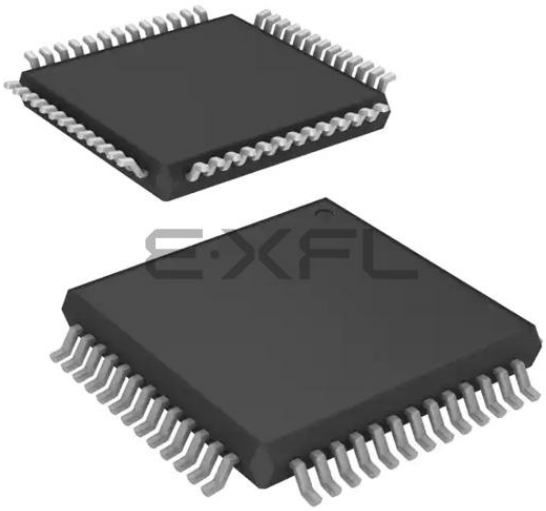


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

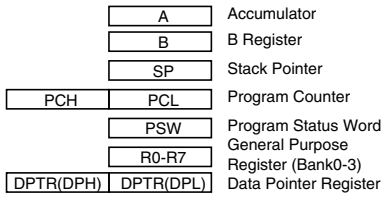
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, UART/USART
Peripherals	LVR, POR, PWM, WDT
Number of I/O	37
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 4x8b
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/upsd3233b-40t6t

Contents

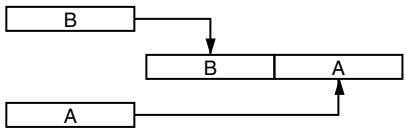
Figure 5. 8032 MCU registers



2.2.1 Accumulator

The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional tests. The Accumulator can be used as a 16-bit register with B Register as shown below.

Figure 6. Configuration of BA 16-bit registers



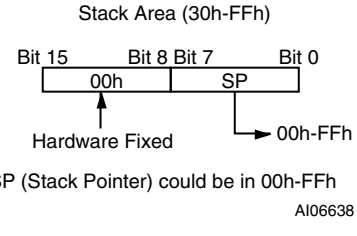
2.2.2 B register

The B Register is the 8-bit general purpose register, used for an arithmetic operation such as multiply, division with Accumulator.

2.2.3 Stack pointer

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07h after reset. This causes the stack to begin at location 08h.

Figure 7. Stack pointer



2.2.4 Program counter

The Program Counter is a 16-bit wide which consists of two 8-bit registers, PCH and PCL. This counter indicates the address of the next instruction to be executed. In $\overline{\text{RESET}}$ state, the program counter has reset routine address (PCH:00h, PCL:00h).

2.2.5 Program status word

The Program Status Word (PSW) contains several bits that reflect the current state of the CPU and select Internal RAM (00h to 1Fh: Bank0 to Bank3). The PSW is described in *Figure 8*. It contains the Carry flag, the Auxiliary Carry flag, the Half Carry (for BCD operation), the General Purpose flag, the Register Bank Select flags, the Overflow flag, and Parity flag.

[Carry flag, CY]. This flag stores any carry or not borrow from the ALU of CPU after an arithmetic operation and is also changed by the Shift Instruction or Rotate Instruction.

[Auxiliary Carry flag, AC]. After operation, this flag is set when there is a carry from Bit 3 of ALU or there is no borrow from Bit 4 of ALU.

[Register Bank Select flags, RS0, RS1]. These flags select one of four banks (00~07H:bank0, 08~0Fh:bank1, 10~17h:bank2, 17~1Fh:bank3) in Internal RAM.

[Overflow flag, OV]. This flag is set to '1' when an overflow occurs as the result of an arithmetic operation involving signs. An overflow occurs when the result of an addition or subtraction exceeds +127 (7Fh) or -128 (80h). The CLRV instruction clears the overflow flag. There is no set instruction. When the BIT instruction is executed, Bit 6 of memory is copied to this flag.

[Parity flag, P]. This flag reflects the number of Accumulator's 1. If the number of Accumulator's 1 is odd, P=0; otherwise, P=1. The sum when adding Accumulator's 1 to P is always even.

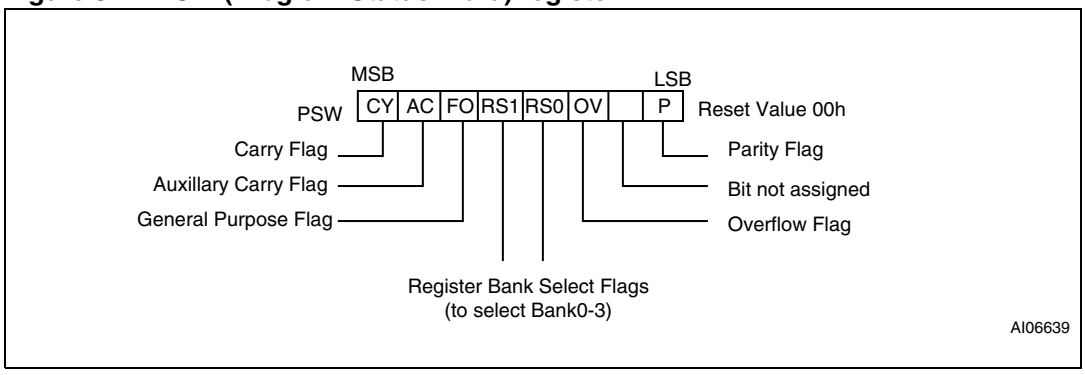
2.2.6 Registers R0~R7

General purpose 8-bit registers that are locked in the lower portion of internal data area.

2.2.7 Data pointer register

Data Pointer Register is 16-bit wide which consists of two-8bit registers, DPH and DPL. This register is used as a data pointer for the data transmission with external data memory in the PSD module.

Figure 8. PSW (Program Status Word) register



2.12.2 External RAM

Table 10 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DTPR.

Note: In all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

2.12.3 Lookup tables

Table 11 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated.

The mnemonic is MOVC for “move constant.” The first MOVC instruction in *Table 11* can accommodate a table of up to 256 entries numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to the beginning of the table. Then:

```
MOVC A, @A+DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired en-try is loaded into the Accumulator, and the subroutine is called:

```
MOV A , ENTRY NUMBER
CALL TABLE
```

The subroutine “TABLE” would look like this:

```
TABLE: MOVC A , @A+PC
RET
```

The table itself immediately follows the RET (return) instruction is Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 cannot be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Table 10. Data transfer instruction that access external data memory space

Address Width	Mnemonic	Operation
8 bits	MOVX A,@Ri	READ external RAM @Ri
8 bits	MOVX @Ri,A	WRITE external RAM @Ri
16 bits	MOVX A,@DPTR	READ external RAM @DPTR
16 bits	MOVX @DPTR,a	WRITE external RAM @DPTR

Table 11. Lookup table READ instruction

Mnemonic	Operation
MOVC A,@A+DPTR	READ program memory at (A+DPTR)
MOVC A,@A+PC	READ program memory at (A+PC)

Table 15. SFR memory map (continued)

88	TCON(1)	TMOD	TL0	TL1	TH0	TH1			8F
80	P0(1)	SP	DPL	DPH				PCON	87

1. Register can be bit addressing

Table 16. List of all SFRs

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
80	P0									FF	Port 0
81	SP									07	Stack Ptr
82	DPL									00	Data Ptr Low
83	DPH									00	Data Ptr High
87	PCON	SMOD	SMOD1	LVREN	ADSFINT	RCLK1	TCLK1	PD	IDLE	00	Power Ctrl
88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00	Timer / Cntr Control
89	TMOD	Gate	C/T	M1	M0	Gate	C/T	M1	M0	00	Timer / Cntr mode Control
8A	TL0									00	Timer 0 Low
8B	TL1									00	Timer 1 Low
8C	TH0									00	Timer 0 High
8D	TH1									00	Timer 1 High
90	P1									FF	Port 1
91	P1SFS	P1S7	P1S6	P1S5	P1S4					00	Port 1 Select Register
93	P3SFS	P3S7	P3S6							00	Port 3 Select Register
94	P4SFS	P4S7	P4S6	P4S5	P4S4	P4S3	P4S2	P4S1	P4S0	00	Port 4 Select Register
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register
98	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	Serial Control Register
99	SBUF									00	Serial Buffer
9A	SCON2	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	2nd UART Ctrl Register
9B	SBUF2									00	2nd UART Serial Buffer
A0	P2									FF	Port 2

Table 16. List of all SFRs (continued)

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A1	PWMCON	PWML	PWMP	PWME	CFG4	CFG3	CFG2	CFG1	CFG0	00	PWM Control Polarity
A2	PWM0									00	PWM0 Output Duty Cycle
A3	PWM1									00	PWM1 Output Duty Cycle
A4	PWM2									00	PWM2 Output Duty Cycle
A5	PWM3									00	PWM3 Output Duty Cycle
A6	WDRST									00	Watch Dog Reset
A7	IEA	EDDC			ES2			EI^{2}C	EUSB	00	Interrupt Enable (2nd)
A8	IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
A9											
AA	PWM4P									00	PWM 4 Period
AB	PWM4W									00	PWM 4 Pulse Width
AE	WDKEY									00	Watch Dog Key Register
B0	P3									FF	Port 3
B1	PSCL0L									00	Prescaler 0 Low (8-bit)
B2	PSCL0H									00	Prescaler 0 High (8-bit)
B3	PSCL1L									00	Prescaler 1 Low (8-bit)
B4	PSCL1H									00	Prescaler 1 High (8-bit)
B7	IPA	PDDC			PS2			PI2C	PUSB	00	Interrupt Priority (2nd)
B8	IP			PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority
C0	P4									FF	New Port 4
C8	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00	Timer 2 Control

5.5 External Int1 interrupt

- The INT1 can be either level active or transition active depending on Bit IT1 in register TCON. The flag that actually generates this interrupt is Bit IE1 in TCON.
- When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition activated.
- If the interrupt was level activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.
- The ADC can take over the External INT1 to generate an interrupt on conversion being completed

5.6 DDC interrupt

- The DDC interrupt is generated either by Bit INTR in the S1STA register for DC2B protocol or by Bit DDC interrupt in the DDCCON register for DDC1 protocol or by Bit SWHINT Bit in the DDCCON register when DDC protocol is changed from DDC1 to DDC2.
- Flags except the INTR have to be cleared by the software. INTR flag is cleared by hardware.

5.7 USB interrupt

- The USB interrupt is generated when endpoint0 has transmitted a packet or received a packet, when Endpoint1 or Endpoint2 has transmitted a packet, when the suspend or resume state is detected and every EOP received.
- When the USB interrupt is generated, the corresponding request flag must be cleared by software. The interrupt service routine will have to check the various USB registers to determine the source and clear the corresponding flag.
- Please see the dedicated interrupt control registers for the USB peripheral for more information.

5.8 USART interrupt

- The USART Interrupt is generated by RI (receive interrupt) OR TI (transmit interrupt).
- When the USART Interrupt is generated, the corresponding request flag must be cleared by software. The interrupt service routine will have to check the various USART registers to determine the source and clear the corresponding flag.
- Both USART's are identical, except for the additional interrupt controls in the Bit 4 of the additional interrupt control registers (A7h, B7h)

6 Power-saving mode

Two software selectable modes of reduced power consumption are implemented.

6.1 Idle mode

In Idle mode, the following functions are switched Off.

- CPU (Halted)

The following functions remain Active during Idle mode:

- External Interrupts
- Timer 0, Timer 1, Timer 2
- DDC Interface
- PWM Units
- USB Interface
- USART
- 8-bit ADC
- I^2C Interface

Note: Interrupt or $\overline{RESET}$ terminates the Idle mode.

6.2 Power-down mode

- System Clock Halted
- LVD Logic Remains Active
- SRAM content remains unchanged
- The SFRs retain their value until a $\overline{RESET}$ is asserted

Note: The only way to exit Power-down mode is through a $\overline{RESET}$.

Table 25. Power-saving mode power consumption

Mode	Addr/data	Ports 1,3,4	PWM	I^2C	DDC	USB
Idle	Maintain Data	Maintain Data	Active	Active	Active	Active
Power-down	Maintain Data	Maintain Data	Disable	Disable	Disable	Disable

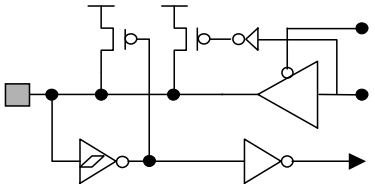
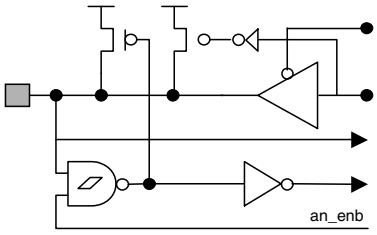
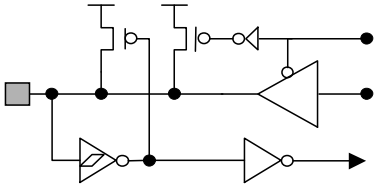
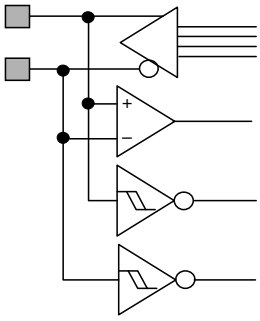
6.3 Power control register

The Idle and Power-down modes are activated by software via the PCON register.

Table 26. Pin status during Idle and Power-down mode

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
87	PCON	SMOD	SMOD1	LVREN	ADSFINT	RCLK1	TCLK1	PD	IDLE	00	Power Ctrl

Figure 17. Port type and description (Part 2)

Symbol	In/ Out	Circuit	Function
PORT1 <3:0>, PORT3, PORT4<7:3,1:0> PORT2	I/O		Bidirectional I/O port with internal pull-ups Schmitt input CMOS compatible interface
PORT1 < 7:4 >	I/O		Bidirectional I/O port with internal pull-ups Schmitt input CMOS compatible interface Analog input option
PORT4.2	I/O		Bidirectional I/O port with internal pull-ups Schmitt input. TTL compatible interface Pull-up when reset Address Latch Enable Program Strobe Enable
USB - , USB +	I/O		Bidirectional I/O port Schmitt input TTL compatible interface

AI06654

Table 44. Description of the SCON bits (continued)

Bit	Symbol	Function
1	TI	Transmit Interrupt Flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the Stop bit in the other modes, in any serial transmission. Must be cleared by software
0	RI	Receive Interrupt Flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the Stop bit in the other modes, in any serial reception (except for SM2). Must be cleared by software

12.2.1 Baud rates

The baud rate in Mode 0 is fixed:

Mode 0 Baud Rate = f_{OSC} / 12

The baud rate in Mode 2 depends on the value of Bit SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate = (2^{SMOD} / 64) x f_{OSC}

In the UPSD323xx devices, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

12.2.2 Using Timer 1 to generate baud rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Modes 1 and 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 TMOD = 0010B). In that case the baud rate is given by the formula:

Modes 1 and 3 Baud Rate = (2^{SMOD} / 32) x (f_{OSC} / (12 x [256 – (TH1)]))

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 Interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 Interrupt to do a 16-bit software reload. *Figure 21* lists various commonly used baud rates and how they can be obtained from Timer 1.

12.2.3 Using Timer/counter 2 to generate baud rates

In the UPSD323xx devices, Timer 2 selected as the baud rate generator by setting TCLK and/or RCLK (see *Figure 21* Timer/ Counter 2 Control Register (T2CON)).

Note: The baud rate for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer into its Baud Rate Generator mode.

The RCLK and TCLK Bits in the T2CON register configure UART 1. The RCLK1 and TCLK1 Bits in the PCON register configure UART 2.

The Baud Rate Generator mode is similar to the Auto-reload Mmode, in that a roll over in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the '1' that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after “WRITE to SBUF.”

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its roll-overs with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not '0,' the receive circuits are reset and the unit goes back to looking for an-other 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the reset of the rest of the frame will proceed.

As data bits come in from the right, '1s' shift out to the left. When the start bit arrives at the left-most position in the shift register (which in Mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. R1 = 0, and
2. Either SM2 = 0, or the received Stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the Stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

12.2.6 More about Modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a Start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a Stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of '0' or '1.' On receive, the data bit goes into RB8 in SCON. The baud rate is programmable to either 1/16 or 1/32 the CPU clock frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figure 30 and *Figure 32* show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “WRITE to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next roll-over in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the “WRITE to SBUF” signal.)

The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a '1' (the Stop bit) into the 9th bit position of the shift register. There-after, only zeros are clocked in. Thus,

as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the Stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by 16 rollover after “WRITE to SUBF.”

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of R-D. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not '0,' the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the Start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, '1s' shift out to the left. When the Start bit arrives at the left-most position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. RI = 0, and
2. Either SM2 = 0, or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

Figure 32. Serial port Mode 3 block diagram

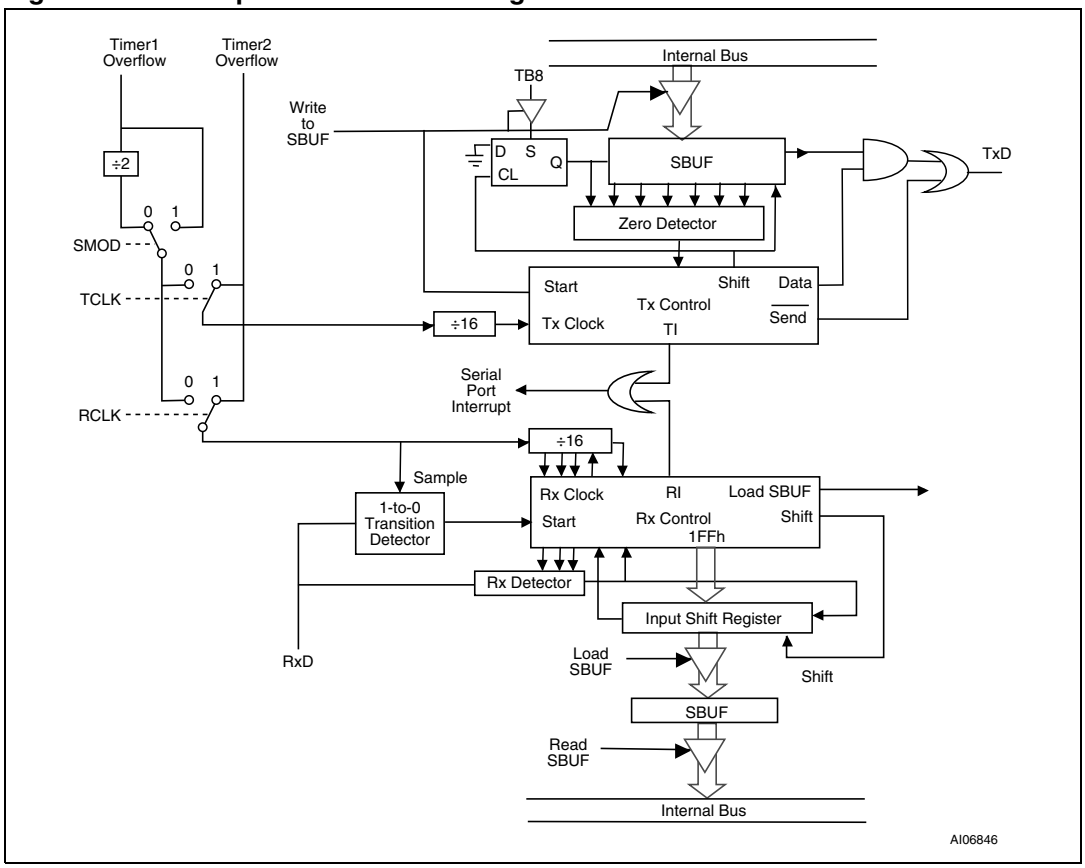


Figure 33. Serial port Mode 3 waveforms

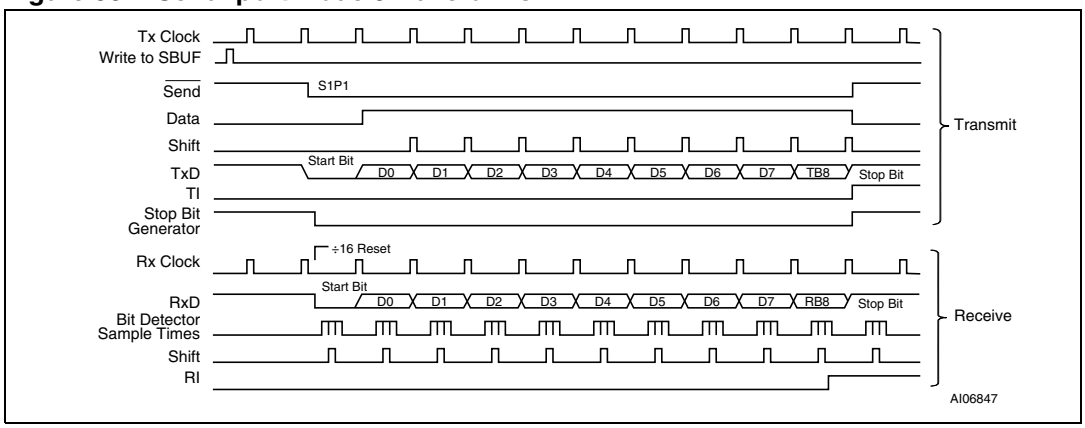


Table 52. Selection of the serial clock frequency SCL in Master mode (continued)

CR2	CR1	CR0	f_{OSC} divisor	Bit rate (kHz) at f_{OSC}			
				12 MHz	24 MHz	36 MHz	40 MHz
1	1	0	480	12.5	25	37.5	41
1	1	1	960	6.25	12.5	18.75	20

15.1 Serial status register (SxSTA: S1STA, S2STA)

SxSTA is a “Read-only” register. The contents of this register may be used as a vector to a service routine. This optimized the response time of the software and consequently that of the I²C-bus. The status codes for all possible modes of the I²C-bus interface are given Table *Table 54*.

This flag is set, and an interrupt is generated, after any of the following events occur.

1. Own slave address has been received during AA = 1: ack_int
2. The general call address has been received while GC(SxADR.0) = 1 and AA = 1:
3. A data byte has been received or transmitted in Master mode (even if arbitration is lost): ack_int
4. A data byte has been received or transmitted as selected slave: ack_int
5. A stop condition is received as selected slave receiver or transmitter: stop_int

15.2 Data shift register (SxDAT: S1DAT, S2DAT)

SxDAT contains the serial data to be transmitted or data which has just been received. The MSB (Bit 7) is transmitted or received first; that is, data shifted from right to left.

Table 53. Serial status register (SxSTA)

7	6	5	4	3	2	1	0
GC	STOP	INTR	TX_MODE	BBUSY	BLOST	/ACK_REP	SLV

18 PSD module

The PSD module provides configurable Program and Data memories to the 8032 CPU core (MCU). In addition, it has its own set of I/O ports and a PLD with 16 macrocells for general logic implementation.

Ports A,B,C, and D are general purpose programmable I/O ports that have a port architecture which is different from the I/O ports in the MCU module.

The PSD module communicates with the MCU module through the internal address, data bus (A0-A15, D0-D7) and control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE, $\overline{RESET}$). The user defines the Decoding PLD in the PSDsoft Development Tool and can map the resources in the PSD module to any program or data address space. *Figure 49* shows the functional blocks in the PSD module.

18.1 Functional overview

- 1 or 2 Mbit Flash memory. This is the main Flash memory. It is divided into eight equal-sized blocks that can be accessed with user-specified addresses.
- Secondary 256 Kbit Flash boot memory. It is divided into four equal-sized blocks thatat can be accessed with user-specified addresses. This secondary memory brings the ability to execute code and update the main Flash *concurrently.*
- 64 Kbit SRAM.
- CPLD with 16 Output Micro Cells (OMCs} and 20 Input Micro Cells (IMCs). The CPLD may be used to efficiently implement a variety of logic functions for internal and external control. Examples include state machines, loadable shift registers, and loadable counters.
- Decode PLD (DPLD) that decodes address for selection of memory blocks in the PSD module.
- Configurable I/O ports (Port A,B,C and D) that can be used for the following functions:
  - MCU I/Os
  - PLD I/Os
  - Latched MCU address output
  - Special function I/Os
  - I/O ports may be configured as open-drain outputs
- Built-in JTAG compliant serial port allows full-chip In-System Programmability (ISP). With it, you can program a blank device or reprogram a device in the factory or the field.
- Internal page register that can be used to expand the 8032 MCU module address space by a factor of 256.
- Internal programmable Power Management Unit (PMU) that supports a low-power mode called Power-down mode. The PMU can automatically detect a lack of the 8032 CPU core activity and put the PSD module into Power-down mode.
- Erase/WRITE cycles:

  Flash memory - 100,000 minimum

  PLD - 1,000 minimum

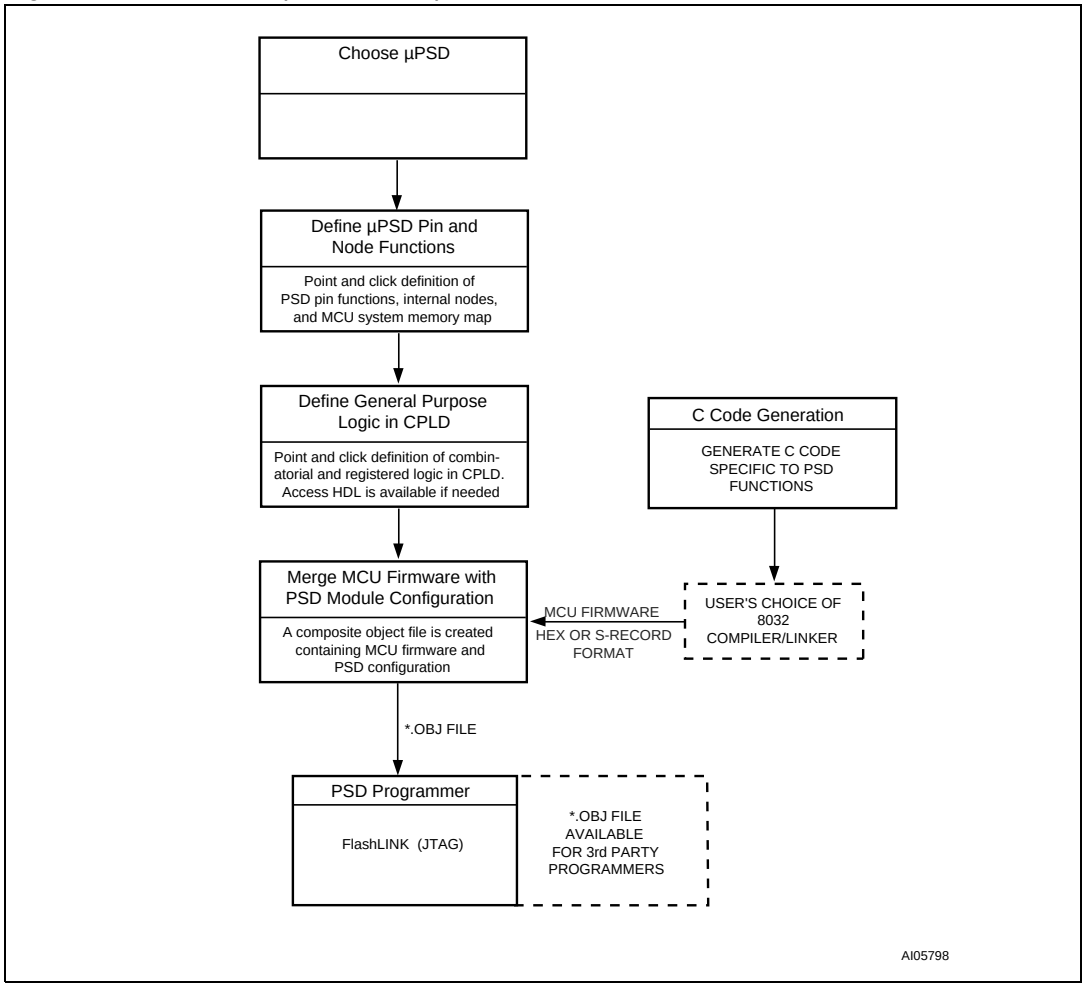
  Data Retention: 15 year minimum (for Main Flash memory, Boot, PLD and Configuration bits)

19 Development system

UPSD323xx devices are supported by PSDsoft, a Windows-based software development tool (Windows-95, Windows-98, Windows-NT). A PSD module design is quickly and easily produced in a point and click environment. The designer does not need to enter Hardware Description Language (HDL) equations, unless desired, to define PSD module pin functions and memory map information. The general design flow is shown in Figure 50. PSDsoft is available from our web site (the address is given on the back page of this data sheet) or other distribution channels.

PSDsoft directly supports a low cost device programmer from ST: FlashLINK (JTAG). The programmer may be purchased through your local distributor/representative. UPSD323xx devices are also supported by third party device programmers. See our web site for the current list.

Figure 50. PSDsoft express development tool



27 Programming in-circuit using the JTAG serial interface

The JTAG Serial Interface pins (TMS, TCK, TDI, and TDO) are dedicated pins on Port C (see *Table 107*). All memory blocks (primary and secondary Flash memory), PLD logic, and PSD module Configuration Register Bits may be programmed through the JTAG Serial Interface block. A blank device can be mounted on a printed circuit board and programmed using JTAG.

The standard JTAG signals (IEEE 1149.1) are TMS, TCK, TDI, and TDO. Two additional signals, $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$, are optional JTAG extensions used to speed up Program and Erase cycles.

By default, on a blank device (as shipped from the factory or after erasure), four pins on Port C are the basic JTAG signals TMS, TCK, TDI, and TDO.

27.1 Standard JTAG Signals

At power-up, the standard JTAG pins are inputs, waiting for a JTAG serial command from an external JTAG controller device (such as FlashLINK or Automated Test Equipment). When the enabling command is received, TDO becomes an output and the JTAG channel is fully functional. The same command that enables the JTAG channel may optionally enable the two additional JTAG signals, $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$.

The $\overline{\text{RESET}}$ input to the uPS3200 should be active during JTAG programming. The active $\overline{\text{RESET}}$ puts the MCU module into RESET mode while the PSD module is being programmed. See Application Note AN1153 for more details on JTAG In-System Programming (ISP).

UPSD323xx devices support JTAG In-System-Configuration (ISC) commands, but not Boundary Scan. The PSDsoft Express software tool and FlashLINK JTAG programming cable implement the JTAG In-System-Configuration (ISC) commands.

Table 107. JTAG port signals

Port C Pin	JTAG Signals	Description
PC0	TMS	Mode Select
PC1	TCK	Clock
PC3	TSTAT	Status (optional)
PC4	TERR	Error Flag (optional)
PC5	TDI	Serial Data In
PC6	TDO	Serial Data Out

27.2 JTAG extensions

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ are two JTAG extension signals enabled by an “ISC_ENABLE” command received over the four standard JTAG signals (TMS, TCK, TDI, and TDO). They are used to speed Program and Erase cycles by indicating status on uPDS signals instead of having to

Figure 81. Peripheral I/O Read timing

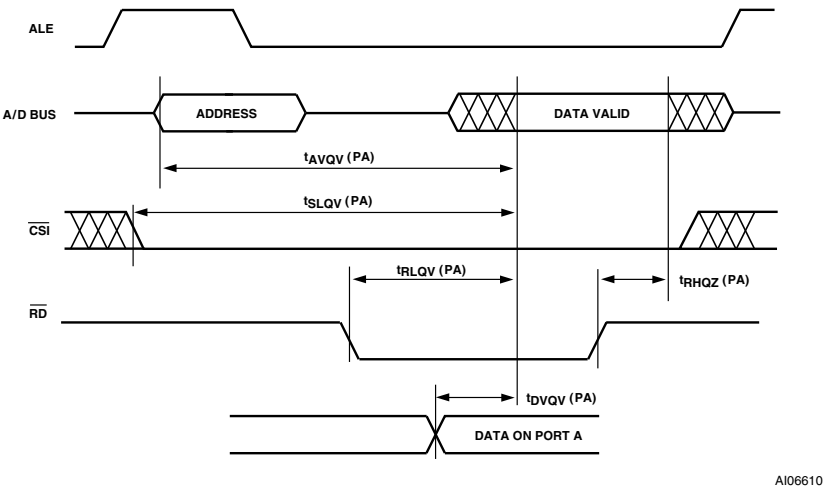


Table 137. Port A peripheral data mode Read timing (5 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Turbo off	Unit
$t_{AVQV-PA}$	Address valid to data valid	(Note 1)		37	+ 10	ns
$t_{SLQV-PA}$	$\overline{CSI}$ valid to data valid			27	+ 10	ns
$t_{RLQV-PA}$	$\overline{RD}$ to data valid	(Note 2)		32		ns
$t_{DVQV-PA}$	Data in to data out valid			22		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to data high-Z			23		ns

1. Any input used to select Port A Data Peripheral mode.
2. Data is already stable on Port A.

Table 138. Port A peripheral data mode Read timing (3 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Turbo off	Unit
$t_{AVQV-PA}$	Address valid to data valid	(Note 1)		50	+ 20	ns
$t_{SLQV-PA}$	$\overline{CSI}$ valid to data valid			37	+ 20	ns
$t_{RLQV-PA}$	$\overline{RD}$ to data valid	(Note 2)		45		ns
$t_{DVQV-PA}$	Data in to data out valid			38		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to data high-Z			36		ns

1. Any input used to select Port A Data Peripheral mode.
2. Data is already stable on Port A.

34 Part numbering

Table 148. Ordering information scheme

Example: UPSD 3 2 3 4 B V – 24 U 6 T

Device type

UPSD = Microcontroller PSD

Family

3 = 8032 core

PLD size

2 = 16 Macrocells

SRAM Size

3 = 8 Kbytes

Main Flash memory size

2 = 64 Kbytes

3 = 128 Kbytes

4 = 256 Kbytes

IP mix

A = USB, I^2C, PWM, DDC, ADC, (2) UARTs, Supervisor (Reset Out, Reset In, LVD, WD)

B = I^2C, PWM, DDC, ADC, (2) UARTs Supervisor (Reset Out, Reset In, LVD, WD)

Operating voltage

blank = V_{CC} = 4.5 to 5.5 V

V = V_{CC} = 3.0 to 3.6 V

Speed

–24 = 24 MHz

–40 = 40 MHz

Package

T = 52-pin LQFP

U = 80-pin LQFP

Temperature range

1 = 0 to 70°C

6 = –40 to 85°C

Shipping options

F = ECOPACK® Package, Tape & Reel Packing

For other options, or for more information on any aspect of this device, please contact the ST Sales Office nearest you.