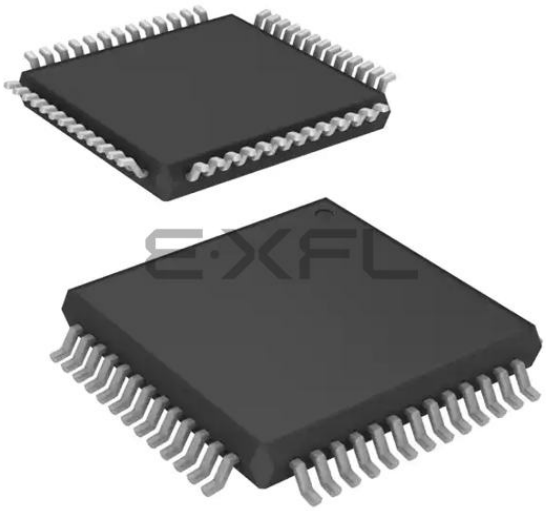


Welcome to E-XFL.COM

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 ~ 3.6V
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/upsd3433evb40t6

List of tables

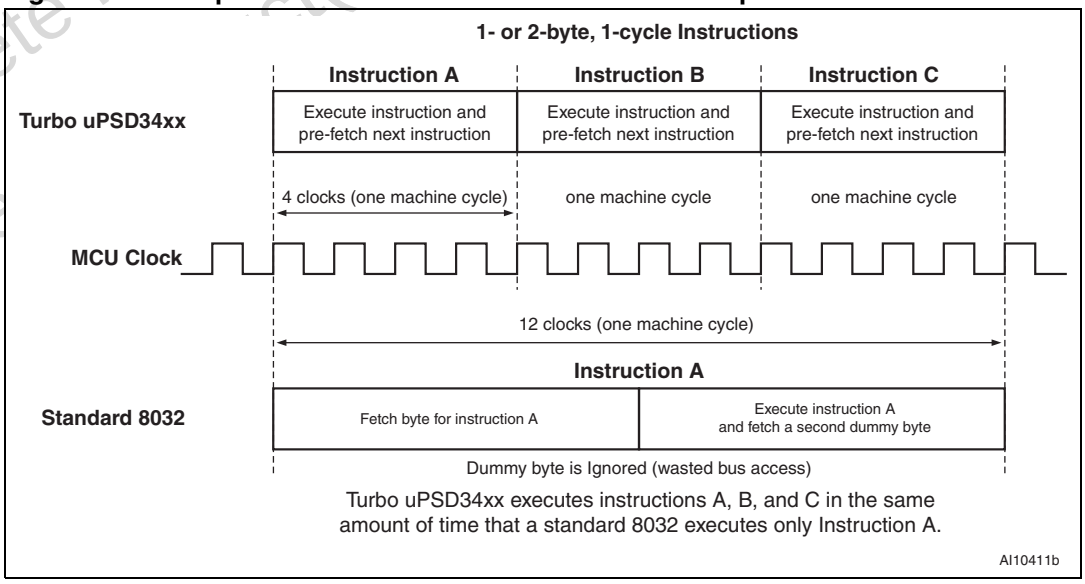
5 8032 MCU core performance enhancements

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



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

- SPI interface registers
  SPICLKD, SPISTAT, SPITDR, SPIRDR, SPICON0, SPICON1
- I^2C interface registers
  S1SETUP, S1CON, S1STA, S1DAT, S1ADR
- Analog to digital converter registers
  ACON, ADCPS, ADAT0, ADAT1
- IrDA interface register
  IRDACON
- USB interface registers
  UADDR, UPAIR, WE0-3, UIF0-3, UCTL, USTA, USEL, UCON, USIZE, UBASEH, UBASEL, USCI, USCV

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

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

Table 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										

Figure 34. UART mode 2, block diagram

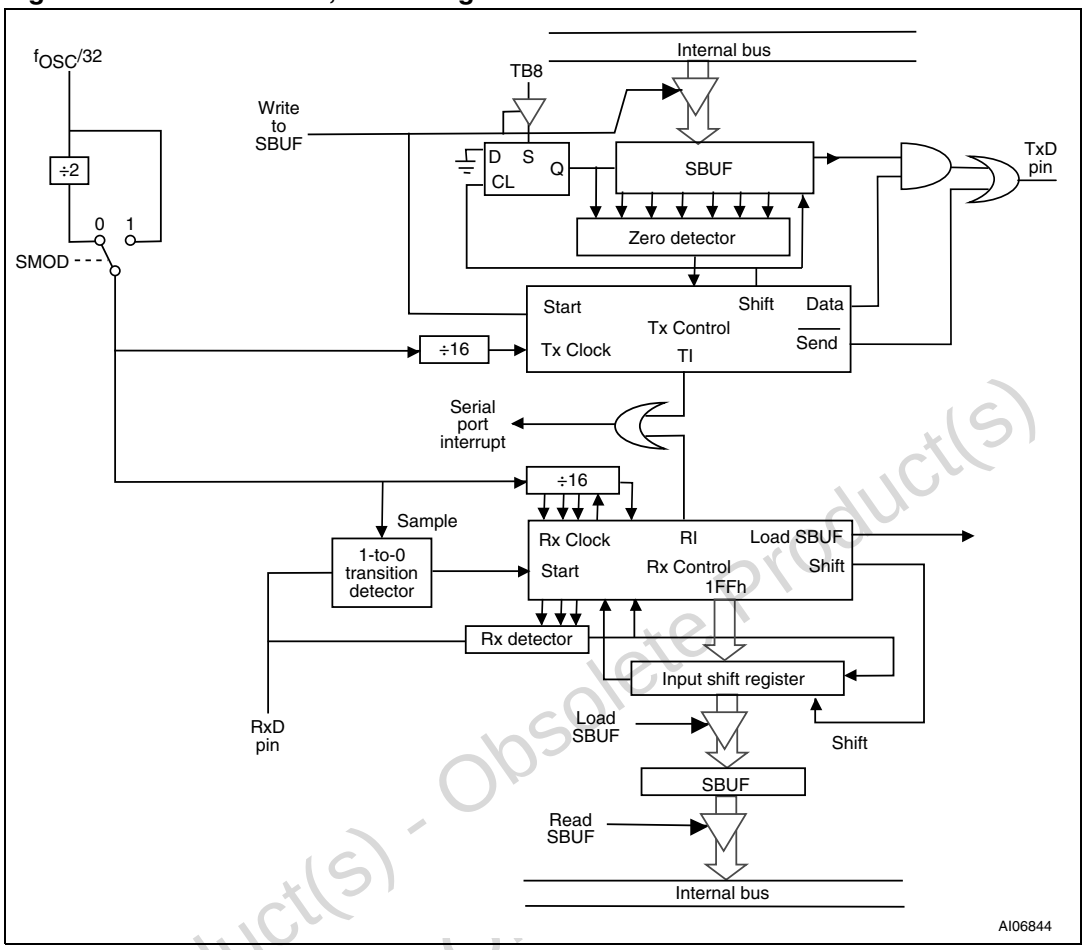
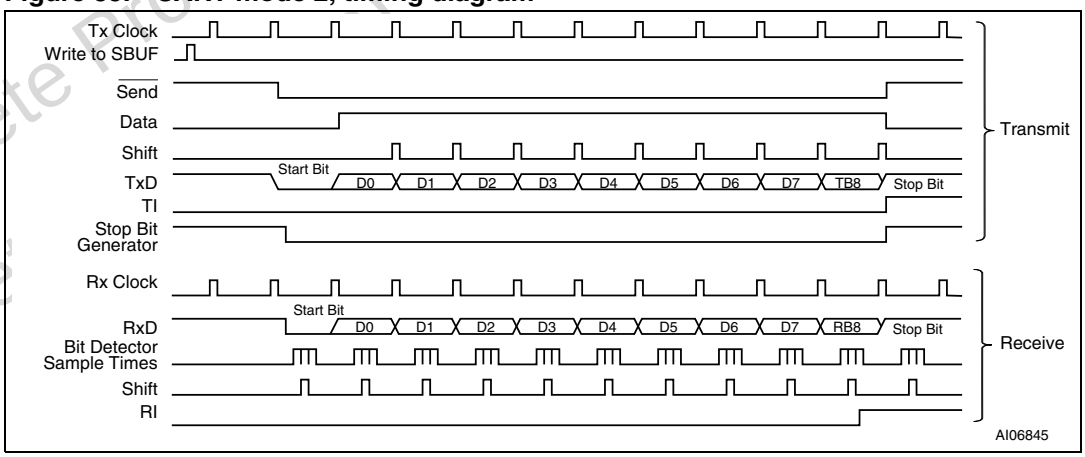


Figure 35. UART mode 2, timing diagram



The interface may operate as either a Master or a Slave within a given application, controlled by firmware writing to SFRs.

By default after a reset, the I²C interface is in Master Receiver mode, and the SDA/P3.6 and SCL/P3.7 pins default to GPIO input mode, high impedance, so there is no I²C bus interference. Before using the I²C interface, it must be initialized by firmware, and the pins must be configured. This is discussed in *Section 23.13: I2C operating sequences on page 135*.

23.4 Bus arbitration

A Master device always samples the I²C bus to ensure a bus line is high whenever that Master is asserting a logic 1. If the line is low at that time, the Master recognizes another device is overriding its own transmission.

A Master may start a transfer only if the I²C bus is not busy. However, it is possible that two or more Masters may generate a Start condition simultaneously. In this case, arbitration takes place on the SDA line each time SCL is high. The Master that first senses that its bus sample does not correspond to what it is driving (SDA line is low while it is asserting a high) will immediately change from Master-Transmitter to Slave-Receiver mode. The arbitration process can carry on for many bit times if both Masters are addressing the same Slave device, and will continue into the data bits if both Masters are trying to be Master-Transmitter. It is also possible for arbitration to carry on into the acknowledge bits if both Masters are trying to be Master-Receiver. Because address and data information on the bus is determined by the winning Master, no information is lost during the arbitration process.

23.5 Clock synchronization

Clock synchronization is used to synchronize arbitrating Masters, or used as a handshake by a devices to slow down the data transfer.

23.5.1 Clock sync during arbitration

During bus arbitration between competing Masters, Master_X, with the longest low period on SCL, will force Master_Y to wait until Master_X finishes its low period before Master_Y proceeds to assert its high period on SCL. At this point, both Masters begin asserting their high period on SCL simultaneously, and the Master with the shortest high period will be the first to drive SCL for the next low period. In this scheme, the Master with the longest low SCL period paces low times, and the Master with the shortest high SCL period paces the high times, making synchronized arbitration possible.

23.5.2 Clock sync during handshaking

This allows receivers in different devices to handle various transfer rates, either at the byte-level, or bit-level.

At the byte-level, a device may pause the transfer between bytes by holding SCL low to have time to store the latest received byte or fetch the next byte to transmit.

At the bit-level, a Slave device may extend the low period of SCL by holding it low. Thus the speed of any Master device will adapt to the internal operation of the Slave.

Table 94. SPICLKD register bit definition

Bit	Symbol	R/W	Definition
7	DIV128	RW	0 = No division 1 = Divide f_{OSC} clock by 128
6	DIV64	RW	0 = No division 1 = Divide f_{OSC} clock by 64
5	DIV32	RW	0 = No division 1 = Divide f_{OSC} clock by 32
4	DIV16	RW	0 = No division 1 = Divide f_{OSC} clock by 16
3	DIV8	RW	0 = No division 1 = Divide f_{OSC} clock by 8
2	DIV4	RW	0 = No division 1 = Divide f_{OSC} clock by 4
1-0	Not Used	–	

Table 95. 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 96. 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

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.

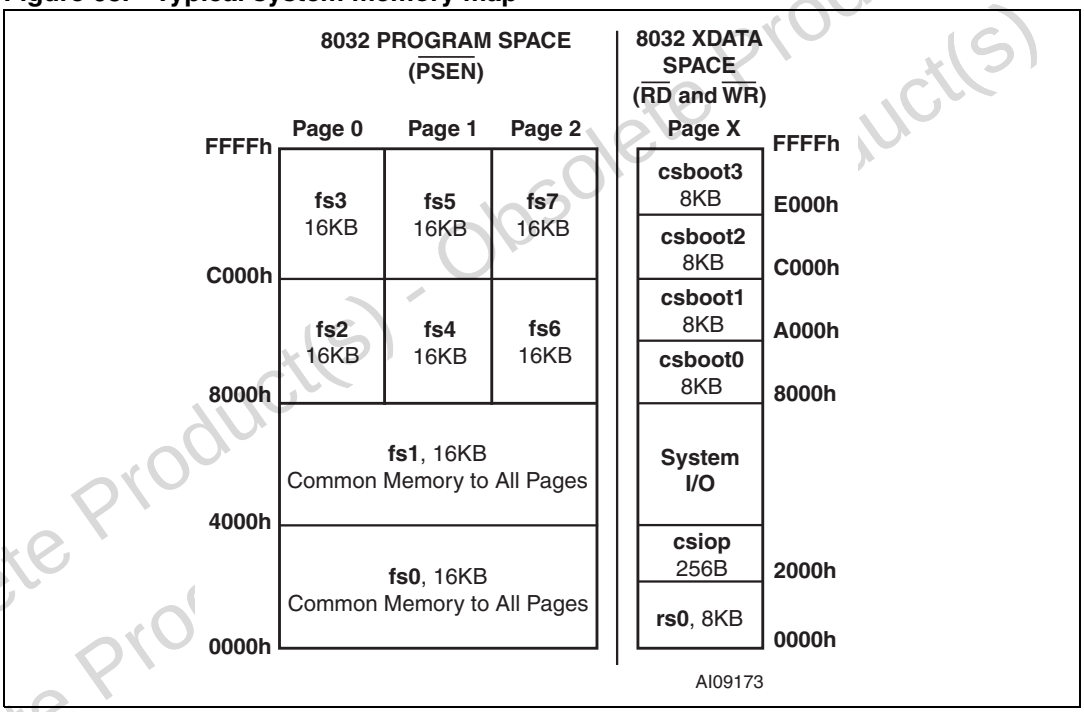
28.2.1 8032 program address space

In the example of *Figure 63*, six sectors of Main Flash memory (fs2.. fs7) are paged across three memory pages in the upper half of program address space, and the remaining two sectors of Main Flash memory (fs0, fs1) reside in the lower half of program address space, and these two sectors are independent of paging (they reside in “common” program address space). This paged memory example is quite common and supported by many 8051 software compilers.

28.2.2 8032 data address space (XDATA)

Four sectors of Secondary Flash memory reside in the upper half of 8032 XDATA space in the example of *Figure 63*. SRAM and csiop registers are in the lower half of XDATA space. The 8032 SFR registers and local SRAM inside the 8032 MCU module do not reside in XDATA space, so it is OK to place PSD module SRAM or csiop registers at an address that overlaps the address of internal 8032 MCU module SRAM and registers.

Figure 63. Typical system memory map



28.2.3 Specifying the memory map with PSDsoft express

The memory map example shown in *Figure 63 on page 198* is implemented using PSDsoft Express in a point-and-click environment. PSDsoft Express will automatically generate Hardware Definition Language (HDL) statements of the ABEL language for the DPLD, such as those shown in *Table 159*.

Specifying these equations using PSDsoft Express is very simple. For example, *Figure 64*, page 84 shows how to specify the chip-select equation for the 16 Kbyte Flash memory segment, fs4. Notice fs4 is on memory page 1. This specification process is repeated for all other Flash memory segments, the SRAM, the csiop register block, and any external chip select signals that may be needed.

signal to latch the incoming signal. Then define an equation for the IMC clock (.ld) or the IMC gate (.le) signal in the “I/O Equations” section.

If the user would like to latch an incoming signal using the gate signal ALE from the 8032, then in PSDsoft Express, for a given input pin on Port A, B, or C, specify “Latched Address” as the pin function.

If it is desired to pass an incoming signal through an IMC directly to the AND-OR array inputs without clocking or gating (this is most common), in PSDsoft Express simply specify “Logic or Address” for the input pin function on Port A, B, or C.

Figure 78. Detail of a single IMC

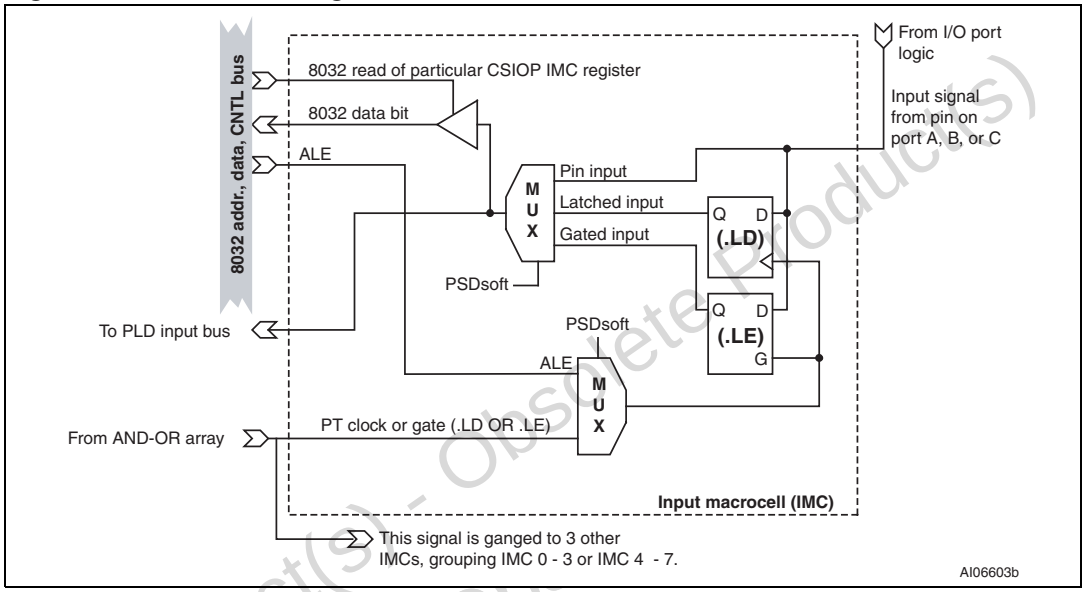


Table 173. Input macrocell port A(1) (address = csiop + offset 0Ah)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IMC PA7	IMC PA6	IMC PA5	IMC PA4	IMC PA3	IMC PA2	IMC PA1	IMC PA0

1. Port A not available on 52-pin UPSD34xx devices.
2. 1 = current state of IMC is logic '1,' 0 = current state is logic ’0’.

Table 174. Input macrocell port B (address = csiop + offset 0Bh)(1)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IMC PB7	IMC PB6	IMC PB5	IMC PB4	IMC PB3	IMC PB2	IMC PB1	IMC PB0

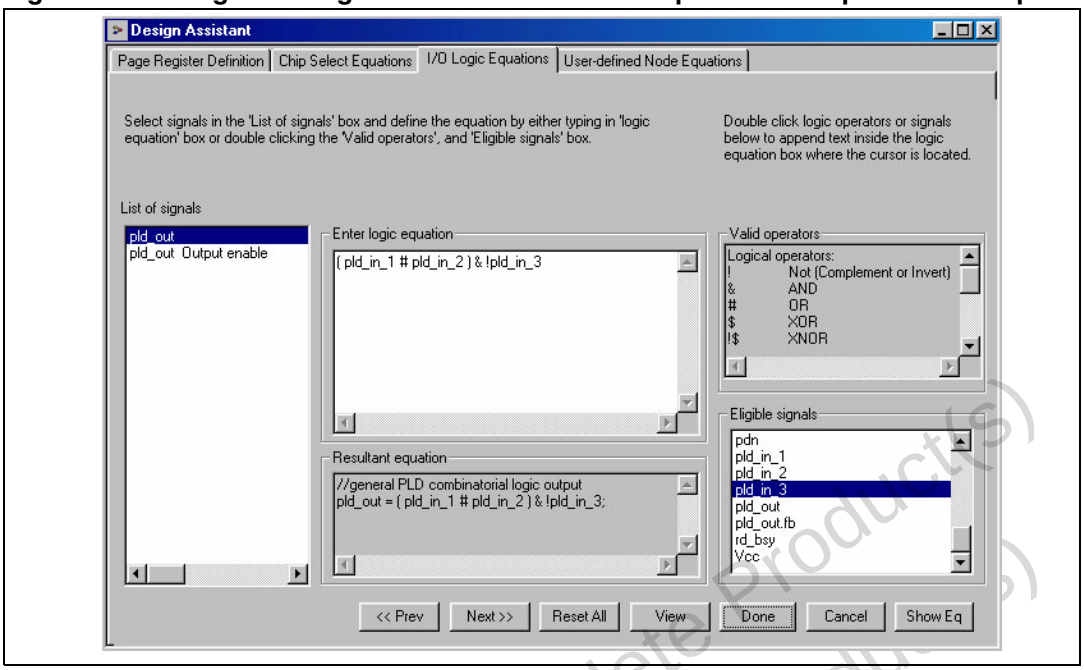
1. 1 = current state of IMC is logic '1,' 0 = current state is logic ’0’.

Table 175. Input macrocell port C (address = csiop + offset 18h)(1) (2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IMC PC7	X	X	IMC PC4	IMC PC3	IMC PC2	X	X

1. X = Not guaranteed value, can be read either '1' or '0.' These are JTAG pins.
2. 1 = current state of IMC is logic '1,' 0 = current state is logic ’0’.

Figure 82. Using the design assistant in PSDsoft Express for simple PLD example



28.5.40 Latched address output mode

In the MCU module, the data bus Bits D0-D15 are multiplexed with the address Bits A0-A15, and the ALE signal is used to separate them with respect to time. Sometimes it is necessary to send de-multiplexed address signals to external peripherals or memory devices. Latched Address Output mode will drive individual demuxed address signals on pins of Ports A or B. Port pins can be designated for this function on a pin-by-pin basis, meaning that an entire port will not be sacrificed if only a few address signals are needed.

To activate this mode, the desired pins on Port A or Port B are designated as “Latched Address Out” in PSDsoft. Then in the 8032 initialization firmware, a logic ’1’ is written to the csiop Control register for Port A or Port B in each bit position that corresponds to the pin of the port driving an address signal. *Table 190* and *Table 191* define the csiop Control register locations and bit assignments.

The latched low address byte A4-A7 is available on both Port A and Port B. The high address byte A8-A15 is available on Port B only. Selection of high or low address byte is specified in PSDsoft Express.

Table 190. Latched address output, port A contro register(1)(2)(3)(address = csiop + offset 02h)l

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7 (addr A7)	PA6 (addr A6)	PA5 (addr A5)	PA4 (addr A4)	PA3 (addr A3)	PA2 (Addr A2)	PA1 (addr A1)	PA0 (addr A0)

1. Port A not available on 52-pin UPSD34xx devices.
2. For each bit, 1 = drive demuxed 8032 address signal on pin, 0 = pin is default mode, MCU I/O.
3. Default state for register is 00h after reset or power-up.

mode, making the pin suitable for input mode (read by the input buffer shown in *Figure 79 on page 232*). *Figure 79* shows the three sources that can control the pin output enable signal: a product term from AND-OR array; the csiop Direction register; or the Peripheral I/O Mode logic (Port A only). The csiop Enable Out registers represent the state of the final output enable signal for each port pin driver, and are defined in *Table 196 on page 242* through *Table 199 on page 242*.

Table 192. Port A pin drive select register(1) (2) (3)(address = csiop + offset 08h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7 open drain	PA6 open drain	PA5 open drain	PA4 open drain	PA3 slew rate	PA2 slew rate	PA1 slew rate	PA0 slew rate

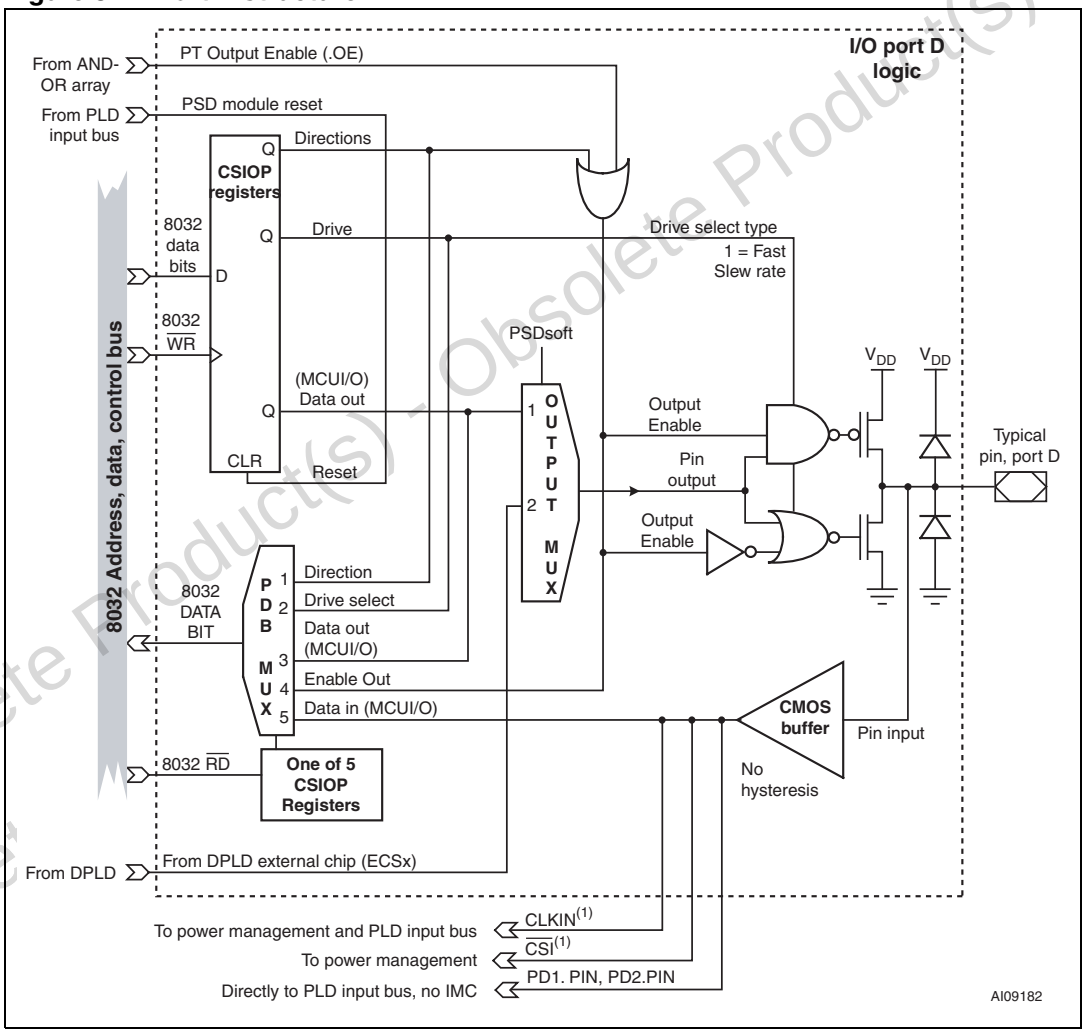
1. Port A not available on 52-pin UPSD34xx devices.
2. For each bit, 1 = pin drive type is selected, 0 = pin drive type is default mode, CMOS push/pull.
3. Default state for register is 00h after reset or power-up.

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 28.1.11: OMCs on page 195*) and also the *Section 28.5.53: Automatic power-down (APD) on page 250*.
- PD2 can be used as a common chip select signal ($\overline{CSI}$) for the Flash and SRAM memories on the PSD module (see *Section 28.5.55: Chip select input (CSI) on page 253*). If driven to logic '1' by an external source, $\overline{CSI}$ will force the Flash memory 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 87. Port D structure



Note: 1 Optional function on a specific Port D pin.

Table 219. USB transceiver specification

Symbol	Parameter	Test conditions(1)	Min.	Typ.	Max.	Unit
UV_{IL}	Low input voltage	V_{DD} = 3.6V			0.8	V
R_{DH}	Output impedance (high state)	(2)	28		43	Ω
R_{DL}	Output impedance (low state)	(2)	28		43	Ω
I_L	Input leakage current	V_{DD} = 3.6 V		±0.1	±5	µA
I_{OZ}	3-state output OFF state current	$V_I = V_{IH}$ or V_{IL}			±10	µA
V_{CR}	Crossover point		1.3		2	V
t_{RISE}	Rise time		4		20	ns
t_{FALL}	Fall time		4		20	ns

1. Temperature range = –45°C to 85°C.
2. *This value includes an external resistor of 24Ω±1%.*

Figure 100. Input to output disable / enable

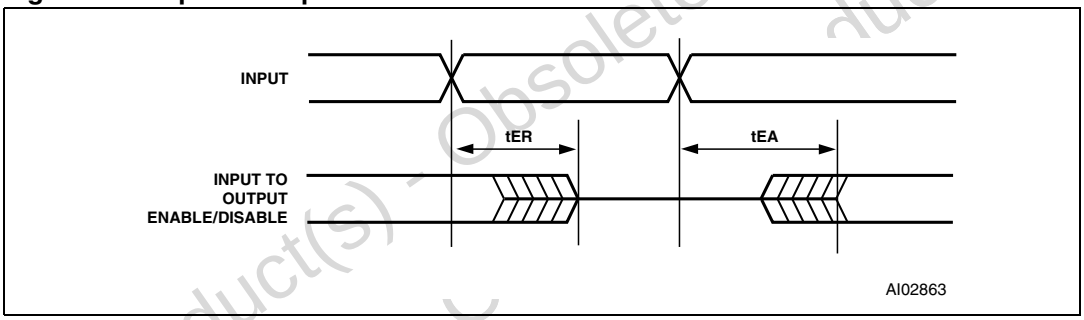


Table 233. Supervisor reset and LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RST\_LO\_IN}$	Reset input duration		1(1)			µs
t_{RST_ACTV}	Generated reset duration	f_{OSC} = 40 MHz	10(2)			ms
t_{RST_FIL}	Reset input spike filter			1		µs
V_{RST_HYS}	Reset input hysteresis	V_{CC} = 3.3 V		0.1		V
V_{RST_THRESH}	LVD trip threshold	V_{CC} = 3.3 V	2.4	2.6	2.8	V

1. 25 µs minimum to abort a Flash memory program or erase cycle in progress.
2. As f_{OSC} decreases, t_{RST_ACTV} increases. Example: t_{RST_ACTV} = 50 ms when f_{OSC} = 8 MHz.

Figure 107. ISC timing

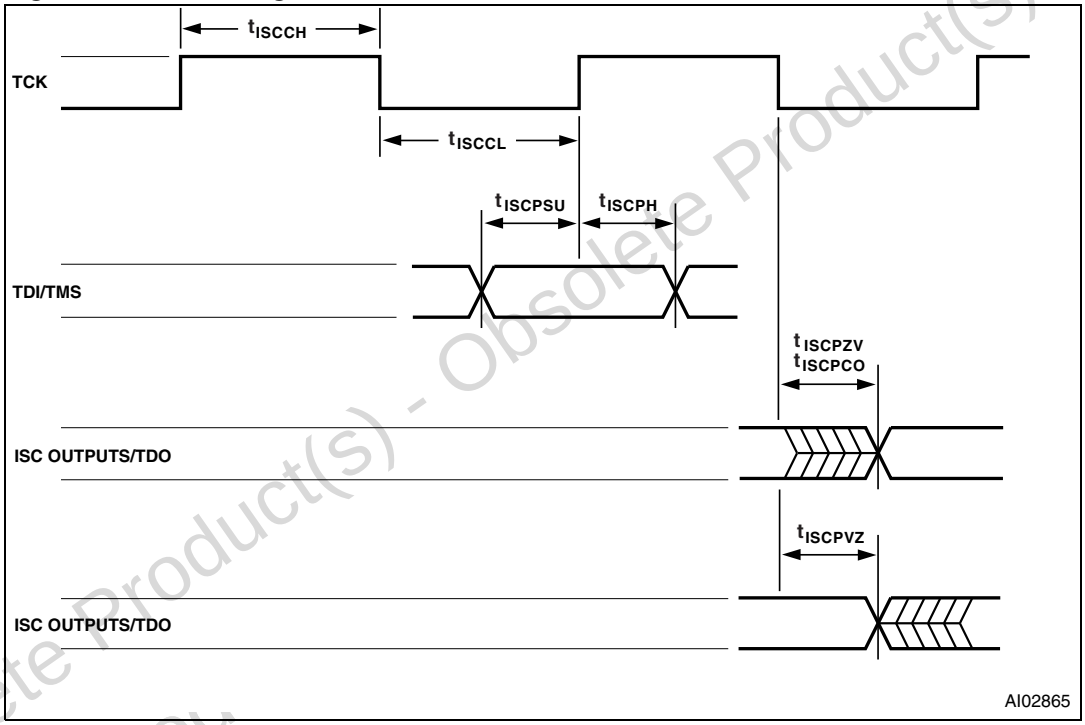


Table 234. ISC timing (5 V PSD module)

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)		23		ns
t_{ISCCL}	Clock (TCK, PC1) low time (except for PLD)		23		ns
t_{ISCCFP}	Clock (TCK, PC1) frequency (PLD only)	(2)		4	MHz
t_{ISCCHP}	Clock (TCK, PC1) high time (PLD only)		90		ns
t_{ISCCLP}	Clock (TCK, PC1) low time (PLD only)		90		ns
t_{ISCPSU}	ISC port setup time		7		ns
t_{ISCPH}	ISC port hold up time		5		ns

32 Package mechanical information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

Table 240. Order codes

Part number	Max MHz	1st Flash	2nd Flash	SRAM	GPIO	8032 bus	V_{CC}	V_{DD}	Package
		(bytes)							
UPSD3422E-40T6	40	64K	32K	4K	35	No	3.3 V	5.0 V	LQFP52
UPSD3422EV-40T6	40	64K	32K	4K	35	No	3.3 V	3.3 V	LQFP52
UPSD3422E-40U6	40	64K	32K	4K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3422EV-40U6	40	64K	32K	4K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3433E-40T6	40	128K	32K	8K	35	No	3.3 V	5.0 V	LQFP52
UPSD3433EV-40T6	40	128K	32K	8K	35	No	3.3 V	3.3 V	LQFP52
UPSD3433E-40U6	40	128K	32K	8K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3433EV-40U6	40	128K	32K	8K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3434E-40T6	40	256K	32K	8K	35	No	3.3 V	5.0 V	LQFP52
UPSD3434EV-40T6	40	256K	32K	8K	35	No	3.3 V	3.3 V	LQFP52
UPSD3434E-40U6	40	256K	32K	8K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3434EV-40U6	40	256K	32K	8K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3454E-40T6	40	256K	32K	32K	35	No	3.3 V	5.0 V	LQFP52
UPSD3454EV-40T6	40	256K	32K	32K	35	No	3.3 V	3.3 V	LQFP52
UPSD3454E-40U6	40	256K	32K	32K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3454EV-40U6	40	256K	32K	32K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3422EB40T6	40	64K	32K	4K	35	No	3.3 V	5.0 V	LQFP52
UPSD3422EVB40T6	40	64K	32K	4K	35	No	3.3 V	3.3 V	LQFP52
UPSD3422EB40U6	40	64K	32K	4K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3422EVB40U6	40	64K	32K	4K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3433EB40T6	40	128K	32K	8K	35	No	3.3 V	5.0 V	LQFP52
UPSD3433EVB40T6	40	128K	32K	8K	35	No	3.3 V	3.3 V	LQFP52
UPSD3433EB40U6	40	128K	32K	8K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3433EVB40U6	40	128K	32K	8K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3434EB40T6	40	256K	32K	8K	35	No	3.3 V	5.0 V	LQFP52
UPSD3434EVB40T6	40	256K	32K	8K	35	No	3.3 V	3.3 V	LQFP52
UPSD3434EB40U6	40	256K	32K	8K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3434EVB40U6	40	256K	32K	8K	46	Yes	3.3 V	3.3 V	LQFP80
UPSD3454EB40T6	40	256K	32K	32K	35	No	3.3 V	5.0 V	LQFP52
UPSD3454EVB40T6	40	256K	32K	32K	35	No	3.3 V	3.3 V	LQFP52
UPSD3454EB40U6	40	256K	32K	32K	46	Yes	3.3 V	5.0 V	LQFP80
UPSD3454EVB40U6	40	256K	32K	32K	46	Yes	3.3 V	3.3 V	LQFP80

Note: Operating temperature is in the Industrial range (–40 °C to 85 °C).

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

Impact on application

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

Workaround

Revision A and B - None identified at this time.

34.10 MCU JTAG ID

Description

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

Revision A MCU JTAG ID - 0451F041f

Revision B MCU JTAG ID - 1451F041h

Impact on application

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

34.11 Port 1 not 5-volt IO tolerant

Description

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

Impact on application

5 V devices should not be connected to port P1.

Workaround

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