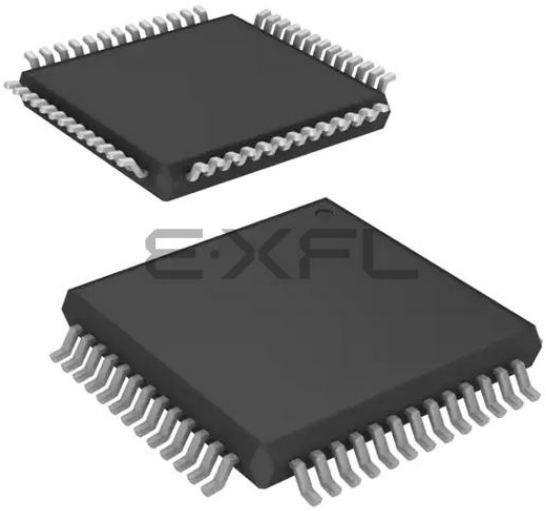


Welcome to E-XFL.COM

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	288KB (288K 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/upsd3434eb40t6

List of figures

7.7.5 Overflow flag (OV)

The OV flag is set when: an ADD, ADDC, or SUBB instruction causes a sign change; a MUL instruction results in an overflow (result greater than 255); a DIV instruction causes a divide-by-zero condition. The OV flag is cleared by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases. The CLRV instruction will clear the OV flag at any time.

7.7.6 Parity flag (P)

The P flag is set if the sum of the eight bits in the Accumulator is odd, and P is cleared if the sum is even.

Table 4. Register bank select addresses

RS1	RS0	Register bank	8032 internal data address
0	0	0	00h - 07h
0	1	1	08h - 0Fh
1	0	2	10h - 17h
1	1	3	18h - 1Fh

Figure 11. Program status word (PSW) register

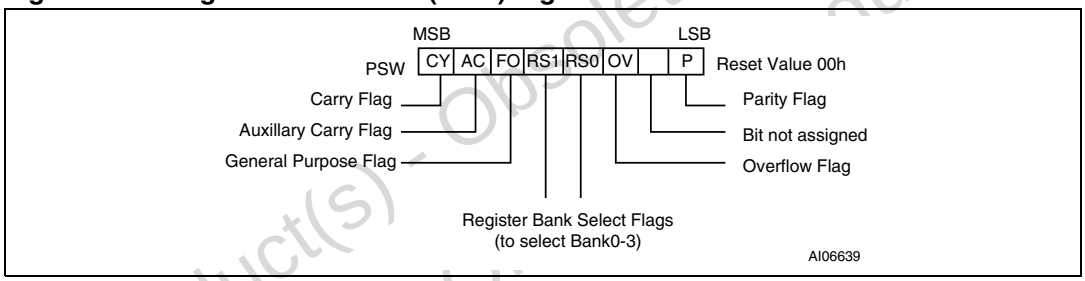


Table 10. Program branching instruction set (continued)

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

1. All mnemonics copyrighted ©Intel Corporation 1980.

Table 11. Miscellaneous instruction set

Mnemonic(1) and Use		Description	Length/Cycles
Miscellaneous			
NOP		No operation	1 byte/1 cycle

1. All mnemonics copyrighted ©Intel Corporation 1980.

Table 12. Notes on instruction set and addressing modes

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

8051 assembly code example

```
        MOV     R7, #COUNT          ; initialize size of data block to transfer
        MOV     DPTR,               ; load XDATA source address base into DPTR0
                #SOURCE_ADDR
        MOV     85h, #01h           ; load DPTC to access DPTR1 pointer
        MOV     DPTR, #DEST_ADDR    ; load XDATA destination address base into DPTR1
        MOV     85h, #40h           ; load DPTC to access DPTR0 pointer and auto
                                    toggle
        MOV     86h, #0Ah           ; load DPTM to auto-increment both pointers
LOOP:   MOVX(1)  A, @DPTR           ; load XDATA byte from source into ACC.
                                    ; after load completes, DPTR0 increments and
                                    DPTR
                                    ; switches DPTR1
        MOVX(1)  @DPTR, A           ; store XDATA byte from ACC to destination.
                                    ; after store completes, DPTR1 increments and
                                    DPTR
                                    ; switches to DPTR0
        DJNZ(1)  R7, LOOP           ; continue until done
        MOV     86h, #00            ; disable auto-increment
        MOV     85h, #00            ; disable auto-toggle, now back to single DPTR
                                    mode
```

Note: 1 The code loop where the data transfer takes place is only 3 lines of code.

If an interrupt flag is not cleared after servicing the interrupt, an unwanted interrupt will occur upon exiting the ISR.

After the interrupt is serviced, the last instruction executed by the ISR is RETI. The RETI informs the MCU that the ISR is no longer in progress and the MCU pops the top two bytes from the stack and loads them into the PC. Execution of the interrupted program continues where it left off.

Note: An ISR must end with a RETI instruction, not a RET. An RET will not inform the interrupt control system that the ISR is complete, leaving the MCU to think the ISR is still in progress, making future interrupts impossible.

Table 17. Interrupt summary

Interrupt source	Polling priority	Vector addr	Flag bit name (SFR.bit position) 1 = Intr pending 0 = No interrupt	Flag bit auto-cleared by hardware?	Enable bit name (SFR.bit position) 1 = Intr Enabled 0 = Intr Disabled	Priority bit name (SFR.bit position) 1= High Priority 0 = Low Priority
Reserved	0 (high)	0063h	–	–	–	–
External Interrupt INT0	1	0003h	IE0 (TCON.1)	Edge - Yes Level - No	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	2	000Bh	TF0 (TCON.5)	Yes	ET0 (IE.1)	PT0 (IP.1)
External Interrupt INT1	3	0013h	IE1 (TCON.3	Edge - Yes Level - No	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	4	001Bh	TF1 (TCON.7)	Yes	ET1 (IE.3)	PT1 (IP.3)
UART0	5	0023h	RI (SCON0.0) TI (SCON0.1)	No	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow or TX2 Pin	6	002Bh	TF2 (T2CON.7) EXF2 (T2CON.6)	No	ET2 (IE.5)	PT2 (IP.5)
SPI	7	0053h	TEISF, RORISF, TISF, RISF (SPISTAT[3:0])	Yes	ESPI (IEA.6)	PSPI (IPA.6)
USB	8	0033h	– (1)	No	EUSB (IEA.0)	PUSB (IPA.0)
I^2C	9	0043h	INTR (S1STA.5)	Yes	EI^2C (IEA.1)	PI^2C (IPA.1)
ADC	10	003Bh	AINTF (ACON.7)	No	EADC (IEA.7)	PADC (IPA.7)
PCA	11	005Bh	OFVx, INTFx (PCASTA[0:7])	No	EPCA (IEA.5)	PPCA (IPA.5)
UART1	12 (low)	004Bh	RI (SCON1.0) TI (SCON1.1)	No	ES1 (IEA.4)	PS1 (IPA.4)

1. See USB interrupt flag registers UIF0-3.

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)

21.1.3 Mode 2

Mode 2 provides asynchronous, full-duplex communication using a total of 11 bits per data byte. Data is transmitted through TxD and received through RxD with: a Start Bit (logic '0'); eight data bits (LSB first); a programmable 9th data bit; and a Stop Bit (logic '1'). Upon Transmit, the 9th data bit (from bit TB8 in SCON) can be assigned the value of '0' or '1.' Or, for example, the Parity Bit (P, in the PSW) could be moved into TB8. Upon receive, the 9th data bit goes into RB8 in SCON, while the Stop Bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of f_{OSC}.

21.1.4 Mode 3

Mode 3 is the same as Mode 2 in all respects except the baud rate is variable like it is in Mode 1.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming Start Bit if REN = 1.

Table 64. UART operating modes

Mode	Synchronization	Bits of SFR, SCON		Baud clock	Data bits	Start/stop bits	See Figure
		SM0	SM1				
0	Synchronous	0	0	$f_{OSC}/12$	8	None	*Figure 30 on page 113*
1	Asynchronous	0	1	Timer 1 or Timer 2 Overflow	8	1 Start, 1 Stop	*Figure 32 on page 115*
2	Asynchronous	1	0	$f_{OSC}/32$ or $f_{OSC}/64$	9	1 Start, 1 Stop	*Figure 34 on page 117*
3	Asynchronous	1	1	Timer 1 or Timer 2 Overflow	9	1 Start, 1 Stop	*Figure 36 on page 118*

21.1.5 Multiprocessor communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into bit RB8, then comes a stop bit. The port can be programmed such that when the stop bit is received, the UART interrupt will be activated only if bit RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multi-processor systems is as follows: When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed leave their SM2 bits set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1, SM2 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

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 73. Baud rate of UART#1 for IrDA interface

BR3	BR2	BR1	BR0	Baud rate (kbps)
0	0	0	0	115.2
0	0	0	1	57.5
0	0	1	0	38.4
0	0	1	1	19.2
0	1	0	0	14.4
0	1	0	1	12.8
0	1	1	0	9.6
0	1	1	1	7.2
1	0	0	0	4.8
1	0	0	1	3.6
1	0	1	0	2.4
1	0	1	1	1.8
1	1	0	0	1.2

22.2 Pulse width selection

The IrDA interface has two ways to modulate the standard UART1 serial stream:

1. An IrDA data pulse will have a constant pulse width for any bit time, regardless of the selected baud rate.
2. An IrDA data pulse will have a pulse width that is proportional to the the bit time of the selected baud rate. In this case, an IrDA data pulse width is 3/16 of its bit time, as shown in *Figure 39 on page 119*.

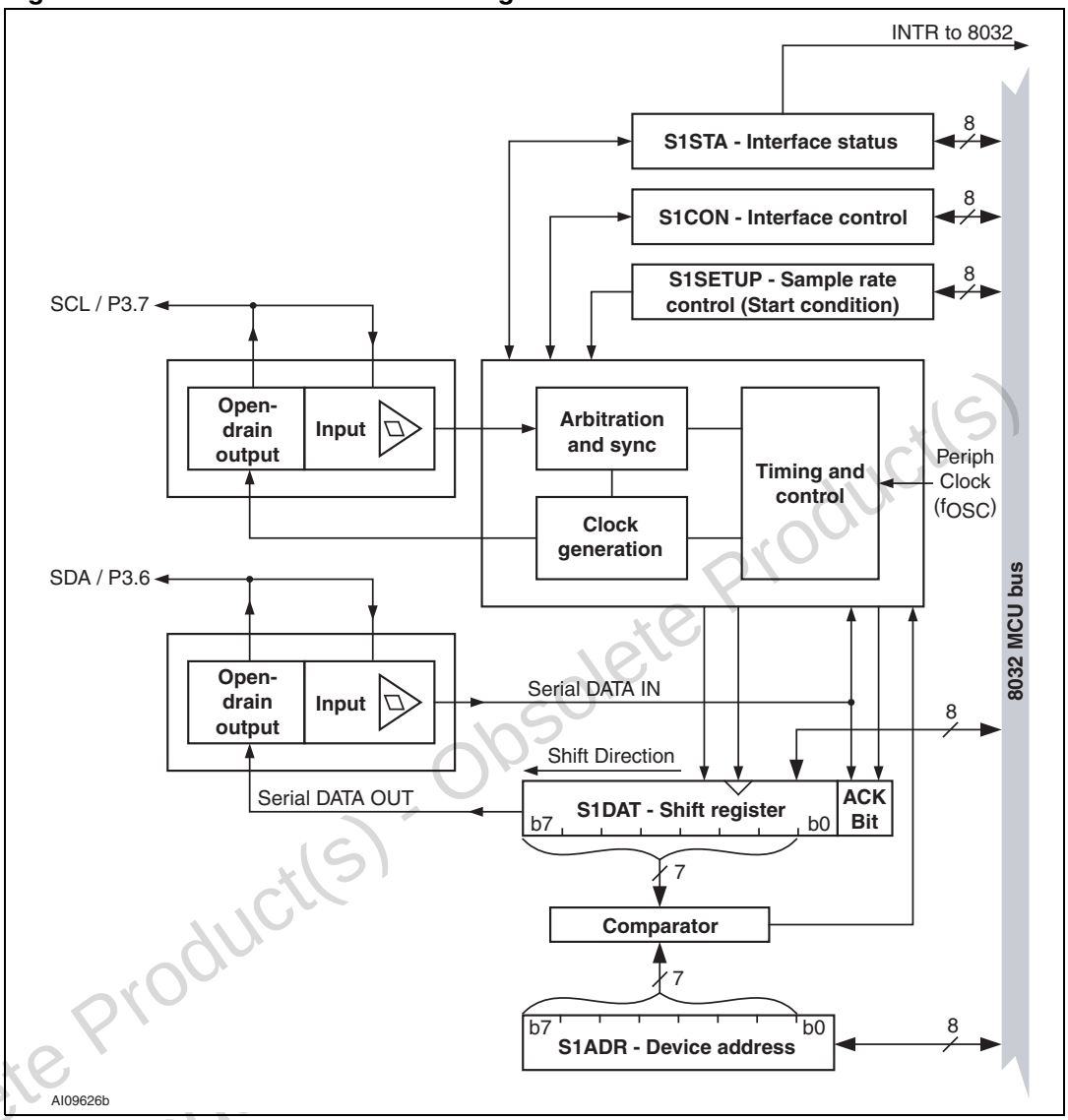
The PULSE bit in the SFR named IRDACON determines which method above will be used.

According to the IrDA physical layer specification, for all baud rates at 115.2k bps and below, the minimum data pulse width is 1.41µs. For a baud rate of 115.2k bps, the maximum pulse width 2.23µs. If a constant pulse width is to be used for all baud rates (PULSE bit = 0), the ideal general pulse width is 1.63µs, derived from the bit time of the fastest baud rate (8.68µs bit time for 115.2k bps rate), multiplied by the proportion, 3/16.

To produce this fixed data pulse width when the PULSE bit = 0, a prescaler is needed to generate an internal reference clock, SIRClk, shown in *Figure 38 on page 119*. SIRClk is derived by dividing the oscillator clock frequency, $f_{OSC,}$ using the five bits CDIV[4:0] in the SFR named IRDACON. A divisor must be chosen to produce a frequency for SIRClk that lies between 1.34 MHz and 2.13 MHz, but it is best to choose a divisor value that produces SIRClk frequency as close to 1.83MHz as possible, because SIRClk at 1.83MHz will produce an fixed IrDA data pulse width of 1.63µs. *Table 74* provides recommended values for CDIV[4:0] based on several different values of f_{OSC}.

For reference, SIRClk of 2.13MHz will generate a fixed IrDA data pulse width of 1.41 µs, and SIRClk of 1.34MHz will generate a fixed data pulse width of 2.23 µs.

Figure 42. I²C interface SIOE block diagram



Else If mode is Master-Receiver:

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

If Yes, Arbitration was lost:

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

If No, Aribitration was not lost, continue:

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

If its from sending Slave address, goto A:

If its from receiving Slave data, goto B:

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

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

If No, an ACK was not received:

- S1CON.STO = 1, set Stop condition

<Stop occurs after ISR exit>

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

If Yes, ACK was received, then continue:

- dummy = S1DAT, read to release bus

Does Master want to receive just one data byte?

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

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

If No, Master can ACK next byte from Slv

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

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

- recv_buf[buffer_index] = S1DAT, read byte

Is this the last data byte to receive from Slave?

If Yes, tell Slave to stop transmitting:

- S1CON.STO = 1, set Stop bus condition

<Stop occurs after ISR exit>

- Exit ISR, finished receiving data from Slave

If No, continue:

- USB FIFO base address high and low registers (UBASEH and UBASEL)

  All 10 Endpoint FIFOs share the same 64-byte address range. The 16-bit base address for the FIFOs is specified using the USB base address registers (see *Table 130* and *Table 132*). The USB endpoint select register (see *Table 124 on page 172*) selects the direction and the Endpoint for the FIFO that is accessed when addressing the 64-bytes of XDATA space starting with the base address specified in the base address registers. The base address is a 64-byte segment where the lower 6 bits of the base register are hardwired to '0.'

  Important note: The USB FIFO base address must be set to an open 64-byte segment in the XDATA space. Care should be taken to ensure that there is no overlap of addresses between the USB FIFOs and the flash memory, SRAM, csiop registers, and anything else accessed in the XDATA space. While the logic in the PSD module handles overlap of flash memory, SRAM, and the csiop registers with a fixed priority (see Section 28.1: PSD module functional description on page 192), this is not the case with the USB FIFOs. Unpredictable results as well as potential damage to the device may occur if there is an overlap of addresses.

Table 130. USB FIFO base address high register (UBASEH 0F3h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BASEADDR[15:8]							

Table 131. UBASEH register bit definition

Bit	Symbol	R/W	Definition
7:0	BASEADDR [15:8]	R/W	The upper 8 bits of the 16-bit base address for USB FIFOs to be mapped in XDATA space

Table 132. USB FIFO base address low register (UBASEL 0F4h, reset value 00h)

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

Table 133. UBASEL register bit definition

Bit	Symbol	R/W	Definition
7:6	BASEADDR [7:6]	R/W	Bits 7 and 6 of the 16-bit base address for the USB FIFOs to be mapped in XDATA space
5:0	BASEADDR [5:0]	R	Hardwired '0'

28.1 PSD module functional description

Major functional blocks are shown in *Figure 61 on page 191*. The next sections describe each major block.

28.1.1 8032 address/data/control interface

These signals attach directly to the MCU module to implement a 16-bit multiplexed 8051-style bus between the two stacked die. The MCU instruction prefetch and branch cache logic resides on the MCU module, leaving a modified 8051-style memory interface on the PSD module.

The active-low reset signal originating from the MCU module goes to the PSD module reset input ($\overline{RST}$). This reset signal can then be routed as an external output from the UPSD34xx to the system PC board, if needed, through any one of the PLD output pins as active-high or active-low logic by specifying logic equations in PSDsoft Express.

The 8032 address and data busses are routed throughout the PSD module as shown in *Figure 61* connecting many elements on the PSD module to the 8032 MCU. The 8032 bus is not only connected to the memories, but also to the General PLD, making it possible for the 8032 to directly read and write individual logic macrocells inside the General PLD.

28.1.2 Dual Flash memories and IAP

UPSD34xx devices contain two independent Flash memory arrays. This means that the 8032 can read instructions from one Flash memory array while erasing or writing the other Flash memory array. Concurrent operation like this enables robust remote updates of firmware, also known as In-Application Programming (IAP). IAP can occur using any UPSD34xx interface (e.g., UART, I2C, SPI). Concurrent memory operation also enables the designer to emulate EEPROM memory within either of the two Flash memory arrays for small data sets that have frequent updates.

The 8032 can erase Flash memories by individual sectors or it can erase an entire Flash memory array at one time. Each sector in either Flash memory may be individually write protected, blocking any WRITEs from the 8032 (good for boot and start-up code protection). The Flash memories automatically go to standby between 8032 READ or WRITE accesses to conserve power. Minimum erase cycles is 100K and minimum data retention is 15 years. Flash memory, as well as the entire PSD module may be programmed with the JTAG In-System Programming (ISP) interface with no 8032 involvement, good for manufacturing and lab development.

28.1.3 Main Flash memory

The Main Flash memory is divided into equal sized sectors that are individually selectable by the Decode PLD output signals, named FSx, one signal for each Main Flash memory sector. Each Flash sector can be located at any address within 8032 program address space (accessed with $\overline{PSEN}$) or data address space, also known as 8032 XDATA space (accessed with $\overline{RD}$ or $\overline{WR}$), as defined with the software development tool, PSDsoft Express. The user only has to specify an address range for each segment and specify if Main Flash memory will reside in 8032 data or program address space, and then $\overline{PSEN}$, $\overline{RD}$, or $\overline{WR}$ are automatically activated for the specified range. 8032 firmware is easily programmed into Main Flash memory using PSDsoft Express or other software tools. See *Table 157 on page 193* for Main Flash sector sizes on the various UPSD34xx devices.

28.1.16 Power management

The PSD module has bits in csiop registers that are configured at run-time by the 8032 to reduce power consumption of the GPLD. The Turbo Bit in the PMMR0 register can be set to logic '1' and both PLDs will go to Non-Turbo mode, meaning it will latch its outputs and go to sleep until the next transition on its inputs. There is a slight penalty in PLD performance (longer propagation delay), but significant power savings are realized. Going to Non-Turbo mode may require an additional wait state in the 8032 SFR, BUSCON, because memory decode signals are also delayed. The default state of the Turbo Bit is logic '0,' meaning by default, the GPLD is in fast Turbo mode until the user turns off Turbo mode.

Additionally, bits in csiop registers PMMR0 and PMMR2 can be set by the 8032 to selectively block signals from entering both PLDs which further reduces power consumption. There is also an Automatic Power Down counter that detects lack of 8032 activity and reduces power consumption on the PSD module to its lowest level (see *Section 28.1.16: Power management on page 197*).

28.1.17 Security and NVM sector protection

A programmable security bit in the PSD module protects its contents from unauthorized viewing and copying. The security bit is specified in PSDsoft Express and programmed into the UPSD34xx with JTAG. Once set, the security bit will block access of JTAG programming equipment to the PSD module Flash memory and PLD configuration, and also blocks JTAG debugging access to the MCU module. The only way to defeat the security bit is to erase the entire PSD module using JTAG (the erase command is the only JTAG command allowed after the security bit has been set), after which the device is blank and may be used again.

Additionally and independently, the contents of each individual Flash memory sector can be write protected (sector protection) by configuration with PSDsoft Express. This is typically used to protect 8032 boot code from being corrupted by inadvertent WRITEs to Flash memory from the 8032.

Status of sector protection bits may be read (but not written) using two registers in csiop space.

28.2 Memory mapping

There many different ways to place (or map) the address range of PSD module memory and I/O depending on system requirements. The DPLD provides complete mapping flexibility. *Figure 63* shows one possible system memory map. In this example, 128 Kbytes of Main Flash memory for a UPSD3433 device is in 8032 program address space, and 32 Kbytes of Secondary Flash memory, the SRAM, and csiop registers are all in 8032 XDATA space.

In *Figure 63*, the nomenclature fs0..fs7 are designators for the individual sectors of Main Flash memory, 16 Kbytes each. CSBOOT0..CSBOOT3 are designators for the individual Secondary Flash memory segments, 8 Kbytes each. *rs0* is the designator for SRAM, and csiop designates the PSD module control register set.

The designer may easily specify memory mapping in a point-and-click software environment using PSDsoft Express, creating a non-volatile configuration when the DPLD is programmed using JTAG.

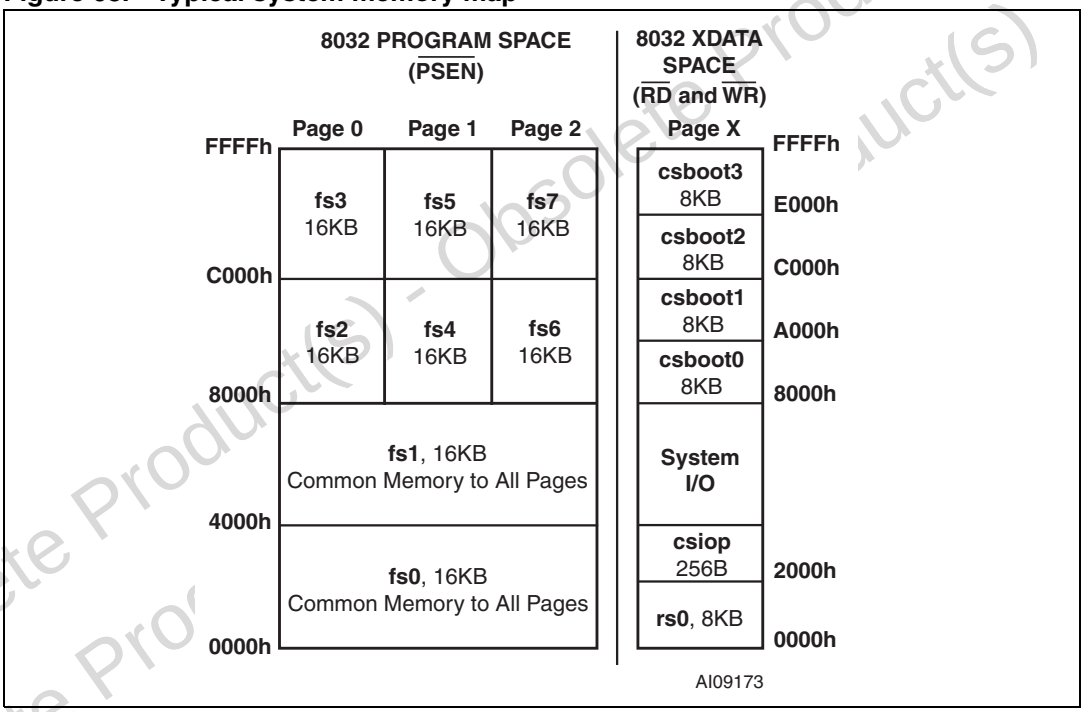
28.2.1 8032 program address space

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.

Table 183. MCU I/O mode port B data out register (address = csiop + offset 05h)(1)(2)

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

1. For each bit, 1 = drive port pin to logic '1,' 0 = drive port pin to logic '0'.
2. Default state of register is 00h after reset or power-up.

Table 184. MCU I/O mode port C data out register (address = csiop + offset 12h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	N/A	N/A	PC4	PC3	PC2	N/A	N/A

1. For each bit, 1 = drive port pin to logic '1,' 0 = drive port pin to logic '0'.
2. Default state of register is 00h after reset or power-up.

Table 185. MCU I/O mode port D data out register (address = csiop + offset 13h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	N/A	N/A	N/A	N/A	PD2(3)	PD1	N/A

1. For each bit, 1 = drive port pin to logic '1,' 0 = drive port pin to logic '0'.
2. Default state of register is 00h after reset or power-up.
3. Not available on 52-pin UPSD34xx devices.

Table 186. MCU I/O mode port A direction register(1) (address = csiop + offset 06h)(2)(3)

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

1. Port A not available on 52-pin UPSD34xx devices.
2. For each bit, 1 = out from UPSD34xx port pin1, 0 = in to PSD34xx port pin.
3. Default state for register is 00h after reset or power-up.

Table 187. MCU I/O mode port B direction in register (address = csiop + offset 07h)(1)(2)

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

1. For each bit, 1 = out from UPSD34xx port pin1, 0 = in to PSD34xx port pin.
2. Default state for register is 00h after reset or power-up.

Table 188. MCU I/O mode port C direction register (address = csiop + offset 14h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	N/A	N/A	PC4	PC3	PC2	N/A	N/A

1. For each bit, 1 = out from UPSD34xx port pin1, 0 = in to PSD34xx port pin.
2. 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.

Figure 98. External READ cycle (80-pin device only)

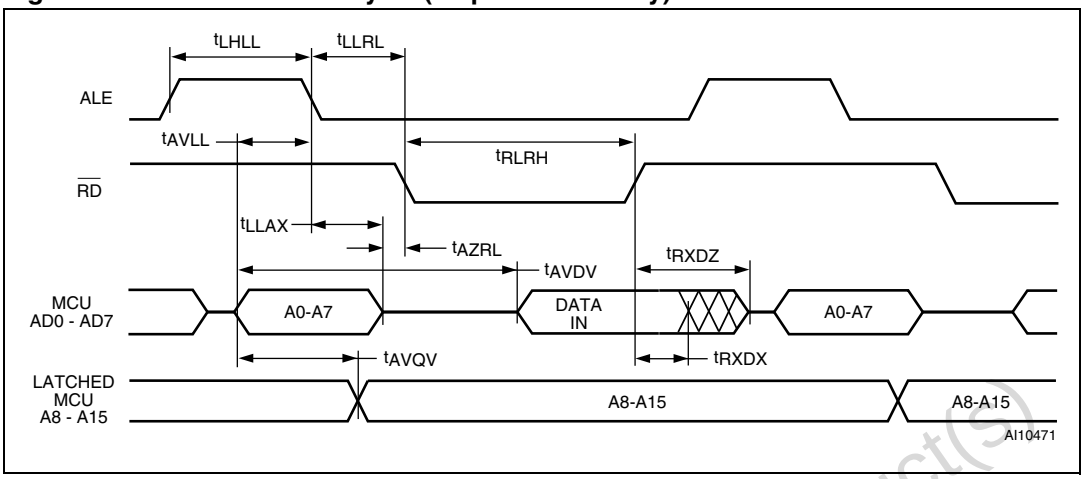


Table 214. External READ cycle AC characteristics (3 V or 5 V device)

Symbol	Parameter	40 MHz oscillator(1)		Variable oscillator $1/t_{CLCL}$ = 3 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{LHLL}	ALE pulse width	17		$t_{CLCL} - 8$		ns
t_{AVLL}	Address setup to ALE	13		$t_{CLCL} - 12$		ns
t_{LLAX}	Address hold after ALE	7.5		$0.5t_{CLCL} - 5$		ns
t_{LLRL}	ALE to $\overline{RD}$	7.5		$0.5t_{CLCL} - 5$		ns
t_{RLRH}	$\overline{RD}$ pulse width(2)	40		$nt_{CLCL} - 10$		ns
t_{RXIX}	Input data hold after $\overline{RD}$	2		2		ns
t_{RHIZ}	Input data float after $\overline{RD}$		10.5		$0.5t_{CLCL} - 2$	ns
t_{AVDX}	Address to valid data in(2)		70		$mt_{CLCL} - 5$	ns
t_{AZRL}	Address float to $\overline{RD}$	–2		–2		ns
t_{AVQV}	Address valid to latched address out on Ports A and B		35.5 (3 V)		$1.5t_{CLCL} - 2$	ns
			28 (5 V)		$t_{CLCL} - 9.5$	ns

1. BUSCON register is configured for 4 PFQCLK.
2. Refer to *Table 215* for “n” and “m” values.

Table 215. n, m, and x, y values

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

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