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	45
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	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3333dv-40u6

3 UPSD33xx hardware description

The UPSD33xx has a modular architecture built from a stacked die process. There are two die, one is designated “MCU module” in this document, and the other is designated “PSD module” (see *Figure 4 on page 27*). In all cases, the MCU module die operates at 3.3 V with 5 V tolerant I/O. The PSD module is either a 3.3 V die or a 5 V die, depending on the UPSD33xx device as described below.

The MCU module consists of a fast 8032 core, that operates with 4 clocks per instruction cycle, and has many peripheral and system supervisor functions. The PSD module provides the 8032 with multiple memories (two Flash and one SRAM) for program and data, programmable logic for address decoding and for general-purpose logic, and additional I/O. The MCU module communicates with the PSD module through internal address and data busses (A8 – A15, AD0 – AD7) and control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE, $\overline{RESET}$).

There are slightly different I/O characteristics for each module. I/Os for the MCU module are designated as Ports 1, 3, and 4. I/Os for the PSD module are designated as Ports A, B, C, and D.

For all 5 V UPSD33xx devices, a 3.3 V MCU module is stacked with a 5 V PSD module. In this case, a 5 V UPSD33xx device must be supplied with 3.3 V_{CC} for the MCU module and 5.0V_{DD} for the PSD module. Ports 3 and 4 of the MCU module are 3.3 V ports with tolerance to 5 V devices (they can be directly driven by external 5 V devices and they can directly drive external 5 V devices while producing a V_{OH} of 2.4 V min and V_{CC} max). Ports A, B, C, and D of the PSD module are true 5 V ports.

For all 3.3 V UPSD33xxV devices, a 3.3 V MCU module is stacked with a 3.3 V PSD module. In this case, a 3.3 V UPSD33xx device needs to be supplied with a single 3.3 V voltage source at both V_{CC} and V_{DD}. I/O pins on Ports 3 and 4 are 5 V tolerant and can be connected to external 5 V peripherals devices if desired. Ports A, B, C, and D of the PSD module are 3.3 V ports, which are not tolerant to external 5 V devices.

Refer to *Table 3 on page 27* for port type and voltage source requirements.

80-pin UPSD33xx devices provide access to 8032 address, data, and control signals on external pins to connect external peripheral and memory devices. 52-pin UPSD33xx devices do not provide access to the 8032 system bus.

All non-volatile memory and configuration portions of the UPSD33xx device are programmed through the JTAG interface and no special programming voltage is needed. This same JTAG port is also used for debugging of the 8032 core at runtime providing breakpoint, single-step, display, and trace features. A non-volatile security bit may be programmed to block all access via JTAG interface for security. The security bit is defeated only by erasing the entire device, leaving the device blank and ready to use again.

Table 3. Port type and voltage source combinations

Device Type	V_{CC} for MCU module	V_{DD} for PSD module	Ports 3 and 4 on MCU module	Ports A, B, C, and D on PSD module
5 V: UPSD33xx	3.3 V	5.0 V	3.3 V but 5 V tolerant	5 V
3.3 V: UPSD33xxV	3.3 V	3.3 V	3.3 V but 5 V tolerant	3.3 V. NOT 5 V tolerant

Figure 4. UPSD33xx functional modules

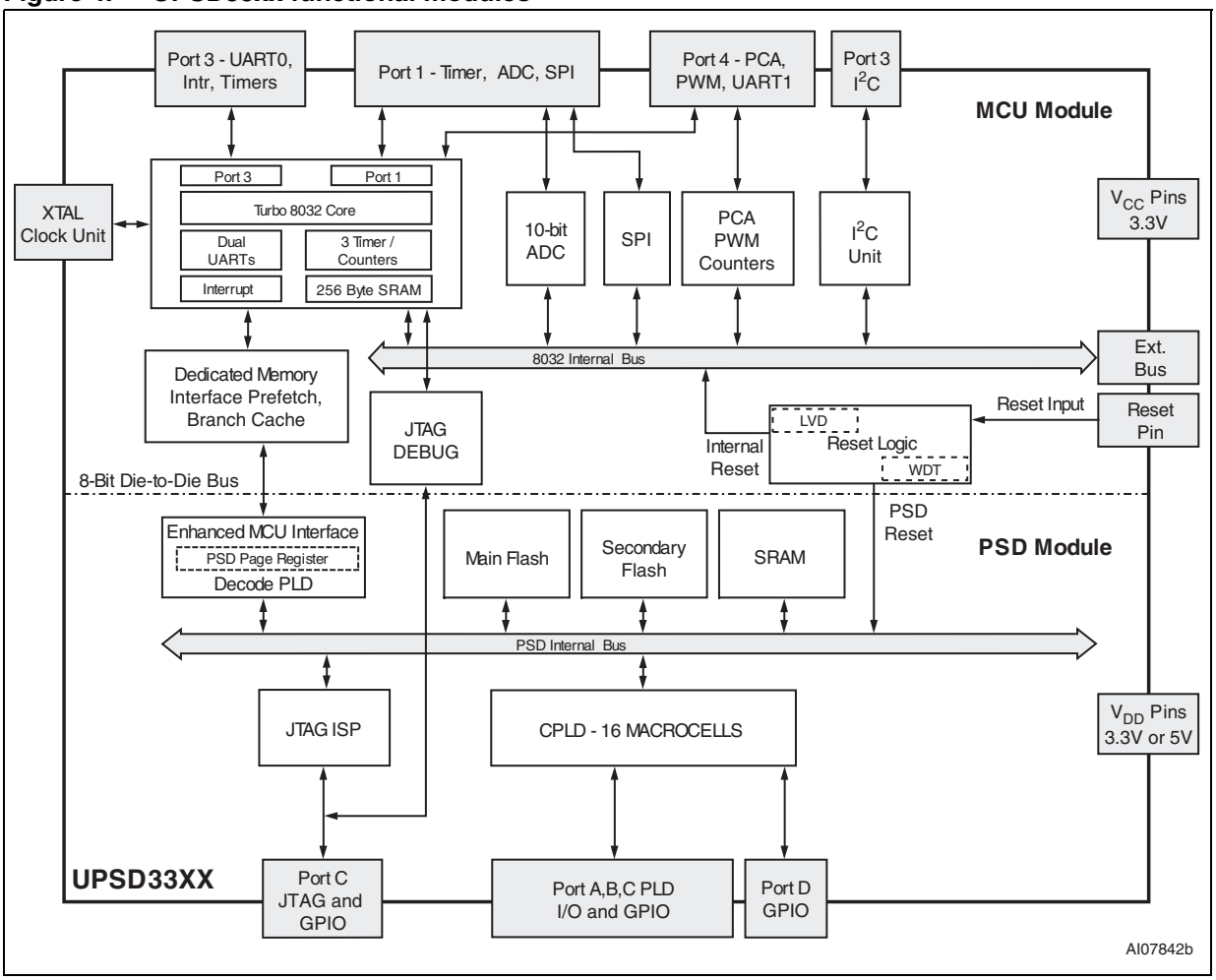
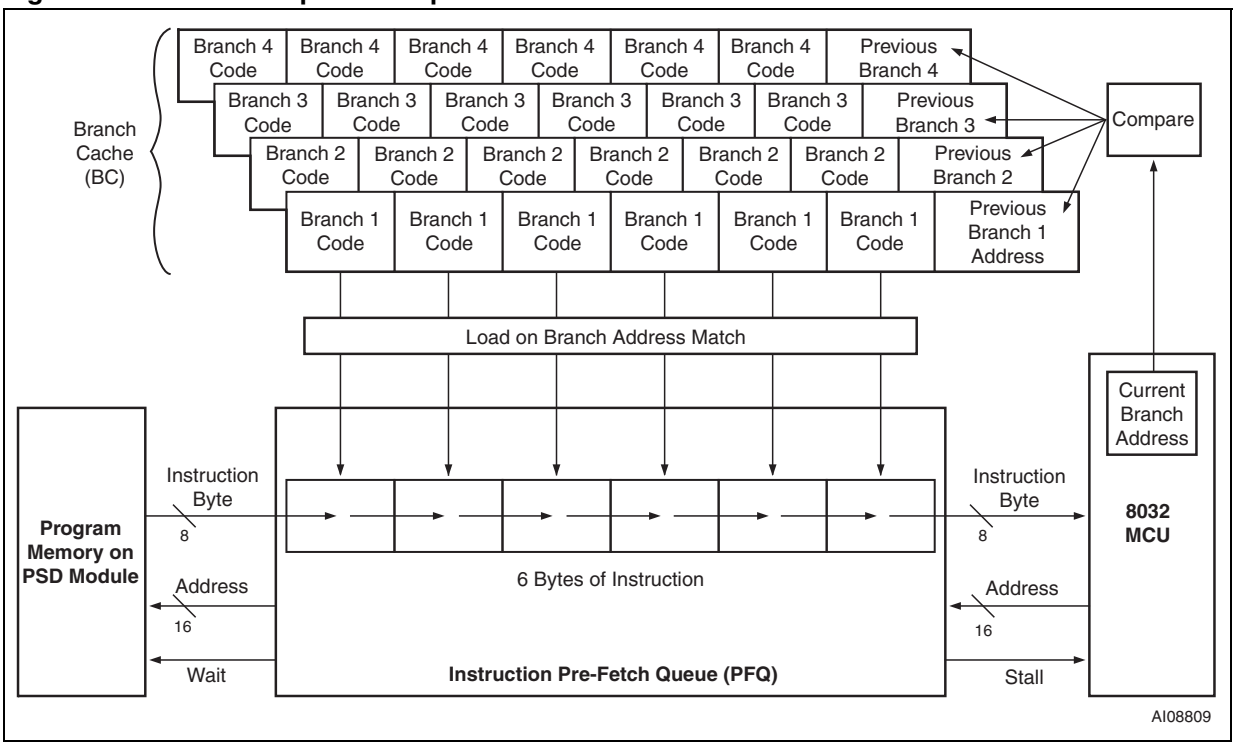


Figure 7. Instruction pre-fetch queue and branch cache



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

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

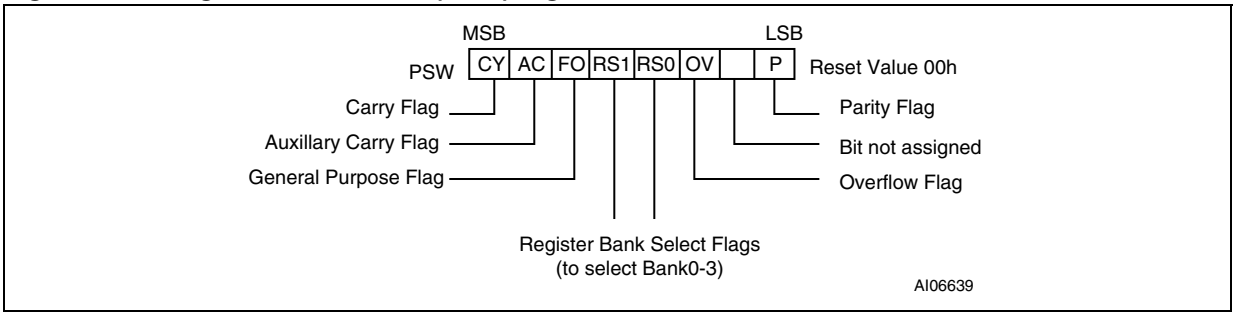


Table 10. Program branching instruction set (continued)

Mnemonic[1] and use		Description	Length/cycles
CJNE	Rn, #data, rel	Compare immediate to register, jump if not equal	3 byte/2 cycle
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
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).

13.1.8 PCA interrupt

The PCA has eight interrupt sources, which are logically ORed together when interrupting the MCU.The ISR must read the flag bits to determine the cause of the interrupt.

- Each of the six TCMs can generate a "match or capture" interrupt on flag bits OFV5..0 respectively.
- Each of the two 16-bit counters can generate an overflow interrupt on flag bits INTF1 and INTF0 respectively.

Table 19, *Table 18*, *Table 19*, and *Table 21* have detailed bit definitions of the interrupt system SFRs.

Table 19. IE: Interrupt Enable register (SFR A8h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	–	ET2	ES0	ET1	EX1	ET0	EX0

Table 20. IE register bit definition

Bit	Symbol	R/W	Function
7	EA	R,W	**Global disable bit** 0 = All interrupts are disabled. 1 = Each interrupt source can be individually enabled or disabled by setting or clearing its enable bit.
6	–	R,W	Do not modify this bit. It is used by the JTAG debugger for instruction tracing. Always read the bit and write back the same bit value when writing this SFR.
5(1)	ET2	R,W	**Enable Timer 2 Interrupt**
4(1)	ES0	R,W	**Enable UART0 Interrupt**
3(1)	ET1	R,W	**Enable Timer 1 Interrupt**
2(1)	EX1	R,W	**Enable External Interrupt INT1**
1(1)	ET0	R,W	**Enable Timer 0 Interrupt**
0(1)	EX0	R,W	**Enable External Interrupt INT0**

1. 1 = Enable Interrupt, 0 = Disable Interrupt

Table 21. IEA: Interrupt Enable Addition register (SFR A7h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EADC	ESPI	EPCA	ES1	–	–	EI^{2}C	–

Table 22. IEA register bit definition

Bit	Symbol	R/W	Function
7(1)	EADC	R,W	**Enable ADC Interrupt**
6(1)	ESPI	R,W	**Enable SPI Interrupt**
5(1)	EPCA	R,W	**Enable Programmable Counter Array Interrupt**

Figure 13. Clock generation logic

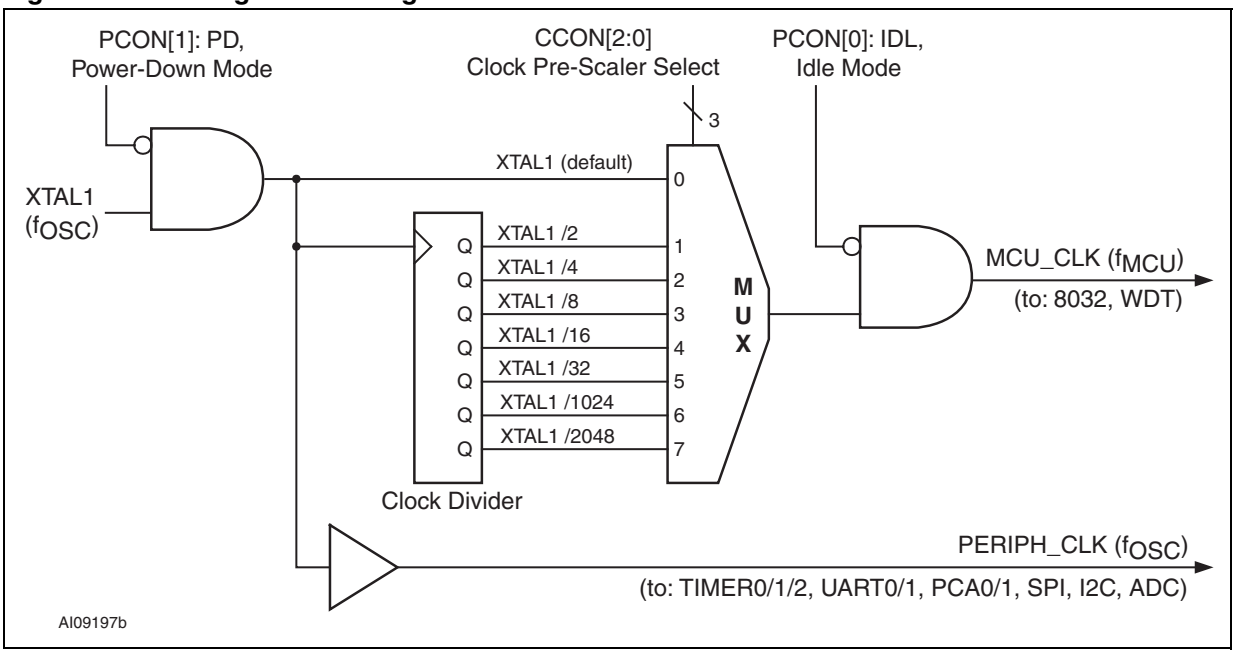


Table 27. CCON0: Clock Control register (SFR F9h, reset value 10h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	DBGCE	CPUAR	CPUPS[2:0]		

Table 28. CCON0 register bit definition

Bit	Symbol	R/W	Definition
7	–	–	Reserved
6	–	–	Reserved
5	–	–	Reserved
4	DBGCE	R,W	**Debug Unit Breakpoint Comparator Enable** 0 = JTAG debug unit comparators are disabled 1 = JTAG debug unit comparators are enabled (Default condition after reset)
3	CPUAR	R,W	**Automatic MCU Clock Recovery** 0 = There is no change of CPUPS[2:0] when an interrupt occurs. 1 = Contents of CPUPS[2:0] automatically become 000b whenever any interrupt occurs.
2:0	CPUPS	R,W	**MCUCLK Pre-Scaler** 000b: $f_{MCU} = f_{OSC}$ (Default after reset) 001b: $f_{MCU} = f_{OSC}/2$ 010b: $f_{MCU} = f_{OSC}/4$ 011b: $f_{MCU} = f_{OSC}/8$ 100b: $f_{MCU} = f_{OSC}/16$ 101b: $f_{MCU} = f_{OSC}/32$ 110b: $f_{MCU} = f_{OSC}/1024$ 111b: $f_{MCU} = f_{OSC}/2048$

15 Power saving modes

The UPSD33xx is a combination of two die, or modules, each module having it's own current consumption characteristics. This section describes reduced power modes for the MCU module. See *Section 27.1.16: Power management on page 170* for reduced power modes of the PSD module. Total current consumption for the combined modules is determined in the DC specifications at the end of this document.

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

- Idle mode
- Power-down mode
- Reduced Frequency mode

15.1 Idle mode

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

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

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

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

To enter Idle mode, the 8032 MCU executes an instruction to set the IDL bit in the SFR named PCON, shown in *Table 31 on page 72*. This is the last instruction executed in normal operating mode before Idle mode is activated. Once in Idle mode, the MCU status is entirely preserved, and there are no changes to: SP, PSW, PC, ACC, SFRs, DATA, IDATA, or XDATA.

The following are factors related to Idle mode exit:

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

Table 39. P3SFS: Port 3 Special Function Select register (SFR 91h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3SFS7	P3SFS6	P3SFS5	P3SFS4	P3SFS3	P3SFS2	P3SFS1	P3SFS0

Table 40. P3SFS register bit definition

Port 3 pin	R/W	Default port function	Alternate port function
		P3SFS[i] - 0; Port 3 pin, i = 0..7	P3SFS[i] - 1; Port 3 pin, i = 0..7
0	R,W	GPIO	UART0 Receive, RXD0
1	R,W	GPIO	UART0 Transmit, TXD0
2	R,W	GPIO	Ext Intr 0/Timer 0 Gate, EXT0INT/TG0
3	R,W	GPIO	Ext Intr 1/Timer 1 Gate, EXT1INT/TG1
4	R,W	GPIO	Counter 0 Input, C0
5	R,W	GPIO	Counter 0 Input, C1
6	R,W	GPIO	I^2C Data, I2CSDA
7	R,W	GPIO	I^2C Clock, I2CCL

Table 41. P1SFS0: Port 1 Special Function Select 0 register (SFR 8Eh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1SF07	P1SF06	P1SF05	P1SF04	P1SF03	P1SF02	P1SF01	P1SF00

Table 42. P1SFS1: Port 1 Special Function Select 1 register (SFR 8Fh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1SF17	P1SF16	P1SF15	P1SF14	P1SF13	P1SF12	P1SF11	P1SF10

Table 43. P1SFS0 and P1SFS1 details

Port 1 pin	R/W	Default port function		Alternate 1 port function
		P1SFS0[i] = 0, P1SFS1[i] = x		P1SFS0[i] = 1, P1SFS1[i] = 0
		Port 1 Pin, i = 0.. 7		Port 1 Pin, i = 0.. 7
0	R,W	GPIO	Timer 2 Count Input, T2	ADC Chn 0 Input, ADC0
1	R,W	GPIO	Timer 2 Trigger Input, TX2	ADC Chn 1 Input, ADC1
2	R,W	GPIO	UART1 Receive, RXD1	ADC Chn 2 Input, ADC2

Note: 1 A $\overline{PSEN}$ bus cycle in progress may be aborted before completion if the PFQ and Branch Cache (BC) determines the current code fetch cycle is not needed.

2 Whenever the same number of MCU_CLK periods is specified in BUSCON for both $\overline{PSEN}$ and $\overline{RD}$ cycles, the bus cycle timing is typically identical for each of these types of bus cycles. In this case, the only time $\overline{PSEN}$ read cycles are longer than $\overline{RD}$ read cycles is when the PFQ issues a stall while reloading. PFQ stalls do not affect $\overline{RD}$ read cycles. By comparison, in many traditional 8051 architectures, $\overline{RD}$ bus cycles are always longer than $\overline{PSEN}$ bus cycles.

18.2 Bus write cycles ($\overline{WR}$)

When the $\overline{WR}$ signal is used, a byte of data is written directly to the PSD module or external device, no PFQ or caching is involved. Bits in the BUSCON register determine the number of MCU_CLK periods for bus write cycles to all addresses. It is not possible to specify in BUSCON a different number of MCU_CLK periods for writes to various address ranges.

18.3 Controlling the PFQ and BC

The BUSCON register allows firmware to enable and disable the PFQ and BC at run-time. Sometimes it may be desired to disable the PFQ and BC to ensure deterministic execution. The dynamic action of the PFQ and BC may cause varying program execution times depending on the events that happen prior to a particular section of code of interest. For this reason, it is not recommended to implement timing loops in firmware, but instead use one of the many hardware timers in the UPSD33xx.

By default, the PFQ and BC are enabled after a reset condition.

*Note: **Important:** Disabling the PFQ or BC will seriously reduce MCU performance.*

Table 49. Number of MCU_CLK periods required to optimize bus transfer rate

MCU clock frequency, MCU_CLK (f_{MCU})	CW[1:0] Clk periods		RDW[1:0] Clk periods		WRW[1:0] Clk periods	
	3.3 V(1)	5 V(1)	3.3 V(1)	5 V(1)	3.3 V(1)	5 V(1)
40 MHz, Turbo mode PSD(2)	5	4	5	4	5	4
40 MHz, Non-Turbo mode PSD	6	5	6	5	6	5
36 MHz, Turbo mode PSD	5	4	5	4	5	4
36 MHz, Non-Turbo mode PSD	6	4	6	4	6	4
32 MHz, Turbo mode PSD	5	4	5	4	5	4
32 MHz, Non-Turbo mode PSD	5	4	5	4	5	4
28 MHz, Turbo mode PSD	4	3	4	4	4	4
28 MHz, Non-Turbo mode PSD	5	4	5	4	5	4
24 MHz, Turbo mode PSD	4	3	4	4	4	4
24 MHz, Non-Turbo mode PSD	4	3	4	4	4	4
20 MHz and below, Turbo mode PSD	3	3	4	4	4	4
20 MHz and below, Non-Turbo mode PSD	3	3	4	4	4	4

1. V_{DD} of the PSD module
2. “Turbo mode PSD” means that the PSD module is in the faster, Turbo mode (default condition). A PSD module in Non-Turbo mode is slower, but consumes less current. See PSD module section, titled “PLD Non-Turbo mode” for details.

Table 59. T2CON register bit definition

Bit	Symbol	R/W	Definition
1	C/$\overline{T2}$	R,W	**Counter or Timer function select.** When C/$\overline{T2}$ = 0, function is timer, clocked by internal clock. When C/$\overline{T2}$ = 1, function is counter, clocked by signal sampled on external pin, T2.
0	CP/$\overline{RL2}$	R,W	**Capture/Reload.** When CP/$\overline{RL2}$ = 1, capture occurs on negative transition at pin T2X if EXEN2 = 1. When CP/$\overline{RL2}$ = 0, auto-reload occurs when Timer 2 overflows, or on negative transition at pin T2X when EXEN2=1. When RCLK = 1 or TCLK = 1, CP/$\overline{RL2}$ is ignored, and Timer 2 is forced to auto-reload upon Timer 2 overflow

1. The RCLK1 and TCLK1 Bits in the SFR named PCON control UART1, and have the exact same function as RCLK and TCLK.

Table 60. Timer/counter 2 operating modes

Mode	Bits in T2CON SFR				Pin T2X (1)	Remarks	Input clock	
	RCLK or TCLK	CP/$\overline{RL2}$	TR2	EXEN2			Timer, internal	Counter, external (pin T2, P1.0)
16-bit Auto-reload	0	0	1	0	x	reload [RCAP2H, RCAP2L] to [TH2, TL2] upon overflow (up counting)	$f_{OSC}/12$	MAX $f_{OSC}/24$
	0	0	1	1	↓	reload [RCAP2H, RCAP2L] to [TH2, TL2] at falling edge on pin T2X		
16-bit Capture	0	1	1	0	x	16-bit Timer/Counter (up counting)	$f_{OSC}/12$	MAX $f_{OSC}/24$
	0	1	1	1	↓	Capture [TH2, TL2] and store to [RCAP2H, RCAP2L] at falling edge on pin T2X		
Baud Rate Generator	1	x	1	0	x	No overflow interrupt request (TF2)	$f_{OSC}/2$	–
	1	x	1	1	↓	Extra Interrupt on pin T2X, sets TF2		
Off	x	x	0	x	x	Timer 2 stops	–	–

1. ↓ = *falling edge*

Table 79. S1ADR register bit definition

Bit	Symbol	R/W	Function
7:1	SLA[6:0]	R/W	Stores desired 7-bit device address, used when SIOE is in Slave mode.
0	–	–	Not used

23.12 I²C START sample setting (S1SETUP)

The S1SETUP register (*Table 80 on page 132*) determines how many times an I²C bus START condition will be sampled before the SIOE validates the START condition, giving the SIOE the ability to reject noise or illegal transmissions.

Because the minimum duration of an START condition varies with I²C bus speed (f_{SCL}), and also because the UPSD33xx may be operated with a wide variety of frequencies (f_{OSC}), it is necessary to scale the number of samples per START condition based on f_{OSC} and f_{SCL}.

In Slave mode, the SIOE recognizes the beginning of a START condition when it detects a 1-to-0 transition on the SDA bus line while the SCL line is high (see *Figure 38 on page 124*). The SIOE must then validate the START condition by sampling the bus lines to ensure SDA remains low and SCL remains high for a minimum amount of hold time, t_{HLDSTA}. Once validated, the SIOE begins receiving the address byte that follows the START condition.

If the EN_SS Bit (in the S1SETUP register) is not set, then the SIOE will sample only once after detecting the 1-to-0 transition on SDA. This single sample is taken $1/f_{OSC}$ seconds after the initial 1-to-0 transition was detected. However, more samples should be taken to ensure there is a valid START condition.

To take more samples, the SIOE should be initialized such that the EN_SS Bit is set, and a value is written to the SMPL_SET[6:0] field of the S1SETUP register to specify how many samples to take. The goal is to take a good number of samples during the minimum START condition hold time, t_{HLDSTA}, but no so many samples that the bus will be sampled after t_{HLDSTA} expires.

Table 82 describes the relationship between the contents of S1SETUP and the resulting number of I²C bus samples that SIOE will take after detecting the 1-to-0 transition on SDA of a START condition.

Note: ***Important:*** *Keep in mind that the time between samples is always $1/f_{OSC}$.*

The minimum START condition hold time, t_{HLDSTA}, is different for the three common I²C speed categories per *Table 83 on page 133*.

Table 80. S1SETUP: I²C START Condition Sample Setup register (SFR DBh, reset value 00h)

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

Table 87. SPICON1: SPI Interface Control register 1 (SFR D7h, Reset Value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	TEIE	RORIE	TIE	RIE

Table 88. SPICON1 register bit definition

Bit	Symbol	R/W	Definition
7-4	–	–	Reserved
3	TEIE	RW	**Transmission End Interrupt Enable** 0 = Disable Interrupt for Transmission End 1 = Enable Interrupt for Transmission End
2	RORIE	RW	**Receive Overrun Interrupt Enable** 0 = Disable Interrupt for Receive Overrun 1 = Enable Interrupt for Receive Overrun
1	TIE	RW	**Transmission Interrupt Enable** 0 = Disable Interrupt for SPITDR empty 1 = Enable Interrupt for SPITDR empty
0	RIE	RW	**Reception Interrupt Enable** 0 = Disable Interrupt for SPIRDR full 1 = Enable Interrupt for SPIRDR full

Table 89. SPICLKD: SPI Prescaler (Clock Divider) register (SFR D2h, Reset Value 04h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIV128	DIV64	DIV32	DIV16	DIV8	DIV4	–	–

Table 90. SPICLKD register bit definition

Bit	Symbol	R/W	Definition
7	DIV128	RW	0 = No division 1 = Divide f_{OSC} clock by 128
6	DIV64	RW	0 = No division 1 = Divide f_{OSC} clock by 64
5	DIV32	RW	0 = No division 1 = Divide f_{OSC} clock by 32
4	DIV16	RW	0 = No division 1 = Divide f_{OSC} clock by 16
3	DIV8	RW	0 = No division 1 = Divide f_{OSC} clock by 8
2	DIV4	RW	0 = No division 1 = Divide f_{OSC} clock by 4
1-0	Not Used	–	

Figure 45. 10-Bit ADC

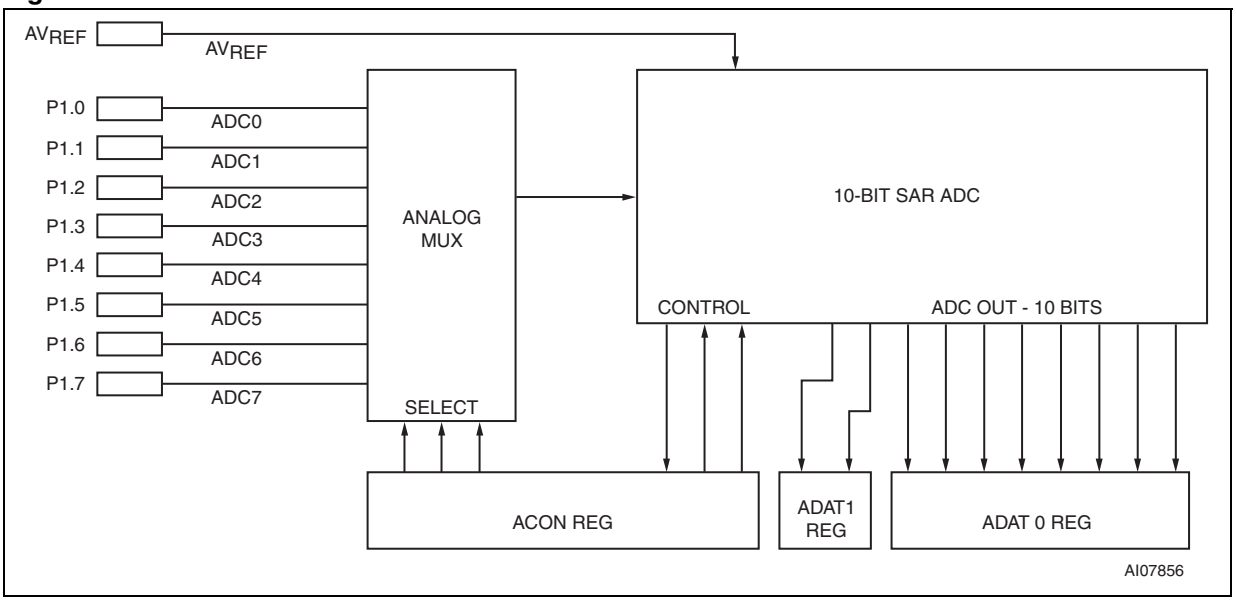


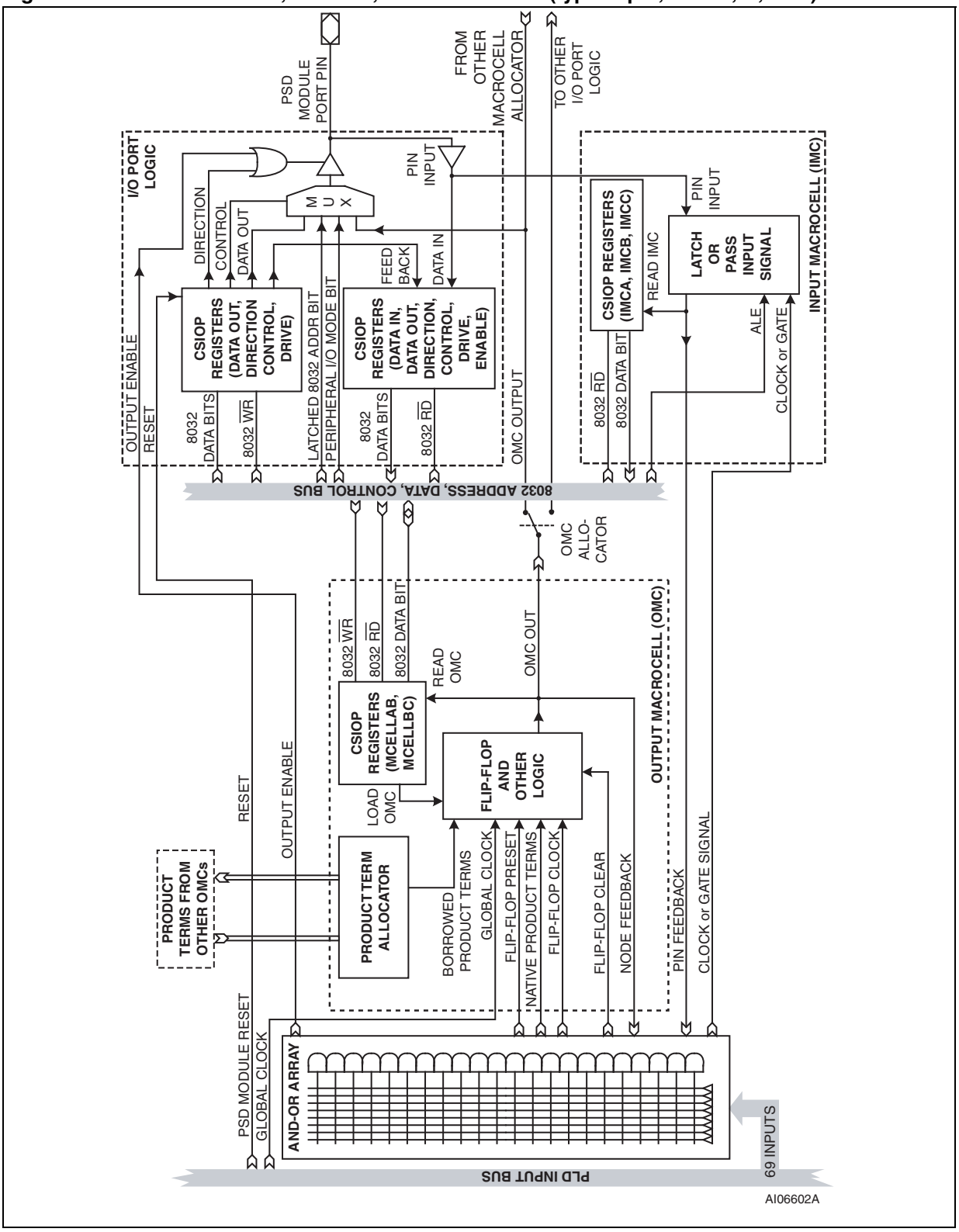
Table 93. ACON register (SFR 97h, Reset Value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AINTF	AINTEN	ADEN	ADS2	ADS1	ADS0	ADST	ADSF

Table 94. ACON register bit definition

Bit	Symbol	Function
7	AINTF	**ADC Interrupt flag**. This bit must be cleared with software. 0 = No interrupt request 1 = The AINTF flag is set when ADSF goes from '0' to '1.' Interrupts CPU when both AINTF and AINTEN are set to '1.'
6	AINTEN	**ADC Interrupt Enable** 0 = ADC interrupt is disabled 1 = ADC interrupt is enabled
5	ADEN	**ADC Enable Bit** 0 = ADC shut off and consumes no operating current 1 = Enable ADC. After ADC is enabled, 16ms of calibration is needed before ADST Bit is set.

Figure 64. GPLD: one OMC, one IMC, and one I/O Port (typical pin, Port A, B, or 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 ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, 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 KΩ), 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 ($\overline{RST}$) 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.

27.4.52 Automatic Power-down (APD)

The APD unit shown in *Figure 62 on page 195* puts the PSD module into power-down mode by monitoring the activity of the 8032 Address Latch Enable (ALE) signal. If the APD unit is enabled by writing a logic '1' to Bit 1 of the csiop PMMR0 register, and if ALE signal activity has stopped (8032 in sleep mode), then the four-bit APD counter starts counting up. If the ALE signal remains inactive for 15 clock periods of the CLKIN signal (pin PD1), then the APD counter will reach maximum count and the power-down indicator signal (PDN) goes to logic '1' forcing the PSD module into power-down mode. During this time, all buffers on the PSD module for 8032 address and data signals are disabled in silicon, preventing the PSD module memories from waking up from standby mode, even if noise or other devices are driving the address lines. The PLDs will also stay in standby mode if the PLDs are in non-Turbo mode and if all other PLD inputs (non-address signals) are static.

However, if the ALE signal has a transition before the APD counter reaches max count, the APD counter is cleared to zero and the PDN signal will not go active, preventing power-down mode. To prevent unwanted APD timeouts during normal 8032 operation (not sleeping), it is important to choose a clock frequency for CLKIN that will NOT produce 15 or more pulses within the longest period between ALE transitions. A 32768 Hz clock signal is quite often an ideal frequency for CLKIN and APD, and this frequency is often available on external supervisor or real-time clock devices.

The "PDN" power-down indicator signal is available to the PLD input bus to use in any PLD equations if desired. The user may want to send this signal as a PLD output to an external device to indicate the PSD module is in power-down mode. PSDsoft Express automatically includes the "PDN" signal in the DPLD chip select equations for FSx, CSBOOTx, RS0, and CSIOP.

The following should be kept in mind when the PSD module is in power-down mode:

- 8032 address and data bus signals are blocked from all memories and both PLDs.
- The PSD module comes out of power-down mode when: ALE starts pulsing again, or the $\overline{\text{CSI}}$ input on pin PD2 transitions from logic '1' to logic '0,' or the PSD module reset signal, $\overline{\text{RST}}$, transitions from logic '0' to logic '1.'
- Various signals can be blocked (prior to power-down mode) from entering the PLDs by using "blocking bits" in csiop PMMR registers.
- All memories enter standby mode, and the state of the PLDs and I/O Ports are unchanged (if no PLD inputs change). *Table 158 on page 233* shows the effects of power-down mode on I/O pins while in various operating modes.
- The 8032 Ports 1,3, and 4 on the MCU module are not affected at all by power-down mode in the PSD module.
- Power-down standby current given in the AC specifications for PSD module assume there are no transitions on any unblocked PLD input, and there are no output pins driving any loads.

The APD counter will count whenever Bit 1 of csiop PMMR0 register is set to logic '1,' and when the ALE signal is steady at either logic '1' or logic '0' (not transitioning). *Figure 78 on page 229* shows the flow leading up to power-down mode. The only action required in PSDsoft Express to enable APD mode is to select the pin function "Common Clock Input, CLKIN" before programming with JTAG.

Table 177. CPLD macrocell synchronous Clock mode timing (3 V PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate(1)	Unit
f_{MAX}	Maximum frequency external feedback	$1/(t_S+t_{CO})$		23.2				MHz
	Maximum frequency internal feedback (f_{CNT})	$1/(t_S+t_{CO}-10)$		30.3				MHz
	Maximum frequency pipelined data	$1/(t_{CH}+t_{CL})$		40.0				MHz
t_S	Input setup time		20		+ 4	+ 15		ns
t_H	Input hold time		0					ns
t_{CH}	Clock high time	Clock Input	15					ns
t_{CL}	Clock low time	Clock Input	10					ns
t_{CO}	Clock to output delay	Clock Input		23			– 6	ns
t_{ARD}	CPLD array delay	Any macrocell		20	+ 4			ns
t_{MIN}	Minimum clock period(2)	$t_{CH}+t_{CL}$	25					ns

1. Fast Slew Rate output available on PA3-PA0, PB3-PB0, and PD2-PD1. Decrement times by given amount.
2. CLKIN (PD1) $t_{CLCL} = t_{CH} + t_{CL}$.

Figure 91. Asynchronous $\overline{\text{RESET}}$ / Preset

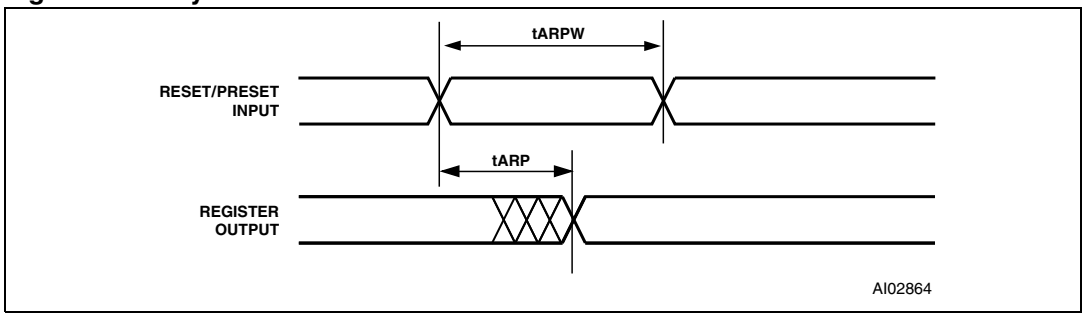


Figure 92. Asynchronous Clock mode timing (product term clock)

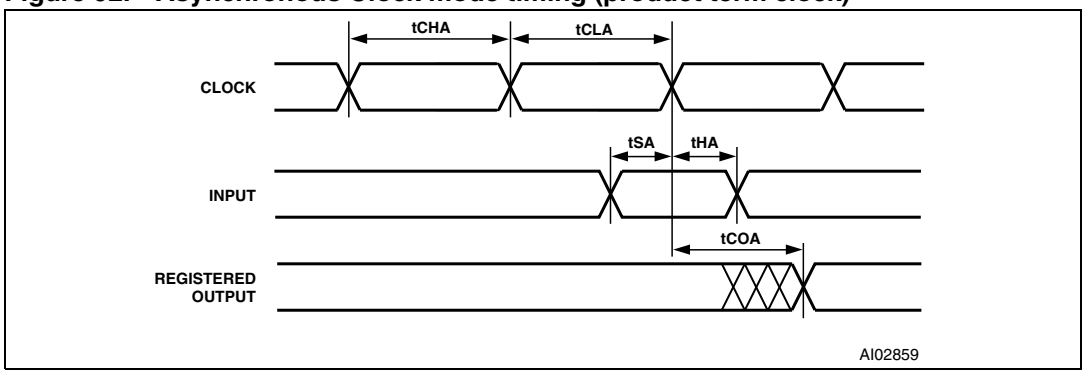


Table 178. CPLD macrocell asynchronous Clock mode timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate	Unit
f_{MAXA}	Maximum frequency external feedback	$1/(t_{SA}+t_{COA})$		38.4				MHz
	Maximum frequency internal feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		62.5				MHz
	Maximum frequency pipelined data	$1/(t_{CHA}+t_{CLA})$		71.4				MHz
t_{SA}	Input setup time		7		+ 2	+ 10		ns
t_{HA}	Input hold time		8					ns
t_{CHA}	Clock input high time		9			+ 10		ns
t_{CLA}	Clock input low time		9			+ 10		ns
t_{COA}	Clock to output delay			21		+ 10	– 2	ns
t_{ARDA}	CPLD array delay	Any macrocell		11	+ 2			ns
t_{MINA}	Minimum clock period	$1/f_{CNTA}$	16					ns

Table 179. CPLD macrocell asynchronous Clock mode timing (3timeV PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate	Unit
f_{MAXA}	Maximum frequency external feedback	$1/(t_{SA}+t_{COA})$		21.7				MHz
	Maximum frequency internal feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		27.8				MHz
	Maximum frequency pipelined data	$1/(t_{CHA}+t_{CLA})$		33.3				MHz
t_{SA}	Input setup time		10		+ 4	+ 15		ns
t_{HA}	Input hold time		12					ns
t_{CHA}	Clock high time		17			+ 15		ns
t_{CLA}	Clock low time		13			+ 15		ns
t_{COA}	Clock to output delay			31		+ 15	– 6	ns
t_{ARD}	CPLD array delay	Any macrocell		20	+ 4			ns
t_{MINA}	Minimum clock period	$1/f_{CNTA}$	36					ns