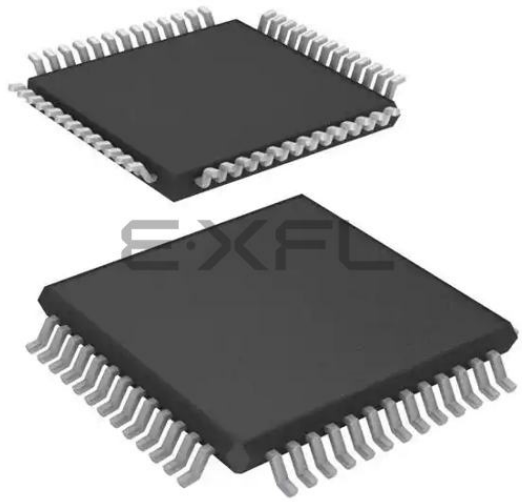


Welcome to E-XFL.COM

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	35
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K 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	52-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3433e-40t6

Obsolete Product(s) - Obsolete Product(s)
Obsolete Product(s) - Obsolete Product(s)

Obsolete Product(s) - Obsolete Product(s)

Obsolete Product(s) - Obsolete Product(s)

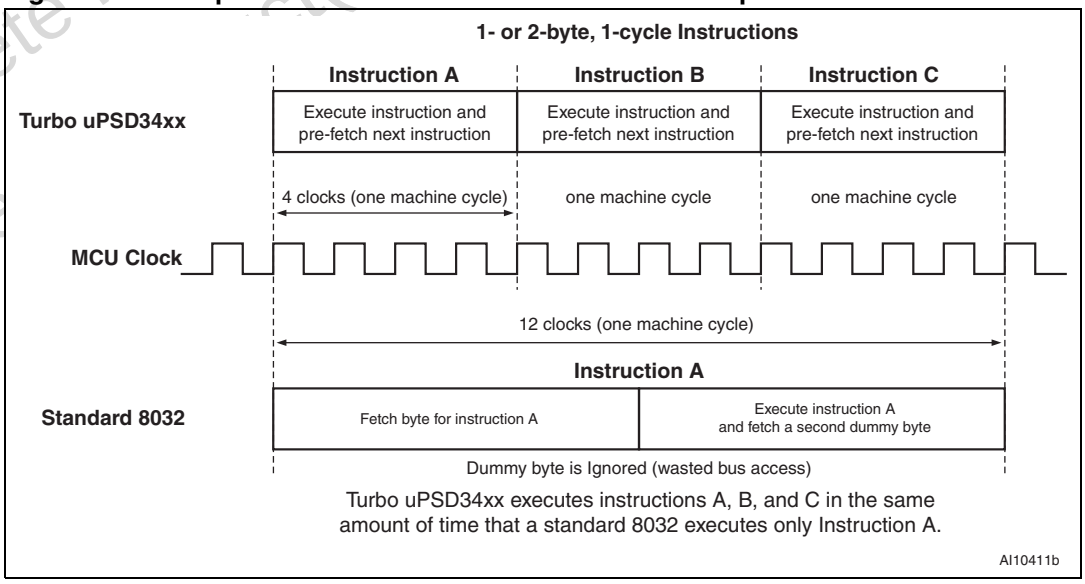
5 8032 MCU core performance enhancements

Before describing performance features of the UPSD34xx, let us first look at standard 8032 architecture. The clock source for the 8032 MCU creates a basic unit of timing called a machine-cycle, which is a period of 12 clocks for standard 8032 MCUs. The instruction set for traditional 8032 MCUs consists of 1, 2, and 3 byte instructions that execute in different combinations of 1, 2, or 4 machine-cycles. For example, there are one-byte instructions that execute in one machine-cycle (12 clocks), one-byte instructions that execute in four machine-cycles (48 clocks), two-byte, two-cycle instructions (24 clocks), and so on. In addition, standard 8032 architecture will fetch two bytes from program memory on almost every machine-cycle, regardless if it needs them or not (dummy fetch). This means for one-byte, one-cycle instructions, the second byte is ignored. These one-byte, one-cycle instructions account for half of the 8032's instructions (126 out of 255 opcodes). There are inefficiencies due to wasted bus cycles and idle bus times that can be eliminated.

The UPSD34xx 8032 MCU core offers increased performance in a number of ways, while keeping the exact same instruction set as the standard 8032 (all opcodes, the number of bytes per instruction, and the native number a machine-cycles per instruction are identical to the original 8032). The first way performance is boosted is by reducing the machine-cycle period to just 4 MCU clocks as compared to 12 MCU clocks in a standard 8032. This shortened machine-cycle improves the instruction rate for one- or two-byte, one-cycle instructions by a factor of three (*Figure 6 on page 33*) compared to standard 8051 architectures, and significantly improves performance of multiple-cycle instruction types.

The example in *Figure 6 on page 33* shows a continuous execution stream of one- or two-byte, one-cycle instructions. The 5 V UPSD34xx will yield 10 MIPS peak performance in this case while operating at 40 MHz clock rate. In a typical application however, the effective performance will be lower since programs do not use only one-cycle instructions, but special techniques are implemented in the UPSD34xx to keep the effective MIPS rate as close as possible to the peak MIPS rate at all times. This is accomplished with an instruction pre-fetch queue (PFQ), a branch cache (BC), and a 16-bit program memory bus as shown in *Figure 7 on page 34*.

Figure 6. Comparison of UPSD34xx with standard 8032 performance



Obsolete Product(s) - Obsolete Product(s)

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										

Obsolete Product(s) - Obsolete Product(s)

17.1.4 Alternate functions

There are five SFRs used to control the mapping of alternate functions onto MCU port pins, and these SFRs are depicted as switches in *Figure 15 on page 80*.

- Port 3 uses the SFR, P3SFS (*Table 41 on page 84*).
- Port 1 uses SFRs, P1SFS0 (*Table 43 on page 84*) and P1SFS1 (*Table 44 on page 84*).
- Port 4 uses SFRs, P4SFS0 (*Table 46 on page 85*) and P4SFS1 (*Table 47 on page 85*).

Since these SFRs are cleared by a reset, then by default all port pins function as GPIO (not the alternate function) until firmware initializes these SFRs.

Each pin on each of the three ports can be independently assigned a different function on a pin-by-pin basis.

The peripheral functions Timer 2, UART1, and I^2C may be split independently between Port 1 and Port 4 for additional flexibility by giving a wider choice of peripheral usage on a limited number of device pins.

When the selected alternate function is UART0, UART1, or SPI, then the related pins are in quasi-bidirectional mode, including the use of the high-side driver for rapid 0-to-1 output transitions. The high-side driver is enabled for just one MCU_CLK period on 0-to-1 transitions by the delay function at the “digital_alt_func_data_out” signal pictured in *Figure 16 on page 80* through *Figure 18 on page 81*.

If the alternate function is Timer 0, Timer 1, Timer 2, or PCA input, then the related pins are in quasi-bidirectional mode, but input only.

If the alternate function is ADC, then for each pin the pull-ups, the high-side driver, and the low-side driver are disabled. The analog input is routed directly to the ADC unit. Only Port 1 supports analog functions (*Figure 16 on page 80*). Port 1 is not 5 V tolerant.

If the alternate function is I^2C, the related pins will be in open drain mode, which is just like quasi-bidirectional mode but the high-side driver is not enabled for one cycle when outputting a 0-to-1 transition. Only the low-side driver and the internal weak pull-ups are used. Only Port 3 supports open-drain mode (*Figure 17 on page 81*). I^2C requires the use of an external pull-up resistor on each bus signal, typically 4.7KΩ to V_{CC}.

If the alternate function is PCA output, then the related pins are in push-pull mode, meaning the pins are actively driven and held to logic '1' by the high-side driver, or actively driven and held to logic '0' by the low-side driver. Only Port 4 supports push-pull mode (*Figure 18 on page 81*). Port 4 push-pull pins can source I_{OH} current when driving logic '1,' and sink I_{OL} current when driving logic '0.' This current is significantly more than the capability of pins on Port 1 or Port 3 (see *Table 211 on page 271*).

For example, to assign these port functions:

- Port 1: UART1, ADC[1:0], P1[7:4] are GPIO
- Port 3: UART0, I^2C, P3[5:2] are GPIO
- Port 4: TCM0, SPI, P4[3:1] are GPIO

The following values need to be written to the SFRs:

P1SFS0 = 00001111b, or 0Fh

P1SFS1 = 00000011b , or 03h

P3SFS = 11000011b, or C3h

P4SFS0 = 11110001b, or F1h

P4SFS1 = 11110000b, or F0h

Obsolete Product(s) - Obsolete Product(s)

A[10:8] and the remaining pins can be configured for other functions such as generating chip selects to the external devices.

Figure 19. Connecting external devices using ports A and B for address AD[15:0]

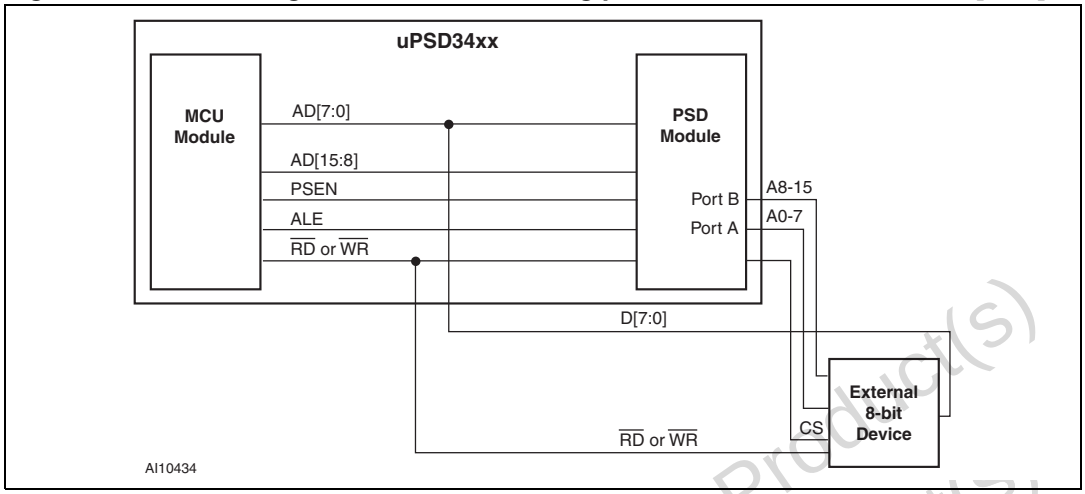
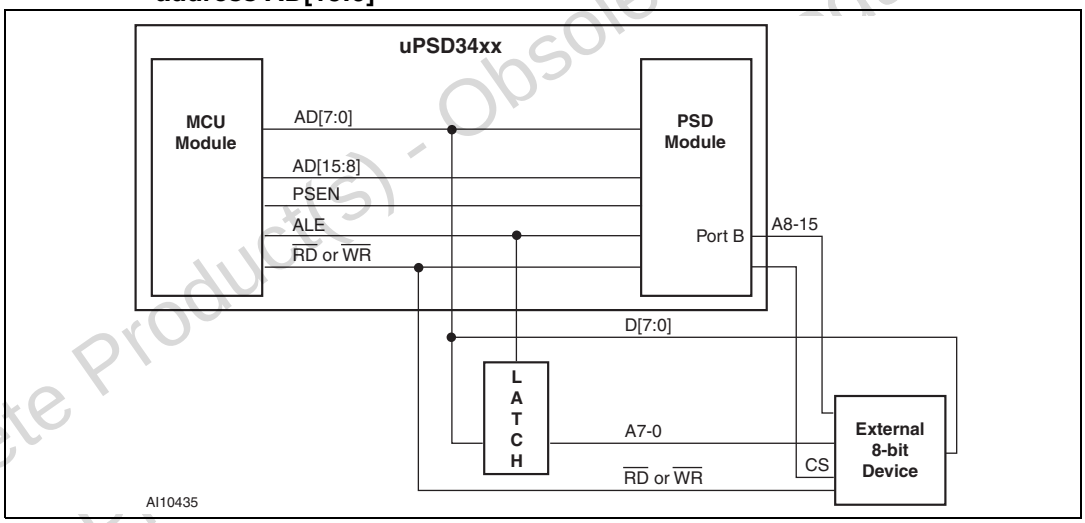


Figure 20. Connecting external devices using port A and an external latch for address AD[15:0]



Obsolete Product(s) - Obsolete Product(s)

18.4 Programmable bus timing

The length of the bus cycles are user programmable at run time. The number of MCU_CLK periods in a bus cycle can be specified in the SFR register named BUSCON (see *Table 49 on page 88*). By default, the BUSCON register is loaded with long bus cycle times (6 MCU_CLK periods) after a reset condition. It is important that the post-reset initialization firmware sets the bus cycle times appropriately to get the most performance, according to *Table 51 on page 90*. Keep in mind that the PSD module has a faster Turbo mode (default) and a slower but less power consuming Non-Turbo mode. The bus cycle times must be programmed in BUSCON to optimize for each mode as shown in *Table 51*. See *Section 28.5: PSD module detailed operation on page 207* for more details.

Enable individual I2C interrupt and set priority
- SFR IEA.I2C = 1
- SFR IPA.I2C = 1 if high priority is desired

Set the Device address for Slave mode
- SFR S1ADR = XXh, desired address

Enable SIOE (as Slave) to return an ACK signal
- SFR S1CON.AA = 1

Master-Transmitter

Disable all interrupts
- SFR IE.EA = 0

Set pointer to global data xmit buffer, set count
- *xmit_buf = *pointer to data
- buf_length = number of bytes to xmit

Set global variables to indicate Master-Xmitter
- I2C_master = 1, I2C_xmitter = 1

Disable Master from returning an ACK
- SFR S1CON.AA = 0

Enable I2C SIOE
- SFR S1CON.INI1 = 1

Transmit Address and R/W bit = 0 to Slave
- Is bus not busy? (SFR S1STA.BBUSY = 0?)

<If busy, then test until not busy>
- SFR S1DAT[7:0] = Load Slave Address & FEh
- SFR S1CON.STA = 1, send Start on bus

<bus transmission begins>

Enable All Interrupts and go do something else
- SFR IE.EA = 1

Master-Receiver

Disable all interrupts
- SFR IE.EA = 0

Set pointer to global data recv buffer, set count
- *recv_buf = *pointer to data
- buf_length = number of bytes to recv

Set global variables to indicate Master-Xmitter
- I2C_master = 1, I2C_xmitter = 0

Disable Master from returning an ACK
- SFR S1CON.AA = 0

Obsolete Product(s) - Obsolete Product(s)

24 SPI (synchronous peripheral interface)

UPSD34xx devices support one serial SPI interface in Master Mode only. This is a three- or four-wire synchronous communication channel, capable of full-duplex operation on 8-bit serial data transfers. The four SPI bus signals are:

- SPIRxD

  Pin P1.5 or P4.5 receives data from the Slave SPI device to the UPSD34xx
- SPITxD

  Pin P1.6 or P4.6 transmits data from the UPSD34xx to the Slave SPI device
- SPICLK

  Pin P1.4 or P4.4 clock is generated from the UPSD34xx to the SPI Slave device
- $\overline{\text{SPISEL}}$

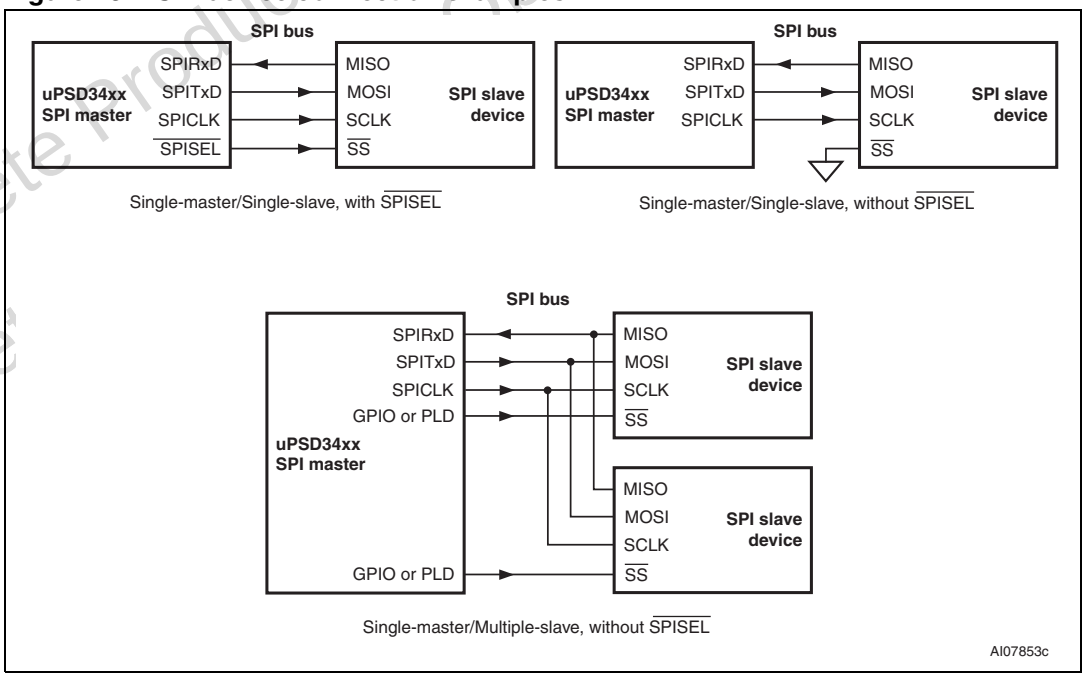
  Pin P1.7 or P4.7 selects the signal from the UPSD34xx to an individual Slave SPI device

This SPI interface supports single-Master/multiple-Slave connections. Multiple-Master connections are not directly supported by the UPSD34xx (no internal logic for collision detection).

If more than one Slave device is required, the $\overline{\text{SPISEL}}$ signal may be generated from UPSD34xx GPIO outputs (one for each Slave) or from the PLD outputs of the PSD module. *Figure 43* illustrates three examples of SPI device connections using the UPSD34xx:

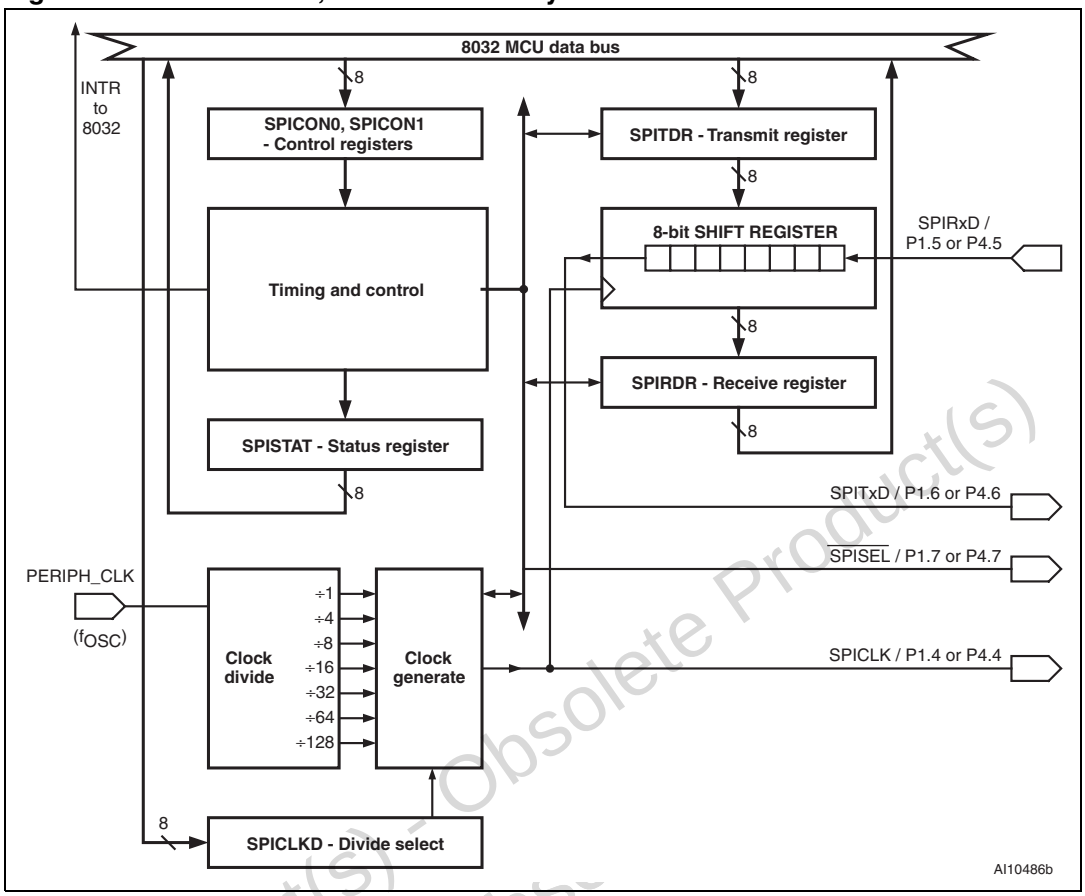
- Single-Master/Single-Slave with $\overline{\text{SPISEL}}$
- Single-Master/Single-Slave without $\overline{\text{SPISEL}}$
- Single-Master/Multiple-Slave without $\overline{\text{SPISEL}}$

Figure 43. SPI device connection examples



Obsolete Product(s) - Obsolete Product(s)

Figure 47. SPI interface, master mode only



24.5 SPI configuration

The SPI interface is reset by the MCU reset, and firmware needs to initialize the SFRs SPICON0, SPICON1, and SPICLKD to define several operation parameters.

The SPO Bit in SPICON0 determines the clock polarity. When SPO is set to '0,' a data bit is transmitted on SPITxD from one rising edge of SPICLK to the next and is guaranteed to be valid during the falling edge of SPICLK. When SPO is set to '1,' a data bit is transmitted on SPITxD from one falling edge of SPICLK to the next and is guaranteed to be valid during the rising edge of SPICLK. The UPSD34xx will sample received data on the appropriate edge of SPICLK as determined by SPO. The effect of the SPO Bit can be seen in *Figure 45* and *Figure 46 on page 145*.

The FLSB Bit in SPICON0 determines the bit order while transmitting and receiving the 8-bit data. When FLSB is '0,' the 8-bit data is transferred in order from MSB (first) to LSB (last). When FLSB Bit is set to '1,' the data is transferred in order from LSB (first) to MSB (last).

The clock signal generated on SPICLK is derived from the internal PERIPH_CLK signal. PERIPH_CLK always operates at the frequency, f_{OSC}, and runs constantly except when stopped in MCU Power Down mode. SPICLK is a result of dividing PERIPH_CLK by a sum of different divisors selected by the value contained in the SPICLKD register. The default value in SPICLKD after a reset divides PERIPH_CLK by a factor of 4. The bits in SPICLKD can be set to provide resulting divisor values in of sums of multiples of 4, such as 4, 8, 12,

Obsolete Product(s) - Obsolete Product(s)

Table 90. SPICON0 register bit definition (continued)

Bit	Symbol	R/W	Definition
3	SSEL	RW	Slave Selection 0 = $\overline{\text{SPISEL}}$ output pin is constant logic '1' (slave device not selected) 1 = $\overline{\text{SPISEL}}$ output pin is logic '0' (slave device is selected) during data transfers
2	FLSB	RW	First LSB 0 = Transfer the most significant bit (MSB) first 1 = Transfer the least significant bit (LSB) first
1	SPO	–	Sampling Polarity 0 = Sample transfer data at the falling edge of clock (SPICLK is '0' when idle) 1 = Sample transfer data at the rising edge of clock (SPICLK is '1' when idle)
0	–	–	Reserved

Table 91. SPICON1: SPI interface control register 1 (SFR D7h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	TEIE	RORIE	TIE	RIE

Table 92. SPICON1 register bit definition

Bit	Symbol	R/W	Definition
7-4	–	–	Reserved
3	TEIE	RW	Transmission End Interrupt Enable 0 = Disable Interrupt for Transmission End 1 = Enable Interrupt for Transmission End
2	RORIE	RW	Receive Overrun Interrupt Enable 0 = Disable Interrupt for Receive Overrun 1 = Enable Interrupt for Receive Overrun
1	TIE	RW	Transmission Interrupt Enable 0 = Disable Interrupt for SPITDR empty 1 = Enable Interrupt for SPITDR empty
0	RIE	RW	Reception Interrupt Enable 0 = Disable Interrupt for SPIRDR full 1 = Enable Interrupt for SPIRDR full

Table 93. SPICLKD: SPI prescaler (clock divider) register (SFR D2h, reset value 04h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIV128	DIV64	DIV32	DIV16	DIV8	DIV4	–	–

Obsolete Product(s) - Obsolete Product(s)

Table 149. PCA0 register bit definition (continued)

Bit	Symbol	Function
3	–	Reserved
2	10B_PWM	0 = Select 16-bit PWM 1 = Select 10-bit PWM
1-0	CLK_SEL [1:0]	00 Select Prescaler clock as Counter clock 01 Select Timer 0 Overflow 10 Select External Clock pin (P4.3 for PCA0) (MAX clock rate = $f_{OSC}/4$)

Table 150. PCA1 control register PCACON1 (SFR 0BCh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	EN_PCA	EOVFI	PCAIDLE	–	–	CLK_SEL[1:0]	

Table 151. PCA1 register bit definition

Bit	Symbol	Function
6	EN_PCA	0 = PCA counter is disabled 1 = PCA counter is enabled EN_PCA Counter Run Control Bit. Set with software to turn the PCA counter on. Must be cleared with software to turn the PCA counter off.
5	EOVFI	1 = Enable Counter Overflow Interrupt if overflow flag (OVF) is set
4	PCAIDLE	0 = PCA operates when CPU is in Idle Mode 1 = PCA stops running when CPU is in Idle Mode
3	–	Reserved
2	10B_PWM	0 = Select 16-bit PWM 1 = Select 10-bit PWM
1-0	CLK_SEL [1:0]	00 Select Prescaler clock as Counter clock 01 Select Timer 0 Overflow 10 Select External Clock pin (P4.7 for PCA1) (MAX clock rate = $f_{OSC}/4$)

Table 152. PCA status register PCASTA (SFR 0A5h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OVF1	INTF5	INTF4	INTF3	OVF0	INTF2	INTF1	INTF0

Table 153. PCASTA register bit definition

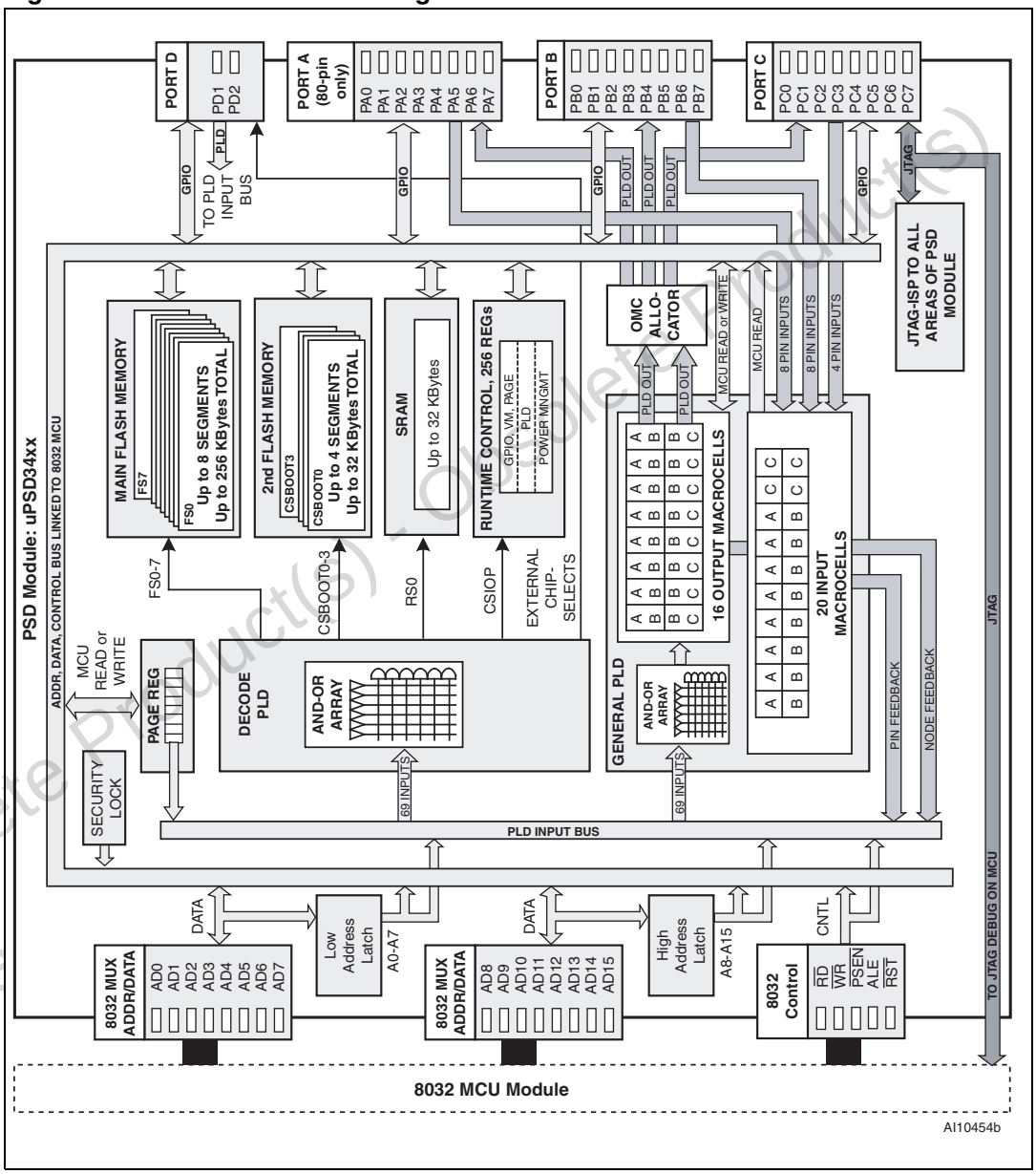
Bit	Symbol	Function
7	OFV1	PCA1 Counter OverFlow flag Set by hardware when the counter rolls over. OVF1 flags an interrupt if Bit EOVFI in PCACON1 is set. OVF1 may be set with either hardware or software but can only be cleared with software.
6	INTF5	TCM5 Interrupt flag Set by hardware when a match or capture event occurs. Must be clear with software.

Obsolete Product(s) - Obsolete Product(s)

28 PSD module

The PSD module is stacked with the MCU module to form the UPSD34xx, see *Section 3: Hardware description on page 28*. Details of the PSD module are shown in *Figure 61*. The two separate modules interface with each other at the 8032 Address, Data, and Control interface blocks in *Figure 61*.

Figure 61. PSD module block diagram



Obsolete Product(s) - Obsolete Product(s)

28.5.29 Output macrocell

The GPLD has 16 OMCs. Architecture of one individual OMC is shown in *Figure 76*. OMCs can be used for internal node feedback (buried registers to build shift registers, etc.), or their outputs may be routed to external port pins. The user can choose any mixture of OMCs used for buried functions and OMCs used to drive port pins.

Referring to *Figure 76*, for each OMC there are native product terms available from the AND-OR Array to form logic, and also borrowed product terms are available (if unused) from other OMCs. The polarity of the final product term output is controlled by the XOR gate. Each OMC can implement sequential logic using the flip-flop element, or combinatorial logic when bypassing the flip-flop as selected by the output multiplexer. An OMC output can drive a port pin through the OMC Allocator, it can also drive the 8032 data bus, and also it can drive a feedback path to the AND-OR Array inputs, all at the same time.

The flip-flop in each OMC can be synthesized as a D, T, JK, or SR type in PSDsoft Express. OMC flip-flops are specified using PSDsoft Express in the “User Defined Nodes” section of the Design Assistant. Each flip-flop’s clock, preset, and clear inputs may be driven individually from a product term of the AND-OR Array, defined by equations in PSDsoft Express for signals *. c, *.pr, and *.re respectively. The preset and clear inputs on the flip-flops are level activated, active-high logic signals. The clock inputs on the flip-flops are rising-edge logic signals.

Optionally, the signal CLKIN (pin PD1) can be used for a common clock source to all OMC flip-flops. Each flip-flop is clocked on the rising edge. A common clock is specified in PSDsoft Express by assigning the function “Common Clock Input” for pin PD1 in the Pin Definition section, and then choosing the signal CLKIN when specifying the clock input (*.c) for individual flip-flops in the “User Defined Nodes” section.

Obsolete Product(s) - Obsolete Product(s)

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.

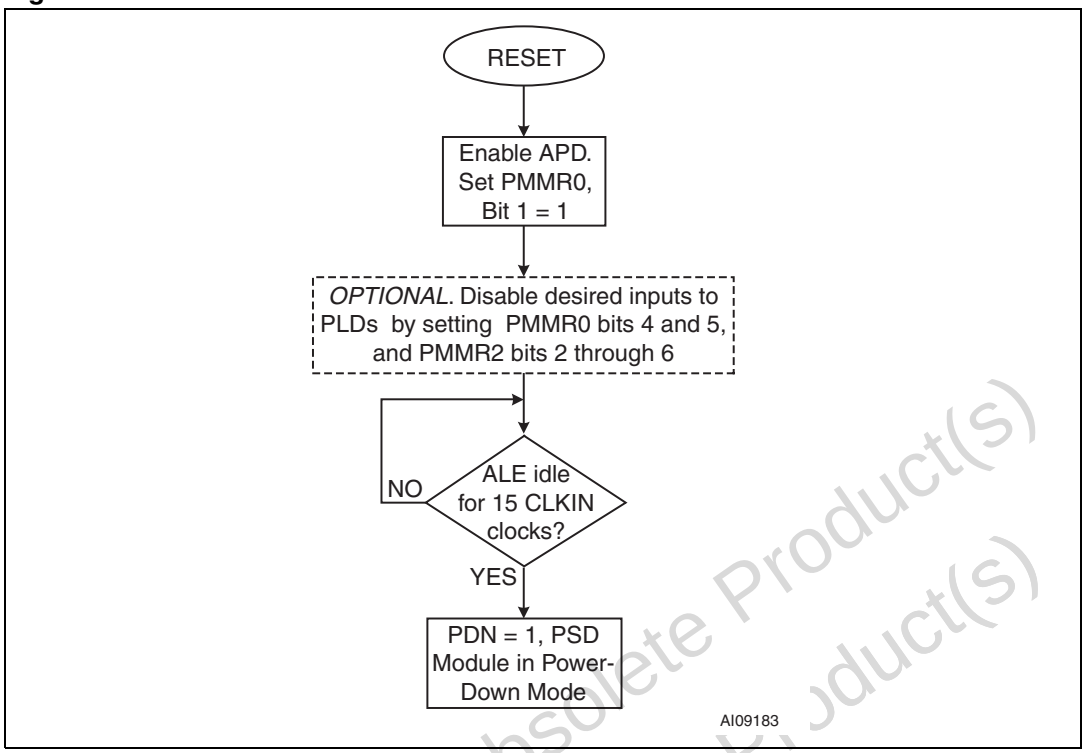
As shown in *Figure 79 on page 232*, a physical port pin is connected to the I/O Port logic and is also separately routed to an IMC, allowing the 8032 to read a port pin by two different methods (MCU I/O input mode or read the IMC).

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

Obsolete Product(s) - Obsolete Product(s)

Figure 89. Power-down mode flowchart



28.5.55 Chip select input ($\overline{CSI}$)

Pin PD2 of Port D can optionally be configured in PSDsoft Express as the PSD module Chip Select Input, $\overline{CSI}$, which is an active-low logic input. By default, pin PD2 does not have the $\overline{CSI}$ function.

When the $\overline{CSI}$ function is specified in PSDsoft Express, the $\overline{CSI}$ signal is automatically included in DPLD chip select equations for FSx, CSBOOTx, RS0, and CSIOP. When the $\overline{CSI}$ pin is driven to logic '0' from an external device, all of these memories will be available for READ and WRITE operations. When $\overline{CSI}$ is driven to logic '1,' none of these memories are available for selection, regardless of the address activity from the 8032, reducing power consumption. The state of the PLD and port I/O pins are not changed when $\overline{CSI}$ goes to logic '1' (disabled).

28.5.56 PLD non-turbo mode

The power consumption and speed of the PLDs are controlled by the Turbo Bit (Bit 3) in the csiop PMMR0 register. By setting this bit to logic '1,' the Turbo mode is turned off and both PLDs consume only standby current when ALL PLD inputs have no transitions for an extended time (65 ns for 5 V devices, 100 ns for 3.3 V devices), significantly reducing current consumption. The PLDs will latch their outputs and go to standby, drawing very little current. When Turbo mode is off, PLD propagation delay time is increased as shown in the AC specifications for the PSD module. Since this additional propagation delay also effects the DPLD, the response time of the memories on the PSD module is also lengthened by that same amount of time. If Turbo mode is off, the user should add an additional wait state to the 8032 BUSCON SFR register if the 8032 clock frequency is higher that a particular value. Please refer to *Table 51 on page 90* in the MCU module section.

Obsolete Product(s) - Obsolete Product(s)

28.5.60 PLD blocking bits

Blocking specific signals from entering the PLDs using bits of the csiop PMMR registers can further reduce PLD AC current consumption by lowering the effective composite frequency of inputs to the PLDs.

28.5.61 Blocking 8032 bus control signals

When the 8032 is active on the MCU module, four bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, and ALE) are constantly transitioning to manage 8032 bus traffic. Each time one of these signals has a transition from logic ’1’ to '0,' or 0 to '1,' it will wake up the PLDs if operating in non-Turbo mode, or when in Turbo mode it will cause the affected PLD gates to draw current. If equations in the DPLD or GPLD do not use the signals $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, or ALE then these signals can be blocked which will reduce the AC current component substantially. These bus control signals are rarely used in DPLD equations because they are routed in silicon directly to the memory arrays of the PSD module, bypassing the PLDs. For example, it is NOT necessary to qualify a memory chip select signal with an MCU write strobe, such as “fs0 = address range & !WR_”. Only “fs0 = address range” is needed.

Each of the 8032 bus control signals may be blocked individually by writing to Bits 2, 3, 4, and 5 of the PMMR2 register shown in *Table 201 on page 249*. Blocking any of these four bus control signals only prevents them from reaching the PLDs, but they will always go to the memories directly.

However, sometimes it is necessary to use these 8032 bus control signals in the GPLD when creating interface signals to external I/O peripherals. But it is still possible to save power by dynamically unblocking the bus signals before reading/writing the external device, then blocking the signals after the communication is complete.

The user can also block an input signal coming from pin PC7 to the PLD input bus if desired by writing to Bit 6 of PMMR2.

28.5.62 Blocking common clock, CLKIN

The input CLKIN (from pin PD1) can be blocked to reduce current consumption. CLKIN is used as a common clock input to all OMC flip-flips, it is a general input to the PLD input bus, and it is used to clock the APD counter. In PSDsoft Express, the function of pin PD1 must be specified as “Common Clock Input, CLKIN” before programming the device with JTAG to get the CLKIN function.

Bit 4 of PMMR0 can be set to logic ’1’ to block CLKIN from reaching the PLD input bus, but CLKIN will still reach the APD counter.

Bit 5 of PMMR0 can be set to logic ’1’ to block CLKIN from reaching the OMC flip-flops only, but CLKIN is still available to the PLD input bus and the APD counter.

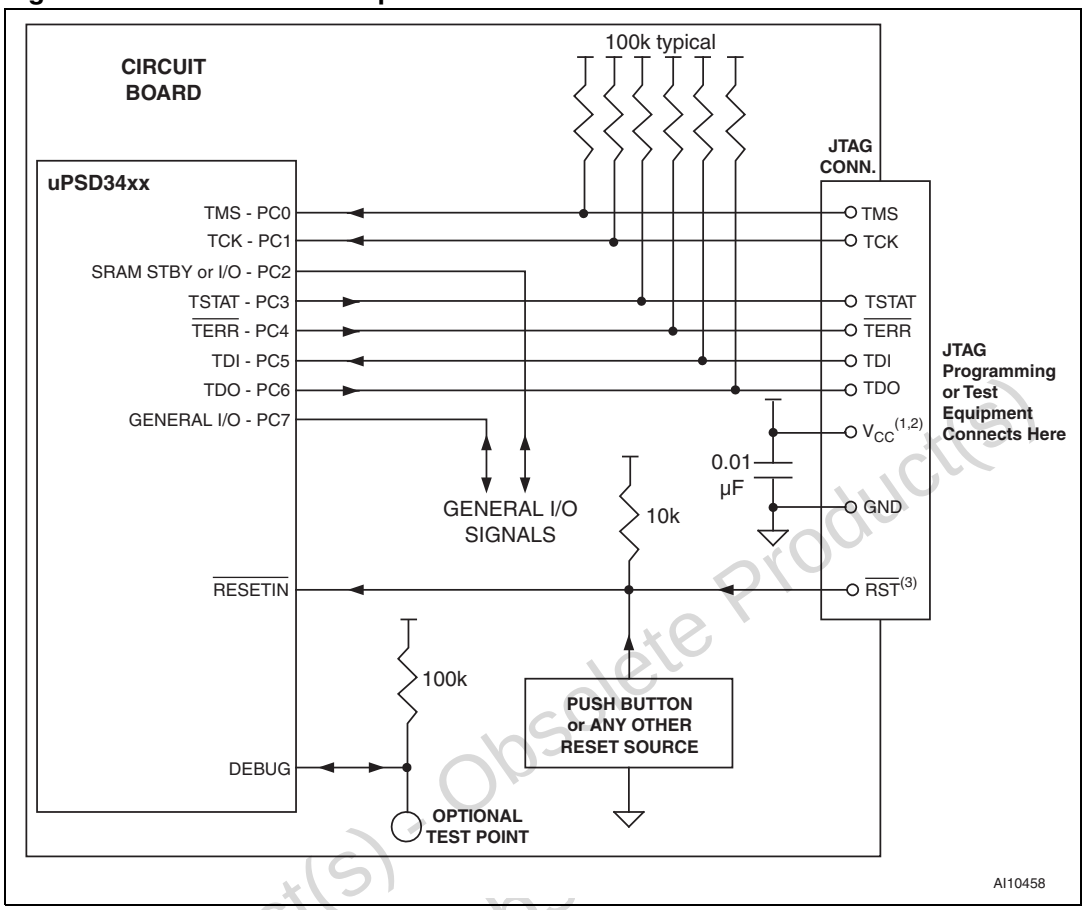
See *Table 200 on page 249* for details.

28.6 PSD module reset conditions

The PSD module receives a reset signal from the MCU module. This reset signal is referred to as the “$\overline{RST}$” input in PSD module documentation, and it is active-low when asserted. The character of the $\overline{RST}$ signal generated from the MCU module is described in *Section 19: Supervisory functions on page 91*.

Obsolete product(s) - Obsolete product(s)

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.

Obsolete Product(s) - Obsolete Product(s)

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).*

Obsolete Product(s) - Obsolete Product(s)
Obsolete Product(s) - Obsolete Product(s)

Table 210. Major parameters

Parameter	Test conditions/comments	5.0 V value	3.3 V value	Unit
Operating voltage	–	4.5 to 5.5 (PSD); 3.0 to 3.6 (MCU)	3.0 to 3.6 (PSD and MCU)	V
Operating temperature	–	–40 to 85	–40 to 85	°C
MCU frequency	8 MHz (min) for I^2C	3 Min, 40 Max	3 Min, 40 Max	MHz
Operating current, typical(1) (20% of PLD used; 25°C operation. Bus control signals are blocked from the PLD in Non-turbo mode.)	40 MHz crystal, Turbo	79	63	mA
	40 MHz crystal, non-Turbo	71	58	mA
	8 MHz crystal, Turbo	32	24	mA
	8 MHz crystal, non-Turbo	17.7	14	mA
Idle current, typical (20% of PLD used; 25°C operation)	40 MHz crystal divided by 2048 internally. All interfaces are disabled.	19	18	mA
Standby current, typical	Power-down mode needs reset to exit.	140	120	µA
I/O sink/source current, ports A, B, C, and D	V_{OL} = 0.45 V (max); V_{OH} = 2.4 V (min)	I_{OL} = 8 (max); I_{OH} = –2 (min)	I_{OL} = 4 (max); I_{OH} = –1 (min)	mA
I/O sink/source current, port 4	V_{OL} = 0.6 V (max); V_{OH} = 2.4 V (min)	I_{OL} = 10 (max); I_{OH} = –10 (min)	I_{OL} = 10 (max); I_{OH} = –10 (min)	mA
PLD macrocells	For registered or combinatorial logic	16	16	–
PLD inputs	Inputs from pins, feedback, or MCU addresses	69	69	–
PLD outputs	Output to pins or internal feedback	18	18	–
PLD propagation delay, typical, Turbo mode	PLD input to output	15	22	ns

1. Operating current is measured while the UPSD34xx is executing a typical program at 40 MHz.

Table 211. MCU module DC characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_{CC}, AV_{CC}	Supply voltage(1)		3.0		3.6	V
V_{IH}	High level input voltage (ports 0, 1, 3, 4, XTAL1, RESET) 5 V tolerant - max voltage 5.5 V	3.0 V < V_{CC} < 3.6 V	0.7V_{CC}		5.5(2)	V
V_{IL}	Low level input voltage (ports 0, 1, 3, 4, XTAL1, RESET)	3.0 V < V_{CC} < 3.6 V	V_{SS} – 0.5		0.3V_{CC}	V
V_{OL1}	Output low voltage (port 4)	I_{OL} = 10 mA			0.6	V
						V

Obsolete Product(s) - Obsolete Product(s)