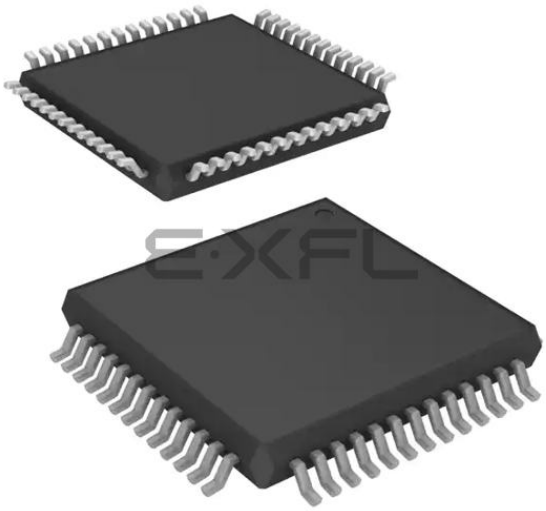


Welcome to E-XFL.COM

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
Peripherals	LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3354d-40t6

Table 10. Program branching instruction set (continued)

Mnemonic[1] and use		Description	Length/cycles
CJNE	Rn, #data, rel	Compare immediate to register, jump if not equal	3 byte/2 cycle
CJNE	@Ri, #data, rel	Compare immediate to indirect, jump if not equal	3 byte/2 cycle
DJNZ	Rn, rel	Decrement register and jump if not zero	2 byte/2 cycle
DJNZ	direct, rel	Decrement direct byte and jump if not zero	3 byte/2 cycle

1. All mnemonics copyrighted ©Intel Corporation 1980.

Table 11. Miscellaneous instruction set

Mnemonic[1] and use		Description	Length/cycles
NOP		No operation	1 byte/1 cycle

1. All mnemonics copyrighted ©Intel Corporation 1980.

Table 12. Notes on instruction set and addressing modes

Rn	Register R0 - R7 of the currently selected register bank.
direct	8-bit address for internal 8032 DATA SRAM (locations 00h - 7Fh) or SFR registers (locations 80h - FFh).
@Ri	8-bit internal 8032 SRAM (locations 00h - FFh) addressed indirectly through contents of R0 or R1.
#data	8-bit constant included within the instruction.
#data16	16-bit constant included within the instruction.
addr16	16-bit destination address used by LCALL and LJMP.
addr11	11-bit destination address used by ACALL and AJMP.
rel	Signed (two-s compliment) 8-bit offset byte.
bit	Direct addressed bit in internal 8032 DATA SRAM (locations 20h to 2Fh) or in SFR registers (88h, 90h, 98h, A8h, B0, B8h, C0h, C8h, D0h, D8h, E0h, F0h).

event signal. If not used, the Debug Event pin should be pulled up to V_{CC} as described in *Section 27.5.8: Debugging the 8032 MCU module on page 240*.

- The duration of a pulse, generated when the Event pin configured as an output, is one MCU clock cycle. This is an active-low signal, so the first edge when an event occurs is high-to-low.
- The clock to the Watchdog timer, ADC, and I^2C interface are not stopped by a breakpoint halt.
- The Watchdog timer should be disabled while debugging with JTAG, else a reset will be generated upon a watchdog timeout.

13.1 Individual interrupt sources

13.1.1 External interrupts Int0 and Int1

External interrupt inputs on pins EXTINT0 and EXTINT1 (pins 3.2 and 3.3) are either edge-triggered or level-triggered, depending on bits IT0 and IT1 in the SFR named TCON.

When an external interrupt is generated from an edge-triggered (falling-edge) source, the appropriate flag bit (IE0 or IE1) is automatically cleared by hardware upon entering the ISR.

When an external interrupt is generated from a level-triggered (low-level) source, the appropriate flag bit (IE0 or IE1) is NOT automatically cleared by hardware.

13.1.2 Timer 0 and 1 overflow interrupt

Timer 0 and Timer 1 interrupts are generated by the flag bits TF0 and TF1 when there is an overflow condition in the respective Timer/Counter register (except for Timer 0 in mode 3).

13.1.3 Timer 2 overflow interrupt

This interrupt is generated to the MCU by a logical OR of flag bits, TF2 and EXE2. The ISR must read the flag bits to determine the cause of the interrupt.

- TF2 is set by an overflow of Timer 2.
- EXE2 is generated by the falling edge of a signal on the external pin, T2X (pin P1.1).

13.1.4 UART0 and UART1 interrupt

Each of the UARTs have identical interrupt structure. For each UART, a single interrupt is generated to the MCU by the logical OR of the flag bits, RI (byte received) and TI (byte transmitted).

The ISR must read flag bits in the SFR named SCON0 for UART0, or SCON1 for UART1 to determine the cause of the interrupt.

13.1.5 SPI interrupt

The SPI interrupt has four interrupt sources, which are logically ORed together when interrupting the MCU. The ISR must read the flag bits to determine the cause of the interrupt.

A flag bit is set for: end of data transmit (TEISF); data receive overrun (RORISF); transmit buffer empty (TISF); or receive buffer full (RISF).

13.1.6 I^2C interrupt

The flag bit INTR is set by a variety of conditions occurring on the I^2C interface: received own slave address (ADDR flag); received general call address (GC flag); received STOP condition (STOP flag); or successful transmission or reception of a data byte.The ISR must read the flag bits to determine the cause of the interrupt.

13.1.7 ADC interrupt

The flag bit AINTF is set when an A-to-D conversion has completed.

19.2 Low V_{CC} voltage detect, LVD

An internal reset is generated by the LVD circuit when V_{CC} drops below the reset threshold, V_{LV_THRESH}. After V_{CC} returns to the reset threshold, the MCU_RESET signal will remain asserted for t_{RST_ACTV} before it is released. The LVD circuit is always enabled (cannot be disabled by SFR), even in Idle mode and Power-down mode. The LVD input has a voltage hysteresis of V_{RST_HYS} and will reject voltage spikes less than a duration of t_{RST_FIL}.

Note: ***Important:*** *The LVD voltage threshold is V_{LV_THRESH}, suitable for monitoring both the 3.3 V V_{CC} supply on the MCU module and the 3.3 V V_{DD} supply on the PSD module for 3.3 V UPSD33xxV devices, since these supplies are one in the same on the circuit board.*

However, for 5 V UPSD33xx devices, V_{LV_THRESH} is not suitable for monitoring the 5 V V_{DD} voltage supply (V_{LV_THRESH} is too low), but good for monitoring the 3.3 V V_{CC} supply. In the case of 5 V UPSD33xx devices, an external means is required to monitor the separate 5 V V_{DD} supply, if desired.

19.3 Power-up reset

At power up, the internal reset generated by the LVD circuit is latched as a logic '1' in the POR bit of the SFR named PCON (*Table 31 on page 72*). Software can read this bit to determine whether the last MCU reset was the result of a power up (cold reset) or a reset from some other condition (warm reset). This bit must be cleared with software.

19.4 JTAG debug Reset

The JTAG debug unit can generate a reset for debugging purposes. This reset source is also available when the MCU is in Idle mode and Power-down mode (the JTAG debugger can be used to exit these modes).

19.5 Watchdog timer (WDT)

When enabled, the WDT will generate a reset whenever it overflows. Firmware that is behaving correctly will periodically clear the WDT before it overflows. Run-away firmware will not be able to clear the WDT, and a reset will be generated.

By default, the WDT is disabled after each reset.

Note: The WDT is not active during Idle mode or Power-down mode.

There are two SFRs that control the WDT, they are WDKEY (*Table 50 on page 92*) and WDRST (*Table 52 on page 92*).

If WDKEY contains 55h, the WDT is disabled. Any value other than 55h in WDKEY will enable the WDT. By default, after any reset condition, WDKEY is automatically loaded with 55h, disabling the WDT. It is the responsibility of initialization firmware to write some value other than 55h to WDKEY after each reset if the WDT is to be used.

The WDT consists of a 24-bit up-counter (*Figure 20*), whose initial count is 000000h by default after every reset. The most significant byte of this counter is controlled by the SFR, WDRST. After being enabled by WDKEY, the 24-bit count is increased by 1 for each MCU machine cycle. When the count overflows beyond FFFFFh (2^{24} MCU machine cycles), a reset is issued and the WDT is automatically disabled (WDKEY = 55h again).

Figure 29. UART mode 1, block diagram

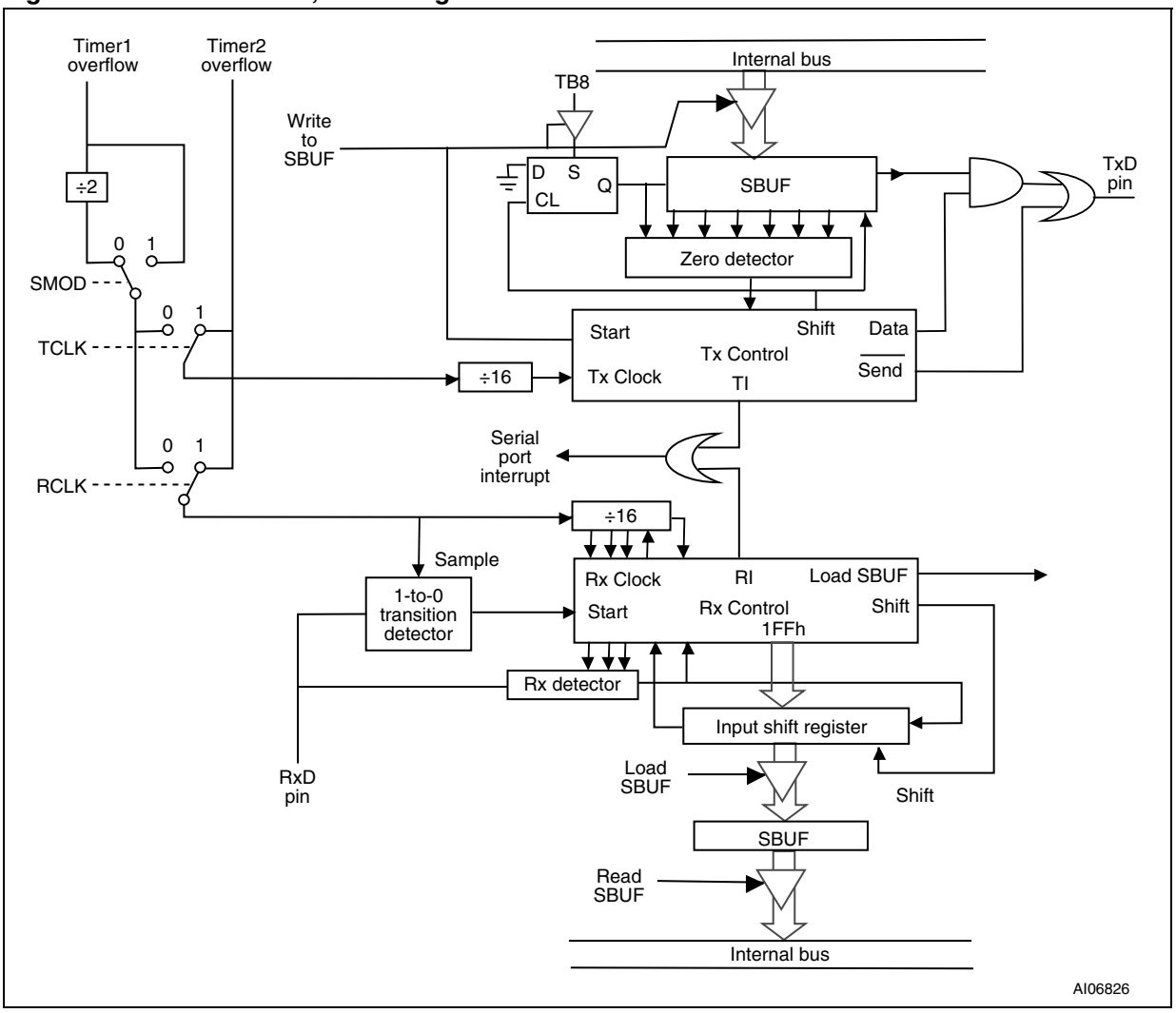
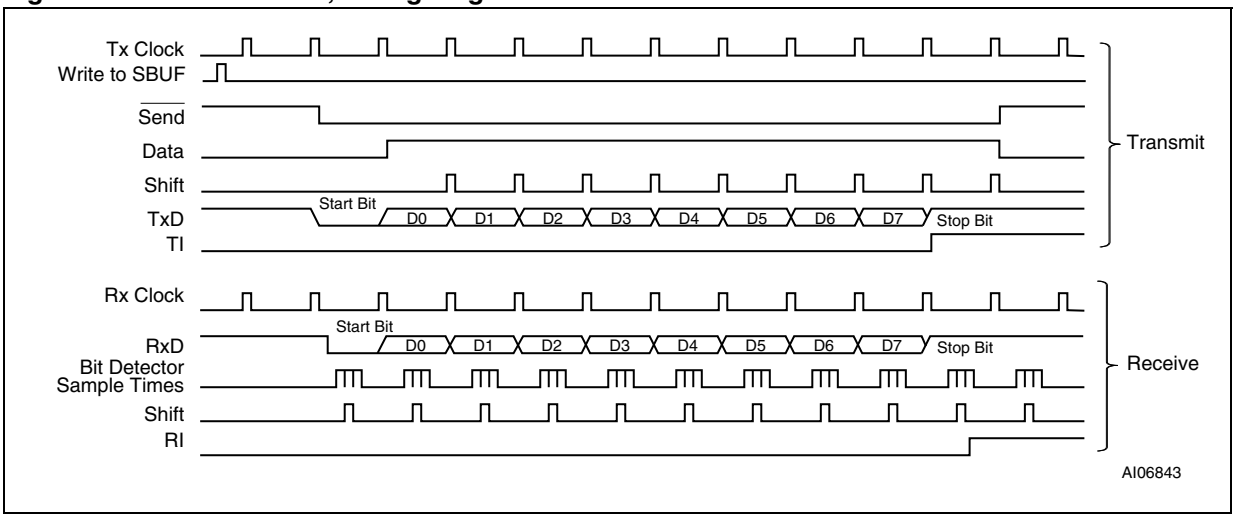


Figure 30. UART mode 1, timing diagram



23 I²C interface

UPSD33xx devices support one serial I²C interface. This is a two-wire communication channel, having a bi-directional data signal (SDA, pin P3.6) and a clock signal (SCL, pin P3.7) based on open-drain line drivers, requiring external pull-up resistors, R_P, each with a typical value of 4.7kΩ (see *Figure 37*).

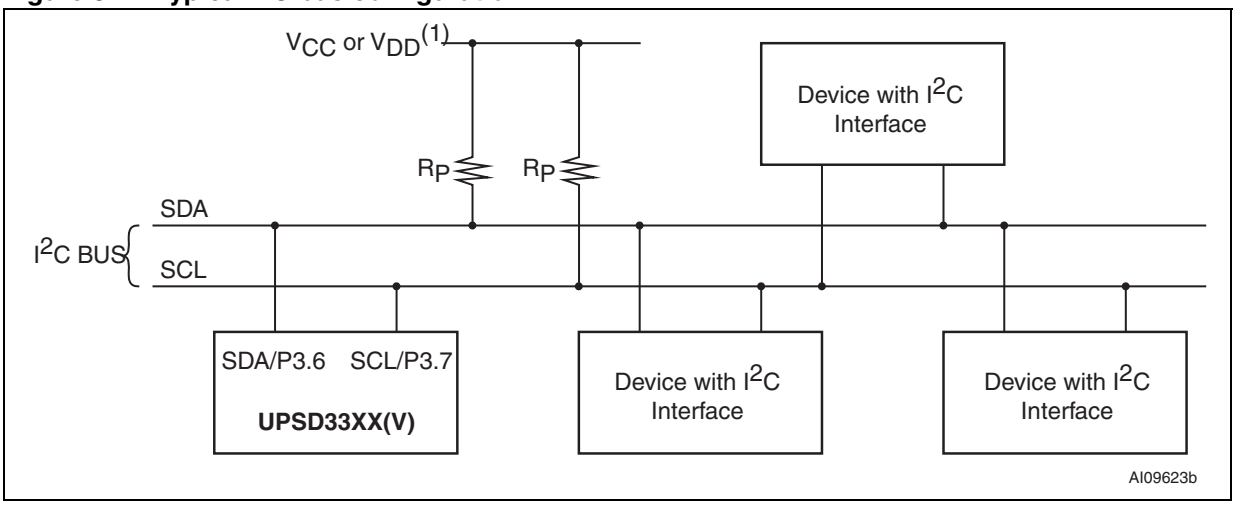
23.1 I²C interface main features

Byte-wide data is transferred, MSB first, between a Master device and a Slave device on two wires. More than one bus Master is allowed, but only one Master may control the bus at any given time. Data is not lost when another Master requests the use of a busy bus because I²C supports collision detection and arbitration. The bus Master initiates all data movement and generates the clock that permits the transfer. Once a transfer is initiated by the Master, any device addressed is considered a Slave. Automatic clock synchronization allows I²C devices with different bit rates to communicate on the same physical bus. A single device can play the role of Master or Slave, or a single device can be a Slave only. Each Slave device on the bus has a unique address, and a general broadcast address is also available. A Master or Slave device has the ability to suspend data transfers if the device needs more time to transmit or receive data.

This I²C interface has the following features:

- Serial I/O Engine (SIOE): serial/parallel conversion; bus arbitration; clock generation and synchronization; and handshaking are all performed in hardware
- Interrupt or Polled operation
- Multi-master capability
- 7-bit Addressing
- Supports standard speed I²C (SCL up to 100kHz), fast mode I²C (101kHz to 400kHz), and high-speed mode I²C (401kHz to 833kHz)

Figure 37. Typical I²C bus configuration



1. For 3.3 V system, connect R_P to 3.3 V V_{CC}. For 5.0 V system, connect R_P to 5.0 V V_{DD}.

Set bus START condition sampling

- SFR S1SETUP[7:0] = number of samples

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

Else If mode is Master-Receiver

Bus Arbitration lost? (status.BLOST=1?)

If Yes, Arbitration was lost:

- S1DAT = dummy, write to release bus
- Exit ISR, SIOE will switch to Slave Recv mode

If No, Aribitration was not lost, continue:

Is this Interrupt from sending an address to Slave, or is it from receiving a data byte from Slave?

If its from sending Slave address, goto A:

If its from receiving Slave data, goto B:

A: (Interrupt is from Master sending addr to Slave)

ACK recvd from Slave? (status.ACK_RESP=0?)

If No, an ACK was not received:

- S1CON.STO = 1, set STOP condition

<STOP occurs after ISR exit>

- dummy = S1DAT, read to release bus
- Exit ISR

If Yes, ACK was received, then continue:

- dummy = S1DAT, read to release bus

Does Master want to receive just one data byte?

If Yes, do not allow Master to ACK on next interrupt: <S1CON.AA is already 0>

- Exit ISR, now ready to recv one byte from Slv

If No, Master can ACK next byte from Slv

- S1CON.AA = 1, allow Master to send ACK
- Exit ISR, now ready to recv data from Slave

B: (Interrupt is from Master recving data from Slv)

- recv_buf[buffer_index] = S1DAT, read byte

Is this the last data byte to receive from Slave?

If Yes, tell Slave to stop transmitting:

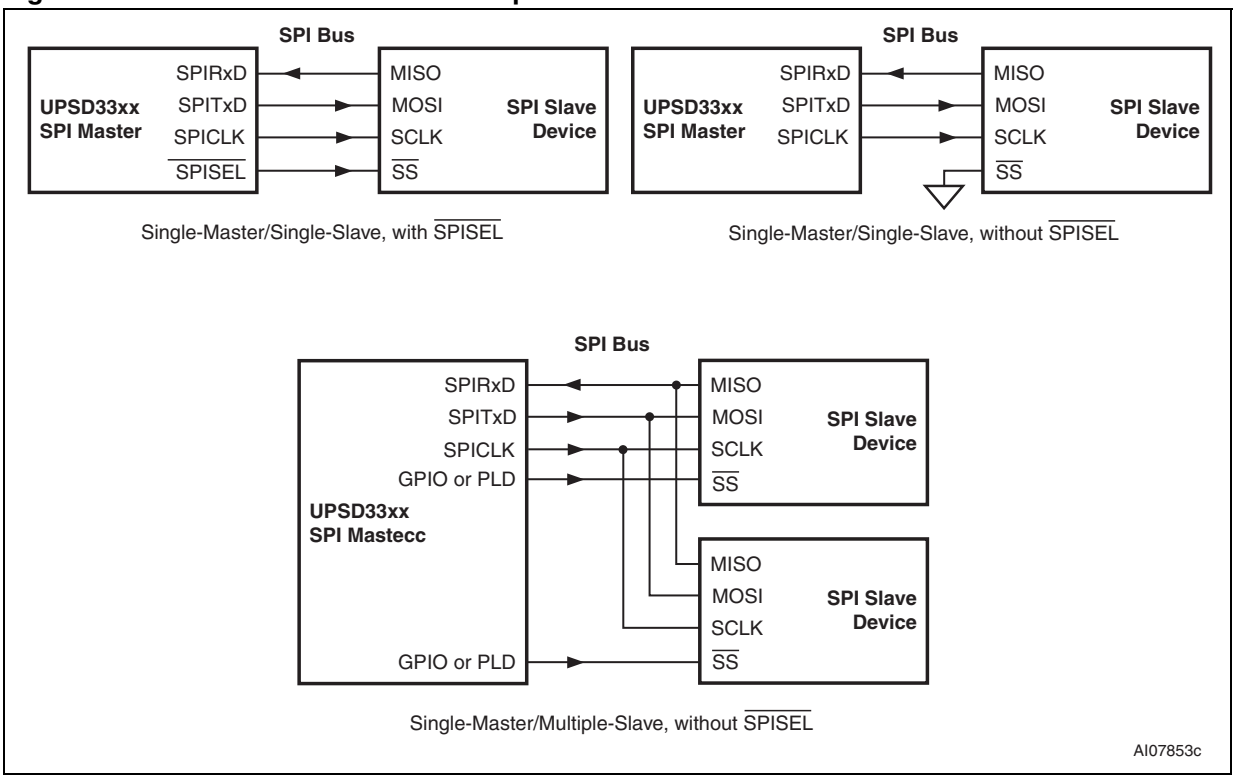
- S1CON.STO = 1, set STOP bus condition

<STOP occurs after ISR exit>

- Exit ISR, finished receiving data from Slave

If No, continue:

Figure 40. SPI device connection examples



24.1 SPI bus features and communication flow

The SPICLK signal is a gated clock generated from the UPSD33xx (Master) and regulates the flow of data bits. The Master may transmit at a variety of baud rates, and the SPICLK signal will clock one period for each bit of transmitted data. Data is shifted on one edge of SPICLK and sampled on the opposite edge.

The SPITxD signal is generated by the Master and received by the Slave device. The SPIRxD signal is generated by the Slave device and received by the Master. There may be no more than one Slave device transmitting data on SPIRxD at any given time in a multi-Slave configuration. Slave selection is accomplished when a Slave's "Slave Select" (SS) input is permanently grounded or asserted active-low by a Master device. Slave devices that are not selected do not interfere with SPI activities. Slave devices ignore SPICLK and keep their MISO output pins in high-impedance state when not selected.

The SPI specification allows a selection of clock polarity and clock phase with respect to data. The UPSD33xx supports the choice of clock polarity, but it does not support the choice of clock phase (phase is fixed at what is typically known as CPHA = 1). See *Figure 42* and *Figure 43 on page 144* for SPI data and clock relationships.

Referring to these figures (*42* and *43*), when the phase mode is defined as such (fixed at CPHA =1), in a new SPI data frame, the Master device begins driving the first data bit on SPITxD at the very first edge of the first clock period of SPICLK.

The Slave device will use this first clock edge as a transmission start indicator, and therefore the Slave's Slave Select input signal may remain grounded in a single-Master/single-Slave

Table 91. SPISTAT: SPI Interface Status register (SFR D3h, Reset Value 02h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	BUSY	TEISF	RORISF	TISF	RISF

Table 92. SPISTAT register bit definition

Bit	Symbol	R/W	Definition
7-5	–	–	Reserved
4	BUSY	R	**SPI Busy** 0 = Transmit or Receive is completed 1 = Transmit or Receive is in process
3	TEISF	R	**Transmission End Interrupt Source flag** 0 = Automatically resets to '0' when firmware reads this register 1 = Automatically sets to '1' when transmission end occurs
2	RORISF	R	**Receive Overrun Interrupt Source flag** 0 = Automatically resets to '0' when firmware reads this register 1 = Automatically sets to '1' when receive overrun occurs
1	TISF	R	**Transfer Interrupt Source flag** 0 = Automatically resets to '0' when SPITDR is full (just after the SPITDR is written) 1 = Automatically sets to '1' when SPITDR is empty (just after byte loads from SPITDR into SPI shift register)
0	RISF	R	**Receive Interrupt Source flag** 0 = Automatically resets to '0' when SPIRDR is empty (after the SPIRDR is read) 1 = Automatically sets to '1' when SPIRDR is full

26.10 PWM mode - fixed frequency, 10-bit

The 10-bit PWM logic requires that all 3 TCMs in PCA0 or PCA1 operate in the same 10-bit PWM mode. The 10-bit PWM operates in a similar manner as the 16-bit PWM, except the PCACHm and PCACLm counters are reconfigured as 10-bit counters. The CAPCOMHn and CAPCOMLn registers become 10-bit registers.

PWM duty cycle of each TCM module can be specified in the 10-bit CAPCOMHn and CAPCOMLn registers. When the 10-bit PCA counter is equal or greater than the values in the 10-bit registers CAPCOMHn and CAPCOMLn, the PWM output switches to a high state. When the 10-bit PCA counter overflows, the PWM pin is switched to a logic low and starts the next PWM pulse.

The most-significant 6 bits in the PCACHm counter and CAPCOMH register are “Don’t cares” and have no effect on the PWM generation.

26.11 Writing to capture/compare registers

When writing a 16-bit value to the PCA Capture/Compare registers, the low byte should always be written first. Writing to CAPCOMLn clears the E_COMP Bit to '0'; writing to CAPCOMHn sets E_COMP to '1' the largest duty cycle is 100% (CAPCOMHn CAPCOMLn = 0x0000), and the smallest duty cycle is 0.0015% (CAPCOMHn CAPCOMLn = 0xFFFF). A 0% duty cycle may be generated by clearing the E_COMP Bit to ‘0’.

26.12 Control register bit definition

Each PCA has its own PCA_CONFIGn, and each module within the PCA block has its own TCM_Mode register which defines the operation of that module (see *Table 103 on page 159* through *Table 105 on page 160*). There is one PCA_STATUS register that covers both PCA0 and PCA1 (see *Table 107 on page 161*).

Table 103. PCA0 Control register PCACON0 (SFR 0A4h, Reset Value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EN-ALL	EN_PCA	EOVFI	PCAIDLE	–	10B_PWM	CLK_SEL[1:0]	

Table 104. PCACON0 register bit definition

Bit	Symbol	Function
7	EN-ALL	0 = No impact on TCM modules 1 = Enable both PCA counters simultaneously (override the EN_PCA Bits) This bit is to start the two 16-bit counters in the PCA. For customers who want 5 PWM, for example, this bit can start all of the PWM outputs.
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.

27.4.26 Decode PLD (DPLD)

The DPLD (*Figure 63 on page 197*) generates the following memory decode signals:

- Eight main Flash memory sector select signals (FS0-FS7) with three product terms each
- Four secondary Flash memory sector select signals (CSBOOT0-CSBOOT3) with three product terms each
- One SRAM select signal (RS0) with two product terms
- One select signal for the base address of 256 PSD module device control and status registers (CSIOP) with one product term
- Two external chip-select output signals for Port D pins, each with one product term (52-pin devices only have one pin on Port D)
- Two chip-select signals (PSEL0, PSEL1) used to enable the 8032 data bus repeater function (Peripheral I/O mode) for Port A on 80-pin devices. Each has one product term.

A product term indicates the logical OR of two or more inputs. For example, three product terms in a DPLD output means the final output signal is capable of representing the logical OR of three different input signals, each input signal representing the logical AND of a combination of the 69 PLD inputs.

Using the signal FS0 for example, the user may create a 3-product term chip select signal that is logic true when any one of three different address ranges are true... FS0 = address range 1 OR address range 2 OR address range 3.

The phrase “one product term” is a bit misleading, but commonly used in this context. One product term is the logical AND of two or more inputs, with no OR logic involved at all, such as the CSIOP signal in *Figure 63 on page 197*.

logic where more product terms may be available. The following list summarizes how product terms are allocated to each OMC, as shown in *Table 122 on page 203*.

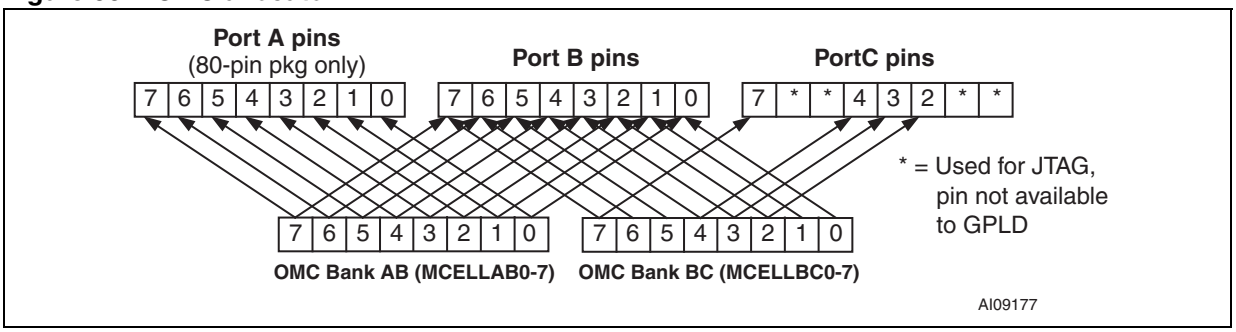
- MCELLAB0-MCELLAB7 each have three native product terms and may borrow up to six more
- MCELLBC0-MCELLBC3 each have four native product terms and may borrow up to five more
- MCELLBC4-MCELLBC7 each have four native product terms and may borrow up to six more.

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 it’s 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 occurrence. 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 66. OMC allocator

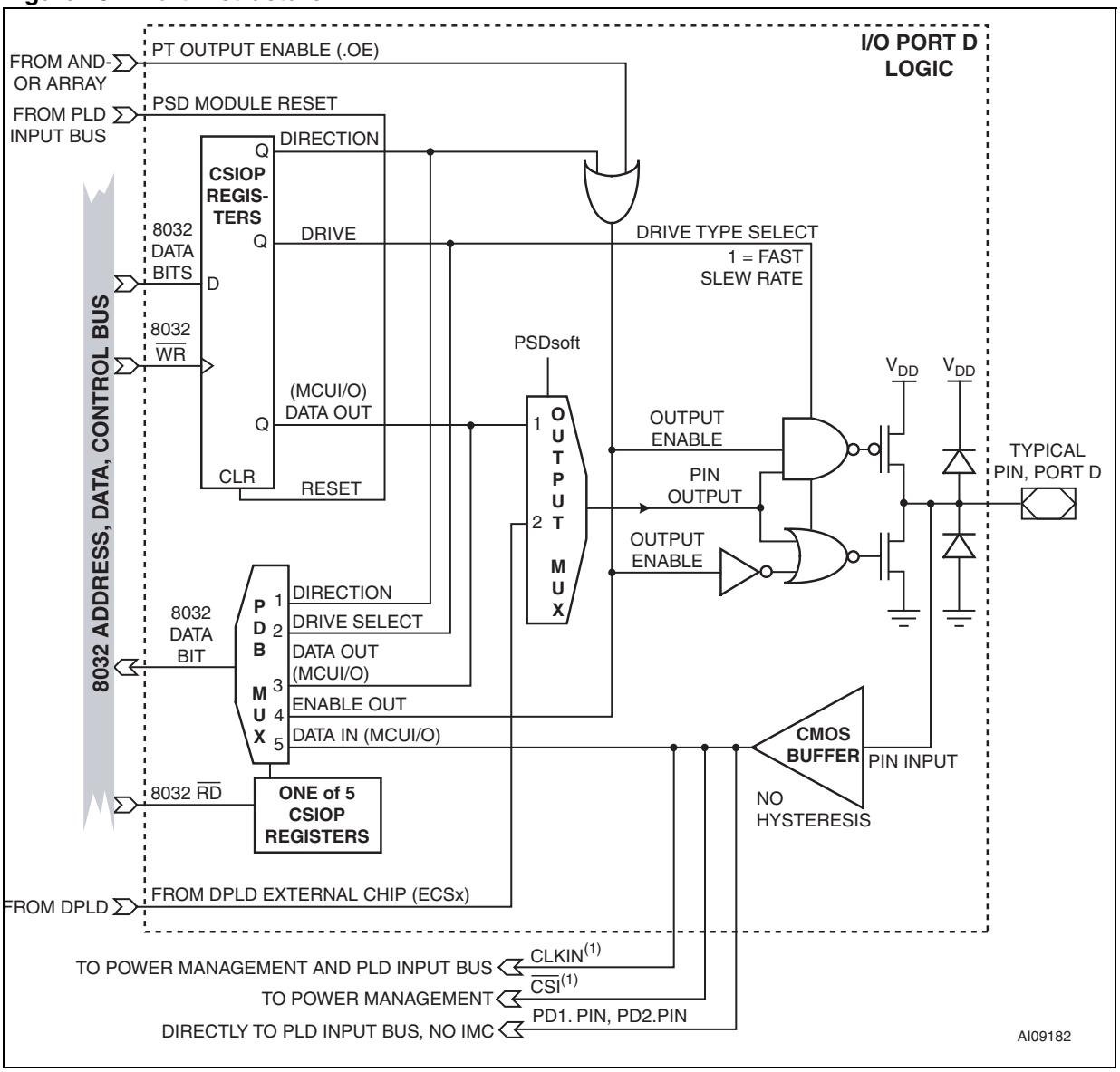


Port D pins can also be configured in PSDsoft as pins for other dedicated functions:

- PD1 can be used as a common clock input to all 16 OMC Flip-flops (see *Section 27.1.11: OMCs on page 168*) and also *Section 27.4.52: Automatic Power-down (APD) on page 227*.
- PD2 can be used as a common chip select signal ($\overline{CSI}$) for the Flash and SRAM memories on the PSD module (see *Section 27.4.54: Chip Select Input (CSI) on page 230*). If driven to logic '1' by an external source, $\overline{CSI}$ will force all memories into standby mode regardless of what other internal memory select signals are doing on the PSD module. This is specified in PSDsoft as "PSD Chip Select Input, $\overline{CSI}$".

Port D also supports the Fast Slew Rate output drive type option using the csiop Drive Select registers.

Figure 76. Port D structure



1. Optional function on a specific Port D pin.

27.5.5 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 81 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 UPSD33xx. 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 81.

Table 170. n, m, and x, y values

# of PFQCLK in BUSCON register	PSEN (code) cycle		READ cycle		WRITE cycle	
	n	m	n	m	x	y
3	1	2	-	-	-	-
4	2	3	2	3	2	1
5	3	4	3	4	3	2
6	4	5	4	5	4	3
7	-	-	5	6	5	4

Figure 88. External WRITE cycle (80-pin device only)

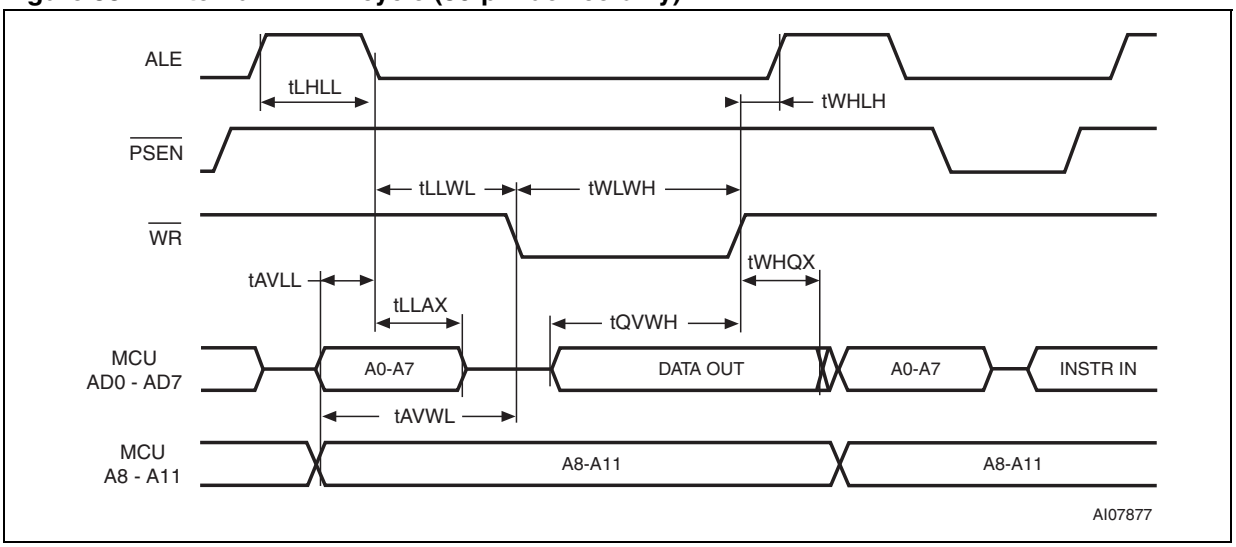


Table 171. External WRITE cycle AC characteristics (3 V or 5 V device)

Symbol	Parameter	40 MHz oscillator(1)		Variable oscillator $1/t_{CLCL}$ = 8 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{LHLL}	ALE pulse width	17		$t_{CLCL} - 8$		ns
t_{AVLL}	Address Setup to ALE	13		$t_{CLCL} - 12$		ns
t_{LLAX}	Address hold after ALE	7.5		$0.5t_{CLCL} - 5$		ns
t_{WLWH}	WR pulse width(2)	40		$xt_{CLCL} - 10$		ns
t_{LLWL}	ALE to $\overline{WR}$	7.5		$0.5t_{CLCL} - 5$		ns
t_{AVWL}	Address valid to $\overline{WR}$	27.5		$1.5t_{CLCL} - 10$		ns
t_{WHLH}	$\overline{WR}$ high to ALE high	6.5	14.5	$0.5t_{CLCL} - 6$	$0.5t_{CLCL} + 2$	ns
t_{QVWH}	Data setup before $\overline{WR}$(y)	20		$yt_{CLCL} - 5$		ns
t_{WHQX}	Data hold after $\overline{WR}$	6.5	14.5	$0.5t_{CLCL} - 6$	$0.5t_{CLCL} + 2$	ns

1. BUSCON register is configured for 4 PFQCLK.
2. Refer to *Table 172* for “n” and “m” values.

Figure 96. ISC timing

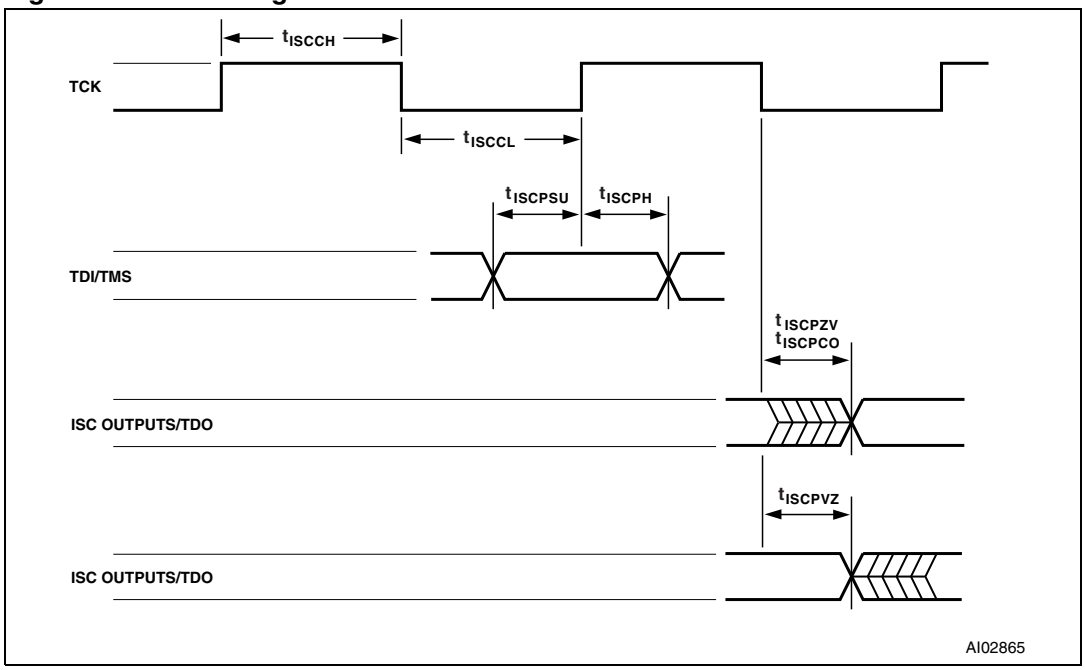


Table 188. ISC timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Unit
t_{ISCCF}	Clock (TCK, PC1) frequency (except for PLD)	(1)		20	MHz
t_{ISCCH}	Clock (TCK, PC1) high time (except for PLD)	(1)	23		ns
t_{ISCCL}	Clock (TCK, PC1) low time (except for PLD)	(1)	23		ns
t_{ISCCFP}	Clock (TCK, PC1) frequency (PLD only)	(2)		4	MHz
t_{ISCCHP}	Clock (TCK, PC1) high time (PLD only)	(2)	90		ns
t_{ISCCLP}	Clock (TCK, PC1) low time (PLD only)	(2)	90		ns
t_{ISCPSU}	ISC Port setup time		7		ns
t_{ISCPH}	ISC Port hold up time		5		ns
t_{ISCPCO}	ISC Port clock to output			21	ns
t_{ISCPZV}	ISC Port high-impedance to valid output			21	ns
t_{ISCPVZ}	ISC Port valid output to high-impedance			21	ns

1. For non-PLD Programming, Erase or in ISC By-pass mode.
2. For Program or Erase PLD only.

34 Revision history

Table 194. Document revision history

Date	Target revision	Revision	Changes
July 1, 2003	Version was kept internal	1.0	First Issue
15-Jul-03	Version was kept internal	1.1	Update register information, electrical characteristics (Table 17, 46, 132, 133, 134, 135; Figure 68)
03-Sep-03	1.3	1.2	Update references for Product Catalog
05-Feb-04	Version was kept internal	2.0	Reformatted; corrected mechanical dimensions (Table 158)
07-May-04	3	3.0	Reformatted; update characteristics (*Figure 2*, *Figure 3*, *Figure 50* through to *Figure 83*; *Table 60*, *Table 95*, *Table 112* through to *Table 155*, *Table 158*, *Table 166* to *Table 168*, *Table 173*)
14-Sep-04	4	4.0	Reformatted; updated Feature Summary; added table (*Table 165*); updated graphics, mechanical dimensions (*Figure 2*, *3*, *36*, *39*, *50*, *75*, *79*; *Table 2*, *3*, *6*, *7*, *8*, *9*, *10*, *11*, *50*, *52*, *56*, *73*, *114*, *121*, *156*, *157*, *158*, *166*, *191*, *192*)
29-Oct-04	5	5.0	Corrected LQFP80 mechanical dimensions (*Table 192*)
21-Jan-05	6	6.0	Updated characteristics, SPI section (*Figure 2*, *40*, *41*, *44*; *Table 85*, *87*, *89*, *91*, *165*, *175*, *177*, *179*, *181*, *182*, *188*, *189*)
13-Jun-05	Version was kept internal	7.0	Added TCM5 signal name to P4.6 in *Table 2 on page 22* Changed register function descriptions for CAPCOMxx registers in *Table 98 on page 154* Added 10B_PWM to bit 2 in *Table 103* and *Table 105 on page 160* Changed values in *Table 175 on page 255* (Turbo Off column) Changed values in *Table 184 on page 261* (Turbo Off column) Changed t_{ISCCFP} value in *Table 188 on page 263* and *Table 189 on page 264* Added *Section 33: Important notes on page 270* *Table 166 on page 249*: modified notes, adding text that Port 1 is not 5 V tolerant, changed parameter description for V_{IH} and V_{IL}.
05-May-09	7	-	Document reformatted and disclaimer text updated. Updated datasheet status to “full datasheet”, and changed RPNs from uPSD33xx to UPSD33xx. New Important note added, *Section 33.2 on page 270* Changed V_{REF} to AV_{REF} throughout document. Changed WDTKEY to WDKEY throughout document. Updated *Figure 13 on page 69* with correct labeling of CCON[2:0]. Updated *Figure 36 on page 120*. Updated supply voltage symbols in tables updated, *Table 160*, *Table 161*, *Table 162*, *Table 166*, *Table 173*. Added ECOPACK text in cover page and in section *Section 31: Package mechanical information*. SRAM standby mode removed. Backup battery feature removed. *Section 31: Package mechanical information on page 266* updated.