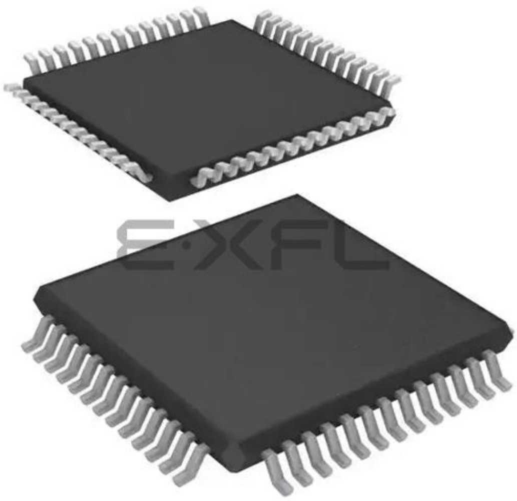
STMicroelectronics - UPSD3333DV-40T6 Datasheet



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
Peripherals	LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3333dv-40t6

4 Memory organization

The 8032 MCU core views memory on the MCU module as “internal” memory and it views memory on the PSD module as “external” memory, see *Figure 5*

Internal memory on the MCU module consists of DATA, IDATA, and SFRs. These standard 8032 memories reside in 384 bytes of SRAM located at a fixed address space starting at address 0x0000.

External memory on the PSD module consists of four types: main Flash (64, 128, or 256 Kbytes), a smaller secondary Flash (16 or 32 Kbytes), SRAM (2, 8, or 32 Kbytes), and a block of PSD module control registers called CSIOP (256 bytes). These external memories reside at programmable address ranges, specified using the software tool PSDsoft Express. See the *Section 27: PSD module on page 164* of this document for more details on these memories.

External memory is accessed by the 8032 in two separate 64 Kbyte address spaces. One address space is for program memory and the other address space is for data memory. Program memory is accessed using the 8032 signal, $\overline{\text{PSEN}}$. Data memory is accessed using the 8032 signals, $\overline{\text{RD}}$ and $\overline{\text{WR}}$. If the 8032 needs to access more than 64 Kbytes of external program or data memory, it must use paging (or banking) techniques provided by the Page register in the PSD module.

Note: When referencing program and data memory spaces, it has nothing to do with 8032 internal SRAM areas of DATA, IDATA, and SFR on the MCU module. Program and data memory spaces only relate to the external memories on the PSD module.

External memory on the PSD module can overlap the internal SRAM memory on the MCU module in the same physical address range (starting at 0x0000) without interference because the 8032 core does not assert the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signals when accessing internal SRAM.

Figure 5. UPSD33xx memories

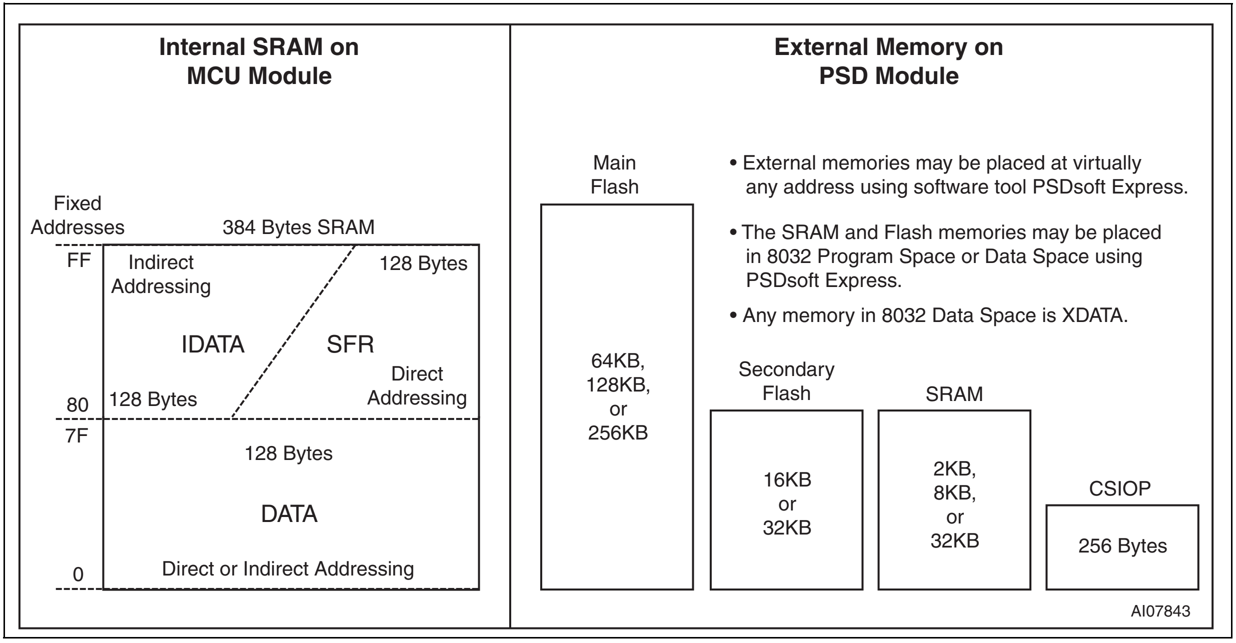
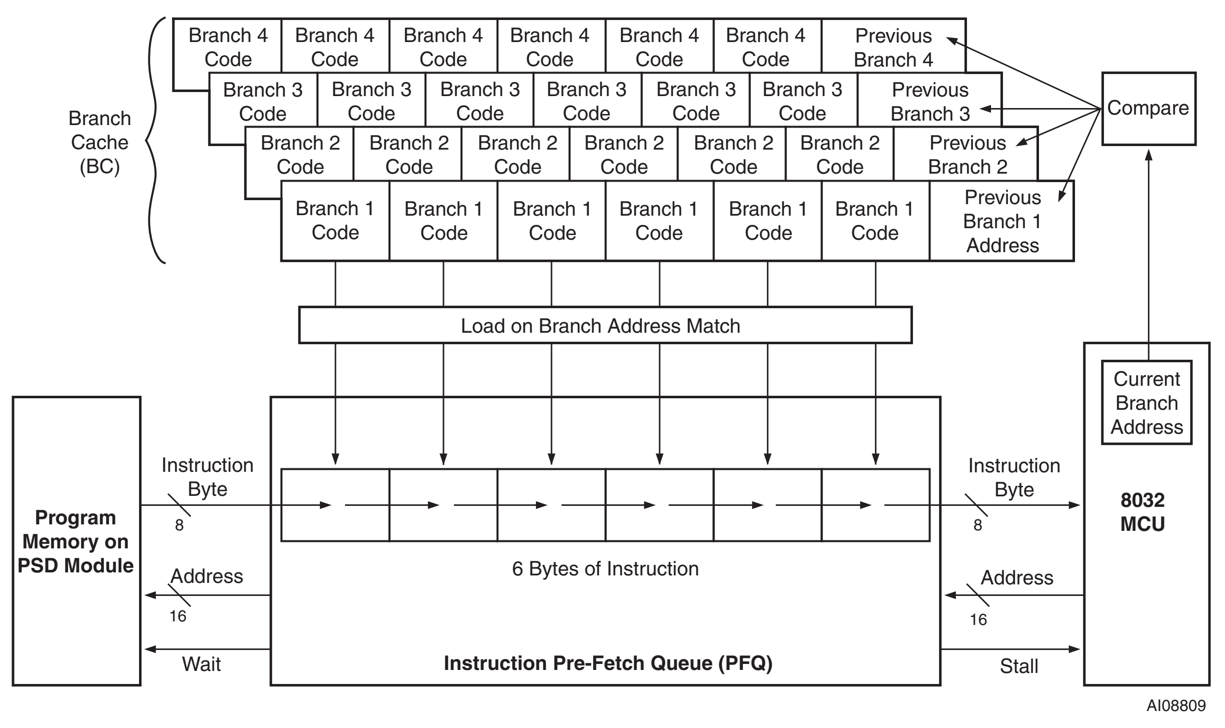

Internal SRAM on MCU Module
Fixed Addresses
384 Bytes SRAM
FF
Indirect Addressing
128 Bytes
IDATA
SFR
Direct Addressing
80
128 Bytes
7F
128 Bytes
DATA
Direct or Indirect Addressing
0
External Memory on PSD Module
Main Flash
64KB, 128KB, or 256KB
• External memories may be placed at virtually any address using software tool PSDsoft Express.
• The SRAM and Flash memories may be placed in 8032 Program Space or Data Space using PSDsoft Express.
• Any memory in 8032 Data Space is XDATA.
Secondary Flash
16KB or 32KB
SRAM
2KB, 8KB, or 32KB
CSIOP
256 Bytes
AI07843

Figure 7. Instruction pre-fetch queue and branch cache



5.1 Pre-Fetch Queue (PFQ) and Branch Cache (BC)

The PFQ is always working to minimize the idle bus time inherent to 8032 MCU architecture, to eliminate wasted memory fetches, and to maximize memory bandwidth to the MCU. The PFQ does this by running asynchronously in relation to the MCU, looking ahead to pre-fetch code from program memory during any idle bus periods. Only necessary bytes will be fetched (no dummy fetches like standard 8032). The PFQ will queue up to six code bytes in advance of execution, which significantly optimizes sequential program performance. However, when program execution becomes non-sequential (program branch), a typical pre-fetch queue will empty itself and reload new code, causing the MCU to stall. The Turbo UPSD33xx diminishes this problem by using a Branch Cache with the PFQ. The BC is a four-way, fully associative cache, meaning that when a program branch occurs, it's branch destination address is compared simultaneously with four recent previous branch destinations stored in the BC. Each of the four cache entries contain up to six bytes of code related to a branch. If there is a hit (a match), then all six code bytes of the matching program branch are transferred immediately and simultaneously from the BC to the PFQ, and execution on that branch continues with minimal delay. This greatly reduces the chance that the MCU will stall from an empty PFQ, and improves performance in embedded control systems where it is quite common to branch and loop in relatively small code localities.

By default, the PFQ and BC are enabled after power-up or reset. The 8032 can disable the PFQ and BC at runtime if desired by writing to a specific SFR (BUSCON).

The memory in the PSD module operates with variable wait states depending on the value specified in the SFR named BUSCON. For example, a 5 V UPSD33xx device operating at a 40 MHz crystal frequency requires four memory wait states (equal to four MCU clocks). In this example, once the PFQ has one or more bytes of code, the wait states become

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

SFR addr (hex)	SFR name	Bit name and <Bit Address>								Reset value (hex)	Reg. descr. with link
		7	6	5	4	3	2	1	0		
FE	RESERVED										
FF	RESERVED										

1. This SFR can be addressed by individual bits (Bit Address mode) or addressed by the entire byte (Direct Address mode).

Figure 18. MCU I/O cell block diagram for Port 4

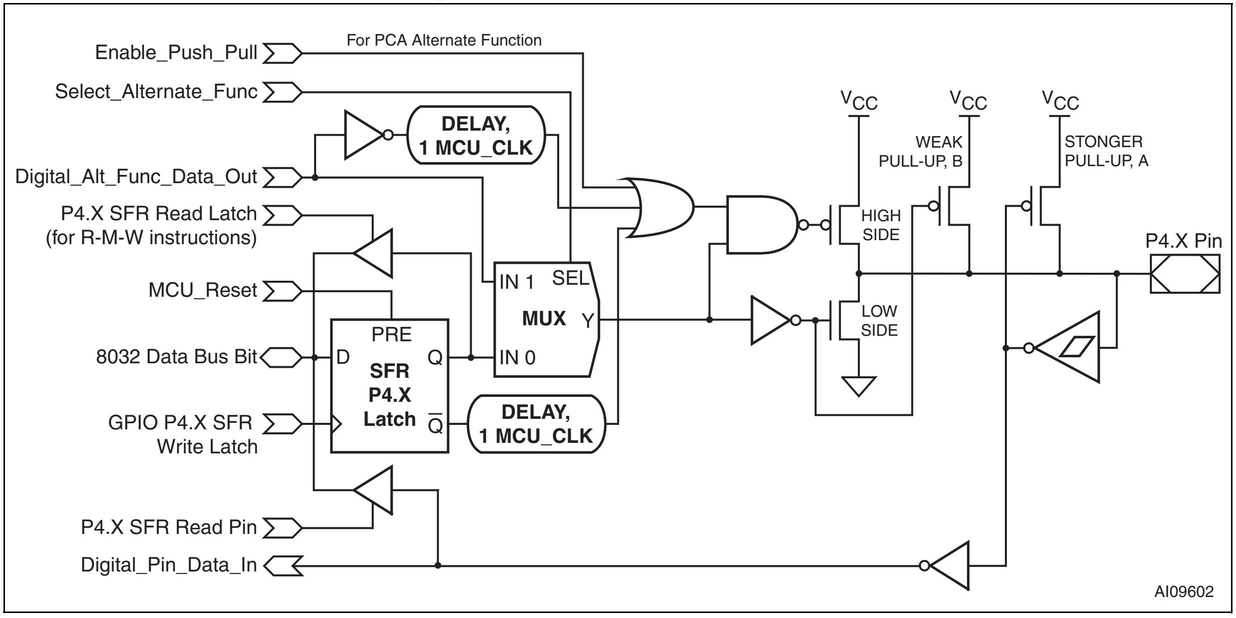


Table 33. P1: I/O Port 1 register (SFR 90h, reset value FFh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Table 34. P1 register bit definition

Bit	Symbol	R/W	Function[1]
7	P1.7	R,W	**Port pin 1.7**
6	P1.6	R,W	**Port pin 1.6**
5	P1.5	R,W	**Port pin 1.5**
4	P1.4	R,W	**Port pin 1.4**
3	P1.3	R,W	**Port pin 1.3**
2	P1.2	R,W	**Port pin 1.2**
1	P1.1	R,W	**Port pin 1.1**
0	P1.0	R,W	**Port pin 1.0**

1. Write '1' or '0' for pin output. Read for pin input, but prior to READ, this bit must have been set to '1' by firmware or by a reset event.

17.1.4 Alternate functions

There are five SFRs used to control the mapping of alternate functions onto MCU port pins, and these SFRs are depicted as switches in *Figure 15 on page 78*.

- Port 3 uses the SFR, P3SFS (*Table 39 on page 83*).
- Port 1 uses SFRs, P1SFS0 (*Table 41 on page 83*) and P1SFS1 (*Table 42 on page 83*).
- Port 4 uses SFRs, P4SFS0 (*Table 44 on page 84*) and P4SFS1 (*Table 45 on page 84*).

Since these SFRs are cleared by a reset, then by default all port pins function as GPIO (not the alternate function) until firmware initializes these SFRs.

Each pin on each of the three ports can be independently assigned a different function on a pin-by-pin basis.

The peripheral functions Timer 2, UART1, and I^2C may be split independently between Port 1 and Port 4 for additional flexibility by giving a wider choice of peripheral usage on a limited number of device pins.

When the selected alternate function is UART0, UART1, or SPI, then the related pins are in quasi-bidirectional mode, including the use of the high-side driver for rapid 0-to-1 output transitions. The high-side driver is enabled for just one MCU_CLK period on 0-to-1 transitions by the delay function at the “digital_alt_func_data_out” signal pictured in *Figure 16* through *Figure 18 on page 80*.

If the alternate function is Timer 0, Timer 1, Timer 2, or PCA input, then the related pins are in quasi-bidirectional mode, but input only.

If the alternate function is ADC, then for each pin the pull-ups, the high-side driver, and the low-side driver are disabled. The analog input is routed directly to the ADC unit. Only Port 1 supports analog functions (*Figure 16 on page 79*). Port 1 is not 5 V tolerant.

If the alternate function is I^2C, the related pins will be in open drain mode, which is just like quasi-bidirectional mode but the high-side driver is not enabled for one cycle when outputting a 0-to-1 transition. Only the low-side driver and the internal weak pull-ups are used. Only Port 3 supports open-drain mode (*Figure 17 on page 79*). I^2C requires the use of an external pull-up resistor on each bus signal, typically 4.7kΩ to V_{CC}.

If the alternate function is PCA output, then the related pins are in push-pull mode, meaning the pins are actively driven and held to logic '1' by the high-side driver, or actively driven and held to logic '0' by the low-side driver. Only Port 4 supports push-pull mode (*Figure 18 on page 80*). Port 4 push-pull pins can source I_{OH} current when driving logic '1,' and sink I_{OL} current when driving logic '0.' This current is significantly more than the capability of pins on Port 1 or Port 3 (see *Table 166 on page 249*).

For example, to assign these port functions:

- Port 1: UART1, ADC[1:0], P1[7:4] are GPIO
- Port 3: UART0, I^2C, P3[5:2] are GPIO
- Port 4: TCM0, SPI, P4[3:1] are GPIO

The following values need to be written to the SFRs:

- P1SFS0 = 00001111b, or 0Fh
- P1SFS1 = 00000011b, or 03h
- P3SFS = 11000011b, or C3h
- P4SFS0 = 11110001b, or F1h
- P4SFS1 = 11110000b, or F0h

20.3 SFR, TCON

Timer 0 and Timer 1 share the SFR, TCON, that controls these timers and provides information about them. See *Table 54 on page 95*.

Bits IE0 and IE1 are not related to Timer/Counter functions, but they are set by hardware when a signal is active on one of the two external interrupt pins, EXTINT0 and EXTINT1. For system information on all of these interrupts, see *Table 18 on page 62*, Interrupt Summary.

Bits IT0 and IT1 are not related to Timer/Counter functions, but they control whether or not the two external interrupt input pins, EXTINT0 and EXTINT1 are edge or level triggered.

20.4 SFR, TMOD

Timer 0 and Timer 1 have four modes of operation controlled by the SFR named TMOD (*Table 56 on page 97*).

20.5 Timer 0 and Timer 1 operating modes

The “Timer” or “Counter” function is selected by the $C/\overline{T}$ control bits in TMOD. The four operating modes are selected by bit-pairs M[1:0] in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 is different.

20.5.1 Mode 0

Putting either Timer/Counter into mode 0 makes it an 8-bit Counter with a divide-by-32 prescaler. *Figure 21 on page 98* shows mode 0 operation as it applies to Timer 1 (same applies to Timer 0).

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all '1s' to all '0s,' it sets the Timer Interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or EXTINT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input pin, EXTINT1, to facilitate pulse width measurements). TR1 is a control bit in the SFR, TCON. GATE is a bit in the SFR, TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag, TR1, does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, C0, TL0, TH0, and EXTINT0 for the corresponding Timer 1 signals in *Figure 21 on page 98*. There are two different GATE Bits, one for Timer 1 and one for Timer 0.

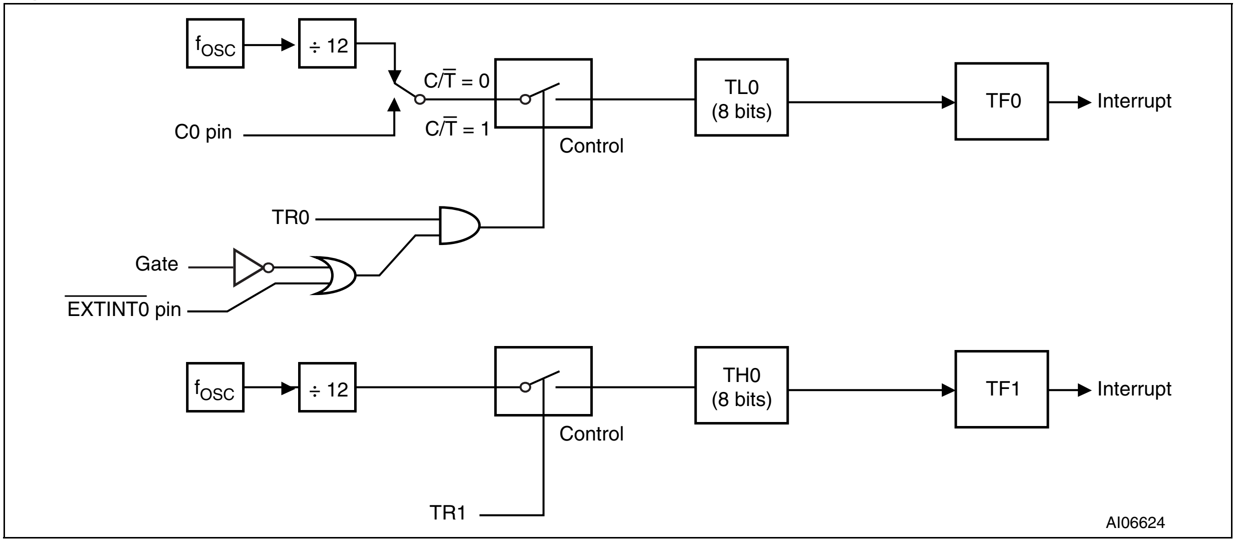
20.5.2 Mode 1

Mode 1 is the same as mode 0, except that the Timer register is being run with all 16 bits.

20.5.3 Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in *Figure 22 on page 98*. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset with firmware. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

Figure 23. Timer/Counter mode 3: Two 8-bit counters



20.6 Timer 2

Timer 2 can operate as either an event timer or as an event counter. This is selected by the bit C/$\overline{T2}$ in the SFR named, T2CON (*Table 58 on page 100*). Timer 2 has three operating modes selected by bits in T2CON, according to *Table 60 on page 101*. The three modes are:

- Capture mode
- Auto re-load mode
- Baud rate generator mode

20.6.1 Capture mode

In Capture mode there are two options which are selected by the bit EXEN2 in T2CON. *Figure 24 on page 104* illustrates Capture mode.

If EXEN2 = 0, then Timer 2 is a 16-bit timer if C/$\overline{T2}$ = 0, or it's a 16-bit counter if C/$\overline{T2}$ = 1, either of which sets the interrupt flag bit TF2 upon overflow.

If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input pin T2X causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2X causes interrupt flag bit EXF2 in T2CON to be set. Either flag TF2 or EXF2 will generate an interrupt and the MCU must read both flags to determine the cause. Flags TF2 and EXF2 are not automatically cleared by hardware, so the firmware servicing the interrupt must clear the flag(s) upon exit of the interrupt service routine.

20.6.2 Auto-reload mode

In the Auto-reload mode, there are again two options, which are selected by the bit EXEN2 in T2CON. *Figure 25 on page 104* shows Auto-reload mode.

If EXEN2 = 0, then when Timer 2 counts up and rolls over from FFFFh it not only sets the interrupt flag TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value contained in registers RCAP2L and RCAP2H, which are preset with firmware.

If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2X will also trigger the 16-bit reload and set the interrupt flag EXF2. Again, firmware servicing the interrupt must read both TF2 and EXF2 to determine the cause, and clear the flag(s) upon exit.

Note: *The UPSD33xx does not support selectable up/down counting in Auto-reload mode (this feature was an extension to the original 8032 architecture).*

Table 58. T2CON: Timer 2 Control register (SFR C8h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{\text{T2}}$	CP/$\overline{\text{RL2}}$

Table 59. T2CON register bit definition

Bit	Symbol	R/W	Definition
7	TF2	R,W	**Timer 2 flag,** causes interrupt if enabled. TF2 is set by hardware upon overflow. Must be cleared by firmware. TF2 will not be set when either RCLK or TCLK =1.
6	EXF2	R,W	**Timer 2 flag,** causes interrupt if enabled. EXF2 is set when a capture or reload is caused by a negative transition on T2X pin and EXEN2 = 1. EXF2 must be cleared by firmware.
5	RCLK[1]	R,W	**UART0 Receive Clock control.** When RCLK = 1, UART0 uses Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK=0, Timer 1 overflow is used for its receive clock
4	TCLK[1]	R,W	**UART0 Transmit Clock control.** When TCLK = 1, UART0 uses Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK=0, Timer 1 overflow is used for transmit clock
3	EXEN2	R,W	**Timer 2 External Enable.** When EXEN2 = 1, capture or reload results when negative edge on pin T2X occurs. EXEN2 = 0 causes Timer 2 to ignore events at pin T2X.
2	TR2	R,W	**Timer 2 run control.** 1 = Timer/Counter 2 is on, 0 = Timer Counter 2 is off.

Table 64. SCON0 register bit definition (continued)

Bit	Symbol	R/W	Definition
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 65. 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

Table 66. SCON1 register bit definition

Bit	Symbol	R/W	Definition
7	SM0	R,W	**Serial Mode Select**, See *Table 62 on page 107*. 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.

Table 69. RDACON register bit definition (continued)

Bit	Symbol	R/W	Definition
5	PULSE	RW	**IrDA Pulse Modulation Select** 0 = 1.627µs 1 = 3/16 bit time pulses
4-0	CDIV[4:0]	RW	**Specify Clock Divider** (see *Figure 70 on page 121*)

22.1 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 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 36 on page 120*.

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 35 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 1.83 MHz as possible, because SIRClk at 1.83 MHz will produce an fixed IrDA data pulse width of 1.63µs. *Table 70* provides recommended values for CDIV[4:0] based on several different values of f_{OSC}.

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

Table 70. Recommended CDIV[4:0] values to generate SIRClk (default CDIV[4:0] = 0Fh, 15 decimal)

f_{OSC} (MHz)	Value in CDIV[4:0]	Resulting f_{SIRCLK} (MHz)
40.00	16h, 22 decimal	1.82
36.864, or 36.00	14h, 20 decimal	1.84, or 1.80
24.00	0Dh, 13 decimal	1.84
11.059, or 12.00	06h, 6 decimal	1.84, or 2.00
7.3728(1)	04h, 4 decimal	1.84

1. When PULSE bit = 0 (fixed data pulse width), this is minimum recommended f_{OSC} because CDIV[4:0] must be 4 or greater.

Table 74. S1STA: I^{2}C Interface Status register (SFR DDh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GC	STOP	INTR	TX_MODE	BBUSY	BLOST	$\overline{\text{ACK\_RESP}}$	SLV

Table 75. S1STA register bit definition

Bit	Symbol	R/W	Function
7	GC	R	**General Call flag** GC = 1 if the General Call address of 00h was received when SIOE is in Slave mode, and GC is cleared by a START or STOP condition on the bus. If the SIOE is in Master mode when GC = 1, the Bus Lost condition exists, and BLOST = 1.
6	STOP	R	**STOP flag** STOP = 1 while SIOE detects a STOP condition on the bus when in Master or Slave mode.
5	INTR	R,W	**Interrupt flag** INTR is set to 1 by any of the five I^{2}C interrupt conditions listed above. INTR must be cleared by firmware.
4	TX_MODE	R	**Transmission Mode flag** TX_MODE = 1 whenever the SIOE is in Master-Transmitter or Slave-Transmitter mode. TX_MODE = 0 when SIOE is in any receiver mode.
3	BBUSY	R	**Bus Busy flag** BBUSY = 1 when the I^{2}C bus is in use. BBUSY is set by the SIOE when a START condition exists on the bus and BBUSY is cleared by a STOP condition.
2	BLOST	R	**Bus Lost flag** BLOST is set when the SIOE is in Master mode and it looses the arbitration process to another Master device on the bus.
1	$\overline{\text{ACK\_RESP}}$	R	**Not Acknowledge Response flag** While SIOE is in Transmitter mode: – After SIOE sends a byte, $\overline{\text{ACK\_RESP}}$ = 1 whenever the external I^{2}C device receives the byte, but that device does NOT assert an accolade signal (external device asserted a high on SDA during the acknowledge bit-time). – After SIOE sends a byte, $\overline{\text{ACK\_RESP}}$ = 0 whenever the external I^{2}C device receives the byte, and that device DOES assert an accolade signal (external device drove a low on SDA during the acknowledge bit-time) *Note: If SIOE is in Master-Transmitter mode, and* $\overline{\text{ACK\_RESP}}$ *= 1 due to a Slave-Transmitter not sending an Acknowledge, a STOP condition will not automatically be generated by the SIOE. The STOP condition must be generated with S1CON.STO = 1.*
0	SLV	R	**Slave Mode flag** SLV = 1 when the SIOE is in Slave mode. SLV = 0 when the SIOE is in Master mode (default).

Table 81. S1SETUP register bit definition

Bit	Symbol	R/W	Function
7	EN_SS	R/W	**Enable Sample Setup** EN_SS = 1 will force the SIOE to sample[1] a START condition on the bus the number of times specified in SMPL_SET[6:0]. EN_SS = 0 means the SIOE will sample[1] a START condition only one time, regardless of the contents of SMPL_SET[6:0].
6:0	SMPL_SET [6:0]	–	**Sample Setting** Specifies the number of bus samples([1] taken during a START condition. See *Table 82 on page 133* for values.

[1] Sampling SCL and SDA lines begins after '1'-to-'0' transition on SDA occurred while SCL is high. Time between samples is $1/f_{OSC}$.

Table 82. Number of I^2C bus samples taken after 1-to-0 transition on SDA (START condition)

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

Table 83. Start condition hold time

I^2C bus speed	Range of I^2C clock speed (f_{SCL})	Minimum START condition hold time (t_{HLDSTA})
Standard	Up to 100kHz	4000ns
Fast	101kHz to 400kHz	600ns
High	401kHz to 833kHz[1]	160ns

[1] 833kHz is maximum for UPSD33xx devices.

Table 84 provides recommended settings for S1SETUP based on various combinations of f_{OSC} and f_{SCL}. Note that the “Total Sample Period” times in *Table 83* are typically slightly less than the minimum START condition hold time, t_{HLDSTA} for a given I^2C bus speed.

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

24 Synchronous peripheral interface (SPI)

UPSD33xx devices support one serial SPI interface in Master mode only. This is a three- or four-wire synchronous communication channel, capable of full-duplex operation on 8-bit serial data transfers. The four SPI bus signals are:

- SPIRxD

  Pin P1.5 or P4.5 receives data from the Slave SPI device to the UPSD33xx
- SPITxD

  Pin P1.6 or P4.6 transmits data from the UPSD33xx to the Slave SPI device
- SPICLK

  Pin P1.4 or P4.4 clock is generated from the UPSD33xx to the SPI Slave device
- $\overline{\text{SPISEL}}$

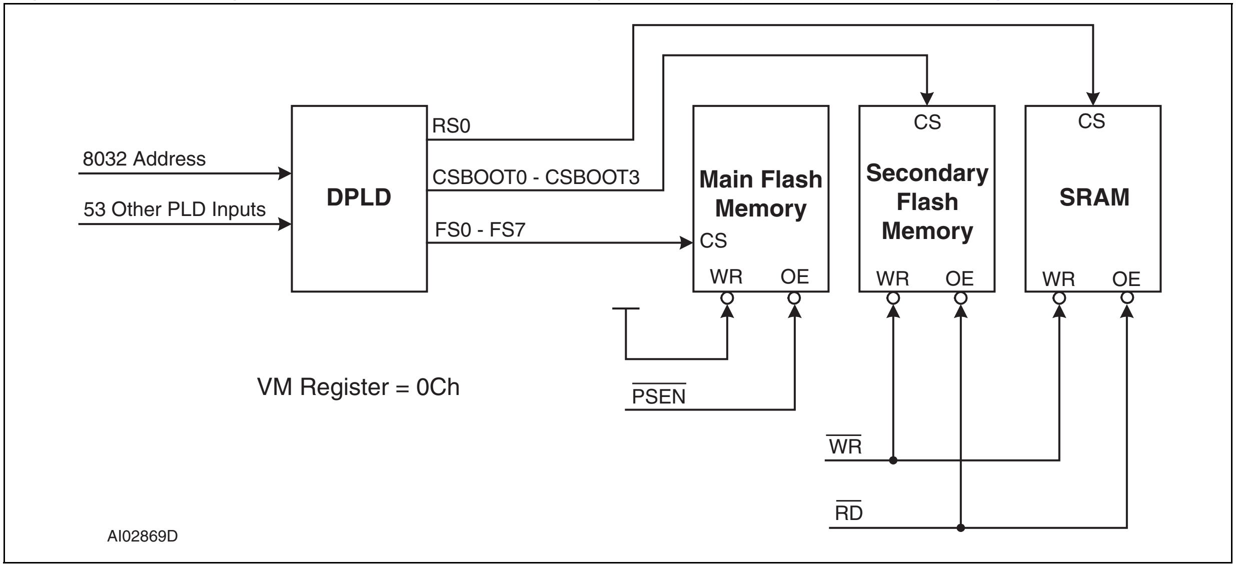
  Pin P1.7 or P4.7 selects the signal from the UPSD33xx to an individual Slave SPI device

This SPI interface supports single-Master/multiple-Slave connections. Multiple-Master connections are not directly supported by the UPSD33xx (no internal logic for collision detection).

If more than one Slave device is required, the $\overline{\text{SPISEL}}$ signal may be generated from UPSD33xx GPIO outputs (one for each Slave) or from the PLD outputs of the PSD module. *Figure 40* illustrates three examples of SPI device connections using the UPSD33xx:

- Single-Master/Single-Slave with $\overline{\text{SPISEL}}$
- Single-Master/Single-Slave without $\overline{\text{SPISEL}}$
- Single-Master/Multiple-Slave without $\overline{\text{SPISEL}}$

Figure 59. VM register example corresponding to memory map example of *Figure 32*



27.3 Runtime control register definitions (CSIOP)

The 39 csiop registers are defined in *Table 116*. The 8032 can access each register by the address offset (specified in *Table 116*) added to the csiop base address that was specified in PSDsoft Express. Do not write to unused locations within the csiop block of 256 registers, they should remain logic zero.

Table 116. CSIOP registers and their Offsets (in hexadecimal

Register name	Port A (80-pin)	Port B	Port C	Port D	Other	Description	Link
Data In	00h	01h	10h	11h		MCU I/O input mode. Read to obtain current logic level of pins on Ports A, B, C, or D. No WRITEs.	*Table 132 on page 210*
Control	02h	03h				Selects MCUI/O or Latched Address Out mode. Logic 0 = MCU I/O, 1 = 8032 Addr Out. Write to select mode. Read for status.	*Table 144 on page 214*
Data Out	04h	05h	12h	13h		MCU I/O output mode. Write to set logic level on pins of Ports A, B, C, or D. Read to check status. This register has no effect if a port pin is driven by an OMC output from PLD.	*Table 136 on page 210*
Direction	06h	07h	14h	15h		MCU I/O mode. Configures port pin as input or output. Write to set direction of port pins. Logic 1 = out, Logic 0 = in. Read to check status.	*Table 140 on page 211*

27.4.3 Reading Flash memory

Under typical conditions, the 8032 may read the Flash memory using READ operations (READ bus cycles) just as it would a ROM or RAM device. Alternately, the 8032 may use READ operations to obtain status information about a Program or Erase operation that is currently in progress. The following sections describe the kinds of READ operations.

27.4.4 Read memory contents

Flash memory is placed in the Read Array mode after Power-up, after a PSD module reset event, or after receiving a Reset Flash memory instruction sequence from the 8032. The 8032 can read Flash memory contents using standard READ bus cycles anytime the Flash array is in Read Array mode. Flash memories will always be in Read Array mode when the array is not actively engaged in a program or erase operation.

27.4.5 Reading the erase/program status bits

The Flash arrays provide several status bits to be used by the 8032 to confirm the completion of an erase or program operation on Flash memory, shown in *Table 118 on page 187*. The status bits can be read as many times as needed until an operation is complete.

The 8032 performs a READ operation to obtain these status bits while an erase or program operation is being executed by the state machine inside each Flash memory array.

27.4.6 Data polling flag (DQ7)

While programming either Flash memory, the 8032 may read the Data Polling Flag Bit (DQ7), which outputs the complement of the D7 Bit of the byte being programmed into Flash memory. Once the program operation is complete, DQ7 is equal to D7 of the byte just programmed into Flash memory, indicating the program cycle has completed successfully. The correct select signal, FSx or CSBOOTx, must be active during the entire polling procedure.

Polling may also be used to indicate when an erase operation has completed. During an erase operation, DQ7 is '0.' After the erase is complete DQ7 is '1.' The correct select signal, FSx or CSBOOTx, must be active during the entire polling procedure.

DQ7 is valid after the fourth instruction byte WRITE operation (for program instruction sequence) or after the sixth instruction byte WRITE operation (for erase instruction sequence).

If all Flash memory sectors to be erased are protected, DQ7 is reset to ’0’ for about 100µs, and then DQ7 returns to the value of D7 of the previously addressed byte. No erasure is performed.

27.4.7 Toggle flag (DQ6)

The Flash memories offer an alternate way to determine when a Flash memory program operation has completed. During the program operation and while the correct sector select FSx or CSBOOTx is active, the Toggle Flag Bit (DQ6) toggles from '0' to '1' and '1' to ’0’ on subsequent attempts to read any byte of the same Flash array.

When the internal program operation is complete, the toggling stops and the data read on the data bus D0-7 is the actual value of the addressed memory byte. The device is now

Table 127. Input Macrocell Port A (address = csiop + offset 0Ah)(1)(2)

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

1. Port A not available on 52-pin UPSD33xx devices
2. 1 = current state of IMC is logic '1,' 0 = current state is logic ’0’

Table 128. Input Macrocell Port B (address = csiop + offset 0Bh)(1)

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

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

Table 129. Input Macrocell Port C (address = csiop + offset 18h)(1)(2)

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

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

27.4.34 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

27.4.35 General port architecture

The general architecture for a single I/O Port pin is shown in *Figure 68 on page 208*. Port structures for Ports A, B, C, and D differ slightly and are shown in *Figure 73 on page 219* though *Figure 76 on page 223*.

Figure 68 on page 208 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).

no PLD inputs are changing, and the PLDs will even use less AC current when inputs do change compared to Turbo mode.

When the Turbo mode is enabled, there is a significant DC current component AND the AC current component is higher than non-Turbo mode, as shown in *Figure 84 on page 242* (5 V) and *Figure 85 on page 243* (3.3 V).

Blocking bits

Significant power savings can be achieved by blocking 8032 bus control signals ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE) from reaching PLD inputs, if these signals are not used in any PLD equations. Blocking is achieved by the 8032 writing to the “blocking bits” in csiop PMMR registers. Current consumption of the PLDs is directly related to the composite frequency of all transitions on PLD inputs, so blocking certain PLD inputs can significantly lower PLD operating frequency and power consumption (resulting in a lower frequency on the graphs of *Figure 84 on page 242* and *Figure 85 on page 243*).

Note: It is recommended to prevent unused inputs from floating on Ports A, B, C, and D by pulling them up to V_{DD} with a weak external resistor (100 $K\Omega$), or by setting the csiop Direction register to “output” at run-time for all unused inputs. This will prevent the CMOS input buffers of unused input pins from drawing excessive current.

The csiop PMMR register definitions are shown in *Table 154* through *Table 156 on page 226*.

Table 154. Power Management Mode register PMMR0 (address = csiop + offset B0h)[1]

Bit num.	Bit name	Value	Description
Bit 0	X	0	Not used, and should be set to zero.
Bit 1	APD Enable	0	Automatic Power-down (APD) counter is disabled.
		1	APD counter is enabled
Bit 2	X	0	Not used, and should be set to zero.
Bit 3	PLD Turbo Disable	0 = on	PLD Turbo mode is on
		1 = off	PLD Turbo mode is off, saving power.
Bit 4	Blocking bit, CLKIN to PLDs[2]	0 = on	CLKIN (pin PD1) to the PLD Input Bus is not blocked. Every transition of CLKIN powers-up the PLDs.
		1 = off	CLKIN input to PLD Input Bus is blocked, saving power. But CLKIN still goes to APD counter.
Bit 5	Blocking bit, CLKIN to OMCs only[2]	0 = on	CLKIN input is not blocked from reaching all OMC’s common clock inputs.
		1 = off	CLKIN input to common clock of all OMCs is blocked, saving power. But CLKIN still goes to APD counter and all PLD logic besides the common clock input on OMCs.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

1. All the bits of this register are cleared to zero following Power-up. Subsequent Reset ($\overline{\text{RST}}$) pulses do not clear the registers.
2. Blocking bits should be set to logic ’1’ only if the signal is not needed in a DPLD or GPLD logic equation.

Figure 77. Automatic Power-down (APD) unit

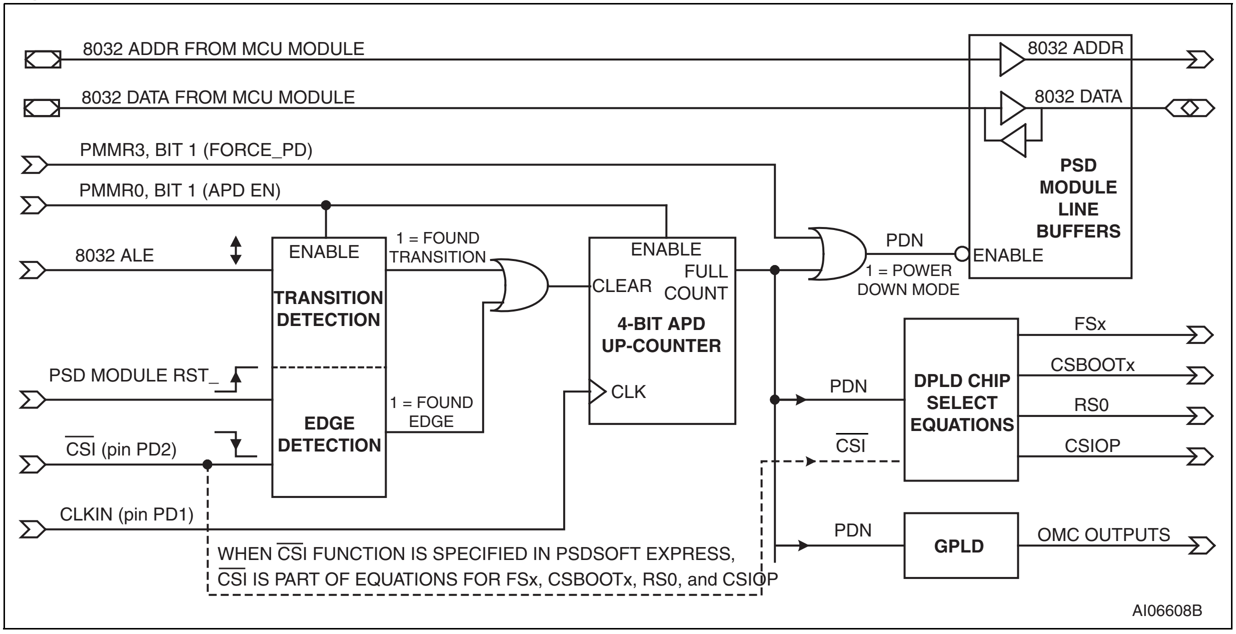
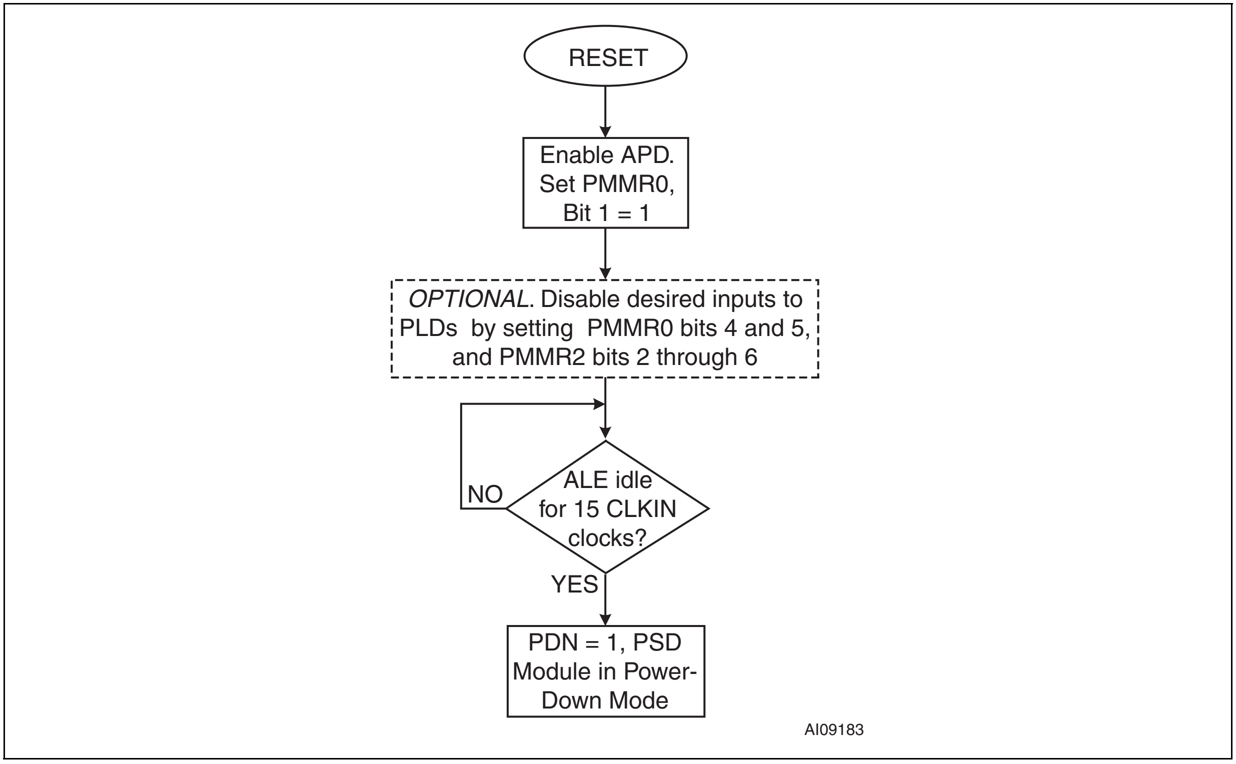


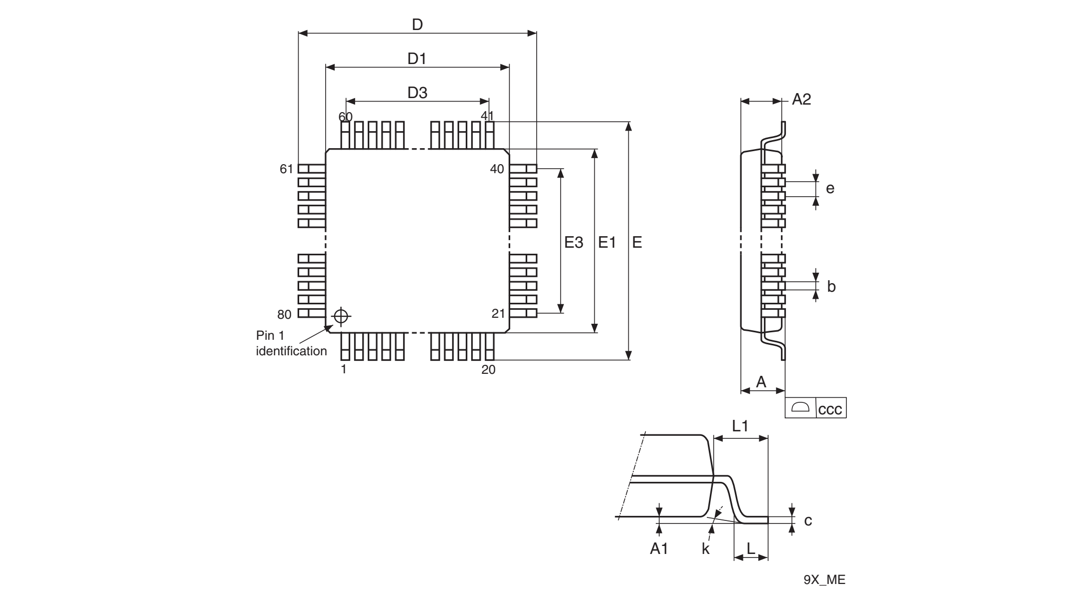
Figure 78. Power-down mode flowchart


RESET
Enable APD. Set PMMR0, Bit 1 = 1
OPTIONAL. Disable desired inputs to PLDs by setting PMMR0 bits 4 and 5, and PMMR2 bits 2 through 6
ALE idle for 15 CLKIN clocks?
NO
YES
PDN = 1, PSD Module in Power-Down Mode
AI09183

31 Package mechanical information

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

Figure 103. LQFP80 – 80-lead plastic thin, quad, flat package outline



1. Drawing not to scale.

Table 192. LQFP80 – 80-lead plastic thin, quad, flat package mechanical data

Symbol	millimeters			inches[1]		
	Typ	Min	Max	Typ	Min	Max
A			1.600			0.0630
A1		0.050	0.150		0.0020	0.0059
A2	1.400	1.350	1.450	0.0551	0.0531	0.0571
b	0.220	0.170	0.270	0.0087	0.0067	0.0106
C		0.090	0.200		0.0035	0.0079
D	14.000			0.5512		
D1	12.000			0.4724		
D3	9.500			0.3740		
E	14.000			0.5512		
E1	12.000			0.4724		
E3	9.500			0.3740		
e	0.500			0.0197		
L	0.600	0.450	0.750	0.0236	0.0177	0.0295
L1	1.000			0.0394		
k		0°	7°		0°	7°
ccc	0.080			0.0031		

1. Values in inches are converted from mm and rounded to 4 decimal digits.