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	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
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/upsd3454eb40u6

Table 2. Pin definitions

Port pin	Signal name	80-pin No.	52-pin No.(1)	In/out	Function		
					Basic	Alternate 1	Alternate 2
MCUAD0	AD0	36	N/A	I/O	External bus multiplexed address/data bus A0/D0		
MCUAD1	AD1	37	N/A	I/O	Multiplexed address/data bus A1/D1		
MCUAD2	AD2	38	N/A	I/O	Multiplexed address/data bus A2/D2		
MCUAD3	AD3	39	N/A	I/O	Multiplexed address/data bus A3/D3		
MCUAD4	AD4	41	N/A	I/O	Multiplexed address/data bus A4/D4		
MCUAD5	AD5	43	N/A	I/O	Multiplexed address/data bus A5/D5		
MCUAD6	AD6	45	N/A	I/O	Multiplexed address/data bus A6/D6		
MCUAD7	AD7	47	N/A	I/O	Multiplexed address/data bus A7/D7		
P1.0	T2 ADC0	52	34	I/O	General I/O port pin	Timer 2 Count input (T2)	ADC Channel 0 input (ADC0)
P1.1	T2X ADC1	54	35	I/O	General I/O port pin	Timer 2 Trigger input (T2X)	ADC Channel 1 input (ADC1)
P1.2	RxD1 ADC2	56	36	I/O	General I/O port pin	UART1 or IrDA Receive (RxD1)	ADC Channel 2 input (ADC2)
P1.3	TXD1 ADC3	58	37	I/O	General I/O port pin	UART or IrDA Transmit (TxD1)	ADC Channel 3 input (ADC3)
P1.4	SPICLK ADC4	59	38	I/O	General I/O port pin	SPI Clock Out (SPICLK)	ADC Channel 4 input (ADC4)
P1.5	SPIRxD ADC5	60	39	I/O	General I/O port pin	SPI Receive (SPIRxD)	ADC Channel 5 input (ADC5)
P1.6	SPITXD ADC6	61	40	I/O	General I/O port pin	SPI Transmit (SPITxD)	ADC Channel 6 input (ADC6)
P1.7	SPISEL ADC7	64	41	I/O	General I/O port pin	SPI Slave Select (SPISEL)	ADC Channel 7 input (ADC7)
P3.0	RxD0	75	23	I/O	General I/O port pin	UART0 Receive (RxD0)	

4.2.1 Program memory

External program memory is addressed by the 8032 using its 16-bit program counter (PC) and is accessed with the 8032 signal, $\overline{PSEN}$. Program memory can be present at any address in program space between 0000h and FFFFh.

After a power-up or reset, the 8032 begins program execution from location 0000h where the reset vector is stored, causing a jump to an initialization routine in firmware. At address 0003h, just following the reset vector are the interrupt service locations. Each interrupt is assigned a fixed interrupt service location in program memory. An interrupt causes the 8032 to jump to that service location, where it commences execution of the service routine. External Interrupt 0 (EXINT0), for example, is assigned to service location 0003h. If EXINT0 is going to be used, its service routine must begin at location 0003h. Interrupt service locations are spaced at 8-byte intervals: 0003h for EXINT0, 000Bh for Timer 0, 0013h for EXINT1, and so forth. If an interrupt service routine is short enough, it can reside entirely within the 8-byte interval. Longer service routines can use a jump instruction to somewhere else in program memory.

4.2.2 Data memory

External data is referred to as XDATA and is addressed by the 8032 using Indirect Addressing via its 16-bit data pointer register (DPTR) and is accessed by the 8032 signals, $\overline{RD}$ and $\overline{WR}$. XDATA can be present at any address in data space between 0000h and FFFFh.

Note: The UPSD34xx has dual data pointers (source and destination) making XDATA transfers much more efficient.

4.2.3 Memory placement

PSD module architecture allows the placement of its external memories into different combinations of program memory and data memory spaces. This means the main Flash, the secondary Flash, and the SRAM can be viewed by the 8032 MCU in various combinations of program memory or data memory as defined by PSDsoft Express.

As an example of this flexibility, for applications that require a great deal of Flash memory in data space (large lookup tables or extended data recording), the larger main Flash memory can be placed in data space and the smaller secondary Flash memory can be placed in program space. The opposite can be realized for a different application if more Flash memory is needed for code and less Flash memory for data.
By default, the SRAM and csiop memories on the PSD module must always reside in data memory space and they are treated by the 8032 as XDATA.
The main Flash and secondary Flash memories may reside in program space, data space, or both. These memory placement choices specified by PSDsoft Express are programmed into non-volatile sections of the UPSD34xx, and are active at power-up and after reset. It is possible to override these initial settings during runtime for In-Application Programming (IAP).

Standard 8032 MCU architecture cannot write to its own program memory space to prevent accidental corruption of firmware. However, this becomes an obstacle in typical 8032 systems when a remote update to firmware in Flash memory is required using IAP. The PSD module provides a solution for remote updates by allowing 8032 firmware to temporarily “reclassify” Flash memory to reside in data space during a remote update, then returning Flash memory back to program space when finished. See the VM register (*Table 160 on page 203*) in the PSD module section of this document for more details.

four MCU clocks). But it is also important to understand PFQ operation on multi-cycle instructions.

5.2 PFQ example, multi-cycle instructions

Let us look at a string of two-byte, two-cycle instructions in *Figure 8 on page 35*. There are three instructions executed sequentially in this example, instructions A, B, and C. Each of the time divisions in the figure is one machine-cycle of four clocks, and there are six phases to reference in this discussion. Each instruction is pre-fetched into the PFQ in advance of execution by the MCU. Prior to Phase 1, the PFQ has pre-fetched the two instruction bytes (A1 and A2) of Instruction A. During Phase one, both bytes are loaded into the MCU execution unit. Also in Phase 1, the PFQ is pre-fetching Instruction B (bytes B1 and B2) from program memory. In Phase 2, the MCU is processing Instruction A internally while the PFQ is pre-fetching Instruction C. In Phase 3, both bytes of instruction B are loaded into the MCU execution unit and the PFQ begins to pre-fetch bytes for the next instruction. In Phase 4 Instruction B is processed.

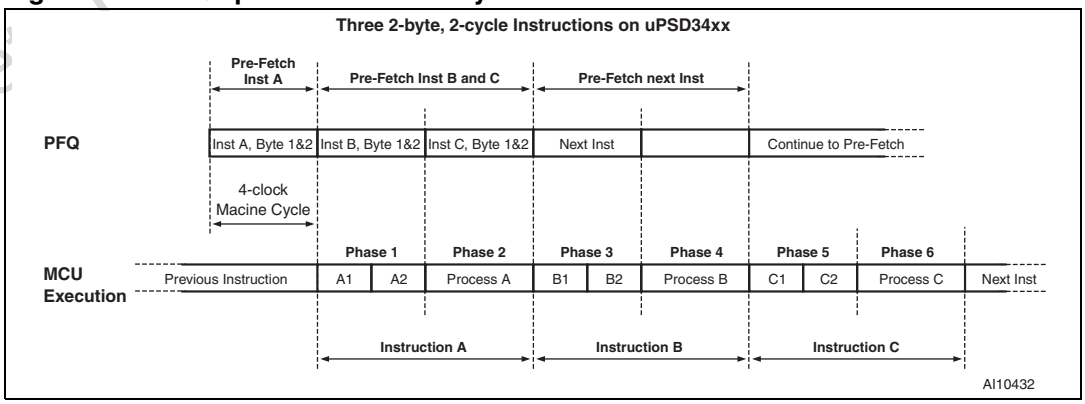
The UPSD34xx MCU instructions are an exact 1/3 scale of all standard 8032 instructions with regard to number of cycles per instruction. *Figure 9 on page 36* shows the equivalent instruction sequence from the example above on a standard 8032 for comparison.

5.3 Aggregate performance

The stream of two-byte, two-cycle instructions in *Figure 8 on page 35*, running on a 40 MHz, 5 V, UPSD34xx will yield 5 MIPs. And we saw the stream of one- or two-byte, one-cycle instructions in *Figure 6 on page 33*, on the same MCU yield 10 MIPs. Effective performance will depend on a number of things: the MCU clock frequency; the mixture of instructions types (bytes and cycles) in the application; the amount of time an empty PFQ stalls the MCU (mix of instruction types and misses on Branch Cache); and the operating voltage. A 5 V UPSD34xx device operates with four memory wait states, but a 3.3 V device operates with five memory wait states yielding 8 MIPS peak compared to 10 MIPs peak for 5 V device. The same number of wait states will apply to both program fetches and to data READ/WRITEs unless otherwise specified in the SFR named BUSCON.

In general, a 3X aggregate performance increase is expected over any standard 8032 application running at the same clock frequency.

Figure 8. PFQ operation on multi-cycle instructions



7.4 Accumulator (ACC)

This is an 8-bit general purpose register which holds a source operand and receives the result of arithmetic operations. The ACC register can also be the source or destination of logic and data movement operations. For MUL and DIV instructions, ACC is combined with the B register to hold 16-bit operands. The ACC is referred to as “A” in the MCU instruction set.

7.5 B register (B)

The B register is a general purpose 8-bit register for temporary data storage and also used as a 16-bit register when concatenated with the ACC register for use with MUL and DIV instructions.

7.6 General purpose registers (R0 - R7)

There are four banks of eight general purpose 8-bit registers (R0 - R7), but only one bank of eight registers is active at any given time depending on the setting in the PSW word (described next). R0 - R7 are generally used to assist in manipulating values and moving data from one memory location to another. These register banks physically reside in the first 32 locations of 8032 internal DATA SRAM, starting at address 00h. At reset, only the first bank of eight registers is active (addresses 00h to 07h), and the stack begins at address 08h.

7.7 Program status word (PSW)

The PSW is an 8-bit register which stores several important bits, or flags, that are set and cleared by many 8032 instructions, reflecting the current state of the MCU core. *Figure 11 on page 40* shows the individual flags.

7.7.1 Carry flag (CY)

This flag is set when the last arithmetic operation that was executed results in a carry (addition) or borrow (subtraction). It is cleared by all other arithmetic operations. The CY flag is also affected by Shift and Rotate Instructions.

7.7.2 Auxiliary carry flag (AC)

This flag is set when the last arithmetic operation that was executed results in a carry into (addition) or borrow from (subtraction) the high-order nibble. It is cleared by all other arithmetic operations.

7.7.3 General purpose flag (F0)

This is a bit-addressable, general-purpose flag for use under software control.

7.7.4 Register bank select flags (RS1, RS0)

These bits select which bank of eight registers is used during R0 - R7 register accesses (see *Table 4*)

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

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		
B0(1)	P3	P3.7 <B7h>	P3.6 <B6h>	P3.5 <B5h>	P3.4 <B4h>	P3.3 <B3h>	P3.2 <B2h>	P3.1 <B1h>	P3.0 <B0h>	FF	Table 37
B1	CAPCOMH1	CAPCOMH1[7:0]								00	Table 143
B2	CAPCOML2	CAPCOML2[7:0]								00	
B3	CAPCOMH2	CAPCOMH2[7:0]								00	
B4	PWMF0	PWMF0[7:0]								00	
B5	RESERVED										
B6	RESERVED										
B7	IPA	PADC	PSPI	PPCA	PS1	–	–	PI2C	–	00	Table 24
B8(1)	IP	–	–	PT2 <BDh>	PS0 <BCh>	PT1 <BBh>	PX1 <BAh>	PT0 <B9h>	PX0 <B8h>	00	Table 22
B9	RESERVED										
BA	PCACL1	PCACL1[7:0]								00	Table 143
BB	PCACH1	PCACH1[7:0]								00	
BC	PCACON1	–	EN_PCA	EOVF1	PCA_IDL	–	–	CLK_SEL[1:0]		00	Table 150
BD	TCMMODE3	EINTF	E_COMP	CAP_PE	CAP_NE	MATCH	TOGGLE	PWM[1:0]		00	Table 154
BE	TCMMODE4	EINTF	E_COMP	CAP_PE	CAP_NE	MATCH	TOGGLE	PWM[1:0]		00	
BF	TCMMODE5	EINTF	E_COMP	CAP_PE	CAP_NE	MATCH	TOGGLE	PWM[1:0]		00	
C0(1)	P4	P4.7 <C7h>	P4.6 <C6h>	P4.5 <C5h>	P4.4 <C4h>	P4.3 <C3h>	P4.2 <C2h>	P4.1 <C1h>	P4.0 <C0h>	FF	Table 39

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

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		
DD	S1STA	GC	STOP	INTR	TX_MD	B_BUSY	B_LOST	ACK_R	SLV	00	*Table 78*
DE	S1DAT	S1DAT[7:0]								00	*Table 80*
DF	S1ADR	S1ADR[7:0]								00	*Table 82*
E0(1)	A	A[7:0] <bit addresses: E7h, E6h, E5h, E4h, E3h, E2h, E1h, E0h>								00	*Section 7.4*
E1	RESERVED										
E2	UADDR	–	USBADDR[6:0]							00	
E3	UPAIR	–	–	–	–	PR3OUT	PR1OUT	PR3IN	PR1IN	00	
E4	UIE0	–	–	–	–	RSTIE	SUSPNDIE	EOPIE	RESUMIE	00	
E5	UIE1	–	–	–	IN4IE	IN3IE	IN2IE	IN1IE	IN0IE	00	
E6	UIE2	–	–	–	OUT4IE	OUT3IE	OUT2IE	OUT1IE	OUT0IE	00	
E7	UIE3	–	–	–	NAK4IE	NAK3IE	NAK2IE	NAK1IE	NAK0IE	00	
E8	UIF0	GLF	INF	OUTF	NAKF	RSTF	SUSPNDF	EOPF	RESUMF	00	
E9	UIF1	–	–	–	IN4F	IN3F	IN2F	IN1F	IN0F	00	
EA	UIF2	–	–	–	OUT4F	OUT3F	OUT2F	OUT1F	OUT0F	00	
EB	UIF3	–	–	–	NAK4F	NAK3F	NAK2F	NAK1F	NAK0F	00	
EC	UCTL	–	–	–	–	–	USBEN	VISIBLE	WAKEUP	00	
ED	USTA	–	–	–	–	RCVT	SETUP	IN	OUT	00	
EE	RESERVED										
EF	USEL	DIR	–	–	–	–	EP[2:0]			00	
F0(1)	B	B[7:0] <bit addresses: F7h, F6h, F5h, F4h, F3h, F2h, F1h, F0h>								00	*Section 7.5*
F1	UCON	–	–	–	–	ENABLE	STALL	TOGGLE	BSY	08	
F2	USIZE	–	SIZE[6:0]							00	
F3	UBASEH	BASEADDR[15:8]								00	
F4	UBASEL	BASEADDR[7:6]		0	0	0	0	0	0	00	
F5	USCI	–	–	–	–	–	USCI[2:0]			00	
F6	USCV	USCV[7:0]								00	
F7	RESERVED										
F8	RESERVED										

Table 36. P1 register bit definition (continued)

Bit	Symbol	R/W	Function(1)
2	P1.2	R,W	Port pin 1.2
1	P1.1	R,W	Port pin 1.1
0	P1.0	R,W	Port pin 1.0

1. Write '1' or '0' for pin output. Read for pin input, but prior to READ, this bit must have been set to '1' by firmware or by a reset event.

Table 37. P3: I/O port 3 register (SFR B0h, reset value FFh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Table 38. P3 register bit definition

Bit	Symbol	R/W	Function(1)
7	P3.7	R,W	Port pin 3.7
6	P3.6	R,W	Port pin 3.6
5	P3.5	R,W	Port pin 3.5
4	P3.4	R,W	Port pin 3.4
3	P3.3	R,W	Port pin 3.3
2	P3.2	R,W	Port pin 3.2
1	P3.1	R,W	Port pin 3.1
0	P3.0	R,W	Port pin 3.0

1. Write '1' or '0' for pin output. Read for pin input, but prior to READ, this bit must have been set to '1' by firmware or by a reset event.

Table 39. P4: I/O port 4 register (SFR C0h, reset value FFh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0

Table 40. P4 register bit definition

Bit	Symbol	R/W	Function(1)
7	P4.7	R,W	Port pin 4.7
6	P4.6	R,W	Port pin 4.6
5	P4.5	R,W	Port pin 4.5
4	P4.4	R,W	Port pin 4.4
3	P4.3	R,W	Port pin 4.3
2	P4.2	R,W	Port pin 4.2
1	P4.1	R,W	Port pin 4.1
0	P4.0	R,W	Port pin 4.0

1. Write '1' or '0' for pin output. Read for pin input, but prior to READ, this bit must have been set to '1' by firmware or by a reset event.

Figure 24. Timer/counter mode 0: 13-bit counter

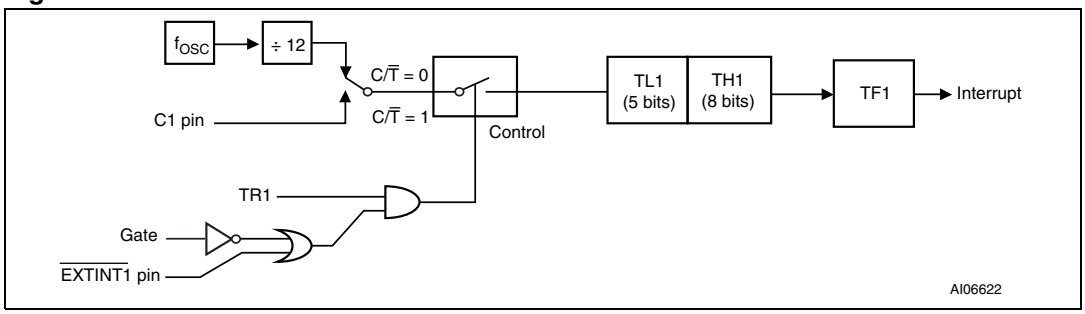


Figure 25. Timer/counter mode 2: 8-bit Auto-reload

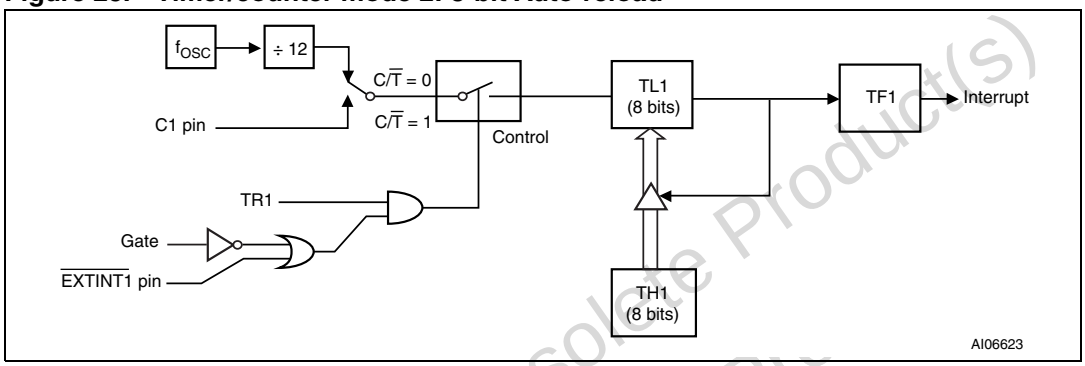
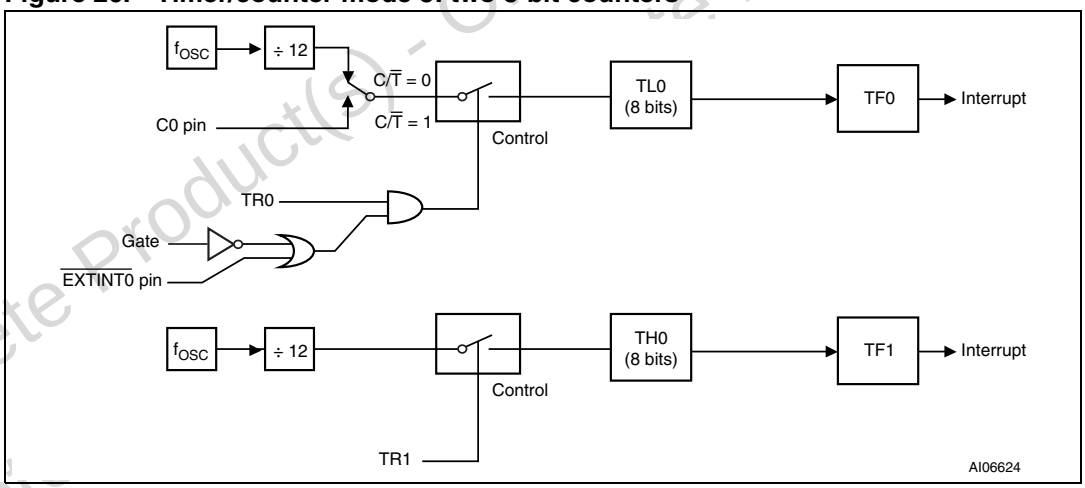


Figure 26. Timer/counter mode 3: two 8-bit counters



20.6 Timer 2

Timer 2 can operate as either an event timer or as an event counter. This is selected by the bit C/$\overline{T2}$ in the SFR named, T2CON (*Table 60 on page 101*). Timer 2 has three operating modes selected by bits in T2CON, according to *Table 62 on page 102*. The three modes are:

- Capture mode
- Auto re-load mode
- Baud rate generator mode

Table 109. UIE2 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	OUT4IE	R/W	Enable Endpoint 4 OUT FIFO interrupt
3	OUT3IE	R/W	Enable Endpoint 3 OUT FIFO interrupt
2	OUT2IE	R/W	Enable Endpoint 2 OUT FIFO interrupt
1	OUT1IE	R/W	Enable Endpoint 1 OUT FIFO interrupt
0	OUT0IE	R/W	Enable Endpoint 0 OUT FIFO interrupt

- USB IN FIFO NAK interrupt enable register (UIE3)

  When an endpoint's IN FIFO is empty and an IN transaction to that endpoint has been received, the SIE sends a NAK handshake token since there is no data ready for it to send.

  The UIE3 register (see *Table 110*) is used to enable each endpoint's IN FIFO NAK Interrupt.

Table 110. USB IN FIFO NAK interrupt enable register (UIE3 0E7h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	NAK4IE	NAK3IE	NAK2IE	NAK1IE	NAK0IE

Table 111. UIE3 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	NAK4IE	R/W	Enable Endpoint 4 IN FIFO NAK interrupt
3	NAK3IE	R/W	Enable Endpoint 3 IN FIFO NAK interrupt
2	NAK2IE	R/W	Enable Endpoint 2 IN FIFO NAK interrupt
1	NAK1IE	R/W	Enable Endpoint 1 IN FIFO NAK interrupt
0	NAK0IE	R/W	Enable Endpoint 0 IN FIFO NAK interrupt

- USB control register (UCTL)

  The USB control register (see *Table 120*) is used to enable the SIE, make the Endpoint FIFOs visible in the XDATA space and for generating a remote wakeup signal. Upon a reset, the USB module is disabled and must be enabled by the CPU for communication with the host over the USB.

Table 120. USB control register (UCTL 0ECh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	USBEN	VISIBLE	WAKEUP

Table 121. UCTL register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	–	–	Reserved
3	–	–	Reserved
2	USBEN	R/W	USB Enable When this bit is set, the USB function is enabled and the SIE responds to tokens from the host. When this bit is clear, the USB function is disabled and does not respond to any tokens from the host. *Note: A USB reset does not clear this bit. Disabling and enabling the SIE using this bit resets part of the USB SIE state machine and some of the bits in the USTA and UCON registers.*
1	VISIBLE	R/W	USB FIFO VISIBLE When this bit is set, the selected USB FIFO is accessible (visible) in the XDATA space.
0	WAKEUP	R/W	Remote Wakeup Enable This bit forces a resume or “K” state on the USB data lines to initiate a remote wake-up. The CPU is responsible for controlling the timing of the forced resume that must be between 10ms and 15ms. Setting this bit will not cause the RESUMF Bit to be set.

- USB setup command index and value registers (USCI and USCV)

  When a Setup/Data packet is received over the USB, the 8 bytes of data received are stored in a command buffer. The USB setup command index register (see *Table 134*) determines which one of the eight bytes in the buffer is read using the USB setup command value register (see *Table 136*).

Table 134. USB setup command index register (USCI 0F5h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	USCI[2:0]		

Table 135. USCI register bit definition

Bit	Symbol	R/W	Definition
7:3	–	–	Reserved
2:0	USCI[2:0]	R/W	Index to access one of the 8 bytes of USB Setup Command Data received with the last Setup transaction

Table 136. USB setup command value register (USCV 0F6h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USCV[7:0]							

Table 137. USCV register bit definition

Bit	Symbol	R/W	Definition
7:0	USCV	R/W	The nth byte of the 8 bytes of USB Setup Command Data received with the last Setup transaction. The nth byte that is read from this register is specified by the index value in the USCI register.

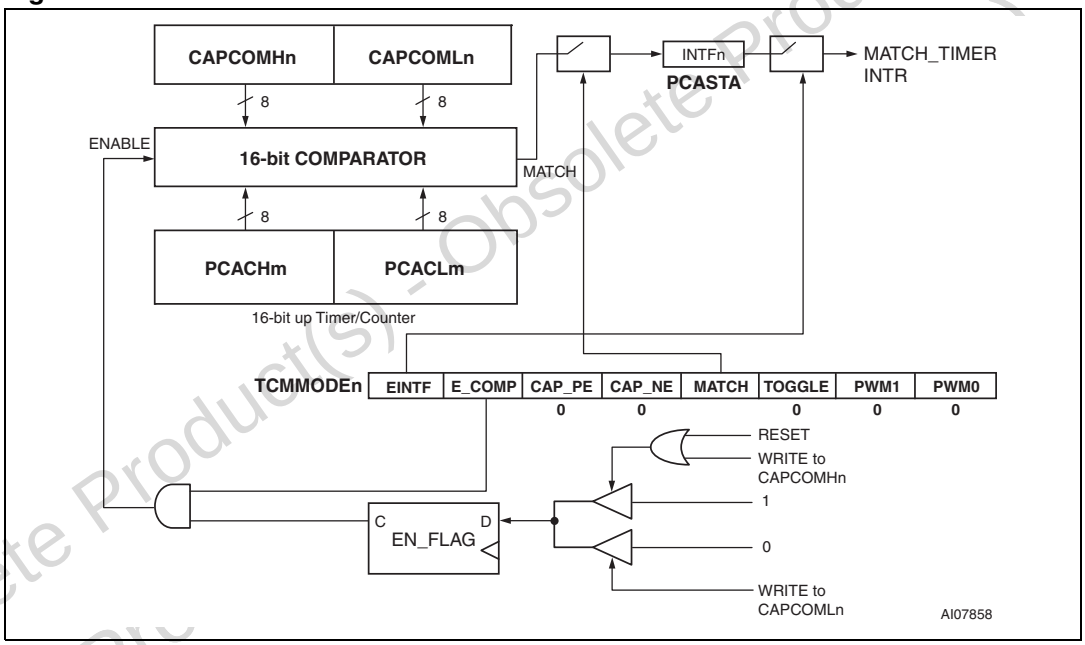
27.6 Toggle mode

In this mode, the user writes a value to the TCM's CAPCOM registers and enables the comparator. When there is a match with the Counter output, the output of the TCM pin toggles. This mode is a simple extension of the Timer Mode.

27.7 PWM mode - (x8), fixed frequency

In this mode, one or all the TCM's can be configured to have a fixed frequency PWM output on the port pins. The PWM frequency depends on when the low byte of the Counter overflows (modulo 256). The duty cycle of each TCM module can be specified in the CAPCOMHn register. When the PCA_Counter_L value is equal to or greater than the value in CAPCOMLn, the PWM output is switched to a high state. When the PCA_Counter_L register overflows, the content in CAPCOMHn is loaded to CAPCOMLn and a new PWM pulse starts.

Figure 58. Timer mode



1. m = 0: n = 0, 1, or 2
   m = 1: n = 3, 4, or 5

Table 163. Flash memory instruction sequences(1)(2) (continued)

Instr. Seq.	Bus Cycle 1	Bus Cycle 2	Bus Cycle 3	Bus Cycle 4	Bus Cycle 5	Bus Cycle 6	Bus Cycle 7	Link
Resume Sector Erase	Write 30h to address that activates FSx or CSBOOTx where desired to resume erase (*command*)							*Resume sector erase on page 217*
Reset Flash	Write F0h to address that activates FSx or CSBOOTx in desired array. (*command*)							*Reset Flash on page 218*

1. All values are in hexadecimal, X = Don't care.
2. 8032 addresses A12 through A15 are "Don't care" during the instruction sequence decoding. Only address bits A0-A11 are used during decoding of Flash memory instruction sequences. The individual sector select signal (FS0 - FS7 or CSBOOT0-CSBOOT3) which is active during the instruction sequence determines the complete address.
3. Directing this command to any individual sector within a Flash memory array will invoke the bulk erase of all Flash memory sectors within that array.

28.5.3 Reading Flash memory

Under typical conditions, the 8032 may read the Flash memory using READ operations (READ bus cycles) just as it would a ROM or RAM device. Alternately, the 8032 may use READ operations to obtain status information about a Program or Erase operation that is currently in progress. The following sections describe the kinds of READ operations.

28.5.4 Read memory contents

Flash memory is placed in the Read Array mode after Power-up, after a PSD module reset event, or after receiving a Reset Flash memory instruction sequence from the 8032. The 8032 can read Flash memory contents using standard READ bus cycles anytime the Flash array is in Read Array mode. Flash memories will always be in Read Array mode when the array is not actively engaged in a program or erase operation.

28.5.5 Reading the erase/program status bits

The Flash arrays provide several status bits to be used by the 8032 to confirm the completion of an erase or program operation on Flash memory, shown in *Table 164 on page 212*. The status bits can be read as many times as needed until an operation is complete.

The 8032 performs a READ operation to obtain these status bits while an erase or program operation is being executed by the state machine inside each Flash memory array.

28.5.6 Data polling flag (DQ7)

While programming either Flash memory, the 8032 may read the Data Polling Flag Bit (DQ7), which outputs the complement of the D7 Bit of the byte being programmed into Flash memory. Once the program operation is complete, DQ7 is equal to D7 of the byte just programmed into Flash memory, indicating the program cycle has completed successfully.

28.5.9 Erase time-out flag (DQ3)

The Erase Time-out Flag Bit (DQ3) reflects the time-out period allowed between two consecutive sector erase instruction sequence bytes. If multiple sector erase commands are desired, the additional sector erase commands (30h) must be sent by the 8032 within 80us after the previous sector erase command. DQ3 is 0 before this time period has expired, indicating it is OK to issue additional sector erase commands. DQ3 will go to logic ’1’ if the time has been longer than 80µs since the previous sector erase command (time has expired), indication that is not OK to send another sector erase command. In this case, the 8032 must start a new sector erase instruction sequence (unlock and command) beginning again after the current sector erase operation has completed.

28.5.10 Programming Flash memory

When a byte of Flash memory is programmed, individual bits are programmed to logic '0.' cannot program a bit in Flash memory to a logic ’1’ once it has been programmed to a logic '0.' A bit must be erased to logic ’1’, and programmed to logic '0.' That means Flash memory must be erased prior to being programmed. A byte of Flash memory is erased to all 1s (FFh). The 8032 may erase the entire Flash memory array all at once, or erase individual sector-by-sector, but not erase byte-by-byte. However, even though the Flash memories cannot be *erased* byte-by-byte, the 8032 may *program* Flash memory byte-by-byte. This means the 8032 does not need to program group of bytes (64, 128, etc.) at one time, like some Flash memories.

Each Flash memory requires the 8032 to send an instruction sequence to program a byte or to erase sectors (see *Table 163 on page 209*).

If the byte to be programmed is in a protected Flash memory sector, the instruction sequence is ignored.

Important note: It is mandatory that a chip-select signal is active for the Flash sector where a programming instruction sequence is targeted. The user must make sure that the correct chip-select equation, FSx or CSBOOTx specified in PSDsoft Express matches the address range that the 8032 firmware is accessing, otherwise the instruction sequence will not be recognized by the Flash array. If memory paging is used, be sure that the 8032 firmware sets the page register to the correct page number before issuing an instruction sequence to the Flash memory segment on a particular memory page, otherwise the correct sector select signal will not become active.

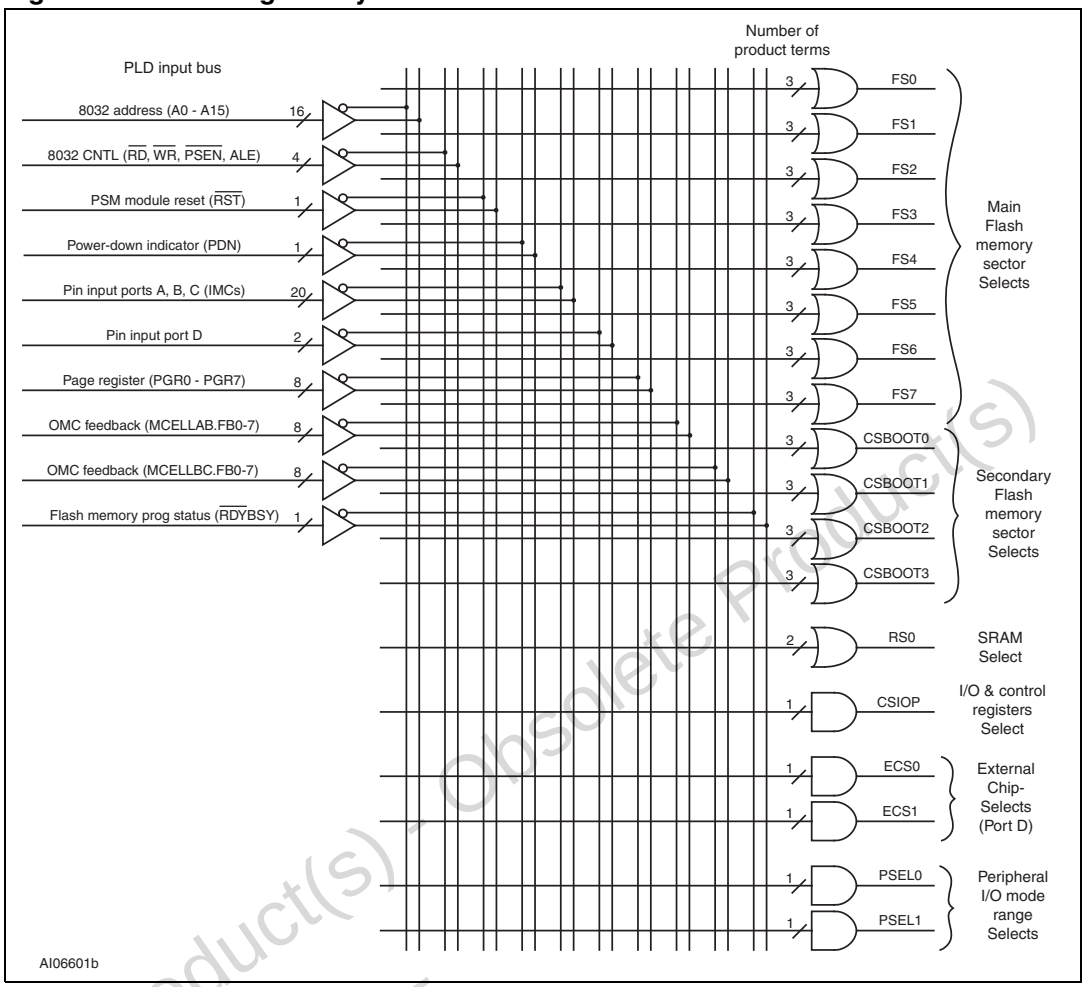
Once the 8032 issues a Flash memory program or erase instruction sequence, it must check the status bits for completion. The embedded algorithms that are invoked inside a Flash memory array provide several ways to give status to the 8032. Status may be checked using any of three methods: Data Polling, Data Toggle, or Ready/$\overline{\text{Busy}}$ (pin PC3).

Table 164. Flash memory status bit definition(1) (2)

Functional block	FSx, or CSBOOTx	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Flash memory	Active (the desired segment is selected)	Data polling	Toggle flag	Error flag	X	Erase timeout	X	X	X

1. X = Not guaranteed value, can be read either '1' or '0.'
2. DQ7-DQ0 represent the 8032 data bus bits, D7-D0.

Figure 74. DPLD logic array



28.5.28 General PLD (GPLD)

The GPLD is used to create general system logic. *Figure 73 on page 221* shows the architecture of the entire GPLD, and *Figure 75 on page 224* shows the relationship between one OMC, one IMC, and one I/O port pin, which is representative of pins on Ports A, B, and C. It is important to understand how these elements work together. A more detailed description will follow for the three major blocks (OMC, IMC, I/O Port) shown in *Figure 75*. *Figure 75* also shows which csiop registers to access for various PLD and I/O functions.

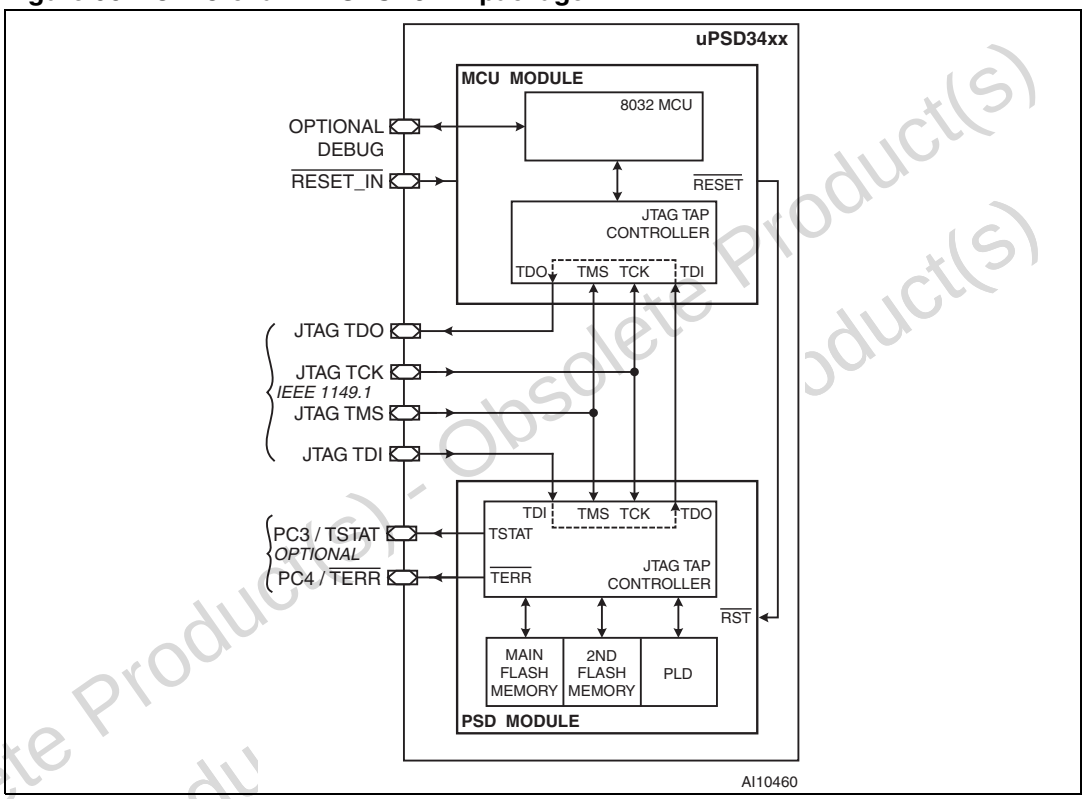
The GPLD contains:

- 16 Output Macrocells (OMC)
- 20 Input Macrocells (IMC)
- OMC Allocator
- Product Term Allocator inside each OMC
- AND-OR Array capable of generating up to 137 product terms
- Three I/O Ports, A, B, and C

PSD module is in BYPASS mode while debugging the MCU module, and the MCU module is in BYPASS mode while performing ISP on the PSD module.

The $\overline{\text{RESET\_IN}}$ input pin on the UPSD34xx package goes to the MCU module, and this module will generate the $\overline{\text{RST}}$ reset signal for the PSD module. These reset signals are totally independent of the JTAG TAP controllers, meaning that the JTAG channel is operational when the modules are held in reset. It is required to assert $\overline{\text{RESET\_IN}}$ during ISP. STMicroelectronics and 3rd party JTAG ISP tools will automatically assert a reset signal during ISP. However, the user must connect this reset signal to $\overline{\text{RESET\_IN}}$ as shown in examples in *Figure 91 on page 259* and *Figure 92 on page 261*.

Figure 90. JTAG chain in UPSD34xx package



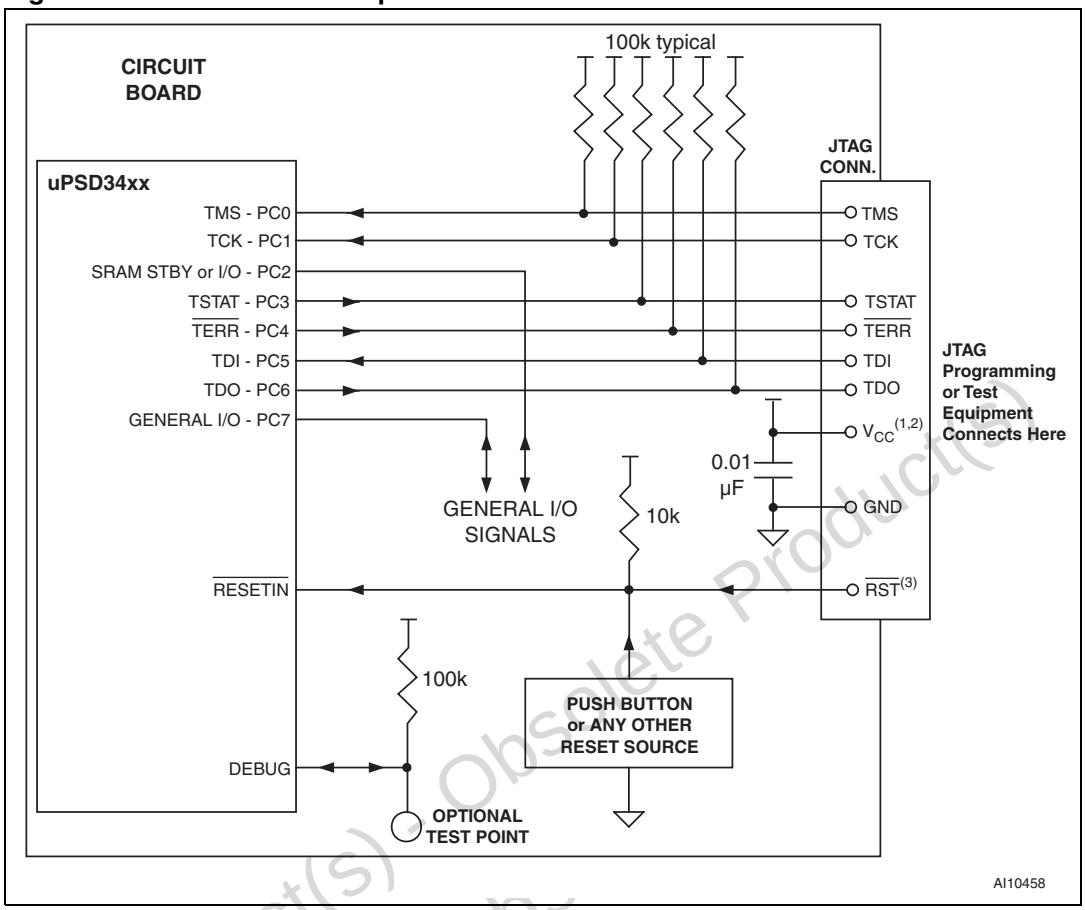
28.6.3 In-system programming

The ISP function can use two different configurations of the JTAG interface:

- 4-pin JTAG: TDI, TDO, TCK, TMS
- 6-pin JTAG: Signals above plus TSTAT, $\overline{\text{TERR}}$

At power-up, the four basic JTAG signals are all inputs, waiting for a command to appear on the JTAG bus from programming or 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 signals, TSTAT and $\overline{\text{TERR}}$.

Figure 92. Recommended 6-pin JTAG connections



1. For 5 V UPSD34xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 5 V system V_{DD}.
2. For 3.3 V UPSD34xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 3.3 V system V_{CC}.
3. This signal is driven by an Open-Drain output in the JTAG equipment, allowing more than one source to activate RESET_IN.

28.6.6 Recommended JTAG connector

There is no industry standard JTAG connector. STMicroelectronics recommends a specific JTAG connector and pinout for uPSD3xxx so programming and debug equipment will easily connect to the circuit board. The user does not have to use this connector if there is a different connection scheme.

The recommended connector scheme can accept a standard 14-pin ribbon cable connector (2 rows of 7 pins on 0.1” centers, 0.025” square posts, standard keying) as shown in *Figure 93*. See the STMicroelectronics “FlashLINK, FL-101 User Manual” for more information.

with an ACK. The host will resend the SETUP packet a number of times and if an ACK is not received from the UPSD3400, the host will issue a USB reset and then enumerate it again. Upon detecting a USB reset, the UPSD3400 firmware will reset and initialize the USB SIE putting the hardware back into the reset/initialized state so that when the next SETUP packet is received, the UPSD3400 will respond with an ACK to the host.

Impact on application

If this occurs during enumeration, the impact is minimal as the host will retry the enumeration. If it happens after enumeration, the communication will break down between the host application and the UPSD3400 and will need to be re-established after the UPSD3400 is reset and enumerated again. In extremely noisy environments, the UPSD3400 may not communicate well over USB with the host application.

Workaround

Revision A and B - None identified at this time.

34.10 MCU JTAG ID

Description

MCU JTAG ID changed to differentiate revision A from revision B silicon through the JTAG port. The PSD JTAG ID remains the same.

Revision A MCU JTAG ID - 0451F041f

Revision B MCU JTAG ID - 1451F041h

Impact on application

There will be no impact on the application. The impact will be to JTAG production programming equipment that may need to distinguish between revision A and B MCU silicon if the firmware is different depending on the revision level.

34.11 Port 1 not 5-volt IO tolerant

Description

The port P1 is shared with the ADC module and as a result Port P1 is not 5 V tolerant.

Impact on application

5 V devices should not be connected to port P1.

Workaround

Revision A and B - Peripherals or GPIO that require 5-Volt IO tolerance should be mapped to Port 3 or Port 4.

35 Revision history

Table 241. Document revision history

Date	Version	Revision details
04-Feb-2005	1	First Edition
30-Mar-2005	2	Added one note in *Section 1: Description on page 20* Added two notes in *Section 25: USB interface on page 150* Changed values in *Table 230 on page 285* (Turbo Off column) Added *Section 34: Important notes on page 294*
25-Oct-2005	3	Changed *Table on page 293* to add sales types with 32K SRAM Changed *Figure 1 on page 21* Changed *Figure 5 on page 30* Corrected Port Pin P1.5 from ADC6 to ADC5 in *Table 2 on page 24* Removed duplicate entry for 80-pin no. 11 in *Table 2 on page 24* Changed *Figure 61 on page 191* Updated *Table 157 on page 193* Updated *Table 239 on page 292*
11-Jul-2006	4	Pin descriptions, *Figure 2 on page 22* and *Figure 3* updated with V_{REF} changed to AV_{REF} V_{REF} changed to AV_{REF} throughout document *Figure 13* updated, correcting CCON[2:0] Clarification of V_{CC}, V_{DD}, AV_{CC} supply voltages in section *Section 30: Maximum rating on page 268* *Section 34: Important notes* updated with differences between silicon revisions A and B, and new Important Notes added. SPI Master Controller corrected to 10MHz in features on first page Latched address out modified, adding A8-A15 to PB0-PB7, *Section Table 2.: Pin definitions* UCON register reset value changed from 00h to 08h throughout Reference to USBCE bit corrected to UPLLCE*Section 14 on page 68* Incorrect references to UART#2 changed to UART#1*Section 22.1 on page 120* UADDR register description enhanced, *Table 100 on page 162* USB interrupts section text expanded, *Section 25.4.3 on page 163* UIFO register table modified, *Table 112 on page 166* UCTL register table enhanced, *Table 120 on page 170* Note added below *Table 122 on page 171* Many modifications made to UCON register description, *Table 126 on page 173* An incorrect reference to CAPCOMHn changed to CAPCOMLn *Section 27.7 on page 184* Part numbering guide updated with B revision information *Section 33 on page 292* *Figure 40 on page 123* updated Document reformatted Note added related to non-support of external indirect addressing, in*Section 9.6* and in *Table 8 on page 54*
26-Jan-2009	5	SRAM standby mode removed. Backup battery feature removed. All products are delivered in ECOPACK-compliant packages. *Section 32: Package mechanical information on page 289* updated. Small text changes including part number capitalization.