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	288KB (288K 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/upsd3334dv-40u6

27.4.38 PLD I/O mode . . . 212
27.4.39 Latched address output mode . . . 214
27.4.40 Peripheral I/O mode . . . 215
27.4.41 JTAG ISP mode . . . 216
27.4.42 Other port capabilities . . . 216
27.4.43 Port pin drive options . . . 216
27.4.44 Drive select registers . . . 216
27.4.45 Enable Out registers . . . 216
27.4.46 Individual port structures . . . 218
27.4.47 Port A structure . . . 218
27.4.48 Port B structure . . . 219
27.4.49 Port C structure . . . 220
27.4.50 Port D structure . . . 222
27.4.51 Power management . . . 224
27.4.52 Automatic Power-down (APD) . . . 227
27.4.53 Forced Power-down (FDP) . . . 228
27.4.54 Chip Select Input (CSI) . . . 230
27.4.55 PLD non-turbo mode . . . 230
27.4.56 PLD current consumption . . . 230
27.4.57 Turbo mode current consumption . . . 230
27.4.58 Non-turbo mode current consumption . . . 231
27.4.59 PLD blocking bits . . . 231
27.4.60 Blocking 8032 bus control signals . . . 231
27.4.61 Blocking common clock, CLKIN . . . 232
27.5 PSD module reset conditions . . . 232
27.5.1 JTAG ISP and JTAG debug . . . 233
27.5.2 JTAG chaining inside the package . . . 234
27.5.3 In-system programming . . . 235
27.5.4 4-pin JTAG ISP (default) . . . 236
27.5.5 6-pin JTAG ISP (optional) . . . 237
27.5.6 Recommended JTAG connector . . . 239
27.5.7 Chaining UPSD33xx devices . . . 239
27.5.8 Debugging the 8032 MCU module . . . 240
27.5.9 JTAG security setting . . . 241
27.5.10 Initial delivery state . . . 241

28 AC/DC parameters . . . 242

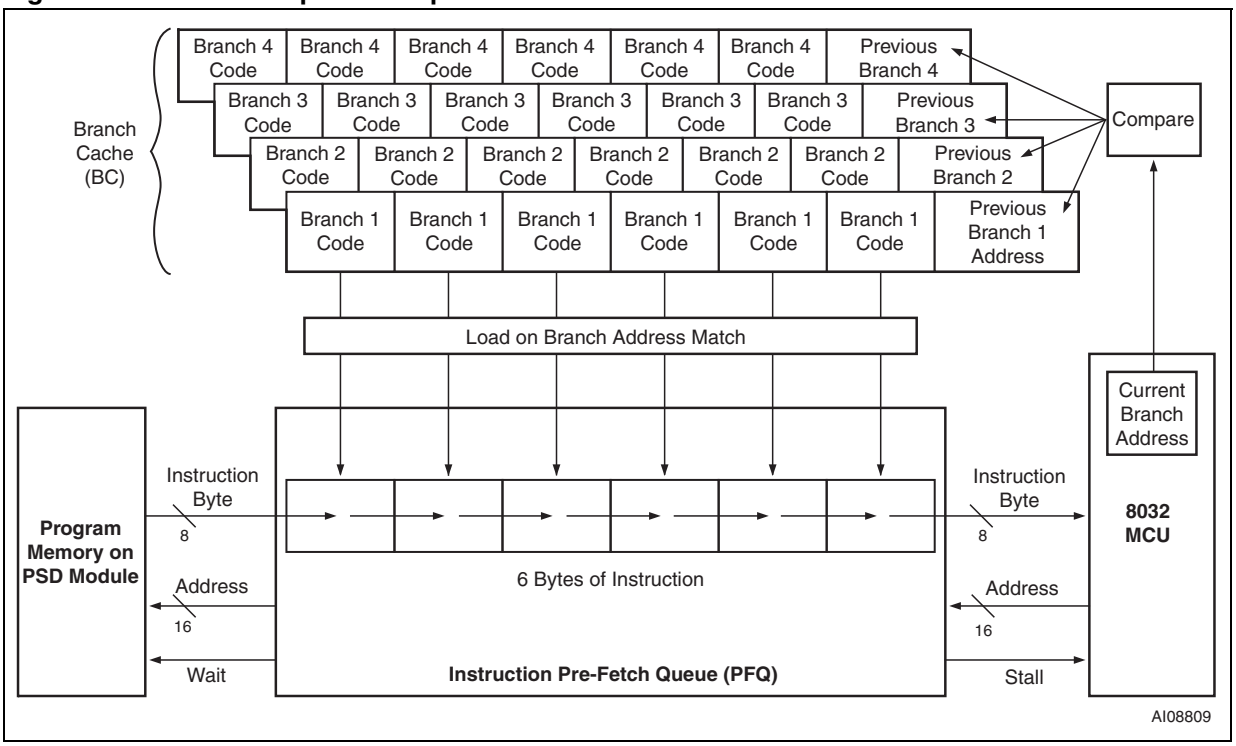
List of tables

Table 1. Device summary . . . 1
Table 2. Pin definitions . . . 22
Table 3. Port type and voltage source combinations . . . 27
Table 4. Register bank select addresses . . . 39
Table 5. SFR memory map with direct address and reset value . . . 41
Table 6. Arithmetic instruction set . . . 50
Table 7. Logical instruction set . . . 51
Table 8. Data transfer instruction set . . . 52
Table 9. Boolean variable manipulation instruction set . . . 53
Table 10. Program branching instruction set . . . 54
Table 11. Miscellaneous instruction set . . . 55
Table 12. Notes on instruction set and addressing modes . . . 55
Table 13. DPTC: Data Pointer Control register (SFR 85h, reset value 00h) . . . 56
Table 14. DPTC register bit definition . . . 56
Table 15. DPTM: Data Pointer Mode register (SFR 86h, reset value 00h) . . . 57
Table 16. DPTM register bit definition . . . 57
Table 17. 8051 assembly code example . . . 58
Table 18. Interrupt summary . . . 62
Table 19. IE: Interrupt Enable register (SFR A8h, reset value 00h) . . . 65
Table 20. IE register bit definition . . . 65
Table 21. IEA: Interrupt Enable Addition register (SFR A7h, reset value 00h) . . . 65
Table 22. IEA register bit definition . . . 65
Table 23. IP: Interrupt Priority register (SFR B8h, reset value 00h) . . . 66
Table 24. IP register bit definition . . . 66
Table 25. IPA: Interrupt Priority Addition register (SFR B7h, reset value 00h) . . . 66
Table 26. IPA register bit definition . . . 66
Table 27. CCON0: Clock Control register (SFR F9h, reset value 10h) . . . 69
Table 28. CCON0 register bit definition . . . 69
Table 29. MCU module port and peripheral status during reduced power modes . . . 72
Table 30. State of 8032 MCU bus Signals during Power-down and Idle modes . . . 72
Table 31. PCON: Power Control register (SFR 87h, reset value 00h) . . . 72
Table 32. PCON register bit definition . . . 72
Table 33. P1: I/O Port 1 register (SFR 90h, reset value FFh) . . . 80
Table 34. P1 register bit definition . . . 80
Table 35. P3: I/O Port 3 register (SFR B0h, reset value FFh) . . . 81
Table 36. P3 register bit definition . . . 81
Table 37. P4: I/O Port 4 register (SFR C0h, reset value FFh) . . . 81
Table 38. P4 register bit definition . . . 81
Table 39. P3SFS: Port 3 Special Function Select register (SFR 91h, reset value 00h) . . . 83
Table 40. P3SFS register bit definition . . . 83
Table 41. P1SFS0: Port 1 Special Function Select 0 register (SFR 8Eh, reset value 00h) . . . 83
Table 42. P1SFS1: Port 1 Special Function Select 1 register (SFR 8Fh, reset value 00h) . . . 83
Table 43. P1SFS0 and P1SFS1 details . . . 83
Table 44. P4SFS0: Port 4 Special Function Select 0 register (SFR 92h, reset value 00h) . . . 84
Table 45. P4SFS1: Port 4 Special Function Select 1 register (SFR 93h, reset value 00h) . . . 84
Table 46. P4SFS0 and P4SFS1 details . . . 84
Table 47. BUSCON: Bus Control register (SFR 9Dh, reset value EBh) . . . 87
Table 48. BUSCON register bit definition . . . 87

Table 2. Pin definitions (continued)

Port pin	Signal name	80-Pin num.	52-Pin num.(1)	In/Out	Function		
					Basic	Alternate 1	Alternate 2
P1.4	SPICLK ADC4	59	38	I/O	General I/O port pin	SPI Clock Out (SPICLK)	ADC Channel 4 input (ADC4)
P1.5	SPIRxD ADC6	60	39	I/O	General I/O port pin	SPI Receive (SPIRxD)	ADC Channel 5 input (ADC5)
P1.6	SPITXD ADC6	61	40	I/O	General I/O port pin	SPI Transmit (SPITxD)	ADC Channel 6 input (ADC6)
P1.7	SPISEL ADC7	64	41	I/O	General I/O port pin	SPI Slave Select ($\overline{\text{SPISEL}}$)	ADC Channel 7 input (ADC7)
P3.0	RxD0	75	23	I/O	General I/O port pin	UART0 Receive (RxD0)	
P3.1	TXD0	77	24	I/O	General I/O port pin	UART0 Transmit (TxD0)	
P3.2	EXINT0 TGO	79	25	I/O	General I/O port pin	Interrupt 0 input (EXTINT0)/Timer 0 gate control (TG0)	
P3.3	INT1	2	26	I/O	General I/O port pin	Interrupt 1 input (EXTINT1)/Timer 1 gate control (TG1)	
P3.4	C0	40	27	I/O	General I/O port pin	Counter 0 input (C0)	
P3.5	C1	42	28	I/O	General I/O port pin	Counter 1 input (C1)	
P3.6	SDA	44	29	I/O	General I/O port pin	I^2C Bus serial data (I^2CSDA)	
P3.7	SCL	46	30	I/O	General I/O port pin	I^2C Bus clock (I^2CSCL)	
P4.0	T2 TCM0	33	22	I/O	General I/O port pin	Program Counter Array0 PCA0-TCM0	Timer 2 Count input (T2)
P4.1	T2X TCM1	31	21	I/O	General I/O port pin	PCA0-TCM1	Timer 2 Trigger input (T2X)
P4.2	RXD1 TCM2	30	20	I/O	General I/O port pin	PCA0-TCM2	UART1 or IrDA Receive (RxD1)
P4.3	TXD1 PCACLK0	27	18	I/O	General I/O port pin	PCACLK0	UART1 or IrDA Transmit (TxD1)
P4.4	SPICLK TCM3	25	17	I/O	General I/O port pin	Program Counter Array1 PCA1-TCM3	SPI Clock Out (SPICLK)
P4.5	SPIRXD TCM4	23	16	I/O	General I/O port pin	PCA1-TCM4	SPI Receive (SPIRxD)

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 26. IPA register bit definition (continued)

Bit	Symbol	R/W	Function
1[1]	PI^{2}C	R,W	**I^{2}C Interrupt priority level**
0	–	–	Reserved

1. 1 = Assigns high priority level, 0 = Assigns low priority level

19.2 Low V_{CC} voltage detect, LVD

An internal reset is generated by the LVD circuit when V_{CC} drops below the reset threshold, V_{LV_THRESH}. After V_{CC} returns to the reset threshold, the MCU_RESET signal will remain asserted for t_{RST_ACTV} before it is released. The LVD circuit is always enabled (cannot be disabled by SFR), even in Idle mode and Power-down mode. The LVD input has a voltage hysteresis of V_{RST_HYS} and will reject voltage spikes less than a duration of t_{RST_FIL}.

Note: ***Important:*** *The LVD voltage threshold is V_{LV_THRESH}, suitable for monitoring both the 3.3 V V_{CC} supply on the MCU module and the 3.3 V V_{DD} supply on the PSD module for 3.3 V UPSD33xxV devices, since these supplies are one in the same on the circuit board.*

However, for 5 V UPSD33xx devices, V_{LV_THRESH} is not suitable for monitoring the 5 V V_{DD} voltage supply (V_{LV_THRESH} is too low), but good for monitoring the 3.3 V V_{CC} supply. In the case of 5 V UPSD33xx devices, an external means is required to monitor the separate 5 V V_{DD} supply, if desired.

19.3 Power-up reset

At power up, the internal reset generated by the LVD circuit is latched as a logic '1' in the POR bit of the SFR named PCON (*Table 31 on page 72*). Software can read this bit to determine whether the last MCU reset was the result of a power up (cold reset) or a reset from some other condition (warm reset). This bit must be cleared with software.

19.4 JTAG debug Reset

The JTAG debug unit can generate a reset for debugging purposes. This reset source is also available when the MCU is in Idle mode and Power-down mode (the JTAG debugger can be used to exit these modes).

19.5 Watchdog timer (WDT)

When enabled, the WDT will generate a reset whenever it overflows. Firmware that is behaving correctly will periodically clear the WDT before it overflows. Run-away firmware will not be able to clear the WDT, and a reset will be generated.

By default, the WDT is disabled after each reset.

Note: The WDT is not active during Idle mode or Power-down mode.

There are two SFRs that control the WDT, they are WDKEY (*Table 50 on page 92*) and WDRST (*Table 52 on page 92*).

If WDKEY contains 55h, the WDT is disabled. Any value other than 55h in WDKEY will enable the WDT. By default, after any reset condition, WDKEY is automatically loaded with 55h, disabling the WDT. It is the responsibility of initialization firmware to write some value other than 55h to WDKEY after each reset if the WDT is to be used.

The WDT consists of a 24-bit up-counter (*Figure 20*), whose initial count is 000000h by default after every reset. The most significant byte of this counter is controlled by the SFR, WDRST. After being enabled by WDKEY, the 24-bit count is increased by 1 for each MCU machine cycle. When the count overflows beyond FFFFFh (2^{24} MCU machine cycles), a reset is issued and the WDT is automatically disabled (WDKEY = 55h again).

Table 73. Selection of the SCL frequency in Master mode based on f_{OSC} examples

CR2	CR1	CR0	f_{OSC} divided by:	Bit rate (kHz) @ f_{OSC}			
				12 MHz f_{OSC}	24 MHz f_{OSC}	36 MHz f_{OSC}	40 MHz f_{OSC}
0	0	0	32	375	750	X(1)	X(1)
0	0	1	48	250	500	750	833
0	1	0	60	200	400	600	666
0	1	1	120	100	200	300	333
1	0	0	240	50	100	150	166
1	0	1	480	25	50	75	83
1	1	0	960	12.5	25	37.5	41
1	1	1	1920	6.25	12.5	18.75	20

1. These values are beyond the bit rate supported by UPSD33xx.

23.9 I²C Interface Status register (S1STA)

The S1STA register provides status regarding immediate activity and the current state of operation on the I²C bus. All bits in this register are read-only except bit 5, INTR, which is the interrupt flag.

23.9.1 Interrupt conditions

If the I²C interrupt is enabled (EI²C = 1 in SFR named IEA, and EA =1 in SFR named IE), and the SIOE is initialized, then an interrupt is automatically generated when any one of the following five events occur:

- When the SIOE receives an address that matches the contents of the SFR, S1ADR. Requirements: SIOE is in Slave mode, and bit AA = 1 in the SFR S1CON.
- When the SIOE receives General Call address. Requirements: SIOE is in Slave mode, bit AA = 1 in the SFR S1CON
- When a complete data byte has been received or transmitted by the SIOE while in Master mode. The interrupt will occur even if the Master looses arbitration.
- When a complete data byte has been received or transmitted by the SIOE while in selected Slave mode.
- A STOP condition on the bus has been recognized by the SIOE while in selected Slave mode.

Selected Slave mode means the device address sent by the Master device at the beginning of the current data transfer matched the address stored in the S1ADR register.

If the I²C interrupt is not enabled, the MCU may poll the INTR flag in S1STA.

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 $\overline{\text{SPISEL}}$

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	–	

Table 104. PCACON0 register bit definition (continued)

Bit	Symbol	Function
5	EOVFI	1 = Enable Counter Overflow Interrupt if overflow flag (OVF) is set
4	PCAIDLE	0 = PCA operates when CPU is in Idle mode 1 = PCA stops running when CPU is in Idle mode
3	–	Reserved
2	10B_PWM	0 = Select 16-bit PWM 1 = Select 10-bit PWM
1-0	CLK_SEL [1:0]	00 Select Prescaler clock as Counter clock 01 Select Timer 0 Overflow 10 Select External Clock pin (P4.3 for PCA0) (MAX clock rate = $f_{OSC}/4$)

Table 105. PCA1 Control register PCACON1 (SFR 0BCh, Reset Value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	EN_PCA	EOVFI	PCAIDLE	–	10B_PWM	CLK_SEL[1:0]	

Table 106. PCACON1 register bit definition

Bit	Symbol	Function
6	EN_PCA	0 = PCA counter is disabled 1 = PCA counter is enabled EN_PCA Counter Run Control Bit. Set with software to turn the PCA counter on. Must be cleared with software to turn the PCA counter off.
5	EOVFI	1 = Enable Counter Overflow Interrupt if overflow flag (OVF) is set
4	PCAIDLE	0 = PCA operates when CPU is in Idle mode 1 = PCA stops running when CPU is in Idle mode
3	–	Reserved
2	10B_PWM	0 = Select 16-bit PWM 1 = Select 10-bit PWM
1-0	CLK_SEL [1:0]	00 Select Prescaler clock as Counter clock 01 Select Timer 0 Overflow 10 Select External Clock pin (P4.7 for PCA1) (MAX clock rate = $f_{OSC}/4$)

27.2.5 Alternative mapping schemes

Here are more possible memory maps for the UPSD3333.

Note: Mapping examples would be slightly different for UPSD3312, UPSD3334, and UPSD3354 because of the different sizes of individual Flash memory sectors and SRAM as defined in Table 119 on page 193.

- ***Figure 54 on page 174*** Place the larger main Flash memory into program space, but split the secondary Flash in half, placing two of it's sectors into XDATA space and remaining two sectors into program space. This method allows the designer to put IAP code (or boot code) into two sectors of secondary Flash in program space, and use the other two secondary Flash sectors for data storage, such as EEPROM emulation in XDATA space.
- ***Figure 55 on page 175*** Place both the Main and secondary Flash memories into program space for maximum code storage, with no Flash memory in XDATA space.

Figure 54. Mapping: split second Flash in half

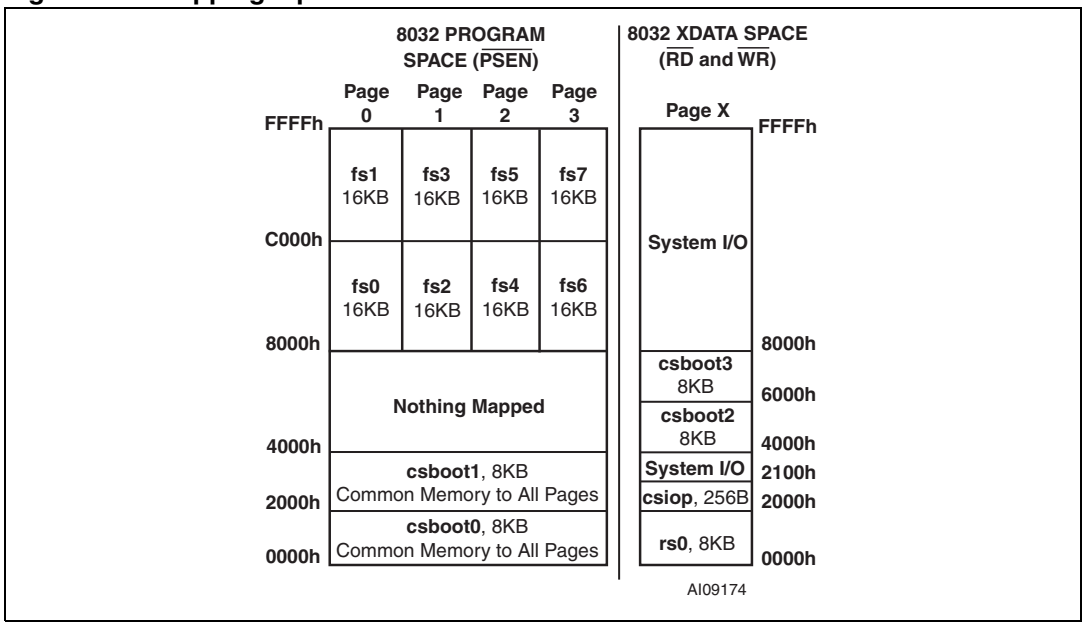
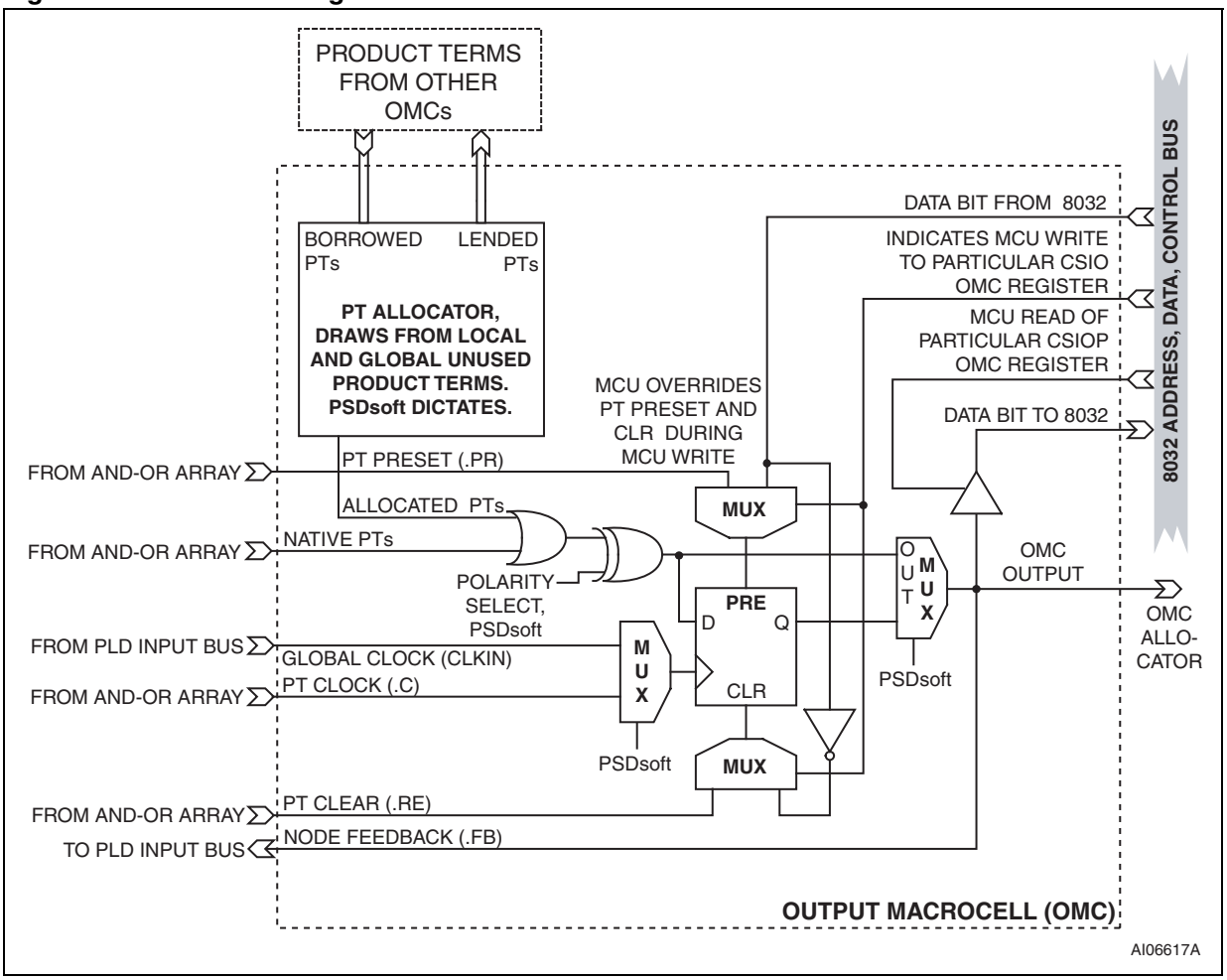


Figure 65. Detail of a Single OMC



27.4.29 OMC allocator

Outputs of the 16 OMCs can be routed to a combination of pins on Port A (80-pin devices only), Port B, or Port C as shown in *Figure 66*. OMCs are routed to port pins automatically after specifying pin numbers in PSDsoft Express. Routing can occur on a bit-by-bit basis, spitting OMC assignment between the ports. However, one OMC can be routed to one only port pin, not both ports.

27.4.30 Product term allocator

Each OMC has a Product Term Allocator as shown in *Figure 65 on page 201*. PSDsoft Express uses PT Allocators to give and take product terms to and from other OMCs to fit a logic design into the available silicon resources. This happens automatically in PSDsoft Express, but understanding how PT allocation works will help the user if the logic design does not “fit,” in which case the user may try selecting a different pin or different OMC for the

Table 122. OMC port and data bit assignments

OMC	Port assignment (1)(2)	Native Product terms from AND-OR array	Maximum borrowed product terms	Data bit on 8032 data bus for loading or reading OMC
MCELLAB0	Port A0 or B0	3	6	D0
MCELLAB1	Port A1 or B1	3	6	D1
MCELLAB2	Port A2 or B2	3	6	D2
MCELLAB3	Port A3 or B3	3	6	D3
MCELLAB4	Port A4 or B4	3	6	D4
MCELLAB5	Port A5 or B5	3	6	D5
MCELLAB6	Port A6 or B6	3	6	D6
MCELLAB7	Port A7 or B7	3	6	D7
MCELLBC0	Port B0	4	5	D0
MCELLBC1	Port B1	4	5	D1
MCELLBC2	Port B or C2	4	5	D2
MCELLBC3	Port B3 or C3	4	5	D3
MCELLBC4	Port B4 or C4	4	6	D4
MCELLBC5	Port B5	4	6	D5
MCELLBC6	Port B6	4	6	D6
MCELLBC7	Port B7 orC7	4	6	D7

1. MCELLAB0-MCELLAB7 can be output to Port A pins only on 80-pin devices. Port A is not available on 52-pin devices
2. Port pins PC0, PC1, PC5, and PC6 are dedicated JTAG pins and are not available as outputs for MCELLBC 0, 1, 5, or 6

27.4.31 Loading and reading OMCs

Each of the two OMC groups (eight OMCs each) occupies a byte in csiop space, named MCELLAB and MCELLBC (see *Table 123* and *Table 124 on page 204*). When the 8032 writes or reads these two OMC registers in csiop it is accessing each of the OMCs through it's 8-bit data bus, with the bit assignment shown in *Table 122 on page 203*. Sometimes it is important to know the bit assignment when the user builds GPLD logic that is accessed by the 8032. For example, the user may create a 4-bit counter that must be loaded and read by the 8032, so the user must know which nibble in the corresponding csiop OMC register the firmware must access. The fitter report generated by PSDsoft Express will indicate how it assigned the OMCs and data bus bits to the logic. The user can optionally force PSDsoft Express to assign logic to specific OMCs and data bus bits if desired by using the 'PROPERTY' statement in PSDsoft Express. Please see the PSDsoft Express User's Manual for more information on OMC assignments.

Loading the OMC flip-flops with data from the 8032 takes priority over the PLD logic functions. As such, the preset, clear, and clock inputs to the flip-flop can be asynchronously overridden when the 8032 writes to the csiop registers to load the individual OMCs.

Table 131. Port configuration setting requirements

Port operating mode	Required action in PSDsoft Express to configure each pin	Value that 8032 writes to csiop Control register at run-time	Value that 8032 writes to csiop Direction register at run-time	Value that 8032 writes to bit 7 (PIO_EN) of csiop VM register at run-time
MCU I/O	Choose the MCU I/O function and declare the pin name	Logic '0' (default)	Logic 1 = Out of UPSD Logic 0 = Into UPSD	N/A
PLD I/O	Choose the PLD function type, declare pin name, and specify logic equation(s)	N/A	Direction register has no effect on a pin if pin is driven from OMC output	N/A
Latched Address Output	Choose Latched Address Out function, declare pin name	Logic '1'	Logic '1' Only	N/A
Peripheral I/O	Choose Peripheral I/O mode function and specify address range in DPLD for PSELx	N/A	N/A	PIO_EN Bit = Logic 1 (default is '0')
4-PIN JTAG ISP	No action required in PSDsoft to get 4-pin JTAG. By default TDO, TDI, TCK, TMS are dedicated JTAG functions.	N/A	N/A	N/A
6-PIN JTAG ISP (faster programming)	Choose JTAG TSTAT function for pin PC3 and JTAG $\overline{\text{TERR}}$ function for pin PC4.	N/A	N/A	N/A

27.4.37 MCU I/O mode

In MCU I/O mode, the 8032 on the MCU module expands its own I/O by using the I/O Ports on the PSD module. The 8032 can read PSD module I/O pins, set the direction of the I/O pins, and change the output state of I/O pins by accessing the Data In, Direction, and Data Out csiop registers respectively at run-time.

To implement MCU I/O mode, each desired pin is specified in PSDsoft Express as *MCU I/O* function and given a pin name. Then 8032 firmware is written to set the Direction bit for each corresponding pin during initialization routines (0 = In, 1 = Out of the chip), then the 8032 firmware simply reads the corresponding Data In register to determine the state of an I/O pin, or writes to a Data Out register to set the state of a pin. The Direction of each pin may be changed dynamically by the 8032 if desired. A mixture of input and output pins within a single port is allowed. *Figure 68 on page 208* shows the Data In, Data Out, and Direction signal paths.

The Data In registers are defined in *Table 132* to *Table 135 on page 210*. The Data Out registers are defined in *Table 136* to *Table 139 on page 211*. The Direction registers are defined in *Table 140* to *Table 27.4.38 on page 212*.

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.

27.4.53 Forced Power-down (FDP)

An alternative to APD is FPD. The resulting power-savings is the same, but the PDN signal in *Figure 77 on page 229* is set and Power-down mode is entered immediately when firmware sets the FORCE_PD Bit to logic '1' in the csiop *register* PMMR3 (Bit 1). FPD will override APD counter activity when FORCE_PD is set. No external clock source for the APD counter is needed. The FORCE_PD Bit is cleared only by a reset condition.

Caution must be used when implementing FPD because code memory goes off-line as soon as PSD module Power-down mode is entered, leaving the MCU with no instruction stream to execute.

The MCU module must put itself into Power-down mode after it puts the PSD module into Power-down mode. How can it do this if code memory goes off-line? The answer is the Pre-Fetch Queue (PFQ) in the MCU module. By using the instruction scheme shown in the 8051 assembly code example in *Table 157 on page 228*, the PFQ will be loaded with the final instructions to command the MCU module to Power-down mode after the PDS module goes to Power-down mode. In this case, even though the code memory goes off-line in the PSD module, the last few MCU instruction are sourced from the PFQ.

Table 157. Forced Power-down example

```
PDOWN:  ANL   A8h, #7Fh      ; disable all interrupts
        ORL   9Dh, #C0h      ; ensure PFQ and BC are enabled
        MOV   DPTR, #xxC7    ; load XDATA pointer to select PMMR3 register (xx = base
                             ; address of csiop registers)
        CLR   A              ; clear A
        JMP   LOOP           ; first loop - fill PFQ/BQ with Power-down instructions
        NOP                  ; second loop - fetch code from PFQ/BC and set Power-
                             ; Down bits for PSD module and then MCU module
LOOP:   MOVX  @DPTR, A       ; set FORCE_PD Bit in PMMR3 in PSD module in second
                             ; loop
        MOV   87h, A         ; set PD Bit in PCON register in MCU module in second
                             ; loop
        MOV   A, #02h        ; set power-down bit in the A register, but not in PMMR3 or
                             ; PCON yet in first loop
        JMP   LOOP           ; UPSD enters into Power-down mode in second loop
```

27.4.54 Chip Select Input ($\overline{CSI}$)

Pin PD2 of Port D can optionally be configured in PSDsoft Express as the PSD module Chip Select Input, $\overline{CSI}$, which is an active-low logic input. By default, pin PD2 does not have the $\overline{CSI}$ function.

When the $\overline{CSI}$ function is specified in PSDsoft Express, the $\overline{CSI}$ signal is automatically included in DPLD chip select equations for FSx, CSBOOTx, RS0, and CSIOP. When the $\overline{CSI}$ pin is driven to logic ’0’ from an external device, all of these memories will be available for READ and WRITE operations. When $\overline{CSI}$ is driven to logic '1,' none of these memories are available for selection, regardless of the address activity from the 8032, reducing power consumption. The state of the PLD and port I/O pins are not changed when $\overline{CSI}$ goes to logic ’1’ (disabled).

27.4.55 PLD non-turbo mode

The power consumption and speed of the PLDs are controlled by the Turbo Bit (Bit 3) in the csiop PMMR0 register. By setting this bit to logic '1,' the Turbo mode is turned off and both PLDs consume only standby current when ALL PLD inputs have no transitions for an extended time (65ns for 5 V devices, 100ns for 3.3 V devices), significantly reducing current consumption. The PLDs will latch their outputs and go to standby, drawing very little current. When Turbo mode is off, PLD propagation delay time is increased as shown in the AC specifications for the PSD module. Since this additional propagation delay also effects the DPLD, the response time of the memories on the PSD module is also lengthened by that same amount of time. If Turbo mode is off, the user should add an additional wait state to the 8032 BUSCON SFR register if the 8032 clock frequency is higher that a particular value. Please refer to *Table 49 on page 88* in the MCU module section.

The default state of the Turbo Bit is logic '0,' meaning Turbo mode is on by default (after power-up and reset conditions) until it is turned off by the 8032 writing to PMMR0.

27.4.56 PLD current consumption

Figure 84 and *Figure 85 on page 243* (5 V and 3.3 V devices respectively) show the relationship between PLD current consumption and the composite frequency of all the transitions on PLD inputs, indicating that a higher input frequency results in higher current consumption.

Current consumption of the PLDs have a DC component and an AC component. Both need to be considered when calculating current consumption for a specific PLD design. When Turbo mode is on, there is a linear relationship between current and frequency, and there is a substantial DC current component consumed by the PSD module when there are no transitions on PLD inputs (composite frequency is zero). The magnitude of this DC current component is directly proportional to how many product terms are used in the equations of both PLDs. PSDsoft Express generates a “fitter” report that specifies how many product terms were used in a design out of a total of 186 available product terms. *Figure 84* and *Figure 85 on page 243* both give two examples, one with 100% of the 186 product terms used, and another with 25% of the 186 product terms used.

27.4.57 Turbo mode current consumption

To determine the AC current component of the specific PLD design with Turbo mode on, the user will have to interpolate from the graph, given the number of product terms specified in the fitter report, and the estimated composite frequency of PLD input signal transitions. For

module (Flash memory and PLD) may be programmed with JTAG ISP, but only the Flash memories may be programmed using IAP.

27.5.2 JTAG chaining inside the package

JTAG protocol allows serial “chaining” of more than one device in a JTAG chain. The UPSD33xx is assembled with a stacked die process combining the PSD module (one die) and the MCU module (the other die). These two die are chained together within the UPSD33xx package. The standard JTAG interface has four basic signals:

- TDI - Serial data into device
- TDO - Serial data out of device
- TCK - Common clock
- TMS - mode Selection

Every device that supports IEEE 1149.1 JTAG communication contains a Test Access Port (TAP) controller, which is a small state machine to manage JTAG protocol and serial streams of commands and data. Both the PSD module and the MCU module each contain a TAP controller.

Figure 79 on page 235 illustrates how these die are chained within a package. JTAG programming/test equipment will connect externally to the four IEEE 1149.1 JTAG pins on Port C. The TDI pin on the UPSD33xx package goes directly to the PSD module first, then exits the PSD module through TDO. TDO of the PSD module is connected to TDI of the MCU module. The serial path is completed when TDO of the MCU module exits the UPSD33xx package through the TDO pin on Port C. The JTAG signals TCK and TMS are common to both modules as specified in IEEE 1149.1. When JTAG devices are chained, typically one devices is in BYPASS mode while another device is executing a JTAG operation. For the UPSD33xx, the PSD module is in BYPASS mode while debugging the MCU module, and the MCU module is in BYPASS mode while performing ISP on the PSD module.

The $\overline{\text{RESET\_IN}}$ input pin on the UPSD33xx package goes to the MCU module, and this module will generate the $\overline{\text{RST}}$ reset signal for the PSD module. These reset signals are totally independent of the JTAG TAP controllers, meaning that the JTAG channel is operational when the modules are held in reset. It is required to assert $\overline{\text{RESET\_IN}}$ during ISP. STMicroelectronics and 3rd party JTAG ISP tools will automatically assert a reset signal during ISP. However, this reset signal must be connected to $\overline{\text{RESET\_IN}}$ as shown in examples in *Figure 80* and *Figure 81 on page 238*.

27.5.5 6-pin JTAG ISP (optional)

The optional signals TSTAT and $\overline{\text{TERR}}$ are programming status flags that can reduce programming time by as much as 30% compared to 4-pin JTAG because this status information does not have to be scanned out of the device serially. TSTAT and $\overline{\text{TERR}}$ must be used as a pair for 6-pin JTAG operation.

- TSTAT (pin PC3) indicates when programming of a single Flash location is complete. Logic 1 = Ready, Logic 0 = busy.
- $\overline{\text{TERR}}$ (pin PC4) indicates if there was a Flash programming error. Logic 1 = no error, Logic 0 = error.

The pin functions for PC3 and PC4 must be selected as “Dedicated JTAG - TSTAT” and “Dedicated JTAG - $\overline{\text{TERR}}$” in PSDsoft Express to enable 6-pin JTAG ISP.

No 8032 firmware is needed to use 6-pin ISP because all ISP functions are controlled from the external JTAG program/test equipment.

TSTAT and $\overline{\text{TERR}}$ are functional only when JTAG ISP operations are occurring, which means they are non-functional during JTAG debugging of the 8032 on the MCU module.

Programming times vary depending on the number of locations to be programmed and the JTAG programming equipment, but typical JTAG ISP programming times are 10 to 25 seconds using 6-pin JTAG. The signals TSTAT and $\overline{\text{TERR}}$ are not included in the IEEE 1149.1 specification.

Figure 81 shows recommended connections on a circuit board to a JTAG program/test tool using 6-pin JTAG. It is required to connect the $\overline{\text{RST}}$ output signal from the JTAG program/test equipment to the $\overline{\text{RESET\_IN}}$ input on the UPSD33xx. The $\overline{\text{RST}}$ signal is driven by the equipment with an Open Drain driver, allowing other sources (like a push button) to drive $\overline{\text{RESET\_IN}}$ without conflict.

Note: The recommended pull-up resistors and decoupling capacitor are illustrated in Figure 81.

Table 167. PSD module DC characteristics (with 5 V V_{DD})

Symbol	Parameter		Test condition (in addition to *Table 166 on page 249*)	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage		4.5 V < V_{DD} < 5.5 V	2		V_{DD} +0.5	V
V_{IL}	Input low voltage		4.5 V < V_{DD} < 5.5 V	–0.5		0.8	V
V_{LKO}	VDD (min) for Flash Erase and Program			2.5		4.2	V
V_{OL}	Output low voltage		I_{OL} = 20 µA, V_{DD} = 4.5 V		0.01	0.1	V
			I_{OL} = 8 mA, V_{DD} = 4.5 V		0.25	0.45	V
V_{OH}	Output high voltage		I_{OH} = –20 µA, V_{DD} = 4.5 V	4.4	4.49		V
			I_{OH} = –2 mA, V_{DD} = 4.5 V	2.4	3.9		V
I_{SB}	Standby supply current for Power-down mode		CSI > V_{DD} – 0.3 V(1)(2)		120	250	µA
I_{LI}	Input leakage current		V_{SS} < V_{IN} < V_{DD}	–1	±0.1	1	µA
I_{LO}	Output leakage current		0.45 < V_{OUT} < V_{DD}	–10	±5	10	µA
I_{CC} (DC)(3)	Operating supply current	PLD only	PLD_TURBO = Off, f = 0 MHz(4)		0		µA/PT
			PLD_TURBO = On, f = 0 MHz		400	700	µA/PT
		Flash memory	During Flash memory WRITE/Erase only		15	30	mA
			Read only, f = 0 MHz		0	0	mA
		SRAM	f = 0 MHz		0	0	mA
I_{CC} (AC)(3)	PLD AC Adder					(4)	
	Flash memory AC Adder				1.5	2.5	mA/MHz
	SRAM AC Adder				1.5	3.0	mA/MHz

1. Internal Power-down mode is active.
2. PLD is in non-Turbo mode, and none of the inputs are switching.
3. I_{OUT} = 0 mA
4. Please see *Figure 84 on page 242* for the PLD current calculation.