

Welcome to [E-XFL.COM](https://www.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	I ² C, IrDA, SPI, UART/USART, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3454e-40t6

Table 49.	BUSCON: bus control register (SFR 9Dh, reset value EBh)	88
Table 50.	BUSCON register bit definition	89
Table 51.	Number of MCU_CLK periods required to optimize bus transfer rate	90
Table 52.	WDKEY: Watchdog timer key register (SFR AEh, reset value 55h)	94
Table 53.	WDKEY register bit definition	94
Table 54.	WDRST: Watchdog timer reset counter register (SFR A6h, reset value 00h)	94
Table 55.	WDRST register bit definition	95
Table 56.	TCON: Timer control register (SFR 88h, reset value 00h)	97
Table 57.	TCON register bit definition	97
Table 58.	TMOD: Timer mode register (SFR 89h, reset value 00h)	99
Table 59.	TMOD register bit definition	99
Table 60.	T2CON: Timer 2 control register (SFR C8h, reset value 00h)	101
Table 61.	T2CON register bit definition	101
Table 62.	Timer/counter 2 operating modes	102
Table 63.	Commonly used baud rates generated from timer2 (T2CON = 34h)	104
Table 64.	UART operating modes	108
Table 65.	SCON0: serial port UART0 control register (SFR 98h, reset value 00h)	109
Table 66.	SCON0 register bit definition	109
Table 67.	SCON1: serial port UART1 control register (SFR D8h, reset value 00h)	109
Table 68.	SCON1 register bit definition	110
Table 69.	Commonly used baud rates generated from timer 1	111
Table 70.	IRDACON register bit definition (SFR CEh, reset value 0Fh)	120
Table 71.	IRDACON register bit definition	120
Table 72.	Baud rate selection register (SFR xxh, reset value xxh)	120
Table 73.	Baud rate of UART#1 for IrDA interface	121
Table 74.	Recommended CDIV[4:0] values to generate SIRCk (default CDIV[4:0] = 0Fh, 15 decimal)	122
Table 75.	Serial control register S1CON (SFR DCh, reset value 00h)	129
Table 76.	S1CON register bit definition	129
Table 77.	Selection of the SCL frequency in Master mode based on f _{OSC} examples	130
Table 78.	S1STA: I ² C interface status register (SFR DDh, reset value 00h)	130
Table 79.	S1STA register bit definition	131
Table 80.	S1DAT: I2C data shift register (SFR DEh, reset value 00h)	132
Table 81.	S1DAT register bit definition	132
Table 82.	S1ADR: I2C address register (SFR DFh, reset value 00h)	132
Table 83.	S1ADR register bit definition	132
Table 84.	S1SETUP: I ² C Start condition sample setup register (SFR DBh, reset value 00h)	133
Table 85.	S1SETUP register bit definition	133
Table 86.	Number of I ² C bus samples taken after 1-to-0 transition on SDA (Start condition)	134
Table 87.	Start condition hold time	134
Table 88.	S1SETUP examples for various I ² C bus speeds and oscillator frequencies	134
Table 89.	SPICON0: control register 0 (SFR D6h, reset value 00h)	147
Table 90.	SPICON0 register bit definition	147
Table 91.	SPICON1: SPI interface control register 1 (SFR D7h, reset value 00h)	148
Table 92.	SPICON1 register bit definition	148
Table 93.	SPICLKD: SPI prescaler (clock divider) register (SFR D2h, reset value 04h)	148
Table 94.	SPICLKD register bit definition	149
Table 95.	SPISTAT: SPI interface status register (SFR D3h, reset value 02h)	149
Table 96.	SPISTAT register bit definition	149
Table 97.	Types of packet IDs	152

Table 150.	PCA1 control register PCACON1 (SFR 0BCh, reset value 00h)	188
Table 151.	PCA1 register bit definition	188
Table 152.	PCA status register PCASTA (SFR 0A5h, reset value 00h)	188
Table 153.	PCASTA register bit definition	188
Table 154.	TCMMODE0 - TCMMODE5 (6 registers, reset value 00h)	189
Table 155.	TCMMODE0 - TCMMODE5 register bit definition	189
Table 156.	TCMMODE register configurations	190
Table 157.	UPSD34xx memory configuration	193
Table 158.	General I/O pins on PSD module	196
Table 159.	HDL statement example generated from PSDsoft express for memory map	199
Table 160.	VM register (address = csiop + offset E2h)	203
Table 161.	Data width in different bus cycles	205
Table 162.	CSIOP registers and their offsets (in hexadecimal)	205
Table 163.	Flash memory instruction sequences	209
Table 164.	Flash memory status bit definition	212
Table 165.	Main Flash memory protection register definition (address = csiop + offset C0h)	219
Table 166.	Secondary Flash memory protection/security register definition (csiop + offset C2h)	219
Table 167.	DPLD and GPLD inputs	220
Table 168.	OMC port and data bit assignments	227
Table 169.	Output macrocell MCELLAB (address = csiop + offset 20h)	228
Table 170.	Output macrocell MCELLBC (address = csiop + offset 21h)	228
Table 171.	Output macrocell MCELLAB mask register (address = csiop + offset 22h)	229
Table 172.	Output macrocell MCELLBC mask register (address = csiop + offset 23h)	229
Table 173.	Input macrocell port A (address = csiop + offset 0Ah)	230
Table 174.	Input macrocell port B (address = csiop + offset 0Bh)	230
Table 175.	Input macrocell port C (address = csiop + offset 18h)	230
Table 176.	Port operating modes	232
Table 177.	Port configuration setting requirements	233
Table 178.	MCU I/O mode port A data in register (address = csiop + offset 00h)	234
Table 179.	MCU I/O mode port B data in register (address = csiop + offset 01h)	234
Table 180.	MCU I/O mode port C data in register (address = csiop + offset 10h)	234
Table 181.	MCU I/O mode port D Data in register (address = csiop + offset 11h)	234
Table 182.	MCU I/O mode port A data out register (address = csiop + offset 04h)	234
Table 183.	MCU I/O mode port B data out register (address = csiop + offset 05h)	235
Table 184.	MCU I/O mode port C data out register (address = csiop + offset 12h)	235
Table 185.	MCU I/O mode port D data out register (address = csiop + offset 13h)	235
Table 186.	MCU I/O mode port A direction register (address = csiop + offset 06h)	235
Table 187.	MCU I/O mode port B direction in register (address = csiop + offset 07h)	235
Table 188.	MCU I/O mode port C direction register (address = csiop + offset 14h)	235
Table 189.	MCU I/O mode port D direction register (address = csiop + offset 15h)	236
Table 190.	Latched address output, port A contro register(address = csiop + offset 02h)I	238
Table 191.	Latched address output, port B contro register (address = csiop + offset 03h)I	239
Table 192.	Port A pin drive select register (address = csiop + offset 08h)	241
Table 193.	Port B pin drive select register (address = csiop + offset 09h)	242
Table 194.	Port C pin drive select register (address = csiop + offset 16h)	242
Table 195.	Port D pin drive select register (address = csiop + offset 17h)	242
Table 196.	Port A enable out register (address = csiop + offset 0Ch)	242
Table 197.	Port B enable out register (address = csiop + offset 0Dh)	242
Table 198.	Port C enable out register (address = csiop + offset 1Ah)	242
Table 199.	Port D enable out register (address = csiop + offset 1Bh)	242
Table 200.	Power management mode register PMMR0 (address = csiop + offset B0h)	249
Table 201.	Power management mode register PMMR2 (address = csiop + offset B4h)	249

List of figures

Figure 1.	Block diagram	21
Figure 2.	LQFP52 connections	22
Figure 3.	LQFP80 connections	23
Figure 4.	Functional modules	29
Figure 5.	UPSD34xx memories	30
Figure 6.	Comparison of UPSD34xx with standard 8032 performance	33
Figure 7.	Instruction pre-fetch queue and branch cache	34
Figure 8.	PFQ operation on multi-cycle instructions	35
Figure 9.	UPSD34xx multi-cycle instructions compared to standard 8032	36
Figure 10.	8032 MCU registers	38
Figure 11.	Program status word (PSW) register	40
Figure 12.	Enabling and polling interrupts	64
Figure 13.	Clock generation logic	70
Figure 14.	Oscillator and clock connections	77
Figure 15.	MCU module port pin function routing	80
Figure 16.	MCU I/O cell block diagram for port 1	80
Figure 17.	MCU I/O cell block diagram for port 3	81
Figure 18.	MCU I/O cell block diagram for port 4	81
Figure 19.	Connecting external devices using ports A and B for address AD[15:0]	87
Figure 20.	Connecting external devices using port A and an external latch for address AD[15:0]	87
Figure 21.	A RD or PSEN bus cycle set to 5 MCU_CLK	88
Figure 22.	Supervisor reset generation	91
Figure 23.	Watchdog counter	93
Figure 24.	Timer/counter mode 0: 13-bit counter	100
Figure 25.	Timer/counter mode 2: 8-bit Auto-reload	100
Figure 26.	Timer/counter mode 3: two 8-bit counters	100
Figure 27.	Timer 2 in capture mode	105
Figure 28.	Timer 2 in auto-reload mode	106
Figure 29.	Timer 2 in baud rate generator mode	106
Figure 30.	UART mode 0, block diagram	113
Figure 31.	UART mode 0, timing diagram	113
Figure 32.	UART mode 1, block diagram	115
Figure 33.	UART mode 1, timing diagram	115
Figure 34.	UART mode 2, block diagram	117
Figure 35.	UART mode 2, timing diagram	117
Figure 36.	UART mode 3, block diagram	118
Figure 37.	UART mode 3, timing diagram	118
Figure 38.	IrDA interface	119
Figure 39.	Pulse shaping by the IrDA interface	119
Figure 40.	Typical I2C bus configuration	123
Figure 41.	Data transfer on an I ² C bus	125
Figure 42.	I ² C interface SIOE block diagram	128
Figure 43.	SPI device connection examples	142
Figure 44.	SPI full-duplex data exchange	144
Figure 45.	SPI receive operation example	144
Figure 46.	SPI transmit operation example	145
Figure 47.	SPI interface, master mode only	146
Figure 48.	USB module block diagram	151

Figure 49.	USB packets in a USB transfer example	153
Figure 50.	IN and OUT bulk transfers	154
Figure 51.	Interrupt transfer	155
Figure 52.	Control transfer	156
Figure 53.	FIFOs with no pairing	158
Figure 54.	FIFO pairing example (1/2 IN paired and 3/4 OUT paired)	159
Figure 55.	Typical self powered example	177
Figure 56.	10-bit ADC	179
Figure 57.	PCA0 block diagram	181
Figure 58.	Timer mode	184
Figure 59.	PWM mode - (x8), fixed frequency	185
Figure 60.	PWM mode - (x8) programmable frequency	186
Figure 61.	PSD module block diagram	191
Figure 62.	Memory page register	194
Figure 63.	Typical system memory map	198
Figure 64.	PSDsoft express memory mapping	199
Figure 65.	Mapping: split second Flash in half	200
Figure 66.	Mapping: all Flash in code space	201
Figure 67.	Mapping: small code / big data	201
Figure 68.	PSD module memory priority	202
Figure 69.	VM register control of memories	204
Figure 70.	VM register example corresponding to memory map example	204
Figure 71.	Data polling flowchart	214
Figure 72.	Data toggle flowchart	215
Figure 73.	DPLD and GPLD	221
Figure 74.	DPLD logic array	223
Figure 75.	GPLD: one OMC, one IMC, and one I/O port (typical pin, port A, B, or C)	224
Figure 76.	Detail of a single OMC	226
Figure 77.	OMC allocator	227
Figure 78.	Detail of a single IMC	230
Figure 79.	Detail of a single I/O port (typical of ports A, B, C)	232
Figure 80.	Simple PLD logic example	237
Figure 81.	Pin declarations in PSDsoft express for simple PLD example	237
Figure 82.	Using the design assistant in PSDsoft Express for simple PLD example	238
Figure 83.	Peripheral I/O mode	239
Figure 84.	Port A structure	243
Figure 85.	Port B structure	244
Figure 86.	Port C structure	246
Figure 87.	Port D structure	247
Figure 88.	Automatic power-down (APD) unit	252
Figure 89.	Power-down mode flowchart	253
Figure 90.	JTAG chain in UPSD34xx package	258
Figure 91.	Recommended 4-pin JTAG connections	259
Figure 92.	Recommended 6-pin JTAG connections	261
Figure 93.	Recommended JTAG connector	262
Figure 94.	Example of chaining UPSD34xx devices	263
Figure 95.	PLD ICC / frequency consumption (5 V range)	265
Figure 96.	PLD ICC / frequency consumption (3 V range)	266
Figure 97.	Switching waveforms – key	270
Figure 98.	External READ cycle (80-pin device only)	275
Figure 99.	External WRITE cycle (80-pin device only)	276
Figure 100.	Input to output disable / enable	278

6 MCU module description

The following sections provide a detailed description of the MCU module system functions and peripherals, including:

- 8032 MCU registers
- Special function registers
- 8032 addressing modes
- UPSD34xx instruction set summary
- Dual data pointers
- Debug unit
- Interrupt system
- MCU clock generation
- Power saving modes
- Oscillator and external components
- I/O ports
- MCU bus interface
- Supervisory functions
- Standard 8032 timer/counters
- Serial UART interfaces
- IrDA interface
- I²C interface
- SPI interface
- Analog to digital converter
- Programmable counter array (PCA)
- USB interface

8 Special function registers (SFR)

A group of registers designated as special function register (SFR) is shown in [Table 5 on page 42](#). SFRs control the operating modes of the MCU core and also control the peripheral interfaces and I/O pins on the MCU module. The SFRs can be accessed only by using the Direct Addressing method within the address range from 80h to FFh of internal 8032 SRAM. Sixteen addresses in SFR address space are both byte- and bit-addressable. The bit-addressable SFRs are noted in [Table 5](#).

106 of a possible 128 SFR addresses are occupied. The remaining unoccupied SFR addresses (designated as “RESERVED” in [Table 5](#)) should not be written. Reading unoccupied locations will return an undefined value.

*Note: There is a separate set of control registers for the PSD module, designated as *csiop*, and they are described in the [Section 28: PSD module on page 191](#). The I/O pins, PLD, and other functions on the PSD module are NOT controlled by SFRs.*

SFRs are categorized as follows:

- MCU core registers:
IP, A, B, PSW, SP, DPTL, DPTH, DPTC, DPTM
- MCU module I/O Port registers:
P1, P3, P4, P1SFS0, P1SFS1, P3SFS, P4SFS0, P4SFS1
- Standard 8032 Timer registers
TCON, TMOD, T2CON, TH0, TH1, TH2, TL0, TL1, TL2, RCAP2L, RCAP2H
- Standard Serial Interfaces (UART)
SCON0, SBUF0, SCON1, SBUF1
- Power, clock, and bus timing registers
PCON, CCON0, CCON1, BUSCON
- Hardware watchdog timer registers
WDKEY, WDRST
- Interrupt system registers
IP, IPA, IE, IEA
- Prog. Counter Array (PCA) control registers
PCACL0, PCACH0, PCACON0, PCASTA, PCACL1, PCACH1, PCACON1, CCON2, CCON3
- PCA capture/compare and PWM registers
CAPCOML0, CAPCOMH0, TCMODE0, CAPCOML1, CAPCOMH1, TCMODE2, CAPCOML2, CAPCOMH2, TCMODE2, CAPCOML3, CAPCOMH3, TCMODE3,

9.4 Immediate addressing

This mode uses 8-bits of data (a constant) contained in the second byte of the instruction, and stores it into the memory location or register indicated by the first byte of the instruction. Thus, the data is immediately available within the instruction. This mode is commonly used to initialize registers and SFRs or to perform mask operations.

There is also a 16-bit version of this mode for loading the DPTR register. In this case, the two bytes following the instruction byte contain the 16-bit value. For example:

```
MOV A, 40#           ; Move the constant, 40h, into
                    ; the accumulator

MOV DPTR, 1234#      ; Move the constant, 1234h, into
                    ; DPTR
```

9.5 External direct addressing

This mode will access external memory (XDATA) by using the 16-bit address stored in the DPTR register. There are only two instructions using this mode and both use the accumulator to either receive a byte from external memory addressed by DPTR or to send a byte from the accumulator to the address in DPTR. The UPSD34xx has a special feature to alternate the contents (source and destination) of DPTR rapidly to implement very efficient memory-to-memory transfers. For example:

```
MOVX A, @DPTR        ; Move contents of accumulator to
                    ; XDATA at address contained in
                    ; DPTR

MOVX @DPTR, A         ; Move XDATA to accumulator
```

Note: See details in [Section 11: Dual data pointers on page 57](#).

9.6 External indirect addressing

This mode will access external memory (XDATA) by using the 8-bit address stored in either register R0 or R1. This is the fastest way to access XDATA (least bus cycles), but because only 8-bits are available for address, this mode limits XDATA to a size of only 256 bytes (the traditional Port 2 of the 8032 MCU is not available in the UPSD34xx, so it is not possible to write the upper address byte).

For example:

```
MOVX @R0,A           ; Move into the accumulator the
                    ; XDATA that is pointed to by
                    ; the address contained in R0.
```

Note: This mode is not supported by UPSD34xx.

Table 34. PCON register bit definition (continued)

Bit	Symbol	R/W	Function
1	PD	R,W	Activate Power-down mode 0 = Not in Power-down mode 1 = Enter Power-down mode
0	IDL	R,W	Activate Idle mode 0 = Not in Idle mode 1 = Enter Idle mode

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

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

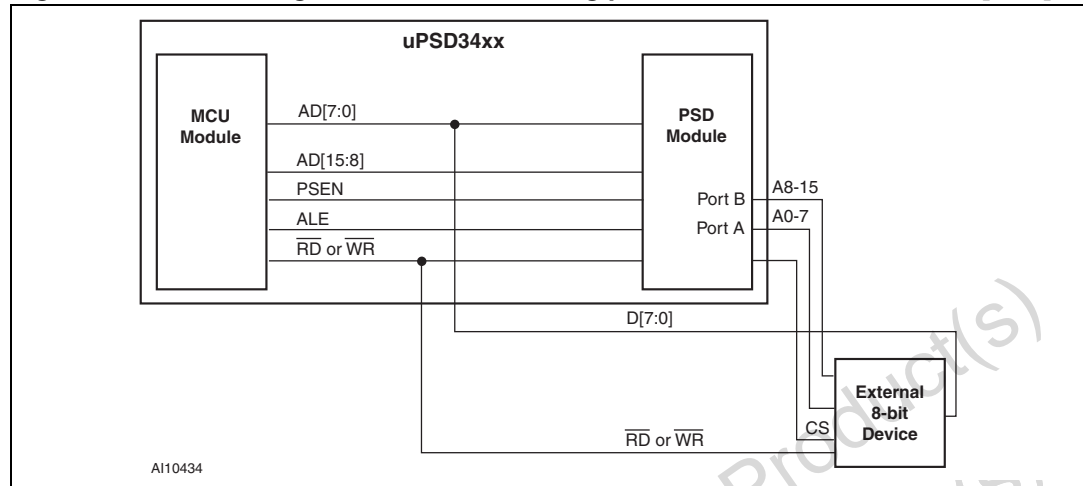
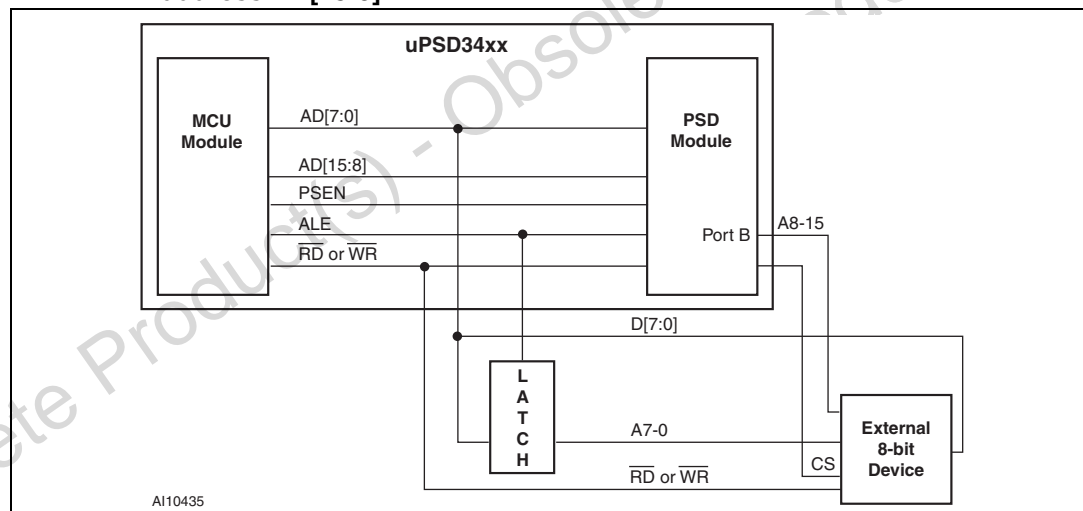


Figure 20. Connecting external devices using port A and an external latch for address AD[15:0]



18.4 Programmable bus timing

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

21.2 Serial port control registers

The SFR SCON0 controls UART0, and SCON1 controls UART1, shown in [Table 65](#) and [Table 67](#). These registers contain not only the mode selection bits, but also the 9th data bit for transmit and receive (bits TB8 and RB8), and the UART Interrupt flags, TI and RI.

Table 65. SCON0: serial port UART0 control register (SFR 98h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Table 66. SCON0 register bit definition

Bit	Symbol	R/W	Definition
7	SM0	R,W	Serial Mode Select, See Table 64 on page 108 . Important, notice bit order of SM0 and SM1. [SM0:SM1] = 00b, Mode 0 [SM0:SM1] = 01b, Mode 1 [SM0:SM1] = 10b, Mode 2 [SM0:SM1] = 11b, Mode 3
6	SM1	R,W	
5	SM2	R,W	Serial Multiprocessor Communication Enable. Mode 0: SM2 has no effect but should remain 0. Mode 1: If SM2 = 0 then stop bit ignored. SM2 = 1 then RI active if stop bit = 1. Mode 2 and 3: Multiprocessor Comm Enable. If SM2=0, 9th bit is ignored. If SM2=1, RI active when 9th bit = 1.
4	REN	R,W	Receive Enable. If REN=0, UART reception disabled. If REN=1, reception is enabled
3	TB8	R,W	TB8 is assigned to the 9th transmission bit in Mode 2 and 3. Not used in Mode 0 and 1.
2	RB8	R,W	Mode 0: RB8 is not used. Mode 1: If SM2 = 0, the RB8 is the level of the received stop bit. Mode 2 and 3: RB8 is the 9th data bit that was received in Mode 2 and 3.
1	TI	R,W	Transmit Interrupt flag. Causes interrupt at end of 8th bit time when transmitting in Mode 0, or at beginning of stop bit transmission in other modes. Must clear flag with firmware.
0	RI	R,W	Receive Interrupt flag. Causes interrupt at end of 8th bit time when receiving in Mode 0, or halfway through stop bit reception in other modes (see SM2 for exception). Must clear this flag with firmware.

Table 67. SCON1: serial port UART1 control register (SFR D8h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Figure 36. UART mode 3, block diagram

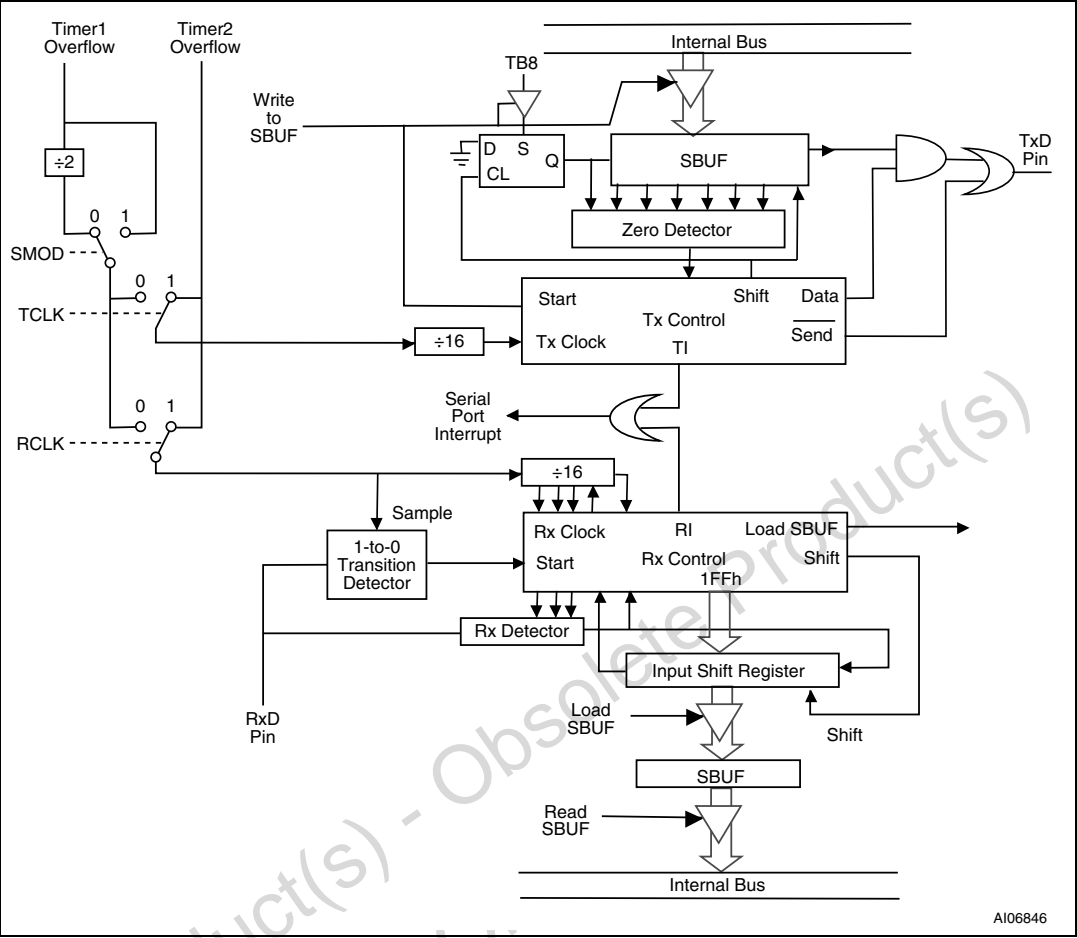
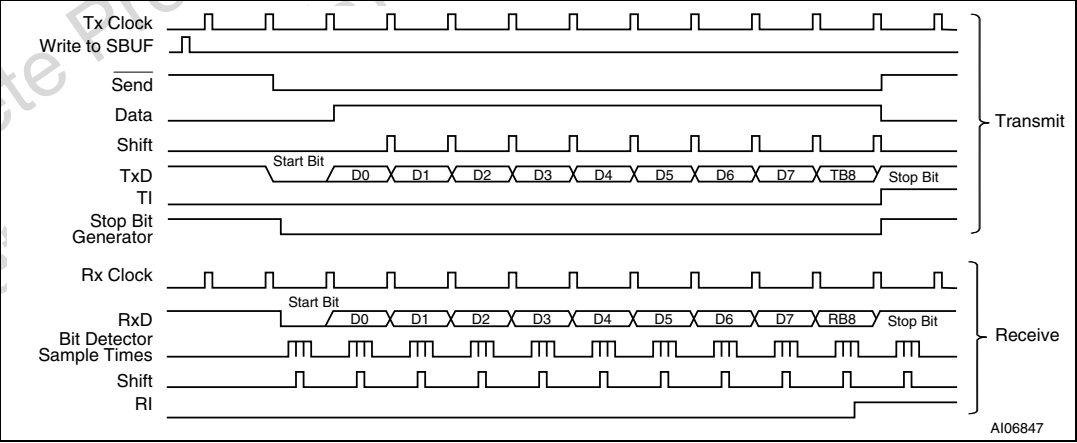


Figure 37. UART mode 3, timing diagram



25.4.2 Endpoint FIFO pairing

Endpoint FIFOs can be paired for double buffering to provide an efficient method for bulk data transfers. With double buffering enabled, the MCU can operate on one data packet while another is being transferred over USB.

When two FIFOs are paired, the active FIFO is automatically toggled by the update of USIZE. The MCU must only use the odd numbered endpoint FIFO when paired in order to access the active FIFO. For example, if endpoints 3 and 4 OUT FIFOs are paired, the active FIFO is accessed via endpoint 3's OUT FIFO (see [Table 102](#)).

Table 100. USB device address register (UADDR 0E2h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	USBADDR[6:0]						

Table 101. UADDR register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6:0	USBADDR	R/W	USB Address of the device. These bits are cleared with a Hardware RESET. When a USB RESET is detected, the address register should be cleared.

Table 102. Pairing control register (UPAIR 0E3h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	PR3OUT	PR1OUT	PR3IN	PR1IN

Table 103. UPAIR register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	–	–	Reserved
3	PR3OUT	R/W	Setting this bit enables double buffering of the OUT FIFOs for Endpoints 3 and 4. Access to the double buffered FIFOs is through Endpoint3's OUT FIFO.
2	PR1OUT	R/W	Setting this bit enables double buffering of the OUT FIFOs for Endpoints 1 and 2. Access to the double buffered FIFOs is through Endpoint1's OUT FIFO.
1	PR3IN	R/W	Setting this bit enables double buffering of the IN FIFOs for Endpoints 3 and 4. Access to the double buffered FIFOs is through Endpoint3's IN FIFO.
0	PR1IN	R/W	Setting this bit enables double buffering of the IN FIFOs for Endpoints 1 and 2. Access to the double buffered FIFOs is through Endpoint1's IN FIFO.

- USB setup command index and value registers (USCI and USCV)
When a Setup/Data packet is received over the USB, the 8 bytes of data received are stored in a command buffer. The USB setup command index register (see [Table 134](#)) determines which one of the eight bytes in the buffer is read using the USB setup command value register (see [Table 136](#)).

Table 134. USB setup command index register (USCI 0F5h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	USCI[2:0]		

Table 135. USCI register bit definition

Bit	Symbol	R/W	Definition
7:3	–	–	Reserved
2:0	USCI[2:0]	R/W	Index to access one of the 8 bytes of USB Setup Command Data received with the last Setup transaction

Table 136. USB setup command value register (USCV 0F6h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USCV[7:0]							

Table 137. USCV register bit definition

Bit	Symbol	R/W	Definition
7:0	USCV	R/W	The nth byte of the 8 bytes of USB Setup Command Data received with the last Setup transaction. The nth byte that is read from this register is specified by the index value in the USCI register.

from the Secondary Flash memory in program space. After the writing is complete, the Main Flash can be “reclassified” back to program space, then execution can continue from the new code in Main Flash memory. The mapping example of [Figure 67](#) will accommodate this operation.

28.2.6 Memory sector select rules

When defining sector select signals (FSx, CSBOOTx, RS0, CSIOP, PSELx) in PSDsoft Express, the user must keep these rules in mind:

- Main Flash and Secondary Flash memory sector select signals may not be larger than their physical sector size as defined in [Table 157 on page 193](#).
- Any Main Flash memory sector select may not be mapped in the same address range as another Main Flash sector select (cannot overlap segments of Main Flash on top of each other).
- Any Secondary Flash memory sector select may not be mapped in the same address range as another Secondary Flash sector select (cannot overlap segments of Secondary Flash on top of each other).
- A Secondary Flash memory sector may overlap a Main Flash memory sector. In the case of overlap, priority is given to the Secondary Flash memory sector.
- SRAM, CSIOP, or PSELx may overlap any Flash memory sector. In the case of overlap, priority is given to SRAM, CSIOP, or PSELx.

Note: PSELx is for optional Peripheral I/O Mode on Port A.

- The address range for sector selects for SRAM, PSELx, and CSIOP must not overlap each other as they have the same priority, causing contention if overlapped.

[Figure 68](#) illustrates the priority scheme of the memory elements of the PSD module. Priority refers to which memory will ultimately produce a byte of data or code to the 8032 MCU for a given bus cycle. Any memory on a higher level can overlap and has priority over any memory on a lower level. Memories on the same level must not overlap.

Example: FS0 is valid when the 8032 produces an address in the range of 8000h to BFFFh. CSBOOT0 is valid from 8000h to 9FFFh. RS0 is valid from 8000h to 87FFh. Any address from the 8032 in the range of RS0 always accesses the SRAM. Any address in the range of CSBOOT0 greater than 87FFh (and less than 9FFFh) automatically addresses Secondary Flash memory. Any address greater than 9FFFh accesses Main Flash memory. One-half of the Main Flash memory segment and one-fourth of the Secondary Flash memory segment cannot be accessed by the 8032.

Figure 68. PSD module memory priority

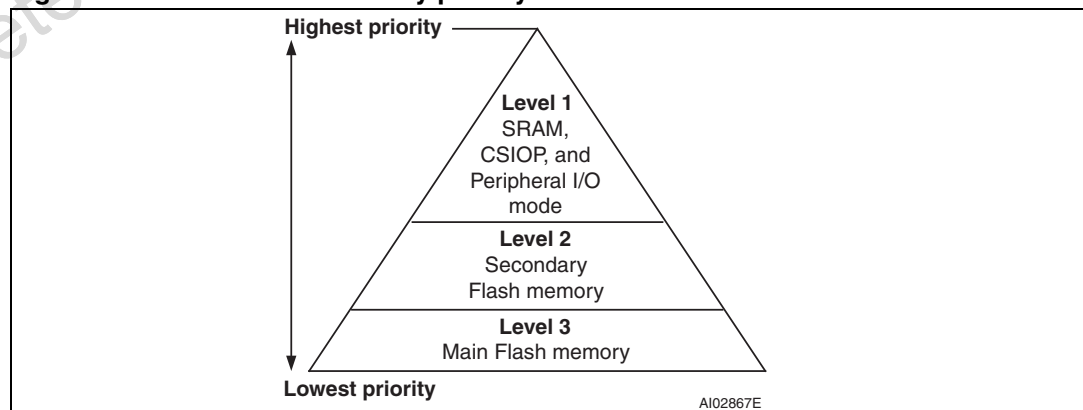
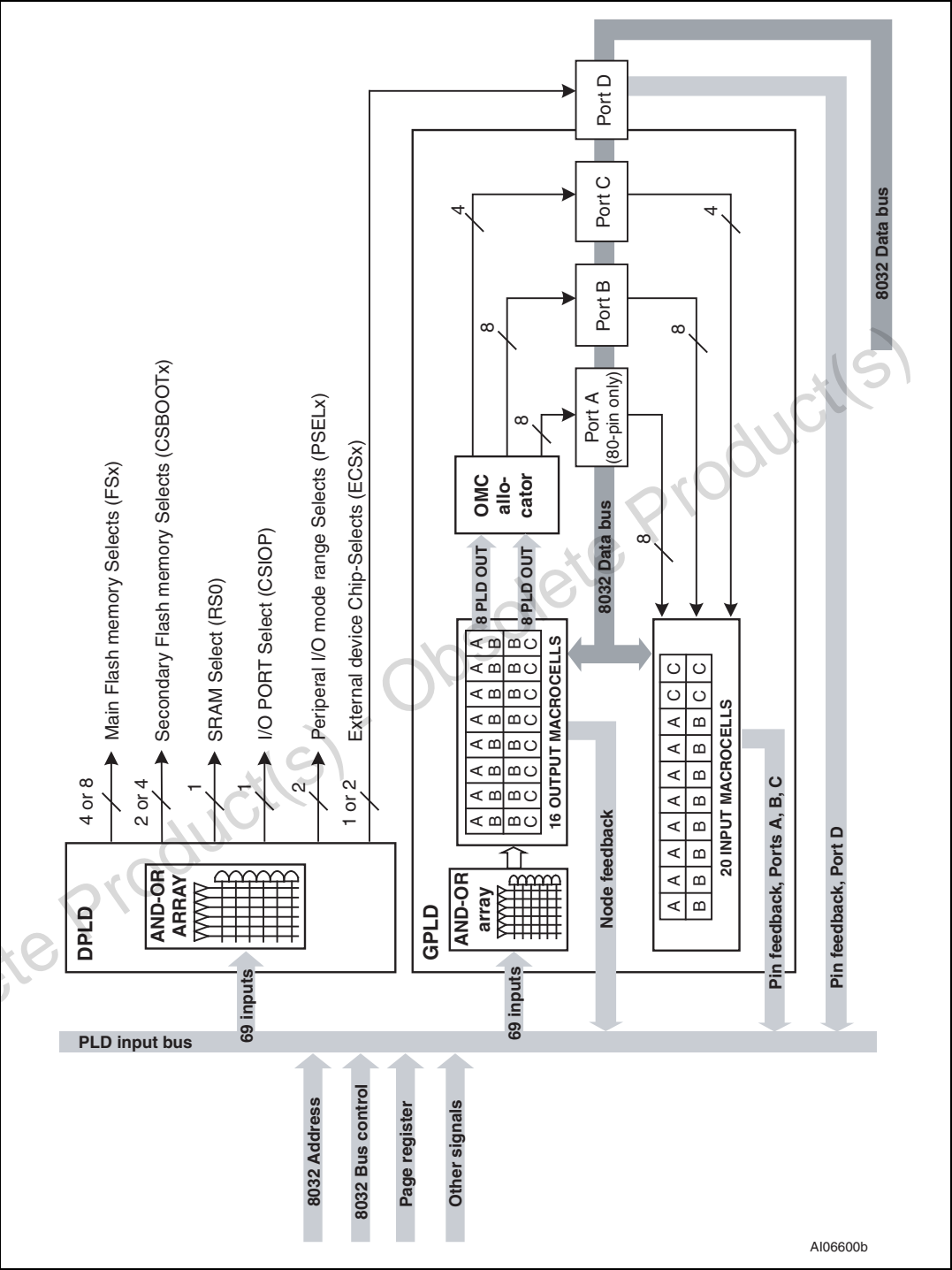


Figure 73. DPLD and GPLD



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](#) through [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:

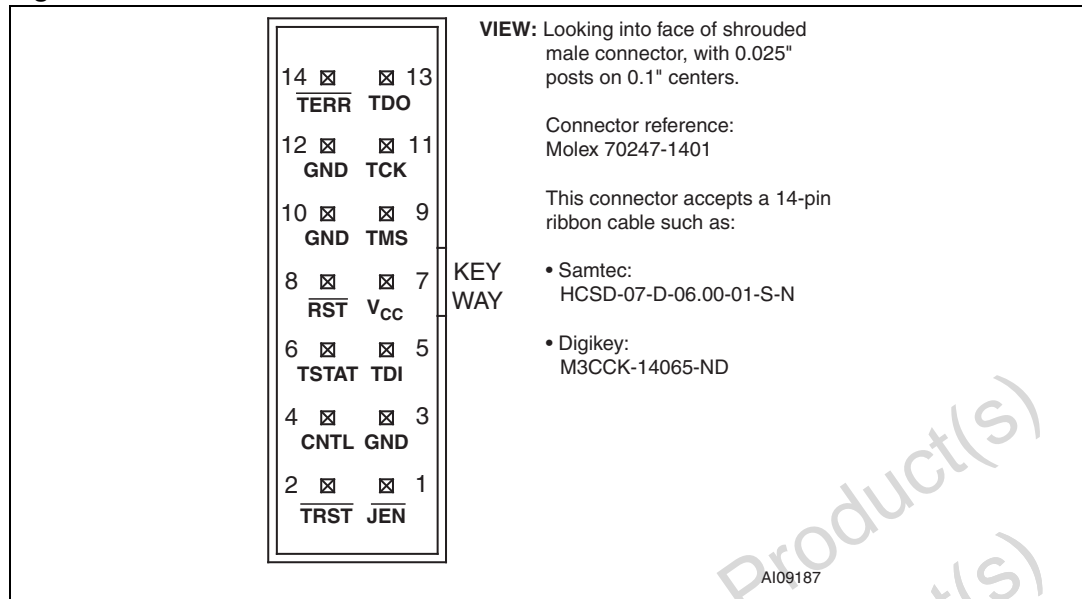
1. MCU I/O signal direction setting (csiop Direction reg)
2. Pin drive type setting (csiop Drive Select reg)
3. Latched Addr Out mode setting (csiop Control reg)
4. MCU I/O pin output setting (csiop Data Out reg)
5. Output Enable of pin driver (csiop Enable Out reg)
6. MCU I/O pin input (csiop Data In 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

Figure 93. Recommended JTAG connector

28.6.7 Chaining UPSD34xx devices

It is possible to chain a UPSD34xx device with other UPSD34xx devices on a circuit board, and also chain with IEEE 1149.1 compliant devices from other manufacturers. [Figure 94 on page 263](#) shows a chaining example. The TDO of one device connects to the TDI of the next device, and so on. Only one device is performing JTAG operations at any given time while the other two devices are in BYPASS mode. Configuration for JTAG chaining can be made in PSDsoft Express by choosing "More than one device" when prompted about chaining devices. Notice in [Figure 94 on page 263](#) that the UPSD34xx devices are chained externally, but also be aware that the two die within each UPSD34xx device are chained internally. This internal chaining of die is transparent to the user and is taken care of by PSDsoft Express and 3rd party JTAG tool software.

The example in [Figure 94 on page 263](#) also shows how to use 6-pin JTAG when chaining devices. The signals TSTAT and $\overline{\text{TERR}}$ are configured as open-drain type signals from PSDsoft Express. This facilitates a wired-OR connection of TSTAT signals from multiple UPSD34xx devices and also a wired-OR connection of $\overline{\text{TERR}}$ signals from those same multiple devices. PSDsoft Express puts TSTAT and $\overline{\text{TERR}}$ signals into open-drain mode by default, requiring external pull-up resistors. Click on 'Properties' in the JTAG-ISP window of PSDsoft Express to change to standard CMOS push-pull outputs if desired, but wired-OR logic is not possible in CMOS output mode.

31 DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristic tables that follow are derived from tests performed under the measurement conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 206. Operating conditions (5 V devices)

Symbol	Parameter	Min.	Max.	Unit
V_{DD}	Supply voltage	4.5	5.5	V
V_{CC} , AV_{CC}	Supply voltage	3.0	3.6	V
T_A	Ambient operating temperature (industrial)	-40	85	°C
	Ambient operating temperature (commercial)	0	70	°C

Table 207. Operating conditions (3.3 V devices)

Symbol	Parameter	Min.	Max.	Unit
V_{CC} , V_{DD} , AV_{CC}	Supply voltage	3.0	3.6	V
T_A	Ambient operating temperature (industrial)	-40	85	°C
	Ambient operating temperature (commercial)	0	70	°C

Table 208. AC signal letters for timing

Letter	Meaning
A	Address
C	Clock
D	Input data
I	Instruction
L	ALE
N	\overline{RESET} input or output
P	PSEN signal
Q	Output data
R	RD signal
W	WR signal
M	Output Macrocell

Note: Example: t_{AVLX} = time from address valid to ALE invalid.

Workaround

Revision A and B - When a USB reset is detected, the USB SIE's registers must be initialized appropriately by the firmware. The 3400 USB firmware examples clear USB SIE's registers if USB reset is detected.

34.4 Data toggle

Description

The data toggle bit is read only.

Impact on application

The IN FIFO data toggle bit is controlled exclusively by the USB SIE; therefore, it is not possible to change the state of the data toggle bit by firmware.

Workaround

Revision A - For cases where the data toggle bit must be reset, such as after a Clear Feature/Stall request, sending the subsequent data on that endpoint twice results in getting the data toggle bit back to the state that it should be.

Revision B - A change in silicon was made so that the data toggle bit is reset by disabling and then enabling the respective endpoint's FIFO.

34.5 USB FIFO accessibility

Description

The USB FIFO is only accessible by firmware and not by a JTAG debugger.

Impact on application

Using a JTAG debugger, it is not possible to view the USB FIFO's contents in a memory dump window.

Workaround

Revision A and B - None identified at this time.

34.6 Erroneous resend of data packet

Description

When a data packet is sent the respective IN FIFO busy bit is not automatically cleared by the USB SIE. This can cause a data packet to be erroneously resent to the host in response to an IN PID immediately after the first correct transmission of this data packet.

Impact on application

Since the Data Toggle in the retransmitted data packet is toggled from when the data was first sent, the host will treat this packet as valid. If the identified workaround is not

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2009 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com