**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 ~ 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/upsd3422eb40u6 |

**Table 2. Pin definitions (continued)**

| Port pin | Signal name | 80-pin No. | 52-pin No.(1) | In/out | Function | | |
|---|---|---|---|---|---|---|---|
| | | | | | Basic | Alternate 1 | Alternate 2 |
| ALE | | 4 | N/A | O | Address Latch signal, external bus | | |
| RESET_IN | | 68 | 44 | I | Active low reset input | | |
| XTAL1 | | 48 | 31 | I | Oscillator input pin for system clock | | |
| XTAL2 | | 49 | 32 | O | Oscillator output pin for system clock | | |
| DEBUG | | 8 | 5 | I/O | I/O to the MCU debug unit | | |
| PA0 | | 35 | N/A | I/O | General I/O port pin | | All Port A pins support:<br>– PLD Macrocell outputs, or<br>– PLD inputs, or<br>– Latched address out (A0-A7), or<br>– Peripheral I/O mode |
| PA1 | | 34 | N/A | I/O | General I/O port pin | | |
| PA2 | | 32 | N/A | I/O | General I/O port pin | | |
| PA3 | | 28 | N/A | I/O | General I/O port pin | | |
| PA4 | | 26 | N/A | I/O | General I/O port pin | | |
| PA5 | | 24 | N/A | I/O | General I/O port pin | | |
| PA6 | | 22 | N/A | I/O | General I/O port pin | | |
| PA7 | | 21 | N/A | I/O | General I/O port pin | | |
| PB0 | | 80 | 52 | I/O | General I/O port pin | | All Port B pins support:<br>– PLD Macrocell outputs, or<br>– PLD inputs, or<br>– Latched address out (A0-A7 or A8-A15) |
| PB1 | | 78 | 51 | I/O | General I/O port pin | | |
| PB2 | | 76 | 50 | I/O | General I/O port pin | | |
| PB3 | | 74 | 49 | I/O | General I/O port pin | | |
| PB4 | | 73 | 48 | I/O | General I/O port pin | | |
| PB5 | | 71 | 46 | I/O | General I/O port pin | | |
| PB6 | | 67 | 43 | I/O | General I/O port pin | | |
| PB7 | | 66 | 42 | I/O | General I/O port pin | | |
| JTAGTMS | TMS | 20 | 13 | I | JTAG pin (TMS) | | |
| JTAGTCK | TCK | 17 | 12 | I | JTAG pin (TCK) | | |
| PC2 | | 16 | 11 | I/O | General I/O port pin | | PLD Macrocell output, or PLD input |
| PC3 | TSTAT | 15 | N/A | I/O | General I/O port pin | Optional JTAG Status (TSTAT) | PLD, Macrocell output, or PLD input |
| PC4 | TERR | 9 | N/A | I/O | General I/O port pin | Optional JTAG Status ($\overline{\text{TERR}}$) | PLD, Macrocell output, or PLD input |
| JTAGTDI | TDI | 7 | 4 | I | JTAG pin (TDI) | | |
| JTAGTDO | TDO | 6 | 3 | O | JTAG pin (TDO) | | |

## 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*)

## 15.2 Power-down mode

Power-down mode will halt the 8032 core and all MCU peripherals (Power-down mode blocks MCU_CLK, USB_CLK, and PERIPH_CLK). This is the lowest power state for the MCU module. When the PSD module is also placed in Power-down mode, the lowest total current consumption for the combined die is achieved for the UPSD34xx. See *Section 28.1.16: Power management on page 197* in the PSD module section for details on how to also place the PSD module in Power-down mode. The sequence of 8032 instructions is important when placing both modules into Power-down mode.

The instruction that sets the PD Bit in the SFR named PCON (*Table 33 on page 74*) is the last instruction executed prior to the MCU module going into Power-down mode. Once in Power-down mode, the on-chip oscillator circuitry and all clocks are stopped. The SFRs, DATA, IDATA, and XDATA are preserved.

Power-down mode is terminated only by a reset from the supervisor, originating from the RESET_IN_ pin, the Low-Voltage Detect circuit (LVD), or a JTAG Debug reset command. Since the clock to the WTD is not active during Power-down mode, it is not possible for the supervisor to generate a WDT reset.

*Table 31 on page 74* summarizes the status of I/O pins and peripherals during Idle and Power-down modes on the MCU module. *Table 32 on page 74* shows the state of 8032 MCU address, data, and control signals during these modes.

## 15.3 Reduced frequency mode

The 8032 MCU consumes less current when operating at a lower clock frequency. The MCU can reduce its own clock frequency at run-time by writing to three bits, CPUPS[2:0], in the SFR named CCON0 described in *Table 27 on page 70*. These bits effectively divide the clock frequency ($f_{OSC}$) coming in from the external crystal or oscillator device. The clock division range is from 1/2 to 1/2048, and the resulting frequency is $f_{MCU}$.

This MCU clock division does not affect any of the peripherals, except for the WTD. The clock driving the WTD is the same clock driving the 8032 MCU core as shown in *Figure 13 on page 70*.

MCU firmware may reduce the MCU clock frequency at run-time to consume less current when performing tasks that are not time critical, and then restore full clock frequency as required to perform urgent tasks.

Returning to full clock frequency is done automatically upon an MCU interrupt, if the CPUAR Bit in the SFR named CCON0 is set (the interrupt will force CPUPS[2:0] = 000). This is an excellent way to conserve power using a low frequency clock until an event occurs that requires full performance. See *Table 27 on page 70* for details on CPUAR.

See the DC Specifications at the end of this document to estimate current consumption based on the MCU clock frequency.

*Note: Some of the bits in the PCON SFR shown in Table 33 on page 74 are not related to power control.*

# 18 MCU bus interface

The MCU module has a programmable bus interface which is a modified 8032 bus with 16 multiplexed address and data lines. The bus supports four types of data transfer (16- or 8-bit), each transfer is to/from a memory location external to the MCU module:

- Code Fetch cycle using the PSEN signal: fetch a 16-bit code word for filling the pre-fetch queue. The CPU fetches a code byte from the PFQ for execution;
- Code Read cycle using PSEN: read a 16-bit code word using the MOVC (Move Constant) instruction. The code word is routed directly to the CPU and by-pass the PFQ;
- XDATA Read cycle using the RD signal: read a data byte using the MOVX (Move eXternal) instruction; and
- XDATA Write cycle using the WR signal: write a data byte using the MOVX instruction

## 18.1 PSEN bus cycles

In a PSEN bus cycle, the MCU module fetches the instruction from the 16-bit program memory in the PSD module. The multiplexed address/data bus AD[15:0] is connected to the PSD module for 16-bit data transfer. The UPSD34xx does not support external PSEN cycles and cannot fetch instruction from other external program memory devices.

## 18.2 READ or WRITE bus cycles

In an XDATA READ or WRITE bus cycle, the MCU's multiplexed AD[15:0] bus is connected to the PSD module, but only the lower bytes AD[7:0] are used for the 8-bit data transfer. The AD[7:0] lines are also connected to pins in the 80-pin package for accessing external devices. If the high address byte A[15:8] is needed for external devices, Port B in the PSD module can be configured to provide the latched A[15:8] address outputs.

## 18.3 Connecting external devices to the MCU bus

The UPSD34xx supports 8-bit only external I/O or Data memory devices. The READ and WRITE data transfer is carried out on the AD[7:0] bus which is available in the 80-pin package. The address lines can be brought out to the external devices in one of three ways:

1. Configure Ports B and A of the PSD module in Address Output mode, as shown in *Figure 19*;
2. Use Port B together with an external latch, as shown in *Figure 20 on page 87*. The external latch latches the low address byte from the AD[7:0] bus with the ALE signal.This configuration is for design where Port A is needed for CPLD functions; and
3. Configure the microcell in the CPLD to output any address line to any of the CPLD output pins. This is the most flexible implementation but requires the use of CPLD resources.

Ports A and B in the PSD module can be configured in the PSDsoft to provide latched MCU address A[7:0] and A[15:8] (see *Section 28.5: PSD module detailed operation on page 207* for details on how to enable Address Output mode). The latched address outputs on the ports are pin configurable. For example, Port B pins PB[2:0] can be enabled to provide

**Table 61. T2CON register bit definition (continued)**

| Bit | Symbol | R/W | Definition |
|---|---|---|---|
| 4 | TCLK(1) | R,W | UART0 Transmit Clock control.<br>When TCLK = 1, UART0 uses Timer 2 overflow pulses for its transmit clock in Modes 1 and 3. TCLK=0, Timer 1 overflow is used for transmit clock |
| 3 | EXEN2 | R,W | Timer 2 External Enable.<br>When EXEN2 = 1, capture or reload results when negative edge on pin T2X occurs. EXEN2 = 0 causes Timer 2 to ignore events at pin T2X. |
| 2 | TR2 | R,W | Timer 2 run control.<br>1 = Timer/Counter 2 is on, 0 = Timer Counter 2 is off. |
| 1 | C/$\overline{T2}$ | R,W | Counter or Timer function select.<br>When C/$\overline{T2}$ = 0, function is timer, clocked by internal clock.<br>When C/$\overline{T2}$ = 1, function is counter, clocked by signal sampled on external pin, T2. |
| 0 | CP/$\overline{RL2}$ | R,W | Capture/Reload.<br>When CP/$\overline{RL2}$ = 1, capture occurs on negative transition at pin T2X if EXEN2 = 1. When CP/$\overline{RL2}$ = 0, auto-reload occurs when Timer 2 overflows, or on negative transition at pin T2X when EXEN2=1. When RCLK = 1 or TCLK = 1, CP/$\overline{RL2}$ is ignored, and Timer 2 is forced to auto-reload upon Timer 2 overflow |

*Note: 1 The RCLK1 and TCLK1 Bits in the SFR named PCON control UART1, and have the exact same function as RCLK and TCLK.*

**Table 62. Timer/counter 2 operating modes**

| Mode | Bits in T2CON SFR | | | | Pin T2X (1) | Remarks | Input clock | |
|---|---|---|---|---|---|---|---|---|
| | RCLK or TCLK | CP/$\overline{RL2}$ | TR2 | EXEN2 | | | Timer, internal | Counter, external (Pin T2, P1.0) |
| 16-bit Auto-reload | 0 | 0 | 1 | 0 | x | reload [RCAP2H, RCAP2L] to [TH2, TL2] upon overflow (upcounting) | $f_{OSC}/12$ | MAX $f_{OSC}/24$ |
| | 0 | 0 | 1 | 1 | ↓ | reload [RCAP2H, RCAP2L] to [TH2, TL2] at falling edge on pin T2X | | |
| 16-bit capture | 0 | 1 | 1 | 0 | x | 16-bit timer/counter (upcounting) | $f_{OSC}/12$ | MAX $f_{OSC}/24$ |
| | 0 | 1 | 1 | 1 | ↓ | Capture [TH2, TL2] and store to [RCAP2H, RCAP2L] at falling edge on pin T2X | | |

## 23.12 I²C Start sample setting (S1SETUP)

The S1SETUP register (*Table 84*) determines how many times an I²C bus Start condition will be sampled before the SIOE validates the Start condition, giving the SIOE the ability to reject noise or illegal transmissions.

Because the minimum duration of an Start condition varies with I²C bus speed ($f_{SCL}$), and also because the UPSD34xx may be operated with a wide variety of frequencies ($f_{OSC}$), it is necessary to scale the number of samples per Start condition based on $f_{OSC}$ and $f_{SCL}$.

In Slave mode, the SIOE recognizes the beginning of a Start condition when it detects a '1'-to-'0' transition on the SDA bus line while the SCL line is high (see *Figure 41 on page 125*). The SIOE must then validate the Start condition by sampling the bus lines to ensure SDA remains low and SCL remains high for a minimum amount of hold time, $t_{HLDSTA}$. Once validated, the SIOE begins receiving the address byte that follows the Start condition.

If the EN_SS Bit (in the S1SETUP register) is not set, then the SIOE will sample only once after detecting the '1'-to-'0' transition on SDA. This single sample is taken $1/f_{OSC}$ seconds after the initial 1-to-0 transition was detected. However, more samples should be taken to ensure there is a valid Start condition.

To take more samples, the SIOE should be initialized such that the EN_SS Bit is set, and a value is written to the SMPL_SET[6:0] field of the S1SETUP register to specify how many samples to take. The goal is to take a good number of samples during the minimum Start condition hold time, $t_{HLDSTA}$, but no so many samples that the bus will be sampled after $t_{HLDSTA}$ expires.

*Table 86 on page 134* describes the relationship between the contents of S1SETUP and the resulting number of I²C bus samples that SIOE will take after detecting the 1-to-0 transition on SDA of a Start condition.

*Important note: Keep in mind that the time between samples is always $1/f_{OSC}$.*

The minimum Start condition hold time, $t_{HLDSTA}$, is different for the three common I²C speed categories per *Table 87 on page 134*.

**Table 84. S1SETUP: I²C Start condition sample setup register (SFR DBh, reset value 00h)**

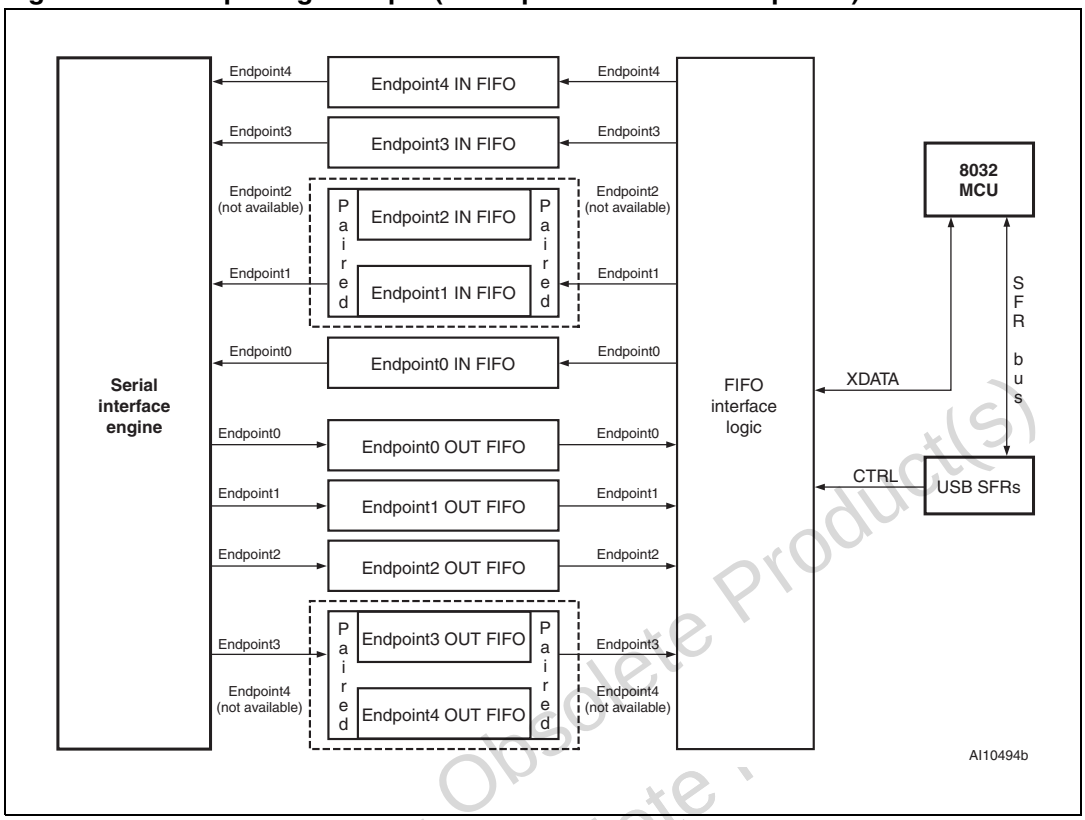
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| EN_SS | SMPL_SET[6:0] | | | | | | |

**Table 85. S1SETUP register bit definition**

| Bit | Symbol | R/W | Function |
|---|---|---|---|
| 7 | EN_SS | R/W | Enable Sample Setup<br>EN_SS = 1 will force the SIOE to sample(1) a Start condition on the bus the number of times specified in SMPL_SET[6:0].<br>EN_SS = 0 means the SIOE will sample(1) a Start condition only one time, regardless of the contents of SMPL_SET[6:0]. |
| 6:0 | SMPL_SET [6:0] | – | Sample Setting<br>Specifies the number of bus samples(1) taken during a Start condition. See *Table 86* for values. |

1. Sampling SCL and SDA lines begins after '1'-to-'0' transition on SDA occurred while SCL is high. Time between samples is $1/f_{OSC}$.

**Figure 54. FIFO pairing example (1/2 IN paired and 3/4 OUT paired)**






## 25.3.4 Reading and writing FIFOs

There are a total of ten 64-byte FIFOs. Each of the five Endpoints has two FIFOs, one IN FIFO for IN transactions and one OUT FIFO for OUT transactions. The FIFOs are accessible by the CPU through a 64-byte segment in the XDATA space when the VISIBLE Bit is set (see *Table 120 on page 170*). If the VISIBLE Bit is not set, the FIFOs are not accessible by the CPU but are still accessible by the SIE. The base address of the 64-byte segment is specified by the USB base address high register (see *Table 130 on page 175*) and the USB base address low register (see *Table 132 on page 175*). When the VISIBLE Bit is set, the FIFO that is accessible in the 64-byte XDATA space segment is the FIFO selected by the USEL register. The USEL register contains two fields used for selecting the accessible FIFO. The EP field determines the Endpoint selected and the DIR Bit selects the IN or OUT FIFO associated with the Endpoint.

## 25.3.5 Accessing FIFO control registers, UCON, and USIZE

Each of the 10 Endpoint FIFOs has an associated USB endpoint control register (UCON, 0F1H) and a USB FIFO valid size register (USIZE, 0F2H). The USB endpoint select register (USEL) is not only used to select the Endpoint FIFO that is accessible in the XDATA space, but also selects the associated Endpoint's UCON and USIZE registers that are accessible at SFR addresses 0F1H and 0F2H.

### 25.3.6 Accessing the setup command buffer

Setup Packets are sent from the host to a device's Endpoint0 and consist of 8 bytes of command data. When the SIE receives a Setup packet from the host, it stores the 8 bytes of data in the Command Buffer. The command buffer is accessed via the indexed USB Setup Command Value register (USCV). The USB Setup Command Index register (USCI) is used to select the byte from the command buffer that is read when accessing the USCV register.

## 25.4 USB registers

The USB module is controlled via registers mapped into the SFR space. The USB SFRs consist of the following:

- UADDR: USB device address
- UPAIR: USB FIFO pairing control
- UIE0~3: USB interrupt enable
- UIF0~3: USB interrupt flags
- UCTL: USB control
- USTA: USB status
- USEL: USB endpoint and direction select
- UCON: USB selected FIFO control register
- USIZE: USB selected FIFO size register
- UBASE: USB base address register
- USCI: USB setup command index
- USCV: USB setup command value

The memory map for the USB SFRs, the individual bit names, and the reset values are shown in *Table 99*.

**Table 99. UPSD34xx USB SFR register map(1)**

| SFR addr (hex) | SFR name | Bit name and <bit address> | | | | | | | | Reset value (hex) | Comment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| E2 | UADDR | – | USBADDR[6:0] | | | | | | | 00 | USB Address |
| E3 | UPAIR | – | – | – | – | PR3OUT | PR1OUT | PR3IN | PR1IN | 00 | USB Pairing Control |
| E4 | UIE0 | – | – | – | – | RSTIE | SUSPNDIE | EOPIE | RESUMIE | 00 | USB Global Interrupt Enable |
| E5 | UIE1 | – | – | – | IN4IE | IN3IE | IN2IE | IN1IE | IN0IE | 00 | USB IN FIFO Interrupt Enable |
| E6 | UIE2 | – | – | – | OUT4IE | OUT3IE | OUT2IE | OUT1IE | OUT0IE | 00 | USB OUT FIFO Interrupt Enable |
| E7 | UIE3 | – | – | – | NAK4IE | NAK3IE | NAK2IE | NAK1IE | NAK0IE | 00 | USB IN FIFO NAK Int. Enable |

**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 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 |

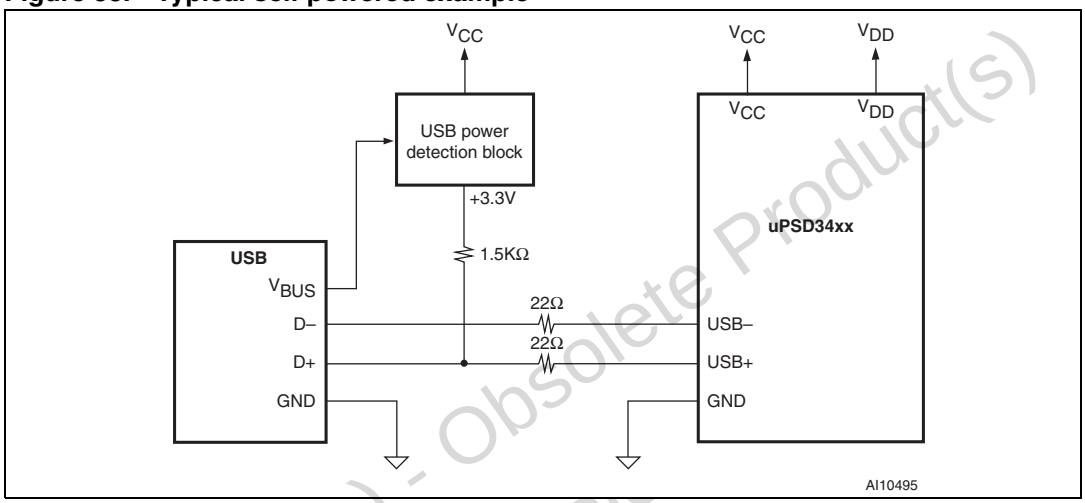
**Table 115. UIF1 register bit definition**

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

## 25.5 Typical connection to USB

Connecting the UPSD34xx to the USB is simple and straightforward. *Figure 55* shows a typical self-powered example requiring only three resistors and a USB power detection circuit. The USB power detection circuit detects when the device has been connected to the USB. When $V_{BUS}$ is detected, it switches 3.3 V to the pull-up resistor on the D+ line. Per the USB specification, the pull-up resistor on D+ is required to signal to the upstream USB port when a full speed device has been connected to the bus. The resistors in series in the D+ and D– lines are recommended per the USB specification to reduce transients on the data lines.

**Figure 55. Typical self powered example**



sequence is typically used when the 8032 is intentionally programming a large number of bytes (such as during IAP). After intentional programming is complete, typically the Bypass mode would be disabled, and full protection is back in place to prevent unwanted WRITEs to Flash memory.

The Bypass Unlock mode is entered by first initiating two Unlock bus cycles. This is followed by a third WRITE operation containing the Bypass Unlock command, 20h (as shown in *Table 163 on page 209*). The Flash memory array that received that sequence then enters the Bypass Unlock mode. After this, a two bus cycle program operation is all that is required to program a byte in this mode. The first bus cycle in this shortened program instruction sequence contains the Bypassed Unlocked Program command, A0h, to any valid address within the unlocked Flash array. The second bus cycle contains the address and data of the byte to be programmed. Programming status is checked using toggle, polling, or Ready/Busy just as before. Additional data bytes are programmed the same way until this Bypass Unlock mode is exited.

To exit Bypass Unlock mode, the system must issue the Reset Bypass Unlock instruction sequence. The first bus cycle of this instruction must write 90h to any valid address within the unlocked Flash Array; the second bus cycle must write 00h to any valid address within the unlocked Flash Array. After this sequence the Flash returns to Read Array mode.

During Bypass Unlock Mode, only the Bypassed Unlock Program instruction, or the Reset Bypass Unlock instruction is valid, other instruction will be ignored.

### 28.5.15 Erasing Flash memory

Flash memory may be erased sector-by-sector, or an entire Flash memory array may be erased with one command (bulk).

### 28.5.16 Flash bulk erase

The Flash Bulk Erase instruction sequence uses six WRITE operations followed by a READ operation of the status register, as described in *Table 163 on page 209*. If any byte of the Bulk Erase instruction sequence is wrong, the Bulk Erase instruction sequence aborts and the device is reset to the Read Array mode. The address provided by the 8032 during the Flash Bulk Erase command sequence may select any one of the eight Flash memory sector select signals FSx or one of the four signals CSBOOTx. An erase of the entire Flash memory array will occur in a particular array even though a command was sent to just one of the individual Flash memory sectors within that array.

During a Bulk Erase, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7). The Error Flag Bit (DQ5) returns a ’1’ if there has been an erase failure. Details of acquiring the status of the Bulk Erase operation are detailed in the section entitled “*Section 28.5.10: Programming Flash memory on page 212*.

During a Bulk Erase operation, the Flash memory does not accept any other Flash instruction sequences.

### 28.5.17 Flash sector erase

The Sector Erase instruction sequence uses six WRITE operations, as described in *Table 163 on page 209*. Additional Flash Sector Erase commands to other sectors within the same Flash array may be issued by the 8032 if the additional commands are sent within a limited amount of time.

Native product terms come from the AND-OR Array. Each OMC may borrow product terms only from certain other OMCs, if they are not in use. Product term allocation does not add any propagation delay to the logic. The fitter report generated by PSDsoft Express will show any PT allocation that has occurred.

If an equation requires more product terms than are available to it through PT allocation, then “external” product terms are required, which consumes other OMCs. This is called product term expansion and also happens automatically in PSDsoft Express as needed. PT expansion causes additional propagation delay because an additional OMC is consumed by the expansion process and its output is rerouted (or fed back) into the AND-OR array. The user can examine the fitter report generated by PSDsoft Express to see resulting PT allocation and PT expansion (expansion will have signal names, such as ‘*.fb_0’ or ‘*.fb_1’). PSDsoft Express will always try to fit the logic design first by using PT allocation, and if that is not sufficient then PSDsoft Express will use PT expansion.

Product term expansion may occur in the DPLD for complex chip select equations for Flash memory sectors and for SRAM, but this is a rare occurence. If PSDsoft Express does use PT expansion in the DPLD, it results in an approximate 15ns additional propagation delay for that chip select signal, which gives 15ns less time for the memory to respond. Be aware of this and consider adding a wait state to the 8032 bus access (using the SFR named, BUSCON), or lower the 8032 clock frequency to avoid problems with memory access time.

**Figure 77. OMC allocator**

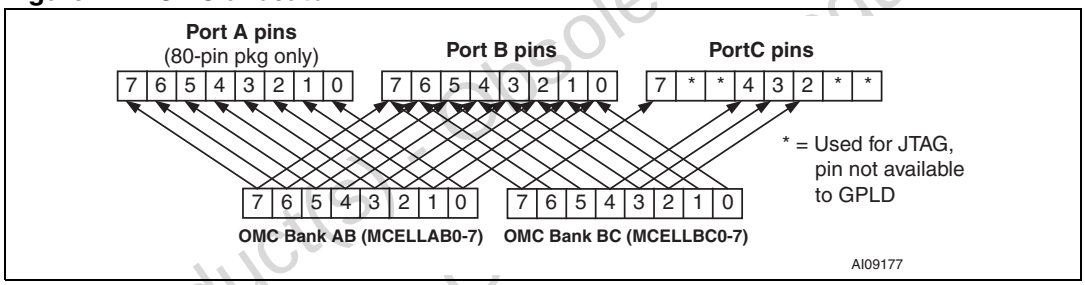



**Table 168. OMC port and data bit assignments**

| OMC | Port assignment(1),(2) | Native product terms from AND-OR array | Maximum borrowed product terms | Data bit on 8032 data bus for loading or reading OMC |
|---|---|---|---|---|
| MCELLAB0 | Port A0 or B0 | 3 | 6 | D0 |
| MCELLAB1 | Port A1 or B1 | 3 | 6 | D1 |
| MCELLAB2 | Port A2 or B2 | 3 | 6 | D2 |
| MCELLAB3 | Port A3 or B3 | 3 | 6 | D3 |
| MCELLAB4 | Port A4 or B4 | 3 | 6 | D4 |
| MCELLAB5 | Port A5 or B5 | 3 | 6 | D5 |
| MCELLAB6 | Port A6 or B6 | 3 | 6 | D6 |
| MCELLAB7 | Port A7 or B7 | 3 | 6 | D7 |
| MCELLBC0 | Port B0 | 4 | 5 | D0 |
| MCELLBC1 | Port B1 | 4 | 5 | D1 |
| MCELLBC2 | Port B or C2 | 4 | 5 | D2 |

*Note: If a particular OMC output is specified as an internal node and not specified as a port pin output in PSDsoft Express, then the port pin that is associated with that OMC can be used for other I/O functions.*

**Figure 80. Simple PLD logic example**

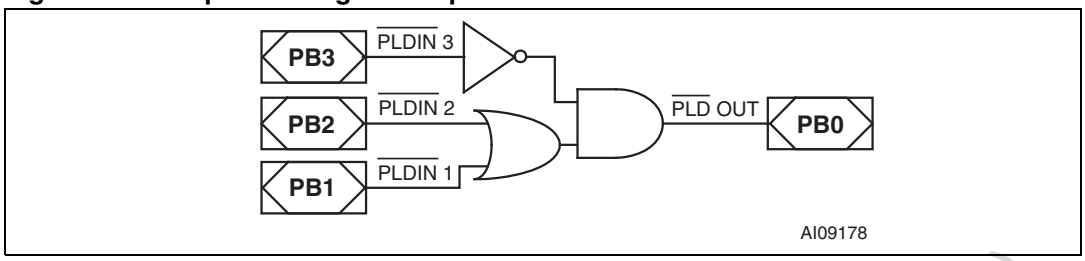
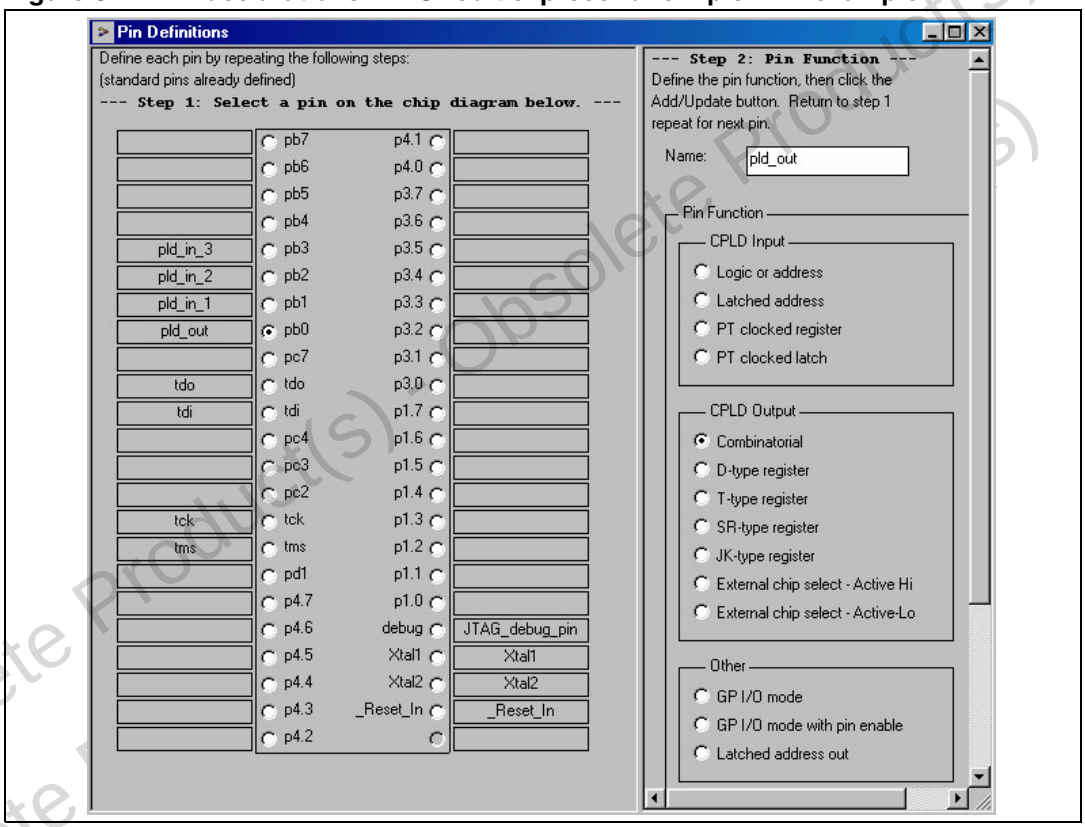



**Figure 81. Pin declarations in PSDsoft express for simple PLD example**



## 28.5.47 Individual port structures

Ports A, B, C, and D have some differences. The structure of each individual port is described in the next sections.

## 28.5.48 Port A structure

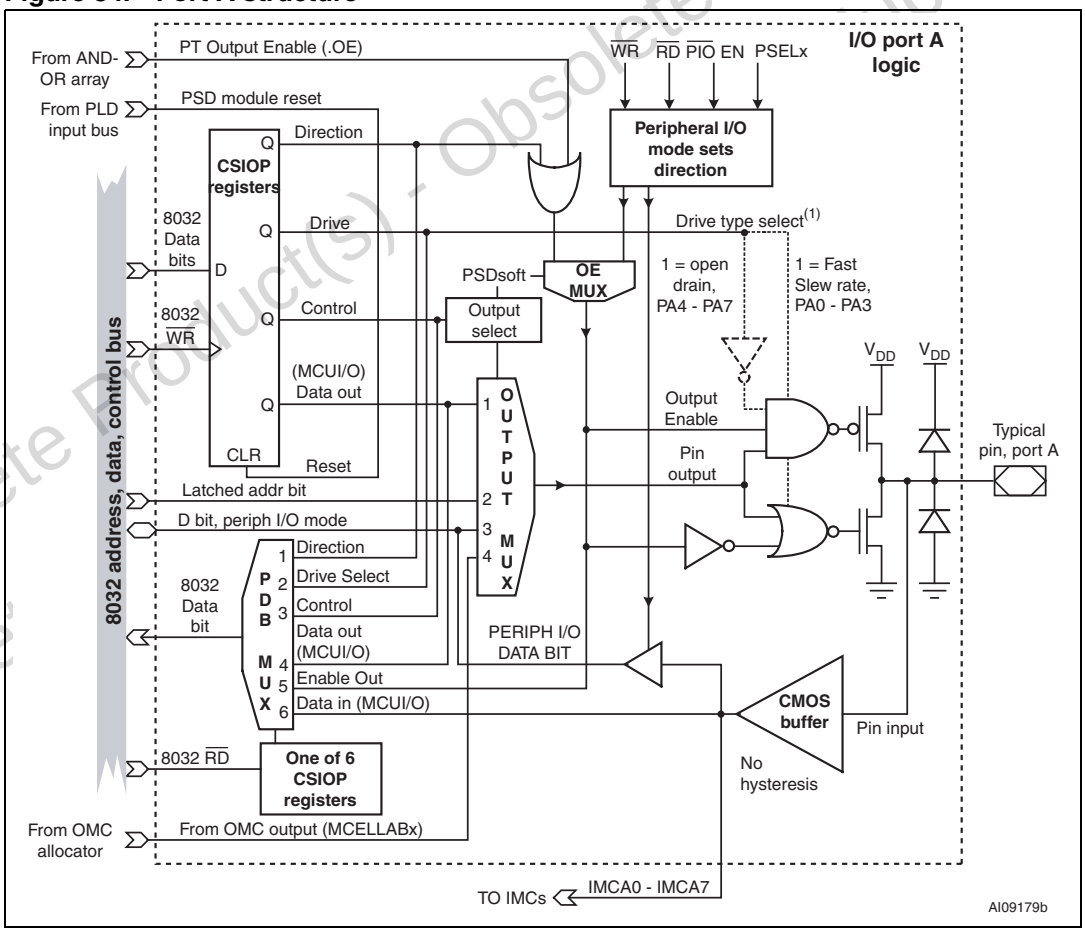
Port A supports the following operating modes:

- MCU I/O Mode
- GPLD Output Mode from Output Macrocells MCELLABx
- GPLD Input Mode to Input Macrocells IMCAx
- Latched Address Output Mode
- Peripheral I/O Mode

Port A also supports Open Drain/Slew Rate output drive type options using csiop Drive Select registers. Pins PA0-PA3 can be configured to fast slew rate, pins PA4-PA7 can be configured to Open Drain Mode.

See *Figure 84* for details.

**Figure 84. Port A structure**



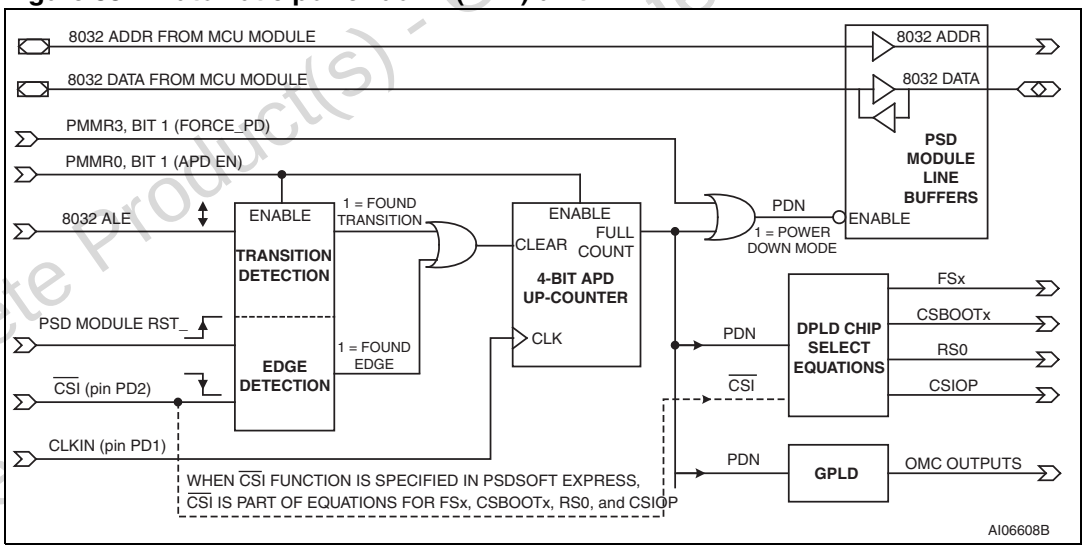

*Note: 1 Port pins PA0-PA3 are capable of Fast Slew Rate output drive option. Port pins PA4-PA7 are capable of Open Drain output option.*

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**



### 28.6.5 6-pin JTAG ISP (optional)

The optional signals TSTAT and $\overline{\text{TERR}}$ are programming status flags that can reduce programming time by as much as 30% compared to 4-pin JTAG because this status information does not have to be scanned out of the device serially. TSTAT and $\overline{\text{TERR}}$ must be used as a pair for 6-pin JTAG operation.

- TSTAT (pin PC3) indicates when programming of a single Flash location is complete. Logic 1 = Ready, Logic 0 = busy.
- $\overline{\text{TERR}}$ (pin PC4) indicates if there was a Flash programming error. Logic 1 = no error, Logic 0 = error.

The pin functions for PC3 and PC4 must be selected as “Dedicated JTAG - TSTAT” and “Dedicated JTAG - $\overline{\text{TERR}}$” in PSDsoft Express to enable 6-pin JTAG ISP.

No 8032 firmware is needed to use 6-pin ISP because all ISP functions are controlled from the external JTAG program/test equipment.

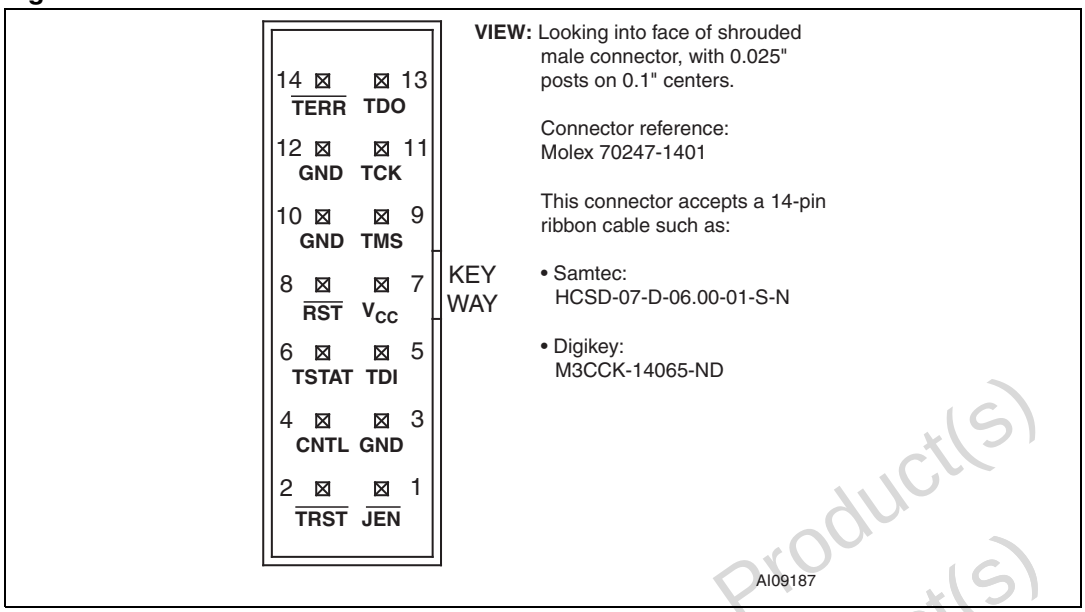
TSTAT and $\overline{\text{TERR}}$ are functional only when JTAG ISP operations are occurring, which means they are non-functional during JTAG debugging of the 8032 on the MCU module.

Programming times vary depending on the number of locations to be programmed and the JTAG programming equipment, but typical JTAG ISP programming times are 10 to 25 seconds using 6-pin JTAG. The signals TSTAT and $\overline{\text{TERR}}$ are not included in the IEEE 1149.1 specification.

*Figure 92 on page 261* shows recommended connections on a circuit board to a JTAG program/test tool using 6-pin JTAG. It is required to connect the $\overline{\text{RST}}$ output signal from the JTAG program/test equipment to the $\overline{\text{RESET\_IN}}$ input on the UPSD34xx. The $\overline{\text{RST}}$ signal is driven by the equipment with an Open Drain driver, allowing other sources (like a push button) to drive $\overline{\text{RESET\_IN}}$ without conflict.

*Note: The recommended pull-up resistors and decoupling capacitor are illustrated in Figure 92.*

**Figure 93. Recommended JTAG connector**




## 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.

**Table 212. PSD module DC characteristics (with 5 V $V_{DD}$)**

| Symbol | Parameter | | Test condition (in addition to those in *Table 211 on page 271*) | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | | 4.5 V < $V_{DD}$ < 5.5 V | 2 | | $V_{DD}$ +0.5 | V |
| $V_{IL}$ | Input low voltage | | 4.5 V < $V_{DD}$ < 5.5 V | –0.5 | | 0.8 | V |
| $V_{LKO}$ | $V_{DD}$ (min) for Flash erase and program | | | 2.5 | | 4.2 | V |
| $V_{OL}$ | Output low voltage | | $I_{OL}$ = 20 µA, $V_{DD}$ = 4.5 V | | 0.01 | 0.1 | V |
| | | | $I_{OL}$ = 8 mA, $V_{DD}$ = 4.5 V | | 0.25 | 0.45 | V |
| $V_{OH}$ | Output high voltage | | $I_{OH}$ = –20 µA, $V_{DD}$ = 4.5 V | 4.4 | 4.49 | | V |
| | | | $I_{OH}$ = –2 mA, $V_{DD}$ = 4.5 V | 2.4 | 3.9 | | V |
| $I_{SB}$ | Standby supply current for power-down mode | | $\overline{CSI}$ > $V_{DD}$ – 0.3 V(1)(2) | | 120 | 250 | µA |
| $I_{LI}$ | Input leakage current | | $V_{SS}$ < $V_{IN}$ < $V_{DD}$ | –1 | ±0.1 | 1 | µA |
| $I_{LO}$ | Output leakage current | | 0.45 < $V_{OUT}$ < $V_{DD}$ | –10 | ±5 | 10 | µA |
| $I_{CC}$ (DC)(3) | Operating supply current | PLD only | PLD_TURBO = Off, f = 0 MHz (3) | | 0 | | µA/PT |
| | | | PLD_TURBO = On, f = 0 MHz | | 400 | 700 | µA/PT |
| | | Flash memory | During Flash memory WRITE/Erase Only | | 15 | 30 | mA |
| | | | Read only, f = 0 MHz | | 0 | 0 | mA |
| | | SRAM | f = 0 MHz | | 0 | 0 | mA |
| $I_{CC}$ (AC)(3) | PLD AC adder | | | | | (4) | |
| | Flash memory AC adder | | | | 1.5 | 2.5 | mA/MHz |
| | SRAM AC adder | | | | 1.5 | 3.0 | mA/MHz |

1. Internal Power-down mode is active.
2. PLD is in non-Turbo mode, and none of the inputs are switching.
3. $I_{OUT}$ = 0 mA.
4. Please see *Figure 95 on page 265* for the PLD current calculation.