
I_{RST}	Reset pin pull-up current ($\overline{RESET}$)	V_{IN} = V_{SS}	–10		–55	µA

Table 116. DC characteristics (3 V devices)

Symbol	Parameter	Test conditions (in addition to those in *Table 111*)	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage (Ports 1, 2, 3, 4[Bits 7,6,5,4,3,1,0], A, B, C, D, XTAL1, $\overline{RESET}$)	3.0 V < V_{CC} < 3.6 V	0.7V_{CC}		V_{CC} + 0.5	V
V_{IH1}	Input high voltage (Port 4[Bit 2])	3.0 V < V_{CC} < 3.6 V	2.0		V_{CC} + 0.5	V
V_{IL}	Input high voltage (Ports 1, 2, 3, 4[Bits 7,6,5,4,3,1,0], XTAL1, $\overline{RESET}$)	3.0 V < V_{CC} < 3.6 V	V_{SS}– 0.5		0.3 V_{CC}	V
V_{IL1}	Input low voltage (Ports A, B, C, D)	3.0 V < V_{CC} < 3.6 V	–0.5		0.8	V
	Input low voltage (Port 4[Bit 2])	3.0 V < V_{CC} < 3.6 V	V_{SS}– 0.5		0.8	V
V_{OL}	Output low voltage (Ports A,B,C,D)	I_{OL} = 20 µA V_{CC} = 3.0 V		0.01	0.1	V
		I_{OL} = 4 mA V_{CC} = 3.0 V		0.15	0.45	V
V_{OL1}	Output low voltage (Ports 1,2,3,4, $\overline{WR}$, $\overline{RD}$)	I_{OL} = 1.6 mA			0.45	V
		I_{OL} = 100 µA			0.3	V
V_{OL2}	Output low voltage (Port 0, ALE, PSEN)	I_{OL} = 3.2 mA			0.45	V
		I_{OL} = 200 µA			0.3	V
V_{OH}	Output high voltage (Ports A,B,C,D)	I_{OH} = –20 µA V_{CC} = 3.0 V	2.9	2.99		V
		I_{OH} = –1 mA V_{CC} = 3.0 V	2.4	2.6		V
V_{OH2}	Output high voltage (Port 0 in ext. Bus mode, ALE, $\overline{PSEN}$)	I_{OH} = –800 µA	2.0			V
		I_{OH} = –80 µA	2.7			V
V_{LVR}	Low voltage reset	0.1 V hysteresis	2.3	2.5	2.7	V
V_{OP}	XTAL open bias voltage (XTAL1, XTAL2)	I_{OL} = 3.2 mA	1.0		2.0	V
V_{LKO}	V_{CC}(min) for Flash Erase and Program		1.5		2.2	V
I_{IL}	Logic '0' input current (Ports 1,2,3,4)	V_{IN} = 0.45 V (0 V for Port 4[pin 2])	–1		–50	µA
I_{TL}	Logic 1-to-0 transition current (Ports 1,2,3,4)	V_{IN} = 3.5 V (2.5 V for Port 4[pin 2])	–25		–250	µA
I_{RST}	Reset pin pull-up current ($\overline{RESET}$)	V_{IN} = V_{SS}	–10		–55	µA
I_{FR}	XTAL feedback resistor current (XTAL1)	XTAL1 = V_{CC} XTAL2 = V_{SS}	–20		–50	µA
I_{LI}	Input leakage current	V_{SS} < V_{IN} < V_{CC}	–1		1	µA
I_{LO}	Output leakage current	0.45 < V_{OUT} < V_{CC}	–10		10	µA

Table 118. External program memory AC characteristics (with the 3 V MCU module)

Symbol	Parameter[1]	24 MHz oscillator		Variable oscillator $1/t_{CLCL}$ = 8 to 24 MHz		Unit
		Min.	Max.	Min.	Max.	
t_{LHLL}	ALE pulse width	43		$2\ t_{CLCL} - 40$		ns
t_{AVLL}	Address set-up to ALE	17		$t_{CLCL} - 25$		ns
t_{LLAX}	Address hold after ALE	17		$t_{CLCL} - 25$		ns
t_{LLIV}	ALE Low to valid instruction in		80		$4\ t_{CLCL} - 87$	ns
t_{LLPL}	ALE to $\overline{PSEN}$	22		$t_{CLCL} - 20$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	95		$3\ t_{CLCL} - 30$		ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in		60		$3\ t_{CLCL} - 65$	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}[2]	Input instruction float after $\overline{PSEN}$		32		$t_{CLCL} - 10$	ns
t_{PXAV}[2]	Address valid after $\overline{PSEN}$	37		$t_{CLCL} - 5$		ns
t_{AVIV}	Address to valid instruction in		148		$5\ t_{CLCL} - 60$	ns
t_{AZPL}	Address float to $\overline{PSEN}$	–10		–10		ns

1. Conditions (in addition to those in *Table 111*, V_{CC} = 3.0 to 3.6 V): V_{SS} = 0 V; C_L for Port 0, ALE and PSEN output is 100 pF, for 5 V devices, and 50 pF for 3 V devices; C_L for other outputs is 80 pF, for 5 V devices, and 50 pF for 3 V devices)
2. Interfacing the UPSD321xx devices to devices with float times up to 35 ns is permissible. This limited bus contention does not cause any damage to Port 0 drivers.

Table 119. External clock drive (with the 5 V MCU module)

Symbol	Parameter[1]	40 MHz oscillator		Variable oscillator $1/t_{CLCL}$ = 24 to 40 MHz		Unit
		Min.	Max.	Min.	Max.	
t_{RLRH}	Oscillator period			25	41.7	ns
t_{WLWH}	High time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{LLAX2}	Low time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{RHDX}	Rise time				10	ns
t_{RHDX}	Fall time				10	ns

1. Conditions (in addition to those in *Table 110*, V_{CC} = 4.5 to 5.5 V): V_{SS} = 0 V; C_L for Port 0, ALE and PSEN output is 100 pF; C_L for other outputs is 80 pF

Figure 78. Peripheral I/O Write timing

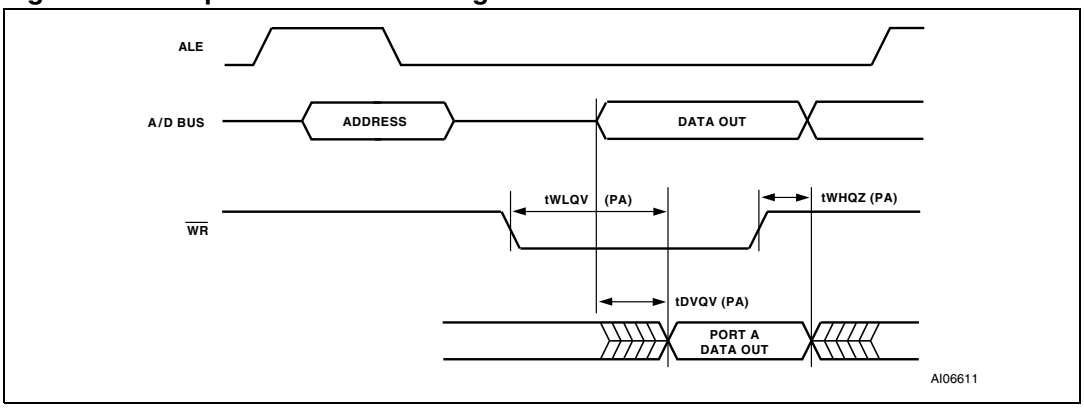


Table 136. Port A peripheral data mode Write timing (5 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to Data Propagation Delay			25	ns
$t_{DVQV-PA}$	Data to Port A Data Propagation Delay	(Note 1)		22	ns
$t_{WHQZ-PA}$	$\overline{WR}$ Invalid to Port A Tri-state			20	ns

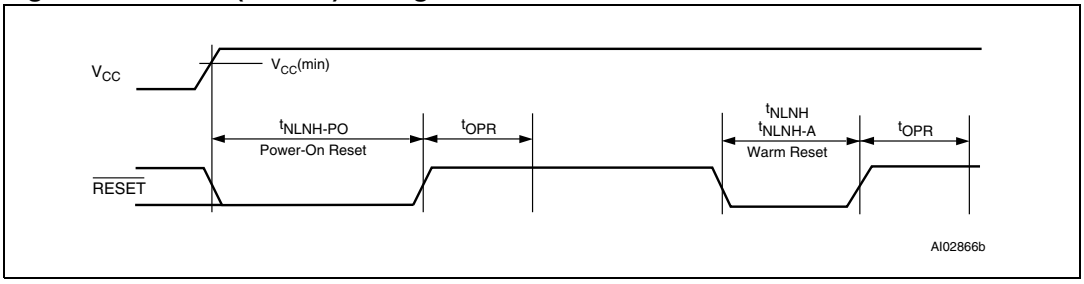
1. Data stable on Port 0 pins to data on Port A.

Table 137. Port A peripheral data mode Write timing (3 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to data propagation delay			42	ns
$t_{DVQV-PA}$	Data to Port A data propagation delay	(Note 1)		38	ns
$t_{WHQZ-PA}$	$\overline{WR}$ invalid to Port A tri-state			33	ns

1. Data stable on Port 0 pins to data on Port A.

Figure 79. Reset ($\overline{RESET}$) timing



34 Revision history

Table 146. Document revision history

Date	Revision	Changes
18-Dec-2002	1.0	First Issue
04-Mar-03	1.1	Updates: port information (Table 30); interface information (Figure 30, Table 44); remove programming guide; PSD module information (Table 82); PLD information (Figure 55); electrical characteristics (Table 114, 115, 131, 132)
02-Sep-03	1.2	Update references for Product Catalog
03-Feb-04	2.0	Reformatted; correct package dimensions (Table 145)
02-July-04	3.0	Reformatted; add EMC characteristics information (Table 106, 107, 108)
04-Nov-04	4.0	Updates per requested data brief changes (Figure 3, 4; Table 1, 2, 113)
03-Dec-04	5.0	Add USB feature to document (Figure 2, 3, 4, 15, 16, 18, 20, 40, 41, 42, 43, 44, 45; Table 1, 2, 15, 16, 18, 19, 21, 23, 24, 25, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 146)
21-Jan-2009	6.0	Removed battery backup feature and related SRAM Standby mode information. Added Ecopack information and updated *Section 32: Package mechanical information on page 177*.