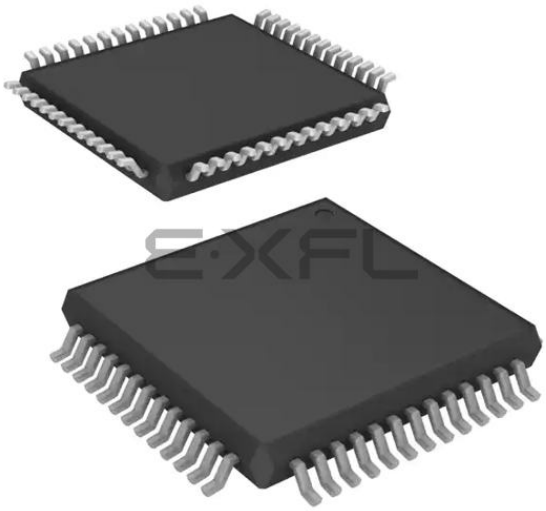


Welcome to E-XFL.COM

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-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3433eb40t6

15 Power saving modes

The UPSD34xx is a combination of two die, or modules, each module having its own current consumption characteristics. This section describes reduced power modes for the MCU module. See *Section 28.1.16: Power management on page 197* for reduced power modes of the PSD module. Total current consumption for the combined modules is determined in the DC specifications at the end of this document.

The MCU module has three software-selectable modes of reduced power operation.

- Idle mode
- Power-down mode
- Reduced frequency mode

15.1 Idle mode

Idle mode will halt the 8032 MCU core while leaving the MCU peripherals active (Idle mode blocks MCU_CLK only). For lowest current consumption in this mode, it is recommended to disable all unused peripherals, before entering Idle mode (such as the ADC and the Debug Unit breakpoint comparators). The following functions remain fully active during Idle mode (except if disabled by SFR settings).

- External Interrupts INT0 and INT1
- Timer 0, Timer 1 and Timer 2
- Supervisor reset from: LVD, JTAG Debug, External RESET_IN_, but **not** the WTD
- ADC
- I^2C Interface
- UART0 and UART1 Interfaces
- SPI Interface
- Programmable Counter Array
- USB Interface

An interrupt generated by any of these peripherals, or a reset generated from the supervisor, will cause Idle mode to exit and the 8032 MCU will resume normal operation.

The output state on I/O pins of MCU ports 1, 3, and 4 remain unchanged during Idle mode.

To enter Idle mode, the 8032 MCU executes an instruction to set the IDL bit in the SFR named PCON, shown in *Table 33 on page 74*. This is the last instruction executed in normal operating mode before Idle mode is activated. Once in Idle mode, the MCU status is entirely preserved, and there are no changes to: SP, PSW, PC, ACC, SFRs, DATA, IDATA, or XDATA.

The following are factors related to Idle mode exit:

- Activation of any enabled interrupt will cause the IDL bit to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following the Return from Interrupt instruction (RETI), the next instruction to be executed will be the one which follows the instruction that set the IDL bit in the PCON SFR.
- After a reset from the supervisor, the IDL bit is cleared, Idle mode is terminated, and the MCU restarts after three MCU machine cycles.

Table 31. MCU module port and peripheral status during reduced power modes

Mode	Ports 1, 3, 4	SPI, I^2C, UART0,1	PCA, Timer 0,1,2	USB	ADC	EXT INT0,1	Supervisory
Idle	Maintain data	Active	Active	Active	Active	Active	Active(1)
Power-down	Maintain data	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled

1. The Watchdog Timer is not active during Idle mode. Other supervisor functions are active: LVD, external reset, JTAG Debug reset.

Table 32. State of 8032 MCU bus signals during power-down and idle modes

Mode	ALE	PSEN_	RD_	WR_	AD0-7	A8-15
Idle	0	1	1	1	FFh	FFh
Power-down	0	1	1	1	FFh	FFh

Table 33. PCON: power control register (SFR 87h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD0	SMOD1	–	POR	RCLK1	TCLK1	PD	IDL

Table 34. PCON register bit definition

Bit	Symbol	R/W	Function
7	SMOD0	R,W	Baud Rate Double Bit (UART0) 0 = No doubling 1 = Doubling (See *Section 21.3: UART baud rates on page 110* for details.)
6	SMOD1	R,W	Baud Rate Double Bit for 2nd UART (UART1) 0 = No doubling 1 = Doubling (See *Section 21.3: UART baud rates on page 110* for details.)
5	–	–	Reserved
4	POR	R,W	Only a power-on reset sets this bit (cold reset). Warm reset will not set this bit. 0 = Cleared to zero with firmware 1 = Is set only by a power-on reset generated by Supervisory circuit (see *Section 19.3: Power-up reset on page 92* for details).
3	RCLK1	R,W	Received Clock Flag (UART1) (See *Table 60 on page 101* for flag description.)
2	TCLK1	R,W	Transmit Clock Flag (UART1) (See *Table 60 on page 101* for flag description)

17 I/O ports of mcu module

The MCU module has three 8-bit I/O ports: Port 1, Port 3, and Port 4. The PSD module has four other I/O ports: Port A, B, C, and D. This section describes only the I/O ports on the MCU module.

I/O ports will function as bidirectional general-purpose I/O (GPIO), but the port pins can have alternate functions assigned at run-time by writing to specific SFRs. The default operating mode (during and after reset) for all three ports is GPIO input mode. Port pins that have no external connection will not float because each pin has an internal weak pull-up (~150 kΩ) to V_{CC}.

I/O ports 3 and 4 are 5 V tolerant, meaning they can be driven/pulled externally up to 5.5 V without damage. The pins on Port 4 have a higher current capability than the pins on Ports 1 and 3.

Three additional MCU ports (only on 80-pin UPSD34xx devices) are dedicated to bring out the 8032 MCU address, data, and control signals to external pins. One port, named MCUAD[7:0], has eight multiplexed address/data bidirectional signals. The third port has MCU bus control outputs: read, write, program fetch, and address latch. These ports are typically used to connect external parallel peripherals and memory devices, but they may NOT be used as GPIO. Notice that the eight upper address signals do not come out to pins on the port. If high-order address signals are required on external pins (MCU addresses A[15:8]), then these address signals can be brought out as needed to PLD output pins or to the Address Out mode pins on PSD module ports. See PSD module section, "*Section 28.5.40: Latched address output mode on page 238* for details.

Figure 15 on page 80 represents the flexibility of pin function routing controlled by the SFRs. Each of the 24 pins on three ports, P1, P3, and P4, may be individually routed on a pin-by-pin basis to a desired function.

17.1 MCU port operating modes

MCU port pins can operate as GPIO or as alternate functions (see *Figure 16 on page 80* through *Figure 18 on page 81*).

Depending on the selected pin function, a particular pin operating mode will automatically be used:

- GPIO - Quasi-bidirectional mode
- UART0, UART1 - Quasi-bidirectional mode
- SPI - Quasi-bidirectional mode
- I2C - Open drain mode
- ADC - Analog input mode
- PCA output - Push-Pull mode
- PCA input - Input only (Quasi-bidirectional)
- Timer 0,1,2 - Input only (Quasi-bidirectional)

21.3.1 Using timer 1 to generate baud rates

When Timer 1 is used as the baud rate generator (bits RCLK = 0, TCLK = 0), the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Mode 1,3 Baud Rate =
$(2^{SMOD} / 32)$ x (Timer 1 overflow rate)

The Timer 1 Interrupt should be disabled in this application. The Timer itself can be configured for either “timer” or “counter” operation, and in any of its 3 running modes. In the most typical applications, it is configured for “timer” operation, in the Auto-reload Mode (high nibble of the SFR TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1,3 Baud Rate =
$(2^{SMOD} / 32) \times (f_{OSC} / (12 \times [256 - (TH1)]))$

Table 69 lists various commonly used baud rates and how they can be obtained from Timer 1.

21.3.2 Using timer/counter 2 to generate baud rates

See *Section 20.6.3: Baud rate generator mode on page 103*.

Table 69. Commonly used baud rates generated from timer 1

UART mode	f_{OSC} MHz	Desired baud rate	Resultant baud rate	Baud rate deviation	SMOD bit in PCON	Timer 1		
						C/$\overline{T}$ Bit in TMOD	Timer mode in TMOD	TH1 reload value (hex)
Mode 0 Max	40.0	3.33MHz	3.33MHz	0	X	X	X	X
Mode 2 Max	40.0	1250 000	1250 000	0	1	X	X	X
Mode 2 Max	40.0	625 000	625 000	0	0	X	X	X
Modes 1 or 3	40.0	19200	18939	-1.36%	1	0	2	F5
Modes 1 or 3	40.0	9600	9470	-1.36%	1	0	2	EA
Modes 1 or 3	36.0	19200	18570	-2.34%	1	0	2	F6
Modes 1 or 3	33.333	57600	57870	0.47%	1	0	2	FD
Modes 1 or 3	33.333	28800	28934	0.47%	1	0	2	FA
Modes 1 or 3	33.333	19200	19290	0.47%	1	0	2	F7
Modes 1 or 3	33.333	9600	9645	0.47%	1	0	2	EE
Modes 1 or 3	24.0	9600	9615	0.16%	1	0	2	F3
Modes 1 or 3	12.0	4800	4808	0.16%	1	0	2	F3
Modes 1 or 3	11.0592	57600	57600	0	1	0	2	FF
Modes 1 or 3	11.0592	28800	28800	0	1	0	2	FE
Modes 1 or 3	11.0592	19200	19200	0	1	0	2	FD
Modes 1 or 3	11.0592	9600	9600	0	1	0	2	FA

Table 86. Number of I²C bus samples taken after 1-to-0 transition on SDA (Start condition)

Contents of S1SETUP		Resulting value for S1SETUP	Resulting number of samples taken after 1-to-0 on SDA line
SS_EN bit	SMPL_SET[6:0]		
0	XXXXXXXb	00h (default)	1
1	0000000b	80h	1
1	0000001b	81h	2
1	0000010b	82h	3
...	...	...	...
1	0001011b	8Bh	12
1	0010111b	97h	24
...	...	...	...
1	1111111b	FFh	128

Table 87. Start condition hold time

I²C bus speed	Range of I²C clock speed (f_{SCL})	Minimum Start condition hold time (t_{HLDSTA})
Standard	Up to 100 kHz	4000 ns
Fast	101 kHz to 400 kHz	600 ns
High	401 kHz to 833 kHz(1)	160 ns

1. 833KHz is maximum for UPSD34xx devices.

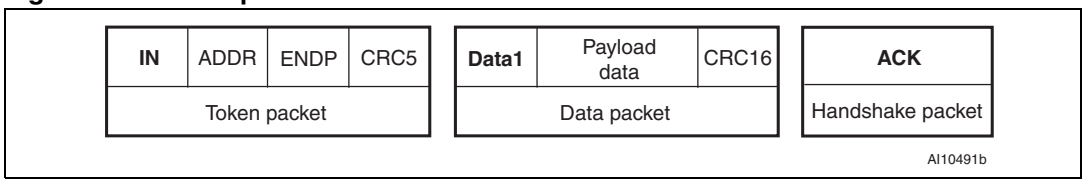
Table 88 provides recommended settings for S1SETUP based on various combinations of f_{OSC} and f_{SCL}. Note that the "Total Sample Period" times in *Table 87 on page 134* are typically slightly less than the minimum Start condition hold time, t_{HLDSTA} for a given I²C bus speed.

Important note: The SCL bit rate f_{SCL} must first be determined by bits CR[2:0] in the SFR S1CON before a value is chosen for SMPL_SET[6:0] in the SFR S1SETUP.

Table 88. S1SETUP examples for various I²C bus speeds and oscillator frequencies

I²C bus speed, f_{SCL}	Parameter	Oscillator frequency, f_{OSC}				
		6 MHz	12 MHz	24 MHz	33 MHz	40 MHz
Standard	Recommended S1SETUP value	93h	A7h	CFh	EEh	FFh
	Number of samples	20	40	80	111	128
	Time between samples	166.6 ns	83.3 ns	41.6 ns	30 ns	25 ns
	Total sampled period	3332 ns	3332 ns	3332 ns	3333 ns	3200 ns

Figure 51. Interrupt transfer



- Control Transfers (see *Figure 52*)

  Control transfers are used to configure and send commands to a device. Control transfers consist of two or three stages:

  – SETUP

  This stage always consists of a data packet with eight bytes of USB CONTROL data.

  – DATA stage (optional)

  If the CONTROL data is such that the host is requesting information from the device, the SETUP stage is followed by a DATA stage. In this case, the host sends an IN token and the device responds with the requested data in the data packet.

  – STATUS stage

  This stage is essentially a handshake informing the device of a successfully completed control operation.

25.2.1 Enumeration

Enumeration is the process that takes place when a device is first connected to the USB. During enumeration, the host requests information from the device about what it is, how many endpoints it has, the power requirements, bus bandwidth requirements, and what driver to load. Once the enumeration process is complete, the device is available for use.

The enumeration process consists of a series of six steps as follows:

1. When a device is first connected to the USB, its address is zero. Upon detecting a new device connected to the USB, the host sends a Get_Descriptor request to address zero, endpoint0.
2. The device, upon receiving a Get_Descriptor request, sends data back to the host identifying what it is.
3. The host resets the device and then sends a Set_Address request. This is a unique address that identifies it from all other devices connected to the USB. This address remains in effect until the device is disconnected from the USB.
4. The host sends more Get_Descriptor requests to the device to gather more detailed information about it and then loads the specified driver.
5. The host will setup and enable the endpoints defined by the device.
6. The device is now configured and ready for use with the host communicating to the device using the assigned address and endpoints.

Table 99. UPSD34xx USB SFR register map[1] (continued)

SFR addr (hex)	SFR name	Bit name and <bit address>								Reset value (hex)	Comment
		7	6	5	4	3	2	1	0		
E8	UIF0	GLF	INF	OUTF	NAKF	RSTF	SUSPNDF	EOPF	RESUMF	00	USB Global Interrupt Flag
E9	UIF1	–	–	–	IN4F	IN3F	IN2F	IN1F	IN0F	00	USB IN FIFO Interrupt Flag
EA	UIF2	–	–	–	OUT4F	OUT3F	OUT2F	OUT1F	OUT0F	00	USB OUT FIFO Interrupt Flag
EB	UIF3	–	–	–	NAK4F	NAK3F	NAK2F	NAK1F	NAK0F	00	USB IN FIFO NAK Int. Flag
EC	UCTL	–	–	–	–	–	USBEN	VISIBLE	WAKEUP	00	USB Control
ED	USTA	–	–	–	–	RCVT	SETUP	IN	OUT	00	USB Status
EE	RESERVED										
EF	USEL	DIR	–	–	–	–	EP[2:0]			00	USB Endpoint Select
F1	UCON	–	–	–	–	ENABLE	STALL	TOGGLE	BSY	08	USB Endpoint Control
F2	USIZE	–	SIZE[6:0]							00	USB FIFO Valid Size
F3	UBASEH	BASEADDR[15:8]								00	USB Base Address High
F4	UBASEL	BASEADDR [7:6]		0	0	0	0	0	0	00	USB Base Address Low
F5	USCI	–	–	–	–	–	USCI[2:0]			00	USB Setup Command Index
F6	USCV	USCV[7:0]								00	USB Setup Command Value

1. Bits marked with a "–" are Reserved.

25.4.1 USB device address register

Initially when a device is connected to the USB, it responds to the host on address 0. Using the Set_Address request, the host assigns a unique address to the device. The firmware writes this address to the USB Device Address register (see *Table 100*), and subsequently the SIE only responds to transactions on that assigned address. This assigned address is in effect until the device or an upstream hub is disconnected from the USB, the host issues a USB Reset, or the host shuts down. The address register is cleared with a Hardware RESET. When a USB RESET is detected, the address register should be cleared.

Table 105. UIE0 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	–	–	Reserved
3	RSTIE	R/W	Enable the USB Reset interrupt
2	SUSPEND IE	R/W	Enable the USB Suspend interrupt
1	EOPIE	R/W	Enable the USB EOP interrupt
0	RESUMIE	R/W	Enable the USB Resume interrupt

- USB IN FIFO interrupt enable register (UIE1)

  When an endpoint's IN FIFO has been successfully sent to the host with an IN transaction, the FIFO becomes empty. The UIE1 register is used to enable each endpoint's IN FIFO interrupt (*Table 106*).

- USB OUT FIFO interrupt enable register (UIE1)

  When an endpoint's OUT FIFO has been filled by an OUT transaction from the host, the FIFO becomes full. The UIE2 register is used to enable each endpoint's OUT FIFO interrupt (*Table 108*).

Table 106. USB IN FIFO interrupt enable register (UIE1 0E5h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	IN4IE	IN3IE	IN2IE	IN1IE	IN0IE

Table 107. UIE1 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	IN4IE	R/W	Enable Endpoint 4 IN FIFO interrupt
3	IN3IE	R/W	Enable Endpoint 3 IN FIFO interrupt
2	IN2IE	R/W	Enable Endpoint 2 IN FIFO interrupt
1	IN1IE	R/W	Enable Endpoint 1 IN FIFO interrupt
0	IN0IE	R/W	Enable Endpoint 0 IN FIFO interrupt

Table 108. USB OUT FIFO interrupt enable register (UIE2 0E6h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	OUT4IE	OUT3IE	OUT2IE	OUT1IE	OUT0IE

27.10 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.

27.11 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 = 0000h), and the smallest duty cycle is 0.0015% (CAPCOMHn CAPCOMLn = FFFFh). A 0% duty cycle may be generated by clearing the E_COMP Bit to ‘0’.

27.12 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 148 on page 187* through *Table 150 on page 188*). There is one PCA_STATUS register that covers both PCA0 and PCA1 (see *Table 152 on page 188*).

Table 148. 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	–	–	CLK_SEL[1:0]	

Table 149. PCA0 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.
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

Table 162. CSIOP registers and their offsets (in hexadecimal) (continued)

Register name	Port A (80-pin)	Port B	Port C	Port D	Other	Description	Link
PMMR0					B0h	Power management register 0. WRITE and READ.	Table 200 on page 249
PMMR2					B4h	Power management register 2. WRITE and READ.	Table 201 on page 249
PMMR3					C7h	Power management register 3. WRITE and READ. However, Bit 1 can be cleared only by a reset condition.	Table 202 on page 250
Page					E0h	Memory page register. WRITE and READ.	Table 62 on page 194
VM (Virtual Memory)					E2h	Places PSD module memories into 8032 Program Address Space and/or 8032 XDATA Address Space. (VM overrides initial non-volatile setting that was specified in PSDsoft Express. Reset restores initial setting)	Table 160 on page 203

28.5 PSD module detailed operation

Specific details are given here for the following key functional areas on the PSD module:

- Flash Memories
- PLDs (DPLD and GPLD)
- I/O Ports
- Power Management
- JTAG ISP and Debug Interface

28.5.1 Flash memory operation

The Flash memories are accessed through the 8032 Address, Data, and Control Bus interfaces. Flash memories (and SRAM) cannot be accessed by any other bus master other than the 8032 MCU (these are not dual-port memories).

The 8032 cannot write to Flash memory as it would an SRAM (supply address, supply data, supply $\overline{WR}$ strobe, assume the data was correctly written to memory). Flash memory must first be “unlocked” with a special instruction sequence of byte WRITE operations to invoke an internal algorithm inside either Flash memory array, then a single data byte is written (programmed) to the Flash memory array, then programming status is checked by a byte READ operation or by checking the Ready/$\overline{Busy}$ pin (PC3). *Table 163 on page 209* lists all of the special instruction sequences to program a byte to either of the Flash memory arrays, erase the arrays, and check for different types of status from the arrays.

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

Instr. Seq.	Bus Cycle 1	Bus Cycle 2	Bus Cycle 3	Bus Cycle 4	Bus Cycle 5	Bus Cycle 6	Bus Cycle 7	Link
Read Memory Contents (Read Array mode)	Read byte from any valid Flash memory addr							*Read memory contents on page 210*
Program (write) a byte to Flash Memory	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write A0h to X555h (*command*)	Write data byte to address				*Programming Flash memory on page 212*
Bypass Unlock	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 20h to X555h (*command*)					*Bypassed unlock sequence on page 215*
Program a byte to Flash Memory with Bypassed Unlock	Write A0h to XXXXh (*command*)	Write data byte to address						*Bypassed unlock sequence on page 215*
Reset Bypass Unlock	Write 90h to XXXXh (*command*)	Write 00h to XXXXh (*command*)						*Bypassed unlock sequence on page 215*
Flash Bulk Erase(3)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 80h to X555h (*command*)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 10h to X555h (*command*)		*Flash bulk erase on page 216*
Flash Sector Erase	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 80h to X555h (*command*)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 30h to desired Sector (*command*)	Write 30h to another Sector (*command*)	*Flash sector erase on page 216*
Suspend Sector Erase	Write B0h to address that activates FSx or CSBOOTx where erase is in progress (*command*)							*Suspend sector erase on page 217*

28.5.11 Data polling

Polling on the Data Polling Flag Bit (DQ7) is a method of checking whether a program or erase operation is in progress or has completed. *Figure 71* shows the Data Polling algorithm.

When the 8032 issues a program instruction sequence, the embedded algorithm within the Flash memory array begins. The 8032 then reads the location of the byte to be programmed in Flash memory to check status. The Data Polling Flag Bit (DQ7) of this location becomes the compliment of Bit D7 of the original data byte to be programmed. The 8032 continues to poll this location, comparing the Data Polling Flag Bit (DQ7) and monitoring the Error Flag Bit (DQ5). When the Data Polling Flag Bit (DQ7) matches Bit D7 of the original data, then the embedded algorithm is complete. If the Error Flag Bit (DQ5) is '1,' the 8032 should test the Data Polling Flag Bit (DQ7) again since the Data Polling Flag Bit (DQ7) may have changed simultaneously with the Error Flag Bit (DQ5) (see *Figure 71*).

The Error Flag Bit (DQ5) is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte (indicating a bad Flash cell) or if the 8032 attempted to program bit to logic ’1’ when that bit was already programmed to logic ’0’ (must erase to achieve logic ’1’).

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte that was written to the Flash memory with the byte that was intended to be written.

When using the Data Polling method during an erase operation, *Figure 71* still applies. However, the Data Polling Flag Bit (DQ7) is '0' until the erase operation is complete. A ’1’ on the Error Flag Bit (DQ5) indicates a time-out condition on the Erase cycle, a ’0’ indicates no error. The 8032 can read any location within the sector being erased to get the Data Polling Flag Bit (DQ7) and the Error Flag Bit (DQ5).

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

During a Sector Erase operation, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7), as detailed in *Section 28.5.5: Reading the erase/program status bits on page 210*.

During a Sector Erase operation, a Flash memory accepts only Reset Flash and Suspend Sector Erase instruction sequences. Erasure of one Flash memory sector may be suspended, in order to read data from another Flash memory sector, and then resumed.

The address provided with the initial Flash Sector Erase command sequence (*Table 163 on page 209*) must select the first desired sector (FSx or CSBOOTx) to erase. Subsequent sector erase commands that are appended within the time-out period must be addressed to other desired segments within the same Flash memory array.

28.5.18 Suspend sector erase

When a Sector Erase operation is in progress, the Suspend Sector Erase instruction sequence can be used to suspend the operation by writing B0h to any valid address within the Flash array that currently is undergoing an erase operation. This allows reading of data from a different Flash memory sector within the same array after the Erase operation has been suspended. Suspend Sector Erase is accepted only during an Erase operation.

There is up to 15µs delay after the Suspend Sector Erase command is accepted and the array goes to Read Array mode. The 8032 will monitor the Toggle Flag Bit (DQ6) to determine when the erase operation has halted and Read Array mode is active.

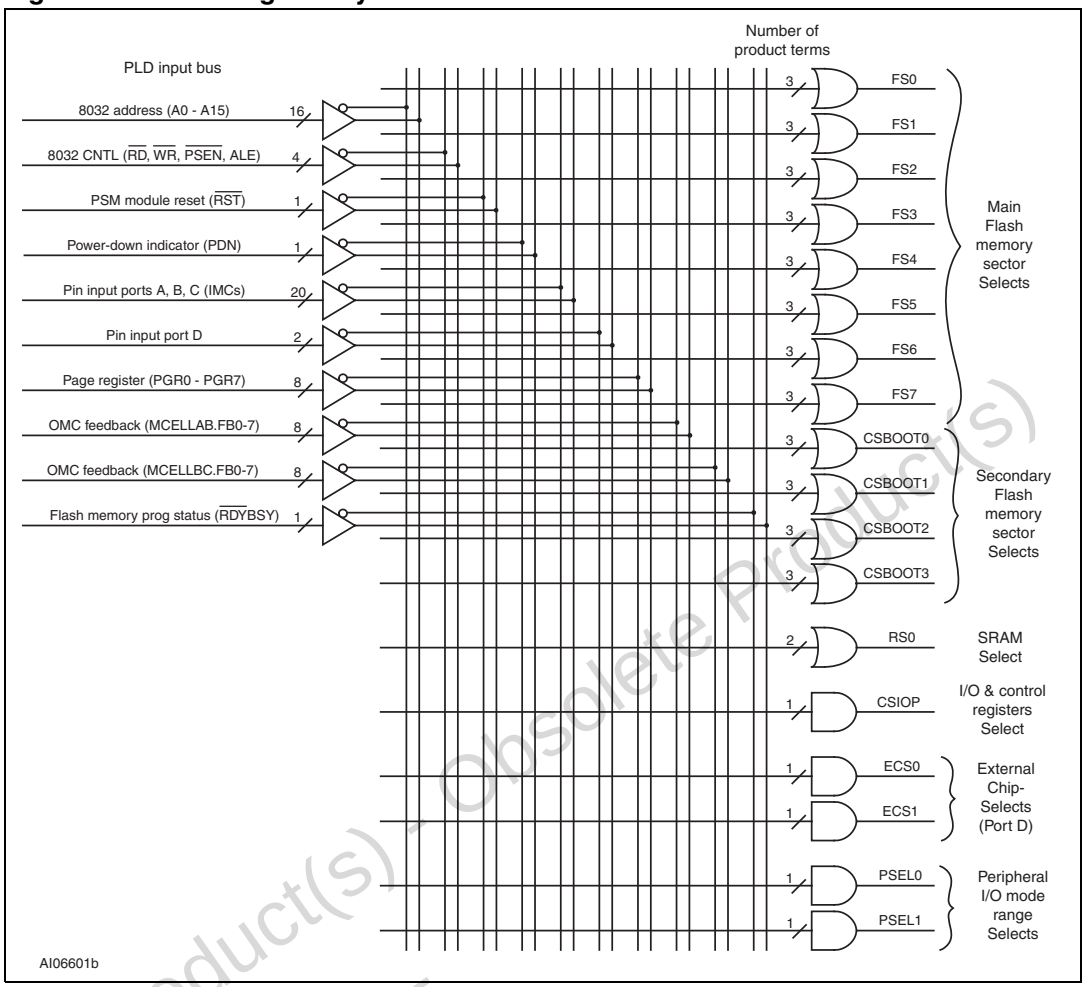
If a Suspend Sector Erase instruction sequence was executed, the following rules apply:

- Attempting to read from a Flash memory sector that was being erased outputs invalid data.
- Reading from a Flash memory sector that was *not* being erased is valid.
- The Flash memory *cannot* be programmed, and only responds to Resume Sector Erase and Reset Flash instruction sequences.
- If a Reset Flash instruction sequence is received, data in the Flash memory sector that was being erased is invalid.

28.5.19 Resume sector erase

If a Suspend Sector Erase instruction sequence was previously executed, the erase cycle may be resumed with this instruction sequence. The Resume Sector Erase instruction sequence consists of writing the command 30h to any valid address within the Flash array that was suspended as shown in *Table 163 on page 209*.

Figure 74. DPLD logic array



28.5.28 General PLD (GPLD)

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

The GPLD contains:

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

Table 168. OMC port and data bit assignments (continued)

OMC	Port assignment(1),(2)	Native product terms from AND-OR array	Maximum borrowed product terms	Data bit on 8032 data bus for loading or reading OMC
MCELLBC3	Port B3 or C3	4	5	D3
MCELLBC4	Port B4 or C4	4	6	D4
MCELLBC5	Port B5	4	6	D5
MCELLBC6	Port B6	4	6	D6
MCELLBC7	Port B7 orC7	4	6	D7

1. MCELLAB0-MCELLAB7 can be output to Port A pins only on 80-pin devices. Port A is not available on 52-pin devices.
2. Port pins PC0, PC1, PC5, and PC6 are dedicated JTAG pins and are not available as outputs for MCELLBC 0, 1, 5, or 6.

28.5.32 Loading and reading OMCs

Each of the two OMC groups (eight OMCs each) occupies a byte in csiop space, named MCELLAB and MCELLBC (see *Table 169* and *Table 170*). When the 8032 writes or reads these two OMC registers in csiop it is accessing each of the OMCs through its 8-bit data bus, with the bit assignment shown in *Table 168 on page 227*. Sometimes it is important to know the bit assignment when the user builds GPLD logic that is accessed by the 8032. For example, the user may create a 4-bit counter that must be loaded and read by the 8032, so the user must know which nibble in the corresponding csiop OMC register the firmware must access. The fitter report generated by PSDsoft Express will indicate how it assigned the OMCs and data bus bits to the logic. The user can optionally force PSDsoft Express to assign logic to specific OMCs and data bus bits if desired by using the 'PROPERTY' statement in PSDsoft Express. Please see the PSDsoft Express User's Manual for more information on OMC assignments.

Loading the OMC flip-flops with data from the 8032 takes priority over the PLD logic functions. As such, the preset, clear, and clock inputs to the flip-flop can be asynchronously overridden when the 8032 writes to the csiop registers to load the individual OMCs.

Table 169. Output macrocell MCELLAB (address = csiop + offset 20h)(1)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCELLAB 7	MCELLAB 6	MCELLAB 5	MCELLAB 4	MCELLAB 3	MCELLAB 2	MCELLAB 1	MCELLAB 0

1. All bits clear to logic '0' at power-on reset, but do not clear after warm reset conditions (non-power-on reset).

Table 170. Output macrocell MCELLBC (address = csiop + offset 21h)(1)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCELLBC 7	MCELLBC 6	MCELLBC 5	MCELLBC 4	MCELLBC 3	MCELLBC 2	MCELLBC 1	MCELLBC 0

1. All bits clear to logic '0' at power-on reset, but do not clear after warm reset conditions (non-power-on reset).

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

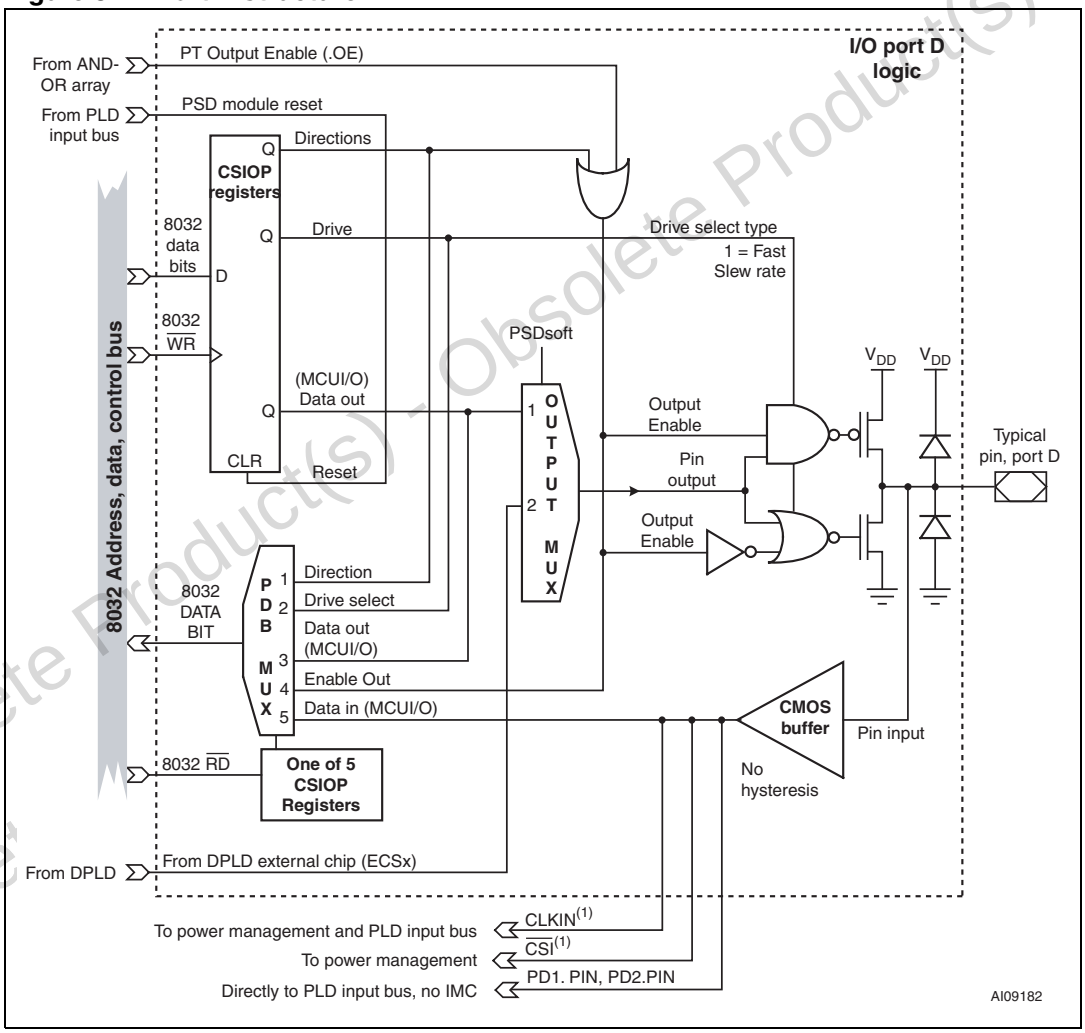
I/O Port logic has several modes of operation. *Table 171 on page 229* summarizes which modes are available on each port. Each of the port operating modes are described in

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.

Figure 110. External clock cycle

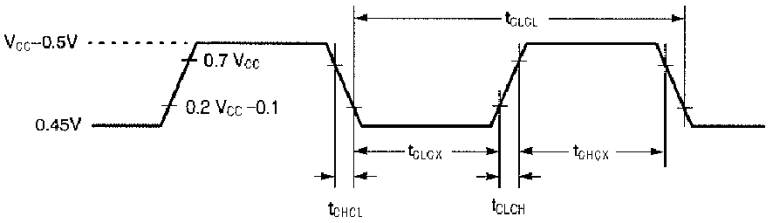


Figure 111. PSD module AC measurement I/O waveform

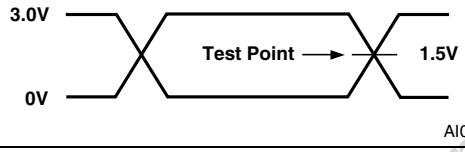


Figure 112. PSD module AC measurement load circuit

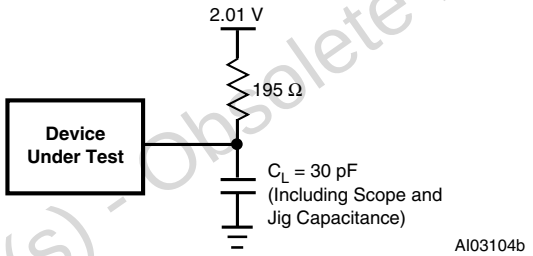


Table 236. I/O pin capacitance

Symbol	Parameter((1)	Test condition	Typ.(2)	Max.	Unit
C_{IN}	Input capacitance (for input pins)	V_{IN} = 0 V	4	6	pF
C_{OUT}	Output capacitance (for input/output pins)(3)	V_{OUT} = 0 V	8	12	pF

1. Sampled only, not 100% tested.
2. Typical values are for T_A = 25 °C and nominal supply voltages.
3. Maximum for MCU Address and Data lines is 20 pF each.

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).

34.12 Incorrect code execution when code banks are switched

Description

When a code bank is switched, the PFQ/BC contain values from the previously selected bank and are not automatically flushed and reloaded from the newly selected code bank.

Impact on application

Depending on the contents of the PFQ/BC when the code bank is switched, improper code execution may result.

Workaround

The PFQ/BC must be flushed when the code bank is changed. Disabling and re-enabling the PFQ/BC will flush them. The following instructions are an example of how to flush the PFQ/BC:

```
ANL    BUSCON,#03Fh    ;Disable PFQ/BC
ORL    BUSCON,#0C0h    ;Enable PFQ/BC
```

Bank switching is typically handled by tool vendors in a file called l51_bank.a51. The uPSD tools offered by Keil and Raisonance now include an updated version of l51_bank.a51 for the uPSD products that flushes the PFQ/BC. The most recent banking examples available from ST's website include the updated l51_bank.a51 files.

34.13 9th received data bit corrupted in UART modes 2 and 3

Description

If the 9th transmit data bit is written by firmware into TB8 at the same time as a received 9th bit is being written by the hardware into RB8, RB8 is not correctly updated. This applies to both UART0 and UART1. Typically, the 9th data bit is used as a parity bit to check for data transmission errors on a byte by byte basis.

Impact on application

UART Modes 2 and 3 can't be used reliably in full-duplex mode.

Workaround

Revision A and B - Some options include:

1. Only use Mode 1 (8 data bits) for full-duplex communication.
2. Use Mode 1 and a packet based communication protocol with a checksum or CRC to detect data transmission errors.
3. Use UART0 in mode 2 or 3 for transmitting data and UART1 in mode 2 or 3 for receiving data.
4. Use some form of handshaking to ensure that data is never transmitted and received simultaneously on a single UART configured in mode 2 or 3.