STMicroelectronics - UPSD3422EV-40U6 Datasheet

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	40MHz
Connectivity	I²C, IrDA, SPI, UART/USART, USB
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	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
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/upsd3422ev-40u6

4 Memory organization

The 8032 MCU core views memory on the MCU module as “internal” memory and it views memory on the PSD module as “external” memory, see *Figure 5*

Internal memory on the MCU module consists of DATA, IDATA, and SFRs. These standard 8032 memories reside in 384 bytes of SRAM located at a fixed address space starting at address 0000h.

External memory on the PSD module consists of four types: main Flash (64 Kbyte, 128 Kbyte, or 256 Kbyte), a smaller secondary Flash (32 Kbyte), SRAM (4 Kbyte, 8 Kbyte or 32 Kbyte), and a block of PSD module control registers called csiop (256 bytes). These external memories reside at programmable address ranges, specified using the software tool PSDsoft Express. See the PSD module section of this document for more details on these memories.

External memory is accessed by the 8032 in two separate 64 Kbyte address spaces. One address space is for program memory and the other address space is for data memory. Program memory is accessed using the 8032 signal, $\overline{\text{PSEN}}$. Data memory is accessed using the 8032 signals, $\overline{\text{RD}}$ and $\overline{\text{WR}}$. If the 8032 needs to access more than 64 Kbytes of external program or data memory, it must use paging (or banking) techniques provided by the page register in the PSD module.

Note: When referencing program and data memory spaces, it has nothing to do with 8032 internal SRAM areas of DATA, IDATA, and SFR on the MCU module. Program and data memory spaces only relate to the external memories on the PSD module.

External memory on the PSD module can overlap the internal SRAM memory on the MCU module in the same physical address range (starting at 0000h) without interference because the 8032 core does not assert the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signals when accessing internal SRAM.

Figure 5. UPSD34xx memories

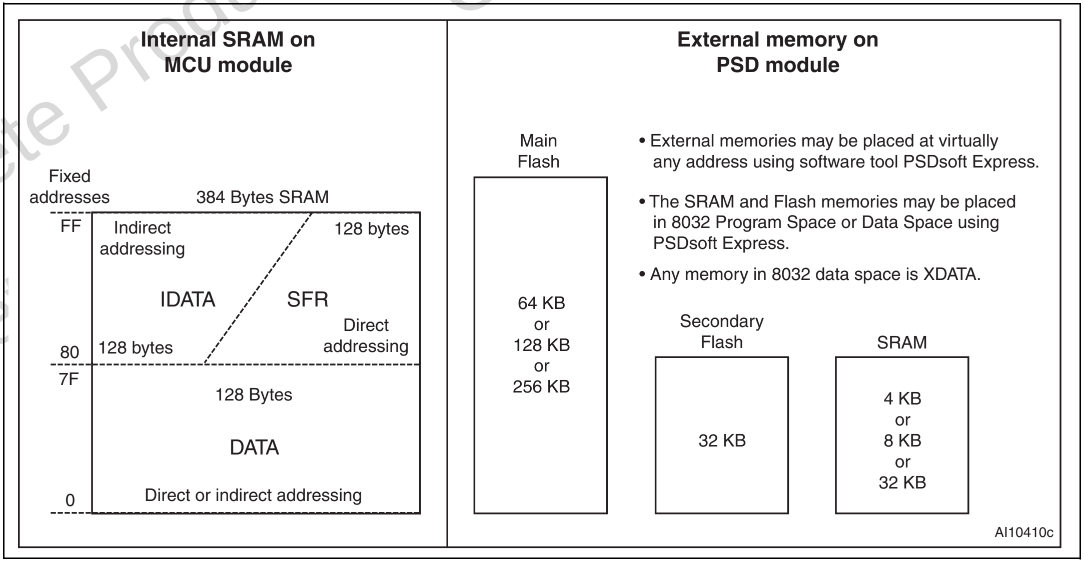
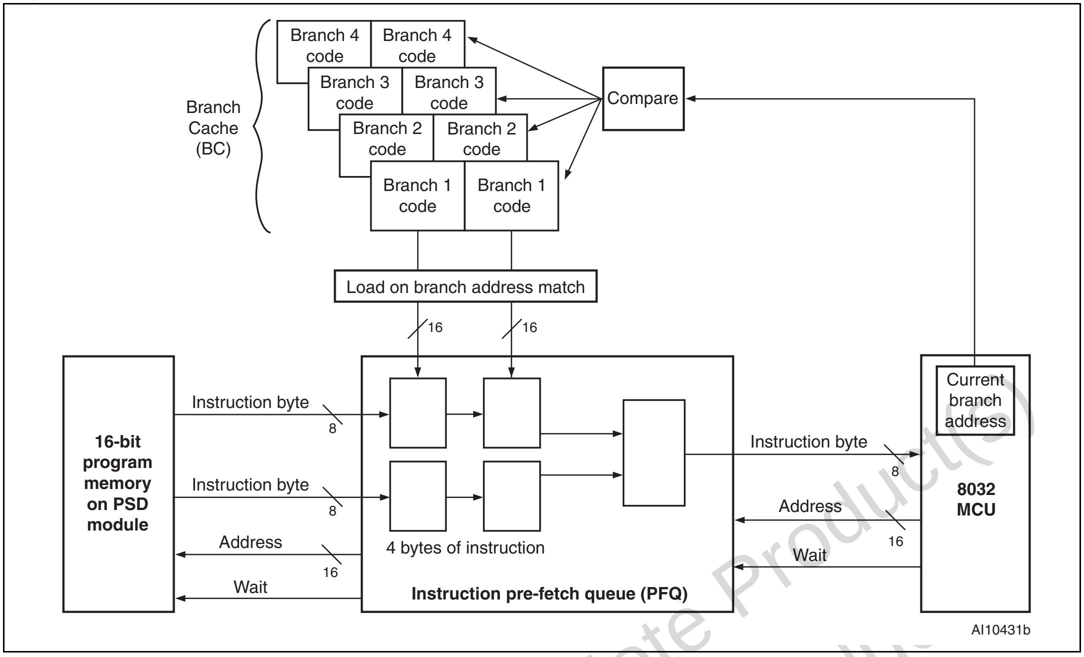

Internal SRAM on MCU module
Fixed addresses
384 Bytes SRAM
FF
Indirect addressing
128 bytes
IDATA
SFR
Direct addressing
80
128 bytes
7F
128 Bytes
DATA
Direct or indirect addressing
0
External memory on PSD module
Main Flash
64 KB or 128 KB or 256 KB
• External memories may be placed at virtually any address using software tool PSDsoft Express.
• The SRAM and Flash memories may be placed in 8032 Program Space or Data Space using PSDsoft Express.
• Any memory in 8032 data space is XDATA.
Secondary Flash
32 KB
SRAM
4 KB or 8 KB or 32 KB
AI10410c

Figure 7. Instruction pre-fetch queue and branch cache



5.1 Pre-fetch queue (PFQ) and branch cache (BC)

The PFQ is always working to minimize the idle bus time inherent to 8032 MCU architecture, to eliminate wasted memory fetches, and to maximize memory bandwidth to the MCU. The PFQ does this by running asynchronously in relation to the MCU, looking ahead to pre-fetch two bytes (word) of code from program memory during any idle bus periods. Only necessary word will be fetched (no dummy fetches like standard 8032). The PFQ will queue up to four code bytes in advance of execution, which significantly optimizes sequential program performance. However, when program execution becomes non-sequential (program branch), a typical pre-fetch queue will empty itself and reload new code, causing the MCU to stall. The Turbo UPSD34xx diminishes this problem by using a Branch Cache with the PFQ. The BC is a four-way, fully associative cache, meaning that when a program branch occurs, its branch destination address is compared simultaneously with four recent previous branch destinations stored in the BC. Each of the four cache entries contain up to four bytes of code related to a branch. If there is a hit (a match), then all four code bytes of the matching program branch are transferred immediately and simultaneously from the BC to the PFQ, and execution on that branch continues with minimal delay. This greatly reduces the chance that the MCU will stall from an empty PFQ, and improves performance in embedded control systems where it is quite common to branch and loop in relatively small code localities.

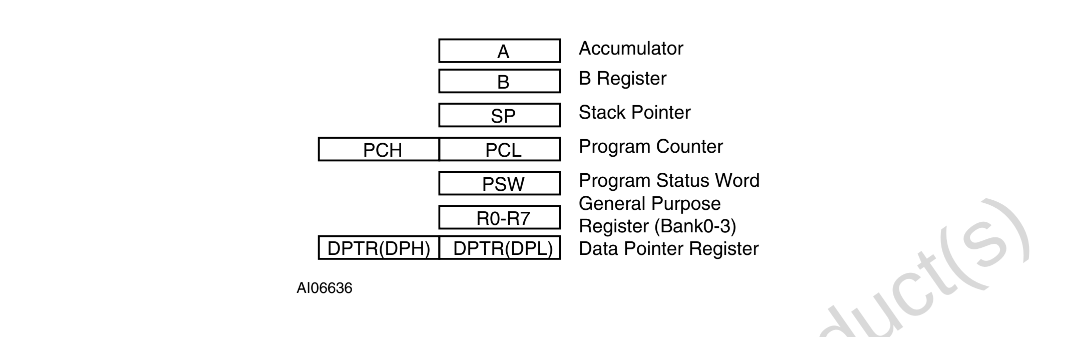
By default, the PFQ and BC are enabled after power-up or reset. The 8032 can disable the PFQ and BC at runtime if desired by writing to a specific SFR (BUSCON).

The memory in the PSD module operates with variable wait states depending on the value specified in the SFR named BUSCON. For example, a 5 V UPSD34xx device operating at a 40 MHz crystal frequency requires four memory wait states (equal to four MCU clocks). In this example, once the PFQ has one word of code, the wait states become transparent and a full 10 MIPS is achieved when the program stream consists of sequential one- or two-byte, one machine-cycle instructions as shown in *Figure 6 on page 33* (transparent because a machine-cycle is four MCU clocks which equals the memory pre-fetch wait time that is also

7 8032 MCU registers

The UPSD34xx has the following 8032 MCU core registers, also shown in *Figure 10*.

Figure 10. 8032 MCU registers



7.1 Stack pointer (SP)

The SP is an 8-bit register which holds the current location of the top of the stack. It is incremented before a value is pushed onto the stack, and decremented after a value is popped off the stack. The SP is initialized to 07h after reset. This causes the stack to begin at location 08h (top of stack). To avoid overlapping conflicts, the user must initialize the top of the stack to 20h if all four banks of registers R0 - R7 are used, as well as the top of stack to 30h if all of the 8032 bit memory locations are used.

7.2 Data pointer (DPTR)

DPTR is a 16-bit register consisting of two 8-bit registers, DPL and DPH. The DPTR register is used as a base register to create an address for indirect jumps, table look-up operations, and for external data transfers (XDATA). When not used for addressing, the DPTR register can be used as a general purpose 16-bit data register.

Very frequently, the DPTR register is used to access XDATA using the external direct addressing mode. The UPSD34xx has a special set of SFR registers (DPTC, DPTM) to control a secondary DPTR register to speed memory-to-memory XDATA transfers. Having dual DPTR registers allows rapid switching between source and destination addresses (see details in *Section 11: Dual data pointers on page 57*).

7.3 Program counter (PC)

The PC is a 16-bit register consisting of two 8-bit registers, PCL and PCH. This counter indicates the address of the next instruction in program memory to be fetched and executed. A reset forces the PC to location 0000h, which is where the reset jump vector is stored.

7.7.5 Overflow flag (OV)

The OV flag is set when: an ADD, ADDC, or SUBB instruction causes a sign change; a MUL instruction results in an overflow (result greater than 255); a DIV instruction causes a divide-by-zero condition. The OV flag is cleared by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases. The CLRV instruction will clear the OV flag at any time.

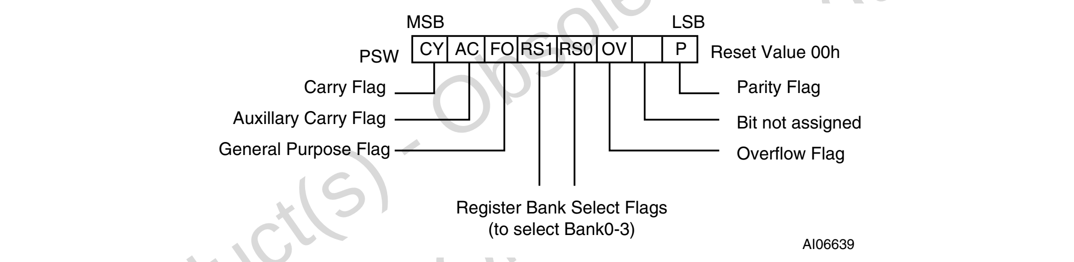
7.7.6 Parity flag (P)

The P flag is set if the sum of the eight bits in the Accumulator is odd, and P is cleared if the sum is even.

Table 4. Register bank select addresses

RS1	RS0	Register bank	8032 internal data address
0	0	0	00h - 07h
0	1	1	08h - 0Fh
1	0	2	10h - 17h
1	1	3	18h - 1Fh

Figure 11. Program status word (PSW) register



CAPCOML4, CAPCOMH4, TCMMODE4, CAPCOML5, CAPCOMH5, TCMMODE5, PWMF0, PMWF1

- SPI interface registers

  SPICLKD, SPISTAT, SPITDR, SPIRDR, SPICON0, SPICON1
- I^2C interface registers

  S1SETUP, S1CON, S1STA, S1DAT, S1ADR
- Analog to digital converter registers

  ACON, ADCPS, ADAT0, ADAT1
- IrDA interface register

  IRDACON
- USB interface registers

  UADDR, UPAIR, WE0-3, UIF0-3, UCTL, USTA, USEL, UCON, USIZE, UBASEH, UBASEL, USCI, USCV

Table 5. SFR memory map with direct address and reset value

SFR addr (hex)	SFR name	Bit name and <bit address>								Reset value (hex)	Reg. descr. with link
		7	6	5	4	3	2	1	0		
80	RESERVED										
81	SP	SP[7:0]								07	*Section 7.1*
82	DPL	DPL[7:0]								00	*Section 7.2*
83	DPH	DPH[7:0]								00	
84	RESERVED										
85	DPTC	–	AT	–	–	–	DPSEL[2:0]			00	*Table 13*
86	DPTM	–	–	–	–	MD1[1:0]		MD0[1:0]		00	*Table 15*
87	PCON	SMOD0	SMOD1	–	POR	RCLK1	TCLK1	PD	IDLE	00	*Table 33*
88(1)	TCON	TF1 <8Fh>	TR1 <8Eh>	TF0 <8Dh>	TR0 <8Ch>	IE1 <8Bh>	IT1 <8Ah>	IE0 <89h>	IT0 <88h>	00	*Table 56*
89	TMOD	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0	00	*Table 58*
8A	TL0	TL0[7:0]								00	*Section 20.1*
8B	TL1	TL1[7:0]								00	
8C	TH0	TH0[7:0]								00	
8D	TH1	TH1[7:0]								00	
8E	P1SFS0	P1SFS0[7:0]								00	*Table 43*
8F	P1SFS1	P1SFS1[7:0]								00	*Table 44*
90(1)	P1	P1.7 <97h>	P1.6 <96h>	P1.5 <95h>	P1.4 <94h>	P1.3 <93h>	P1.2 <92h>	P1.1 <91h>	P1.0 <90h>	FF	*Table 35*
91	P3SFS	P3SFS[7:0]								00	*Table 41*
92	P4SFS0	P4SFS0[7:0]								00	*Table 46*
93	P4SFS1	P4SFS1[7:0]								00	*Table 47*

Table 45. P1SFS0 and P1SFS1 details (continued)

Port 1 Pin	R/W	Default port function	Alternate 1 port function	Alternate 2 port function

		P1SFS0[i] = 0
		P1SFS1[i] = x
			P1SFS0[i] = 1
			P1SFS1[i] = 0
				P1SFS0[i] = 1
				P1SFS1[i] = 1

		Port 1 Pin, i = 0.. 7	Port 1 Pin, i = 0.. 7	Port 1 Pin, i = 0.. 7
6	R,W	GPIO	SPI Transmit, SPITXD	ADC Chn 6 input, ADC6
7	R,W	GPIO	SPI Select, SPISEL_	ADC Chn 7 input, ADC7

Table 46. P4SFS0: Port 4 special function select 0 register (SFR 92h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4SF07	P4SF06	P4SF05	P4SF04	P4SF03	P4SF02	P4SF01	P4SF00

Table 47. P4SFS1: Port 4 special function select 1 register (SFR 93h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4SF17	P4SF16	P4SF15	P4SF14	P4SF13	P4SF12	P4SF11	P4SF10

Table 48. P4SFS0 and P4SFS1 details

Port 4 pin	R/W	Default port function	Alternate 1 port function	Alternate 2 port function

		P4SFS0[i] = 0
		P4SFS1[i] = x
			P4SFS0[i] = 1
			P4SFS1[i] = 0
				P4SFS0[i] = 1
				P4SFS1[i] = 1

		Port 4 Pin, i = 0.. 7	Port 4 Pin, i = 0.. 7	Port 4 Pin, i = 0.. 7
0	R,W	GPIO	PCA0 Module 0, TCM0	Timer 2 count input, T2
1	R,W	GPIO	PCA0 Module 1, TCM1	Timer 2 trigger input, TX2
2	R,W	GPIO	PCA0 Module 2, TCM2	UART1 Receive, RXD1
3	R,W	GPIO	PCA0 ext clock, PCACLK0	UART1 Transmit, TXD1
4	R,W	GPIO	PCA1 Module 3, TCM3	SPI Clock, SPICLK
5	R,W	GPIO	PCA1 Module 4, TCM4	SPI Receive, SPIRXD
6	R,W	GPIO	PCA1 Module 5, TCM5	SPI Transmit, SPITXD
7	R,W	GPIO	PCA1 ext clock, PCACLK1	SPI Select, SPISEL_

20.4 SFR, TMOD

Timer 0 and Timer 1 have four modes of operation controlled by the SFR named TMOD (*Table 58*).

20.5 Timer 0 and Timer 1 operating modes

The “Timer” or “Counter” function is selected by the $C/\overline{T}$ control bits in TMOD. The four operating modes are selected by bit-pairs M[1:0] in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 is different.

20.5.1 Mode 0

Putting either Timer/Counter into Mode 0 makes it an 8-bit Counter with a divide-by-32 prescaler. *Figure 24* shows Mode 0 operation as it applies to Timer 1 (same applies to Timer 0).

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all '1s' to all '0s,' it sets the Timer Interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or EXTINT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input pin, EXTINT1, to facilitate pulse width measurements). TR1 is a control bit in the SFR, TCON. GATE is a bit in the SFR, TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag, TR1, does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, C0, TL0, TH0, and EXTINT0 for the corresponding Timer 1 signals in *Figure 24*. There are two different GATE Bits, one for Timer 1 and one for Timer 0.

20.5.2 Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

20.5.3 Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in *Figure 25 on page 100*. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset with firmware. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

20.5.4 Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in *Figure 26 on page 100*. TL0 uses the Timer 0 control Bits: $C/\overline{T}$, GATE, TR0, and TF0, as well as the pin EXTINT0. TH0 is locked into a timer function (counting at a rate of 1/12 f_{OSC}) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the “Timer 1“ interrupt flag.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter (see *Figure 26 on page 100*). With Timer 0 in Mode 3, a UPSD34xx device can look like it has three Timer/Counters (not including the PCA). When Timer 0 is in Mode 3, Timer 1 can be

23.6 General call address

A General Call (GC) occurs when a Master-Transmitter initiates a transfer containing a Slave address of 0000000b, and the R/$\overline{\text{W}}$ bit is logic 0. All Slave devices capable of responding to this broadcast message will acknowledge the GC simultaneously and then behave as a Slave-Receiver. The next byte transmitted by the Master will be accepted and acknowledged by all Slaves capable of handling the special data bytes. A Slave that cannot handle one of these data bytes must ignore it by not acknowledging it. The I^{2}C specification lists the possible meanings of the special bytes that follow the first GC address byte, and the actions to be taken by the Slave device(s) upon receiving them. A common use of the GC by a Master is to dynamically assign device addresses to Slave devices on the bus capable of a programmable device address.

The UPSD34xx can generate a GC as a Master-Transmitter, and it can receive a GC as a Slave. When receiving a GC address (00h), an interrupt will be generated so firmware may respond to the special GC data bytes if desired.

23.7 Serial I/O engine (SIOE)

At the heart of the I^{2}C interface is the hardware SIOE, shown in *Figure 42*. The SIOE automatically handles low-level I^{2}C bus protocol (data shifting, handshaking, arbitration, clock generation and synchronization) and it is controlled and monitored by five SFRs.

The five SFRs shown in *Figure 42* are:

- S1CON - Interface control (*Table 75 on page 129*)
- S1STA - Interface status (*Table 78 on page 130*)
- S1DAT - Data shift register (*Table 80 on page 132*)
- S1ADR - Device address (*Table 82 on page 132*)
- S1SETUP - Sampling rate (*Table 84 on page 133*)

- USB IN FIFO interrupt flag (UIF1)

  The USB IN FIFO Interrupt Flag register (see *Table 114*) contains flags that indicate when an IN Endpoint FIFO that was full becomes empty. Once set, firmware must clear the flag by writing a '0' to the appropriate bit. When FIFOs are paired, only the odd numbered FIFO Interrupt flags are active.

Table 114. USB IN FIFO interrupt flag (UIF1 0E9h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	IN4F	IN3F	IN2F	IN1F	IN0F

Table 115. UIF1 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	IN4F	R/W	Endpoint 4 IN FIFO Interrupt flag This bit is set when the FIFO status changes from full to empty.
3	IN3F	R/W	Endpoint 3 IN FIFO Interrupt flag This bit is set when the FIFO status changes from full to empty.
2	IN2F	R/W	Endpoint 2 IN FIFO Interrupt flag This bit is set when the FIFO status changes from full to empty.
1	IN1F	R/W	Endpoint 1 IN FIFO Interrupt flag This bit is set when the FIFO status changes from full to empty.
0	IN0F	R/W	Endpoint 0 IN FIFO Interrupt flag This bit is set when the FIFO status changes from full to empty.

Table 139. ACON register bit definition (continued)

Bit	Symbol	Function
1	ADST	ADC start bit 0 = Force to zero 1 = Start ADC, then after one cycle, the bit is cleared to '0.'
0	ADSF	ADC Status Bit 0 = ADC conversion is not completed 1 = ADC conversion is completed. The bit can also be cleared with software.

Table 140. ADCPS register details (SFR 94h, Reset Value 00h)

Bit	Symbol	Function
7:4	–	Reserved
3	ADCCE	ADC conversion reference clock enable 0 = ADC reference clock is disabled (default) 1 = ADC reference clock is enabled
2:0	ADCPS[2:0]	ADC reference clock prescaler Only three Prescaler values are allowed: ADCPS[2:0] = 0, for f_{OSC} frequency 16MHz or less. Resulting ADC clock is f_{OSC}. ADCPS[2:0] = 1, for f_{OSC} frequency 32MHz or less. Resulting ADC clock is $f_{OSC}/2$. ADCPS[2:0] = 2, for f_{OSC} frequency 32MHz > 40MHz. Resulting ADC clock is $f_{OSC}/4$.

Table 141. ADAT0 register (SFR 95h, reset value 00h)

Bit	Symbol	Function
7:0	–	Store ADC output, Bit 7 - 0

Table 142. ADAT1 register (SFR 96h, reset value 00h)

Bit	Symbol	Function
7:2	–	Reserved
1.. 0	–	Store ADC output, Bit 9, 8

28.5.11 Data polling

Polling on the Data Polling Flag Bit (DQ7) is a method of checking whether a program or erase operation is in progress or has completed. *Figure 71* shows the Data Polling algorithm.

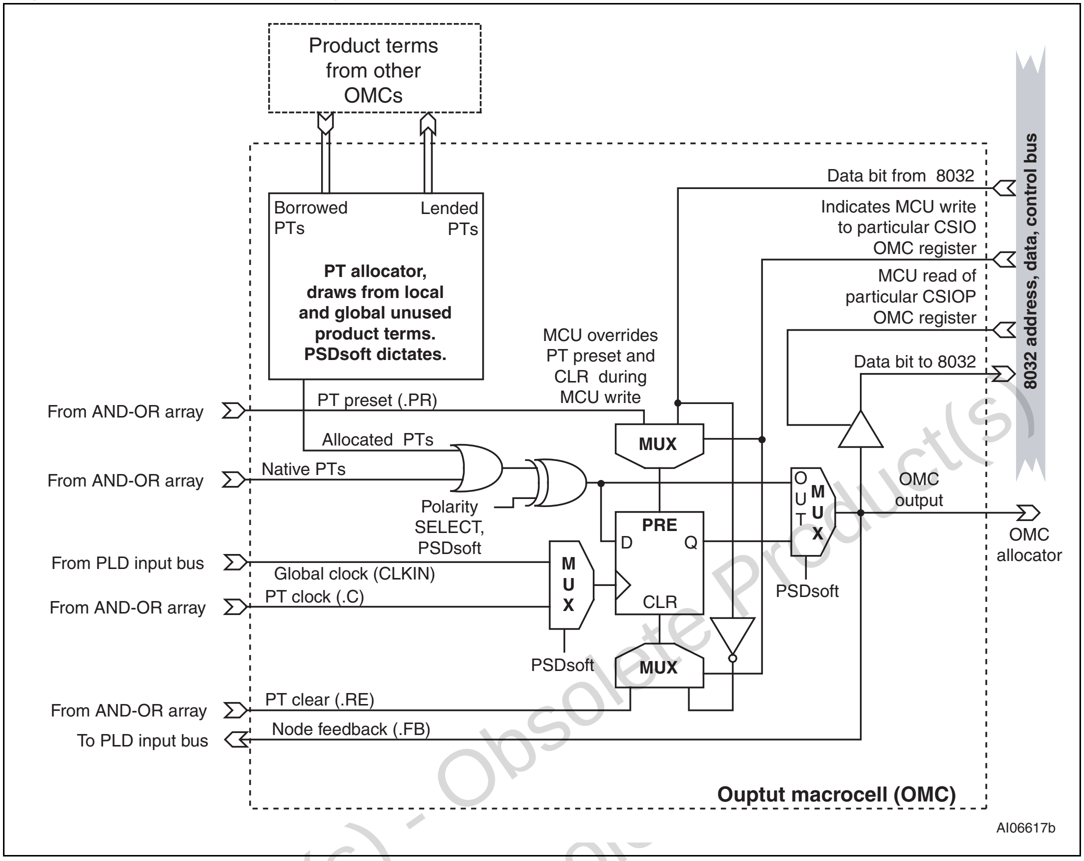
When the 8032 issues a program instruction sequence, the embedded algorithm within the Flash memory array begins. The 8032 then reads the location of the byte to be programmed in Flash memory to check status. The Data Polling Flag Bit (DQ7) of this location becomes the compliment of Bit D7 of the original data byte to be programmed. The 8032 continues to poll this location, comparing the Data Polling Flag Bit (DQ7) and monitoring the Error Flag Bit (DQ5). When the Data Polling Flag Bit (DQ7) matches Bit D7 of the original data, then the embedded algorithm is complete. If the Error Flag Bit (DQ5) is '1,' the 8032 should test the Data Polling Flag Bit (DQ7) again since the Data Polling Flag Bit (DQ7) may have changed simultaneously with the Error Flag Bit (DQ5) (see *Figure 71*).

The Error Flag Bit (DQ5) is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte (indicating a bad Flash cell) or if the 8032 attempted to program bit to logic ’1’ when that bit was already programmed to logic ’0’ (must erase to achieve logic ’1’).

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte that was written to the Flash memory with the byte that was intended to be written.

When using the Data Polling method during an erase operation, *Figure 71* still applies. However, the Data Polling Flag Bit (DQ7) is '0' until the erase operation is complete. A ’1’ on the Error Flag Bit (DQ5) indicates a time-out condition on the Erase cycle, a ’0’ indicates no error. The 8032 can read any location within the sector being erased to get the Data Polling Flag Bit (DQ7) and the Error Flag Bit (DQ5).

Figure 76. Detail of a single OMC



28.5.30 OMC allocator

Outputs of the 16 OMCs can be routed to a combination of pins on Port A (80-pin devices only), Port B, or Port C as shown in *Figure 77*. OMCs are routed to port pins automatically after specifying pin numbers in PSDsoft Express. Routing can occur on a bit-by-bit basis, spitting OMC assignment between the ports. However, one OMC can be routed to one only port pin, not both ports.

28.5.31 Product term allocator

Each OMC has a Product Term Allocator as shown in *Figure 76 on page 226*. PSDsoft Express uses PT Allocators to give and take product terms to and from other OMCs to fit a logic design into the available silicon resources. This happens automatically in PSDsoft Express, but understanding how PT allocation works will help if the logic design does not “fit”, in which case the user may try selecting a different pin or different OMC for the logic where more product terms may be available. The following list summarizes how product terms are allocated to each OMC, as shown in *Table 168 on page 227*.

- MCELLAB0-MCELLAB7 each have three native product terms and may borrow up to six more
- MCELLBC0-MCELLBC3 each have four native product terms and may borrow up to five more
- MCELLBC4-MCELLBC7 each have four native product terms and may borrow up to six more.

28.5.35 I/O ports

There are four programmable I/O ports on the PSD module: Port A (80-pin device only), Port B, Port C, and Port D. Ports A and B are eight bits each, Port C is four bits, and Port D is two bits for 80-pin devices or 1-bit for 52-pin devices. Each port pin is individually configurable, thus allowing multiple functions per port. The ports are configured using PSDsoft Express then programming with JTAG, and also by the 8032 writing to csiop registers at run-time.

Topics discussed in this section are:

- General port architecture
- Port operating modes
- Individual port structure

28.5.36 General port architecture

The general architecture for a single I/O Port pin is shown in *Figure 79 on page 232*. Port structures for Ports A, B, C, and D differ slightly and are shown in *Figure 84 on page 243* though *Figure 87 on page 247*.

Figure 79 on page 232 shows four csiop registers whose outputs are determined by the value that the 8032 writes to csiop Direction, Drive, Control, and Data Out. The I/O Port logic contains an output mux whose mux select signal is determined by PSDsoft Express and the csiop Control register bits at run-time. Inputs to this output mux include the following:

1. Data from the csiop Data Out register for MCU I/O output mode (All ports)
2. Latched de-multiplexed 8032 Address for Address Output mode (Ports A and B only)
3. Peripheral I/O mode data bit (Port A only)
4. GPLD OMC output (Ports A, B, and C).

The Port Data Buffer (PDB) provides feedback to the 8032 and allows only one source at a time to be read when the 8032 reads various csiop registers. There is one PDB for each port pin enabling the 8032 to read the following on a pin-by-pin basis:

1. MCU I/O signal direction setting (csiop Direction reg)
2. Pin drive type setting (csiop Drive Select reg)
3. Latched Addr Out mode setting (csiop Control reg)
4. MCU I/O pin output setting (csiop Data Out reg)
5. Output Enable of pin driver (csiop Enable Out reg)
6. MCU I/O pin input (csiop Data In reg)

A port pin's output enable signal is controlled by a two input OR gate whose inputs come from: a product term of the AND-OR array; the output of the csiop direction register. If an output enable from the AND-OR Array is not defined, and the port pin is not defined as an OMC output, and if Peripheral I/O mode is not used, then the csiop direction register has sole control of the OE signal.

As shown in *Figure 79 on page 232*, a physical port pin is connected to the I/O Port logic and is also separately routed to an IMC, allowing the 8032 to read a port pin by two different methods (MCU I/O input mode or read the IMC).

28.5.37 Port operating modes

I/O Port logic has several modes of operation. *Table 171 on page 229* summarizes which modes are available on each port. Each of the port operating modes are described in assembly code example in *Table* , the PFQ will be loaded with the final instructions to command the MCU module to Power Down mode after the PDS Module goes to Power-Down mode. In this case, even though the code memory goes off-line in the PSD module, the last few MCU instruction are sourced from the PFQ.

Forced power-down example

```
PDOWN:  ANL     A8h, #7Fh    ; disable all interrupts
        ORL     9Dh, #C0h    ; ensure PFQ and BC are enabled
        MOV     DPTR, #xxC7  ; load XDATA pointer to select PMMR3 register (xx = base
                             ; address of csiop registers)
        CLR     A            ; clear A
        JMP     LOOP         ; first loop - fill PFQ/BQ with Power Down instructions
        NOP                  ; second loop - fetch code from PFQ/BC and set Power-
                             ; Down bits for PSD module and then MCU module
LOOP:   MOVX    @DPTR, A     ; set FORCE_PD Bit in PMMR3 in PSD module in second
                             ; loop
        MOV     87h, A       ; set PD Bit in PCON register in MCU module in second
                             ; loop
        MOV     A, #02h      ; set power-down bit in the A register, but not in PMMR3 or
                             ; PCON yet in first loop
        JMP     LOOP         ; uPSD enters into Power-Down mode in second loop
```

Figure 88. Automatic power-down (APD) unit

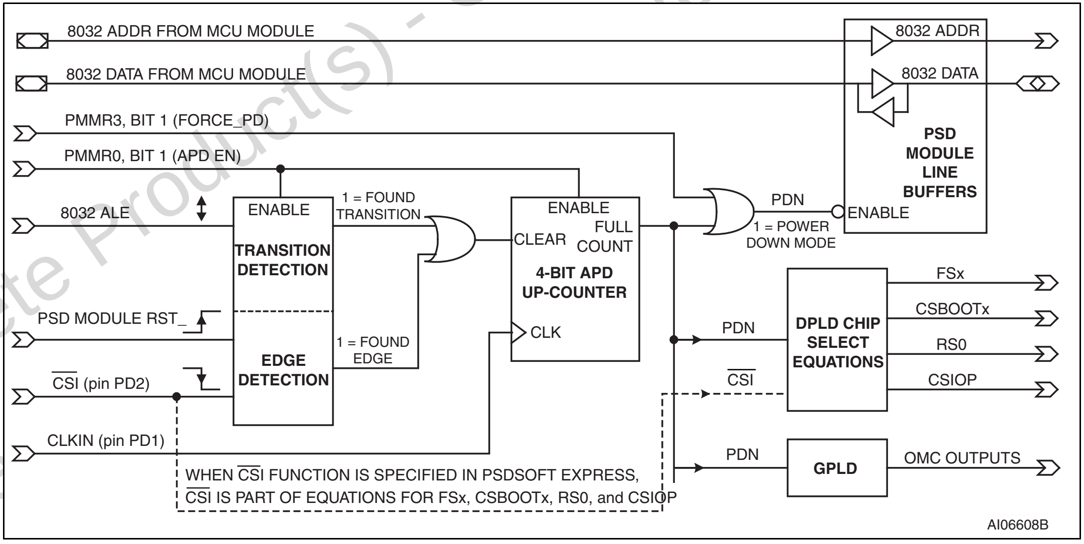
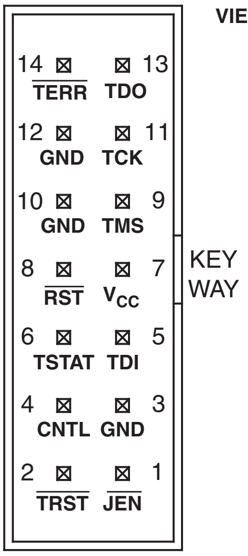


Figure 93. Recommended JTAG connector


14 ☒ ☒ 13
$\overline{\text{TERR}}$ TDO
12 ☒ ☒ 11
GND TCK
10 ☒ ☒ 9
GND TMS
8 ☒ ☒ 7
$\overline{\text{RST}}$ V_{CC}
6 ☒ ☒ 5
TSTAT TDI
4 ☒ ☒ 3
CNTL GND
2 ☒ ☒ 1
$\overline{\text{TRST}}$ $\overline{\text{JEN}}$
KEY WAY

VIEW: Looking into face of shrouded male connector, with 0.025" posts on 0.1" centers.

Connector reference:
Molex 70247-1401

This connector accepts a 14-pin ribbon cable such as:

- Samtec: HCSD-07-D-06.00-01-S-N
- Digikey: M3CCK-14065-ND

AI09187

28.6.7 Chaining UPSD34xx devices

It is possible to chain a UPSD34xx device with other UPSD34xx devices on a circuit board, and also chain with IEEE 1149.1 compliant devices from other manufacturers. *Figure 94 on page 263* shows a chaining example. The TDO of one device connects to the TDI of the next device, and so on. Only one device is performing JTAG operations at any given time while the other two devices are in BYPASS mode. Configuration for JTAG chaining can be made in PSDsoft Express by choosing “More than one device” when prompted about chaining devices. Notice in *Figure 94 on page 263* that the UPSD34xx devices are chained externally, but also be aware that the two die within each UPSD34xx device are chained internally. This internal chaining of die is transparent to the user and is taken care of by PSDsoft Express and 3rd party JTAG tool software.

The example in *Figure 94 on page 263* also shows how to use 6-pin JTAG when chaining devices. The signals TSTAT and $\overline{\text{TERR}}$ are configured as open-drain type signals from PSDsoft Express. This facilitates a wired-OR connection of TSTAT signals from multiple UPSD34xx devices and also a wired-OR connection of $\overline{\text{TERR}}$ signals from those same multiple devices. PSDsoft Express puts TSTAT and $\overline{\text{TERR}}$ signals into open-drain mode by default, requiring external pull-up resistors. Click on 'Properties' in the JTAG-ISP window of PSDsoft Express to change to standard CMOS push-pull outputs if desired, but wired-OR logic is not possible in CMOS output mode.

30 Maximum rating

Stressing the device above the rating listed in the absolute maximum ratings table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the operating sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 205. Absolute maximum ratings

Symbol	Parameter	Min.	Max.	Unit
T_{STG}	Storage temperature	–65	125	°C
T_{LEAD}	Lead temperature during soldering (20 seconds max.)[1]		235	°C
V_{IO}	Input and output voltage (Q = V_{OH} or Hi-Z)	–0.5	6.5	V
V_{CC}, V_{DD}, AV_{CC}	Supply voltage	–0.5	6.5	V
V_{ESD}	Electrostatic discharge voltage (human body model)[2]	–2000	2000	V

1. IPC/JEDEC J-STD-020A.
2. *JEDEC Std JESD22-A114A (C1=100pF, R1=1500 Ω, R2=500 Ω).*

Table 220. CPLD combinatorial timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo off	Slew rate[1]	Unit
t_{PD}[2]	CPLD input pin/feedback to CPLD combinatorial output			20	+ 2	+ 10	– 2	ns
t_{EA}	CPLD input to CPLD Output Enable			21		+ 10	– 2	ns
t_{ER}	CPLD input to CPLD Output Disable			21		+ 10	– 2	ns
t_{ARP}	CPLD register clear or preset delay			21		+ 10	– 2	ns
t_{ARPW}	CPLD register clear or preset pulse width		10			+ 10		ns
t_{ARD}	CPLD array delay	Any macrocell		11	+ 2			ns

1. Fast Slew Rate output available on PA3-PA0, PB3-PB0, and PD2-PD1. Decrement times by given amount.
2. t_{PD} for MCU address and control signals refers to delay from pins on Port 0, Port 2, $\overline{RD}$ $\overline{WR}$, $\overline{PSEN}$ and ALE to CPLD combinatorial output (80-pin package only).

Table 221. CPLD combinatorial timing (3 V PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo off	Slew rate[1]	Unit
t_{PD}[2]	CPLD input pin/feedback to CPLD combinatorial output			35	+ 4	+ 15	– 6	ns
t_{EA}	CPLD input to CPLD Output Enable			38		+ 15	– 6	ns
t_{ER}	CPLD input to CPLD Output Disable			38		+ 15	– 6	ns
t_{ARP}	CPLD register clear or preset delay			35		+ 15	– 6	ns
t_{ARPW}	CPLD register clear or preset pulse width		18			+ 15		ns
t_{ARD}	CPLD array delay	Any macrocell		20	+ 4			ns

1. Fast Slew Rate output available on PA3-PA0, PB3-PB0, and PD2-PD1. Decrement times by given amount.
2. t_{PD} for MCU address and control signals refers to delay from pins on Port 0, Port 2, $\overline{RD}$ $\overline{WR}$, $\overline{PSEN}$ and ALE to CPLD combinatorial output (80-pin package only).

Table 230. Port A peripheral data mode READ timing (3 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Turbo off	Unit
$t_{AVQV–PA}$	Address valid to data valid	(1)		50	+ 15	ns
$t_{SLQV–PA}$	$\overline{\text{CSI}}$ valid to data valid			37	+ 15	ns
$t_{RLQV–PA}$	$\overline{\text{RD}}$ to data valid	(2)		45		ns
$t_{DVQV–PA}$	Data in to data out valid			38		ns
$t_{RHQZ–PA}$	$\overline{\text{RD}}$ to data High-Z			36		ns

1. Any input used to select Port A data peripheral mode.
2. Data is already stable on Port A.

Figure 106. Peripheral I/O WRITE timing

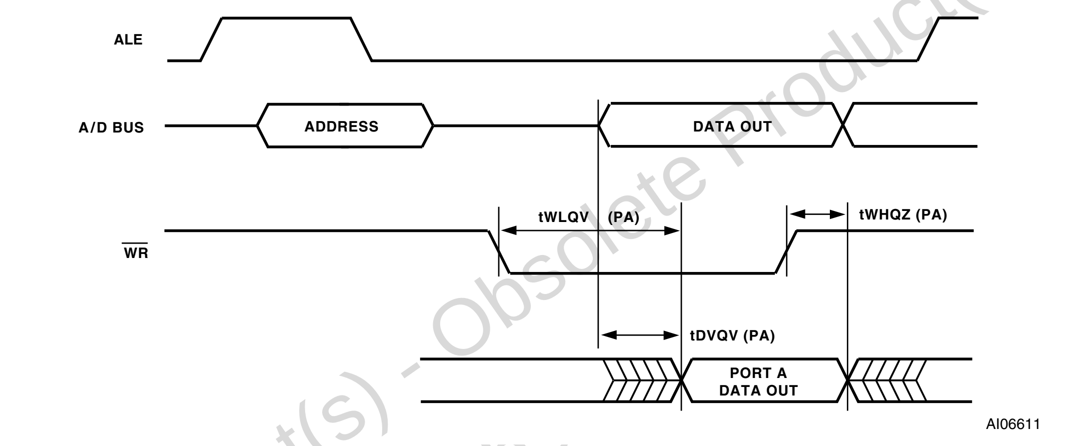


Table 231. Port A peripheral data mode WRITE timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV–PA}$	$\overline{\text{WR}}$ to data propagation delay			25	ns
$t_{DVQV–PA}$	Data to port A data propagation delay	(1)		22	ns
$t_{WHQZ–PA}$	$\overline{\text{WR}}$ Invalid to port a tri-state			20	ns

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

Table 232. Port A peripheral data mode WRITE timing (3 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV–PA}$	$\overline{\text{WR}}$ to data propagation delay			42	ns
$t_{DVQV–PA}$	Data to port A data propagation delay	(1)		38	ns
$t_{WHQZ–PA}$	$\overline{\text{WR}}$ Invalid to port a tri-state			33	ns

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