µPSD3251F

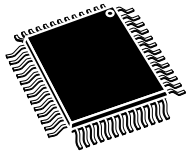
8032 MCU with Programmable Logic

FEATURES SUMMARY

- The µPSD3251F device combines a Flash PSD architecture with an 8032 microcontroller core.

  The µPSD3251F device of Flash PSDs feature dual banks of Flash memory, SRAM, general purpose I/O and programmable logic, supervisory functions and access via I^2C, ADC, and an on-board 8032 microcontroller core, with two UARTs, three 16-bit Timer/Counters and two External Interrupts. As with other Flash PSD families, the µPSD3251F device are also in-system programmable (ISP) via a JTAG ISP interface.
- Large 2KByte SRAM with battery back-up option
- Dual bank Flash memories
  - 64KByte main Flash memory
  - 16KByte secondary Flash memory
- Content Security
  - Block access to Flash memory
- Programmable Decode PLD for flexible address mapping of all memories within 8032 space.
- High-speed clock standard 8032 core (12-cycle)
- I^2C interface for peripheral connections
- Analog-to-Digital Converter (ADC)
- Six I/O ports with up to 37 I/O pins
- 3000 gate PLD with 16 macrocells
- Supervisor functions with Watchdog Timer
- In-System Programming (ISP) via JTAG
- Zero-Power Technology
- Single Supply Voltage
  - 4.5 to 5.5V

Figure 1. 52-lead, Thin, Quad, Flat Package



TQFP52 (T)

TABLE OF CONTENTS

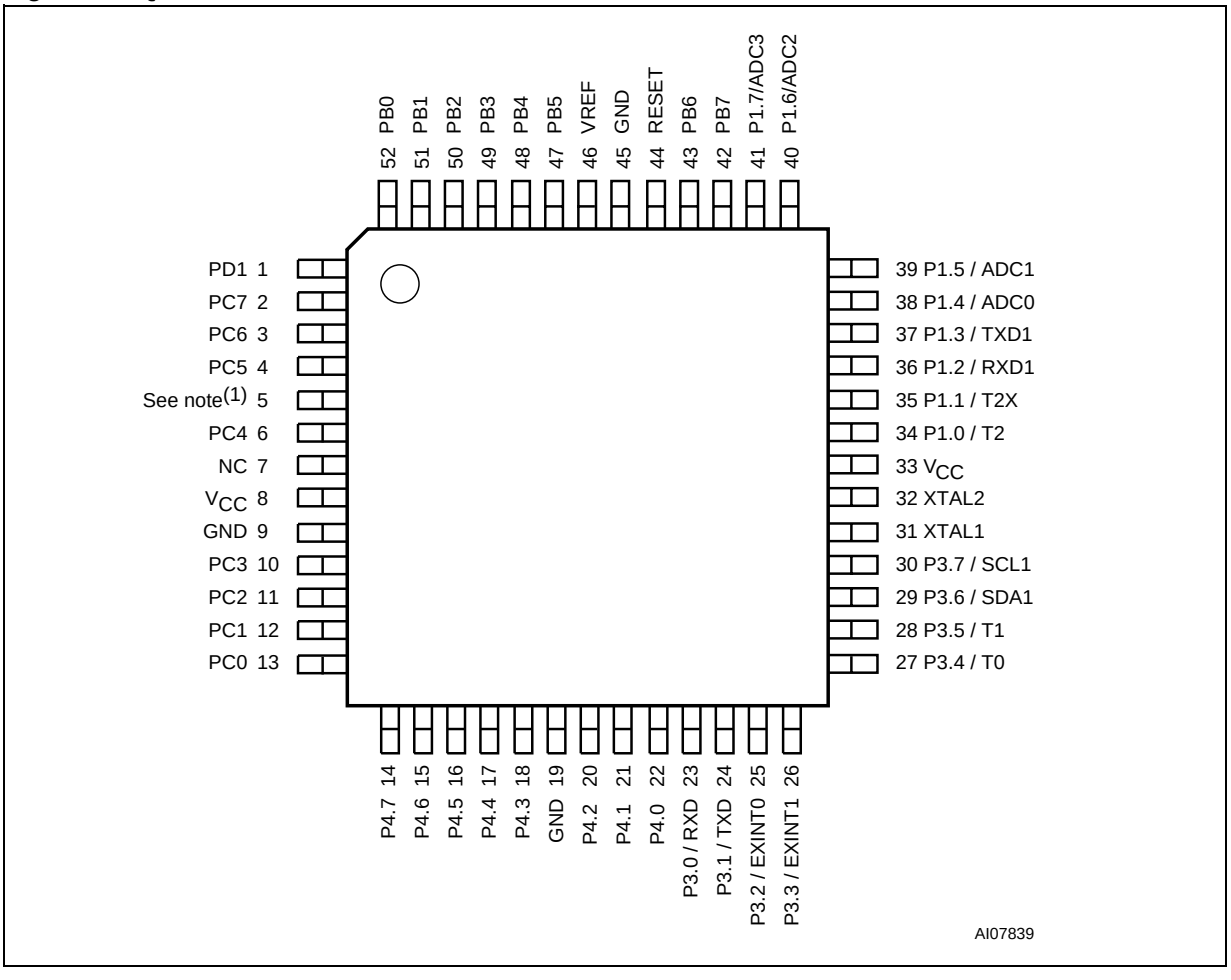
SUMMARY DESCRIPTION

- Dual bank Flash memories
  - Concurrent operation, read from memory while erasing and writing the other. In-Application Programming (IAP) for remote updates
  - Large 64KByte main Flash memory for application code, operating systems, or bit maps for graphic user interfaces
  - Large 16KByte secondary Flash memory divided in small sectors. Eliminate external EEPROM with software EEPROM emulation
  - Secondary Flash memory is large enough for sophisticated communication protocol during IAP while continuing critical system tasks
- Large SRAM with battery back-up option
  - 2KByte SRAM for RTOS, high-level languages, communication buffers, and stacks
- Programmable Decode PLD for flexible address mapping of all memories
  - Place individual Flash and SRAM sectors on any address boundary
  - Built-in page register breaks restrictive 8032 limit of 64KByte address space
  - Special register swaps Flash memory segments between 8032 “program” space and “data” space for efficient In-Application Programming
- High-speed clock standard 8032 core (12-cycle)
  - 40MHz operation at 5V
  - 2 UARTs with independent baud rate, three 16-bit Timer/Counters and two External Interrupts
- I^2C interface for peripheral connections
  - Capable of master or slave operation
- 4-channel, 8-bit Analog-to-Digital Converter (ADC) with analog supply voltage (V_{REF})
- Six I/O ports with up to 37 I/O pins
  - Multifunction I/O: GPIO, I^2C, PLD I/O, supervisor, and JTAG
  - Eliminates need for external latches and logic
- 3000 gate PLD with 16 macrocells
  - Create glue logic, state machines, delays, etc.
  - Eliminate external PALs, PLDs, and 74HCxx
  - Simple PSDsoft Express software... Free
- Supervisor functions
  - Generates reset upon low voltage or watchdog time-out. Eliminate external supervisor device
  - $\overline{\text{RESET}}$ Input pin; Reset output via PLD
- In-System Programming (ISP) via JTAG
  - Program entire chip in 10 - 25 seconds with no involvement of 8032
  - Allows efficient manufacturing, easy product testing, and Just-In-Time inventory
  - Eliminate sockets and pre-programmed parts
  - Program with FlashLINK™ cable and any PC
- Content Security
  - Programmable Security Bit blocks access of device programmers and readers
- Zero-Power Technology
  - Memories and PLD automatically reach standby current between input changes
- Packages
  - 52-pin TQFP

Table 1. µPSD3251F Device Product Matrix

Part No.	Main Flash (bit)	Sec. Flash (bit)	SRAM (bit)	Macro-Cells	I/O Pins	Timer/ Ctr	UART Ch.	I^2C	ADC Ch.	V_{CC}	MHz	Pins
uPSD3251F-40T6	512K	128K	16K	16	37	3	2	1	4	5V	40	52

Figure 2. TQFP52 Connections



Note: 1. Pull-up resistor of 7.5kΩ required on pin 5.
2. NC = Not Connected.

Table 2. 52-Pin Package Pin Description

Port Pin	Signal Name	Pin No.	In/Out	Function	
				Basic	Alternate
P1.0	T2	34	I/O	General I/O port pin	Timer 2 Count input
P1.1	T2EX	35	I/O	General I/O port pin	Timer 2 Trigger input
P1.2	RxD2	36	I/O	General I/O port pin	2nd UART Receive
P1.3	TxD2	37	I/O	General I/O port pin	2nd UART Transmit
P1.4	ADC0	38	I/O	General I/O port pin	ADC Channel 0 input
P1.5	ADC1	39	I/O	General I/O port pin	ADC Channel 1 input
P1.6	ADC2	40	I/O	General I/O port pin	ADC Channel 2 input
P1.7	ADC3	41	I/O	General I/O port pin	ADC Channel 3 input
P3.0	RxD1	23	I/O	General I/O port pin	UART Receive
P3.1	TxD1	24	I/O	General I/O port pin	UART Transmit
P3.2	INTO	25	I/O	General I/O port pin	Interrupt 0 input / Timer 0 gate control
P3.3	INT1	26	I/O	General I/O port pin	Interrupt 1 input / Timer 1 gate control
P3.4	T0	27	I/O	General I/O port pin	Counter 0 input
P3.5	T1	28	I/O	General I/O port pin	Counter 1 input
P3.6	SDA1	29	I/O	General I/O port pin	I^2C Bus serial data I/O
P3.7	SCL1	30	I/O	General I/O port pin	I^2C Bus clock I/O
P4.0		22	I/O	General I/O port pin	
P4.1		21	I/O	General I/O port pin	
P4.2		20	I/O	General I/O port pin	
P4.3		18	I/O	General I/O port pin	
P4.4		17	I/O	General I/O port pin	
P4.5		16	I/O	General I/O port pin	
P4.6		15	I/O	General I/O port pin	
P4.7		14	I/O	General I/O port pin	
	PUP	5	I/O	Pull-up resistor required (7.5kΩ)	
	AVREF	46	O	Reference Voltage input for ADC	
	RESET_	44	I	Active low $\overline{\text{RESET}}$ input	
	XTAL1	31	I	Oscillator input pin for system clock	
	XTAL2	32	O	Oscillator output pin for system clock	

Port Pin	Signal Name	Pin No.	In/Out	Function	
				Basic	Alternate
PB0		52	I/O	General I/O port pin	1. PLD Macro-cell outputs 2. PLD inputs 3. Latched Address Out (A0-A7)
PB1		51	I/O	General I/O port pin	
PB2		50	I/O	General I/O port pin	
PB3		49	I/O	General I/O port pin	
PB4		48	I/O	General I/O port pin	
PB5		47	I/O	General I/O port pin	
PB6		43	I/O	General I/O port pin	
PB7		42	I/O	General I/O port pin	
PC0	TMS	13	I	JTAG pin	1. PLD Macro-cell outputs 2. PLD inputs 3. SRAM stand by voltage input (V_{STBY}) 4. SRAM battery-on indicator (PC4) 5. JTAG pins are dedicated pins
PC1	TCK	12	I	JTAG pin	
PC2	V_{STBY}	11	I/O	General I/O port pin	
PC3	TSTAT	10	I/O	General I/O port pin	
PC4	TERR	6	I/O	General I/O port pin	
PC5	TDI	4	I	JTAG pin	
PC6	TDO	3	O	JTAG pin	
PC7		2	I/O	General I/O port pin	
PD1	CLKIN	1	I/O	General I/O port pin	1. PLD I/O 2. Clock input to PLD and APD
V_{CC}		8			
V_{CC}		33			
GND		9			
GND		19			
GND		45			
NC		7			

ARCHITECTURE OVERVIEW

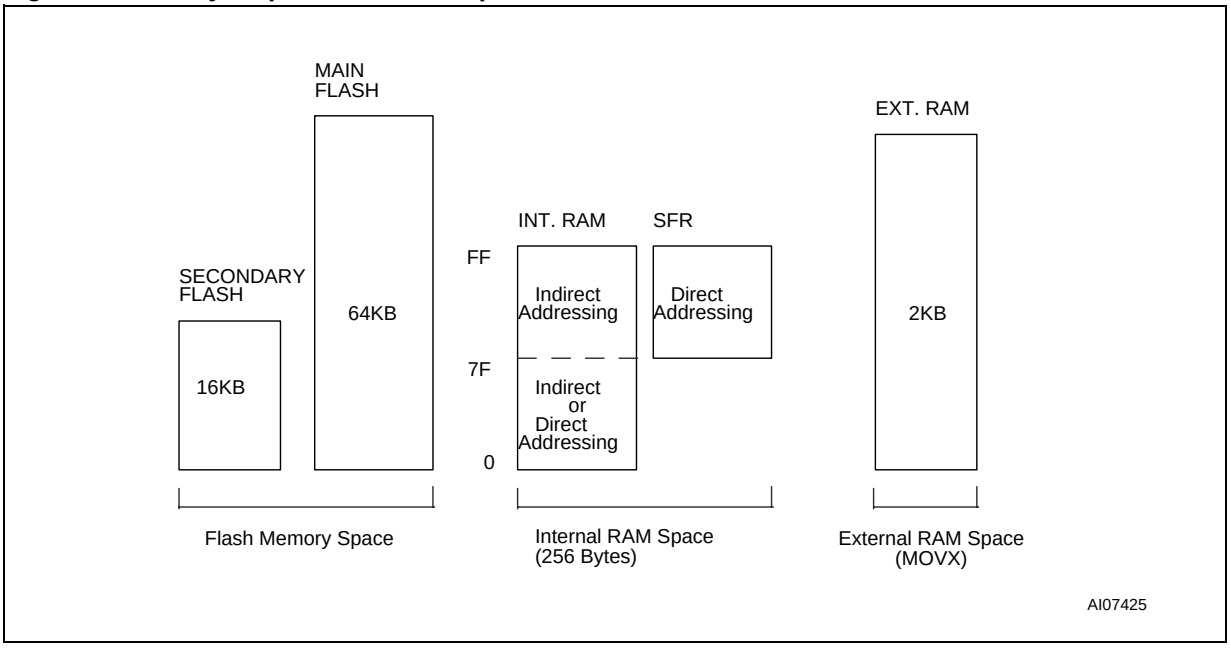
Memory Organization

The µPSD3251F device's standard 8032 Core has separate 64KB address spaces for Program memory and Data Memory. Program memory is where the 8032 executes instructions from. Data memory is used to hold data variables. Flash memory can be mapped in either program or data space. The Flash memory consists of two flash memory blocks: the main Flash (512Kbit) and the Secondary Flash (128Kbit). Except during flash memory programming or update, Flash memory can only be read, not written to. A Page Register is used to access memory beyond the 64K bytes address space. Refer to the PSD Module for details on mapping of the Flash memory.

The 8032 core has two types of data memory (internal and external) that can be read and written. The internal SRAM consists of 256 bytes, and includes the stack area.

The SFR (Special Function Registers) occupies the upper 128 bytes of the internal SRAM, the registers can be accessed by Direct addressing only. Another 2K bytes resides in the PSD Module that can be mapped to any address space defined by the user.

Figure 3. Memory Map and Address Space



Registers

The 8032 has several registers; these are the Program Counter (PC), Accumulator (A), B Register (B), the Stack Pointer (SP), the Program Status Word (PSW), General purpose registers (R0 to R7), and DPTR (Data Pointer register).

Accumulator. The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional tests. The Accumulator can be used as a 16-bit register with B Register as shown in Figure 4.

B Register. The B Register is the 8-bit general purpose register, used for an arithmetic operation such as multiply, division with the Accumulator (see Figure 5).

Stack Pointer. The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07h after reset. This causes the stack to begin at location 08h (see Figure 6).

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

Program Status Word. The Program Status Word (PSW) contains several bits that reflect the current state of the CPU and select Internal RAM (00h to 1Fh: Bank0 to Bank3). The PSW is described in Figure 7, page 16. It contains the Carry Flag, the Auxiliary Carry Flag, the Half Carry (for BCD operation), the general purpose flag, the Register Bank Select Flags, the Overflow Flag, and Parity Flag.

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

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

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

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

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

R0~R7. General purpose 8-bit registers that are locked in the lower portion of internal data area.

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

Figure 4. 8032 MCU Registers

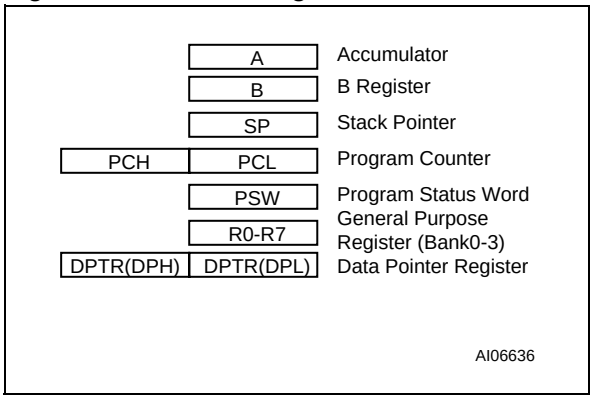


Figure 5. Configuration of BA 16-bit Registers

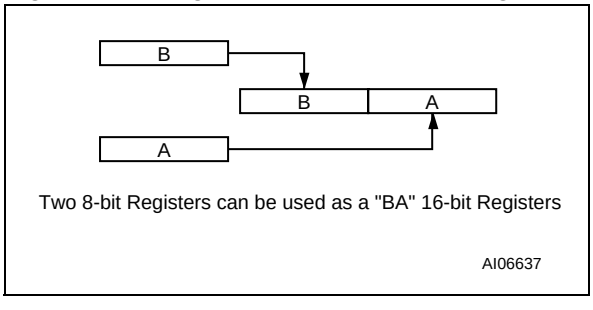


Figure 6. Stack Pointer

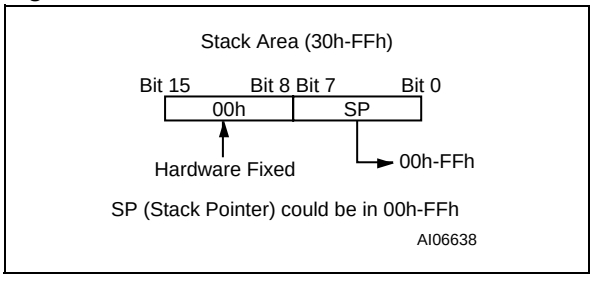
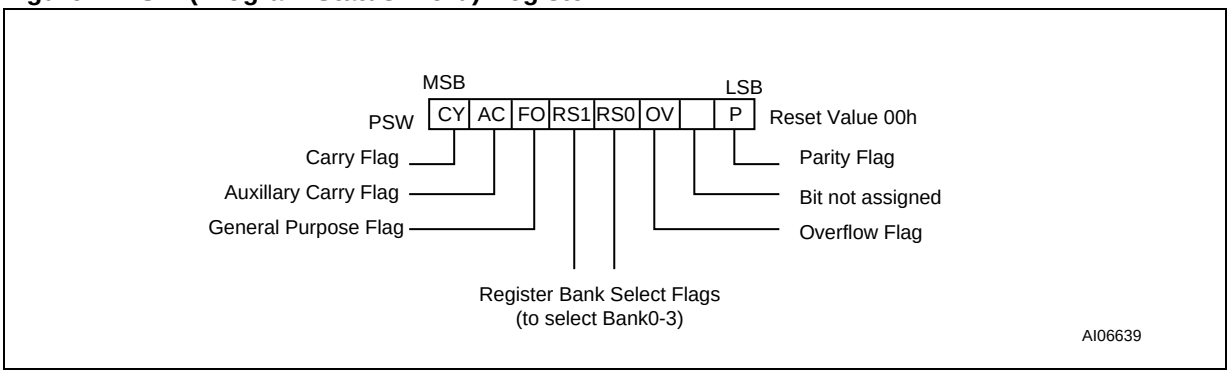


Figure 7. PSW (Program Status Word) Register



Program Memory

The program memory consists of two Flash memory: 64KByte Main Flash and 16KByte of Secondary Flash. The Flash memory can be mapped to any address space as defined by the user in the PSDsoft Tool. It can also be mapped to Data memory space during Flash memory update or programming.

After reset, the CPU begins execution from location 0000h. As shown in Figure 8, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003h. If External Interrupt 0 is going to be used, its service routine must begin at location 0003h. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003h for External Interrupt 0, 000Bh for Timer 0, 0013h for External Interrupt 1, 001Bh for Timer 1 and so forth. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval (see Figure 8). Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

Data memory

The internal data memory is divided into four physically separated blocks: 256 bytes of internal RAM, 128 bytes of Special Function Registers (SFRs) areas and 2K bytes (XRAM-PSD) in the PSD Module.

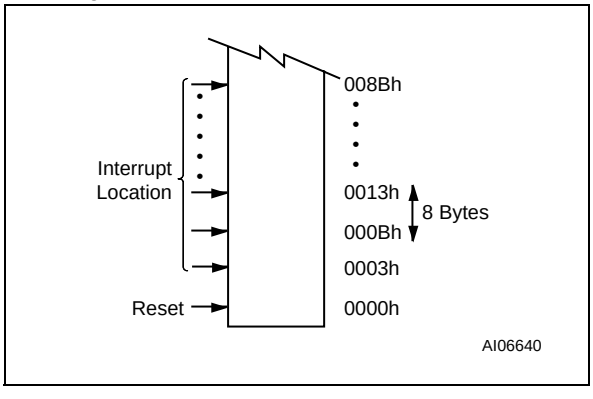
RAM

Four register banks, each 8 registers wide, occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack depth is only limited by the available internal RAM space of 256 bytes.

XRAM-PSD

The 2K bytes of XRAM-PSD resides in the PSD Module and can be mapped to any address space through the DPLD (Decoding PLD) as defined by the user in PSDsoft Development tool. The XRAM-PSD has a battery backup feature that allow the data to be retained in the event of a power lost. The battery is connected to the Port C PC2 pin. This pin must be configured in PSDSoft to be battery back-up.

Figure 8. Interrupt Location of Program Memory



SFR

The SFRs can only be addressed directly in the address range from 80h to FFh. Table 15, page 29 gives an overview of the Special Function Registers. Sixteen address in the SFRs space are both-byte and bit-addressable. The bit-addressable SFRs are those whose address ends in 0h and 8h. The bit addresses in this area are 80h to FFh.

Table 3. RAM Address

Byte Address (in Hexadecimal)	msb			Bit Address (Hex)			lsb	Byte Address (in Decimal)
↓								↓
FFh								255
30h								48
2Fh	7F	7E	7D	7C	7B	7A	79	78	47
2Eh	77	76	75	74	73	72	71	70	46
2Dh	6F	6E	6D	6C	6B	6A	69	68	45
2Ch	67	66	65	64	63	62	61	60	44
2Bh	5F	5E	5D	5C	5B	5A	59	58	43
2Ah	57	56	55	54	53	52	51	50	42
29h	4F	4E	4D	4C	4B	4A	49	48	41
28h	47	46	45	44	43	42	41	40	40
27h	3F	3E	3D	3C	3B	3A	39	38	39
26h	37	36	35	34	33	32	31	30	38
25h	2F	2E	2D	2C	2B	2A	29	28	37
24h	27	26	25	24	23	22	21	20	36
23h	1F	1E	1D	1C	1B	1A	19	18	35
22h	17	16	15	14	13	12	11	10	34
21h	0F	0E	0D	0C	0B	0A	09	08	33
20h	07	06	05	04	03	02	01	00	32
1Fh	Register Bank 3								31
18h									24
17h	Register Bank 2								23
10h									16
0Fh	Register Bank 1								15
08h									8
07h	Register Bank 0								7
00h									0

Addressing Modes

The addressing modes in the µPSD3251F device instruction set are as follows

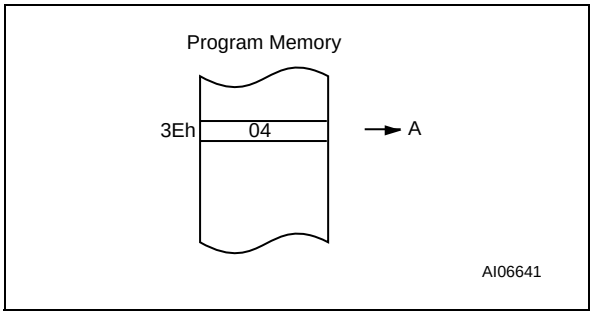
- Direct addressing
- Indirect addressing
- Register addressing
- Register-specific addressing
- Immediate constants addressing
- Indexed addressing

(1) Direct addressing. In a direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs (80~FFH RAM) can be directly addressed.

Example:

mov A, 3EH ;A <----- RAM[3E]

Figure 9. Direct Addressing

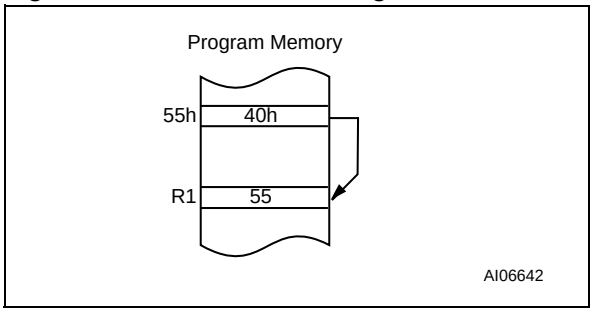


(2) Indirect addressing. In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed. The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit “data pointer” register, DPTR.

Example:

mov @R1, #40 H ;[R1] <-----40H

Figure 10. Indirect Addressing



(3) Register addressing. The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of four banks is selected at execution time by the two bank select bits in the PSW.

Example:

```
mov PSW, #0001000B ; select Bank0
mov A, #30H
mov R1, A
```

(4) Register-specific addressing. Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point it. The opcode itself does that.

(5) Immediate constants addressing. The value of a constant can follow the opcode in Program memory.

Example:

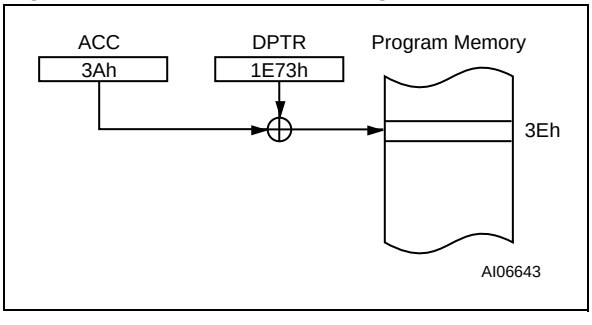
```
mov A, #10H.
```

(6) Indexed addressing. Only Program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program memory is formed by adding the Accumulator data to the base pointer (see Figure 11).

Example:

```
movc A, @A+DPTR
```

Figure 11. Indexed Addressing



Arithmetic Instructions

The arithmetic instructions is listed in Table 4, page 19. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as:

```
ADD a, 7FH (direct addressing)
ADD A, @R0 (indirect addressing)
ADD a, R7 (register addressing)
ADD A, #127 (immediate constant)
```

Note: Any byte in the internal Data Memory space can be incremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operations is

a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

In shift operations, dividing a number by 2n shifts its “n” bits to the right. Using DIV AB to perform the division completes the shift in 4?s and leaves the B register holding the bits that were shifted out. The DAA instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DAA operation, to ensure that the result is also in BCD.

Note: DAA will not convert a binary number to BCD. The DAA operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 4. Arithmetic Instructions

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
ADD A,<byte>	A = A + <byte>	X	X	X	X
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X
SUBB A,<byte>	A = A – <byte> – C	X	X	X	X
INC	A = A + 1	Accumulator only			
INC <byte>	<byte> = <byte> + 1	X	X	X	
INC DPTR	DPTR = DPTR + 1	Data Pointer only			
DEC	A = A – 1	Accumulator only			
DEC <byte>	<byte> = <byte> – 1	X	X	X	
MUL AB	B:A = B x A	Accumulator and B only			
DIV AB	A = Int[A / B] B = Mod[A / B]	Accumulator and B only			
DA A	Decimal Adjust	Accumulator only			

Logical Instructions

Table 5, page 20 shows list of the µPSD3251F device's logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and byte contains 01010011B, then:

ANL A, <byte>

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 5.

The ANL A, <byte> instruction may take any of the forms:

ANL A,7FH(direct addressing)

ANL A, @R1 (indirect addressing)

ANL A,R6 (register addressing)

ANL A,#53H (immediate constant)

Note: Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in

XRL P1, #0FFH.

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to push it onto the stack in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

MOVE B,#10

DIV AB

SWAP A

ADD A,B

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

Table 5. Logical Instructions

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
ANL A,<byte>	A = A .AND. <byte>	X	X	X	X
ANL <byte>,A	A = <byte> .AND. A	X			
ANL <byte>,#data	A = <byte> .AND. #data	X			
ORL A,<byte>	A = A .OR. <byte>	X	X	X	X
ORL <byte>,A	A = <byte> .OR. A	X			
ORL <byte>,#data	A = <byte> .OR. #data	X			
XRL A,<byte>	A = A .XOR. <byte>	X	X	X	X
XRL <byte>,A	A = <byte> .XOR. A	X			
XRL <byte>,#data	A = <byte> .XOR. #data	X			
CRL A	A = 00h	Accumulator only			
CPL A	A = .NOT. A	Accumulator only			
RL A	Rotate A Left 1 bit	Accumulator only			
RLC A	Rotate A Left through Carry	Accumulator only			
RR A	Rotate A Right 1 bit	Accumulator only			
RRC A	Rotate A Right through Carry	Accumulator only			
SWAP A	Swap Nibbles in A	Accumulator only			

Data Transfers

Internal RAM. Table 6 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember, the Upper 128 bytes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.

Note: In the µPSD3251F device, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128 bytes of RAM, if they are implemented, but not into SFR space.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory.

The XCH A, <byte> instruction causes the Accumulator and ad-dressed byte to exchange data. The XCHD A, @Ri instruction is similar, but only the low nibbles are involved in the exchange. To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting and 8-digit BCD number two digits to the right. Table 8 shows how this can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes. The same operation with XCHs uses only 9 bytes and executes almost twice as fast. To right-shift by an odd number of digits, a one-digit must be executed. Table 9 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the accumulator are shown alongside each instruction.

Table 6. Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
MOV A,<src>	A = <src>	X	X	X	X
MOV <dest>,A	<dest> = A	X	X	X	
MOV <dest>,<src>	<dest> = <src>	X	X	X	X
MOV DPTR,#data16	DPTR = 16-bit immediate constant				X
PUSH <src>	INC SP; MOV “@SP”,<src>	X			
POP <dest>	MOV <dest>,”@SP”; DEC SP	X			
XCH A,<byte>	Exchange contents of A and <byte>	X	X	X	
XCHD A,@Ri	Exchange low nibbles of A and @Ri		X		

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not equal) is a loop control that will be described later. The loop executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH, and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

Table 7. Shifting a BCD Number Two Digits to the Right (using direct MOVs: 14 bytes)

		2A	2B	2C	2D	2E	ACC
MOV	A,2Eh	00	12	34	56	78	78
MOV	2Eh,2Dh	00	12	34	56	56	78
MOV	2Dh,2Ch	00	12	34	34	56	78
MOV	2Ch,2Bh	00	12	12	34	56	78
MOV	2Bh,#0	00	00	12	34	56	78

Table 8. Shifting a BCD Number Two Digits to the Right (using direct XCHs: 9 bytes)

		2A	2B	2C	2D	2E	ACC
CLR	A	00	12	34	56	78	00
XCH	A,2Bh	00	00	34	56	78	12
XCH	A,2Ch	00	00	12	56	78	34
XCH	A,2Dh	00	00	12	34	78	56
XCH	A,2Eh	00	00	12	34	56	78

Table 9. Shifting a BCD Number One Digit to the Right

			2A	2B	2C	2D	2E	ACC
	MOV	R1,#2Eh	00	12	34	56	78	xx
	MOV	R0,#2Dh	00	12	34	56	78	xx
	; loop for R1 = 2Eh							
LOOP:	MOV	A,@R1	00	12	34	56	78	78
	XCHD	A,@R0	00	12	34	58	78	76
	SWAP	A	00	12	34	58	78	67
	MOV	@R1,A	00	12	34	58	67	67
	DEC	R1	00	12	34	58	67	67
	DEC	R0	00	12	34	58	67	67
	CNJE	R1,#2Ah,LOOP	00	12	34	58	67	67
	; loop for R1 = 2Dh		00	12	38	45	67	45
	; loop for R1 = 2Ch		00	18	23	45	67	23
	; loop for R1 = 2Bh		08	01	23	45	67	01
	CLR	A	08	01	23	45	67	00
	XCH	A,2Ah	00	01	23	45	67	08

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

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

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

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

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

copies the desired table entry into the Accumulator.

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

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

The subroutine “TABLE” would look like this:

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

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

Table 10. Data Transfer Instruction that Access External Data Memory Space

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

Table 11. Lookup Table READ Instruction

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

Boolean Instructions

The µPSD3251F device contains a complete Boolean (single-bit) processor. One page of the internal RAM contains 128 addressable bits, and the SFR space can support up to 128 addressable bits as well. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

The instruction set for the Boolean processor is shown in Table 12. All bits accesses are by direct addressing.

Bit addresses 00h through 7Fh are in the Lower 128, and bit addresses 80h through FFh are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV C,FLAG
MOV P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the Flag Bit is '1' or '0.'

The Carry Bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry Bit as C assemble as Carry-specific instructions (CLR C, etc.). The Carry Bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note: The Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

```
C = bit 1 .XRL. bit2
```

The software to do that could be as follows:

```
MOV C , bit1
JNB bit2, OVER
CPL C
OVER: (continue)
```

First, Bit 1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, Bit 1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1, C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, Bit 2 is being tested, and if bit2 = 0, the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation. All the PSW bits are directly addressable, so the Parity Bit, or the general-purpose flags, for example, are also available to the bit-test instructions.

Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program memory. How-ever, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore -128 to +127 Program Memory bytes relative to the first byte following the instruction.

Table 12. Boolean Instructions

Mnemonic	Operation
ANL C, bit	C = A .AND. bit
ANL C, /bit	C = C .AND. .NOT. bit
ORL C, bit	C = A .OR. bit
ORL C, /bit	C = C .OR. .NOT. bit
MOV C, bit	C = bit
MOV bit,C	bit = C
CLR C	C = 0
CLR bit	bit = 0
SETB C	C = 1
SETB bit	bit = 1
CPL C	C = .NOT. C
CPL bit	bit = .NOT. bit
JC rel	Jump if C =1
JNC rel	Jump if C = 0
JB bit,rel	Jump if bit =1
JNB bit,rel	Jump if bit = 0
JBC bit,rel	Jump if bit = 1; CLR bit

Jump Instructions

Table 13 shows the list of unconditional jump instructions. The table lists a single “JMP add” instruction, but in fact there are three SJMP, LJMP, and AJMP, which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is en-coded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of -128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a “Destination out of range” message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically. DPTR is set up with the address of a jump table. In a 5-way branch, for ex-ample, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV DPTR,#JUMP TABLE
MOV A,INDEX_NUMBER
RL A
JMP @A+DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP TABLE:
AJMP CASE 0
AJMP CASE 1
AJMP CASE 2
AJMP CASE 3
AJMP CASE 4
```

Table 13 shows a single “CALL addr” instruction, but there are two of them, LCALL and ACALL, which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case, the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 13. Unconditional Jump Instructions

Mnemonic	Operation
JMP addr	Jump to addr
JMP @A+DPTR	Jump to A+DPTR
CALL addr	Call Subroutine at addr
RET	Return from subroutine
RETI	Return from Interrupt
NOP	No operation

Table 14 shows the list of conditional jumps available to the µPSD3251F device user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of -128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero Bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
MOV COUNTER,#10
LOOP: (begin loop)
  •
  •
  •
(end loop)
DJNZ COUNTER, LOOP
(continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Table 9. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Table 9 Shifting a BCD Number One Digits to the Right, the two bytes were data in R1 and the constant 2Ah. The initial data in R1 was 2Eh.

Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2Ah.

Another application of this instruction is in “greater than, less than” comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry Bit is set (1). If the first is greater than or equal to the second, then the Carry Bit is cleared.

Machine Cycles

A machine cycle consists of a sequence of six states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus, a machine cycle takes 12 oscillator periods or 1µs if the oscillator frequency is 12MHz. Refer to Figure 12, page 27.

Each state is divided into a Phase 1 half and a Phase 2 half. State Sequence in µPSD3251F device shows that retrieve/execute sequences in states and phases for various kinds of instructions.

Normally two program retrievals are generated during each machine cycle, even if the instruction being executed does *not* require it. If the instruction being executed does not need more code bytes, the CPU simply ignores the extra retrieval, and the Program Counter is not incremented.

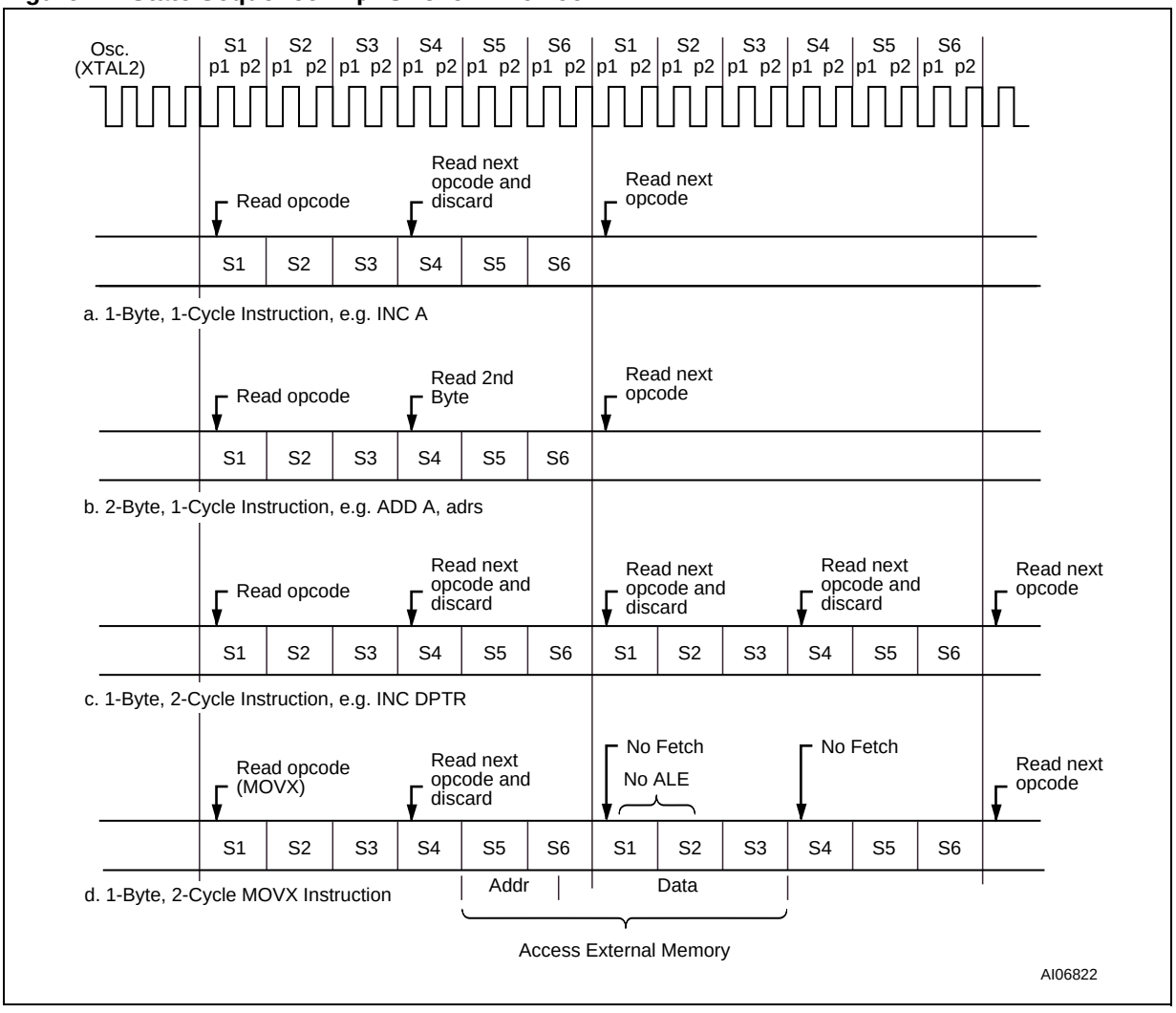
Execution of a one-cycle instruction (Figure 12, page 27) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second retrieve occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program retrieval is generated during the second cycle of a MOVX instruction. This is the only time program retrievals are skipped. The retrieve/execute sequence for MOVX instruction is shown in Figure 12, page 27 (d).

Table 14. Conditional Jump Instructions

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
JZ rel	Jump if A = 0	Accumulator only			
JNZ rel	Jump if A ≠ 0	Accumulator only			
DJNZ <byte>,rel	Decrement and jump if not zero	X		X	
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X	

Figure 12. State Sequence in µPSD3251F Device

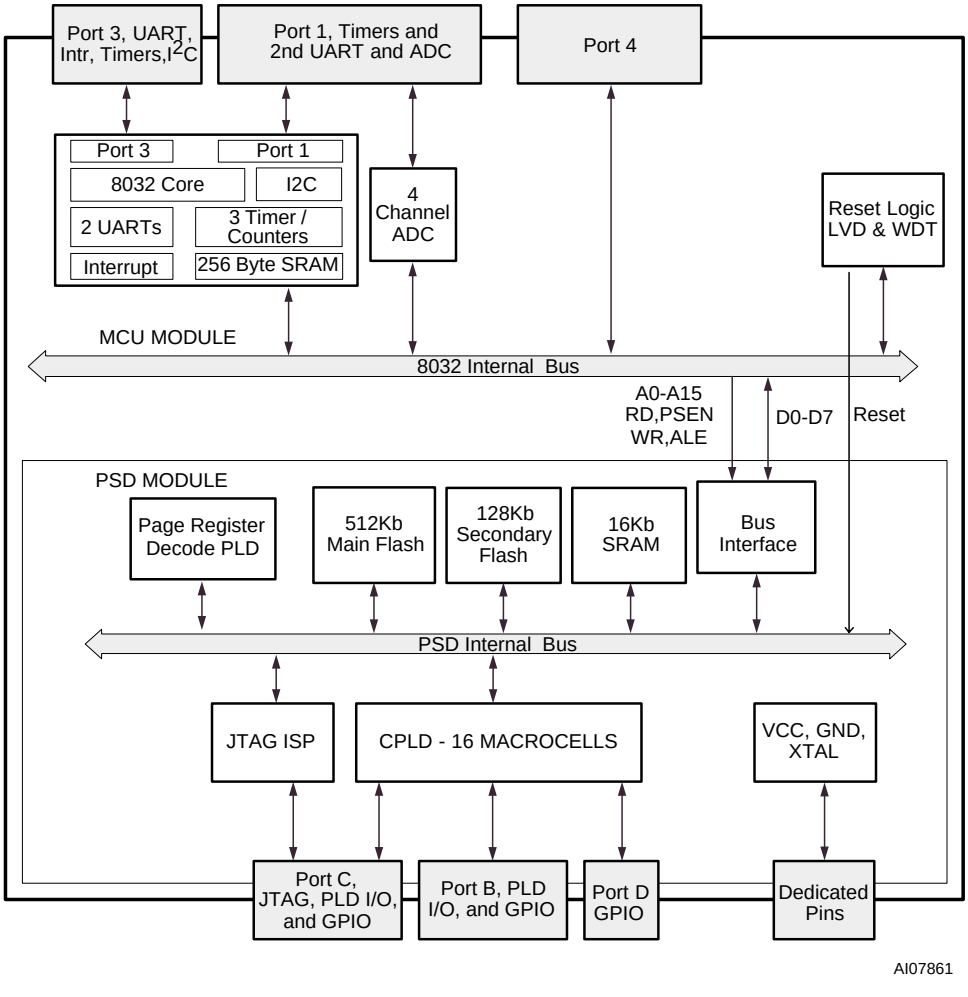


µPSD3251F HARDWARE DESCRIPTION

The µPSD3251F device has a modular architecture with two main functional modules: the MCU Module and the PSD Module. The MCU Module consists of a standard 8032 core, peripherals and other system supporting functions. The PSD Module provides configurable Program and Data memories to the 8032 CPU core. In addition, it has its own set of I/O ports and a PLD with 16 macrocells for general logic implementation. Ports B, C, and D are general purpose programmable I/O ports that have a port architecture which is different from Ports in the MCU Module.

The PSD Module communicates with the CPU Core through the internal address, data bus (A0-A15, D0-D7) and control signals (RD_, WR_, PSEN_ , ALE, RESET_). The user defines the Decoding PLD in the PSDsoft Development Tool and can map the resources in the PSD Module to any program or data address space.

Figure 13. µPSD3251F Device Functional Modules



MCU MODULE DISCRIPTION

This section provides a detail description of the MCU Module system functions and Peripherals, including:

- Special Function Registers
- Timers/Counter
- Interrupts
- Supervisory Function (LVD and Watchdog)
- USART
- Power Saving Modes
- I^2C Bus
- On-chip Oscillator
- ADC
- I/O Ports

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 15.

Note: In the SFRs not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. READ accesses to these addresses will in general return random data, and WRITE accesses will have no effect. User software should write '0s' to these unimplemented locations.

Table 15. SFR Memory Map

F8									FF
F0	B(1)								F7
E8									EF
E0	ACC(1)								E7
D8					S2CON	S2STA	S2DAT	S2ADR	DF
D0	PSW(1)								D7
C8	T2CON(1)	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CF
C0	P4(1)								C7
B8	IP(1)								BF
B0	P3(1)	PSCL0L	PSCL0H	PSCL1L	PSCL1H			IPA	B7
A8	IE(1)						WDKEY		AF
A0							WDRST	IEA	A7
98	SCON	SBUF	SCON2	SBUF2					9F
90	P1(1)	P1SFS		P3SFS		ASCL	ADAT	ACON	97
88	TCON(1)	TMOD	TL0	TL1	TH0	TH1			8F
80		SP	DPL	DPH				PCON	87

Note: 1. Register can be bit addressing

Table 16. List of all SFR

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
80											
81	SP									07	Stack Ptr
82	DPL									00	Data Ptr Low
83	DPH									00	Data Ptr High
87	PCON	SMOD	SMOD1	LVREN	ADSFINT	RCLK1	TCLK1	PD	IDLE	00	Power Ctrl
88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00	Timer / Cntr Control
89	TMOD	Gate	C/T	M1	M0	Gate	C/T	M1	M0	00	Timer / Cntr Mode Control
8A	TL0									00	Timer 0 Low
8B	TL1									00	Timer 1 Low
8C	TH0									00	Timer 0 High
8D	TH1									00	Timer 1 High
90	P1									FF	Port 1
91	P1SFS	P1S7	P1S6	P1S5	P1S4					00	Port 1 Select Register
93	P3SFS	P3S7	P3S6							00	Port 3 Select Register
94											
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register
98	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	Serial Control Register
99	SBUF									00	Serial Buffer
9A	SCON2	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	2nd UART Ctrl Register
9B	SBUF2									00	2nd UART Serial Buffer
A0											
A1											
A2											
A3											
A4											
A5											

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A6	WDRST									00	Watch Dog Reset
A7	IEA				ES2			EI^{2}C		00	Interrupt Enable (2nd)
A8	IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
A9											
AA											
AB											
AE	WDKEY									00	Watch Dog Key Register
B0	P3									FF	Port 3
B1	PSCL0L									00	Prescaler 0 Low (8-bit)
B2	PSCL0H									00	Prescaler 0 High (8-bit)
B3	PSCL1L									00	Prescaler 1 Low (8-bit)
B4	PSCL1H									00	Prescaler 1 High (8-bit)
B7	IPA				PS2			PI2C		00	Interrupt Priority (2nd)
B8	IP			PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority
C0	P4									FF	New Port 4
C8	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00	Timer 2 Control
C9	T2MOD								DCEN	00	Timer 2 Mode
CA	RCAP2L									00	Timer 2 Reload low
CB	RCAP2H									00	Timer 2 Reload High
CC	TL2									00	Timer 2 Low byte
CD	TH2									00	Timer 2 High byte
D0	PSW	CY	AC	FO	RS1	RS0	OV		P	00	Program Status Word
D1											
D2	S2SETUP									00	I^{2}C (S2) Setup
D4											

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
D5											
D6											
D7											
D8											
D9											
DA											
DB											
DC	S2CON	CR2	EN1	STA	STO	ADDR	AA	CR1	CR0	00	I^2C Bus Control Reg
DD	S2STA	GC	Stop	Intr	TX-Md	Bbusy	Blost	ACK_R	SLV	00	I^2C Bus Status
DE	S2DAT									00	Data Hold Register
DF	S2ADR									00	I^2C address
E0	ACC									00	Accumulator
F0	B									00	B Register

Table 17. PSD Module Register Address Offset

CSIOP Addr Offset	Register Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
01	Data In (Port B)										
03	Control (Port B)									00	
05	Data Out (Port B)									00	
07	Direction (Port B)									00	
09	Drive (Port B)									00	
0B	Input Macrocell (Port B)										
0D	Enable Out (Port B)										
10	Data In (Port C)										
12	Data Out (Port C)									00	
14	Direction (Port C)									00	
16	Drive (Port C)									00	
18	Input Macrocell (Port C)										
1A	Enable Out (Port C)										
11	Data In (Port D)	*	*	*	*	*		*			Only Bit 1 is used
13	Data Out (Port D)	*	*	*	*	*		*		00	Only Bit 1 is used
15	Direction (Port D)	*	*	*	*	*		*		00	Only Bit 1 is used
17	Drive (Port D)	*	*	*	*	*		*		00	Only Bit 1 is used
1B	Enable Out (Port D)	*	*	*	*	*		*			Only Bit 1 is used
20	Output Macrocells AB										
21	Output Macrocells BC										
22	Mask Macrocells AB										
23	Mask Macrocells BC										
C0	Primary Flash Protection					Sec3_ Prot	Sec2_ Prot	Sec1_ Prot	Sec0_ Prot		Bit = 1 sector is protected
C2	Secondary Flash Protection	Security _Bit	*	*	*	*	*	Sec1_ Prot	Sec0_ Prot		Security Bit = 1 device is secured

CSIOP Addr Offset	Register Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
B0	PMMR0	*	*	PLD Mcells clk	PLD array-clk	PLD Turbo	*	APD enable	*	00	Control PLD power consumption
B4	PMMR2	*		PLD array Ale	PLD array Cntl2	PLD array Cntl1	PLD array Cntl0	*	*	00	Blocking inputs to PLD array
E0	Page									00	Page Register
E2	VM	Periph-mode	*	*	FL_ data	Boot_ data	FL_ code	Boot_ code	SR_ code		Configure 8032 Program and Data Space

Note: (Register address = csiop address + address offset; where csiop address is defined by user in PSDsoft)
* indicates bit is not used and need to set to '0.'

INTERRUPT SYSTEM

There are interrupt requests from 10 sources as follows (see Figure 14, page 36).

- INT0 External Interrupt
- 2nd USART Interrupt
- Timer 0 Interrupt
- I^2C Interrupt
- INT1 External Interrupt (or ADC Interrupt)
- Timer 1 Interrupt
- USART Interrupt
- Timer 2 Interrupt

External Int0

– The INT0 can be either level-active or transition-active depending on Bit IT0 in register TCON. The flag that actually generates this interrupt is Bit IE0 in TCON.

– When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition activated.

– If the interrupt was level activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Timer 0 and 1 Interrupts

– Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1 which are set by an overflow of their respective Timer/Counter registers (except for Timer 0 in Mode 3).

– These flags are cleared by the internal hardware when the interrupt is serviced.

Timer 2 Interrupt

– Timer 2 Interrupt is generated by TF2 which is set by an overflow of Timer 2. This flag has to be cleared by the software - not by hardware.

– It is also generated by the T2EX signal (Timer 2 External Interrupt P1.1) which is controlled by EXEN2 and EXF2 Bits in the T2CON register.

I^2C Interrupt

– The interrupt of the I^2C is generated by Bit INTR in the register S2STA.

– This flag is cleared by hardware.

External Int1

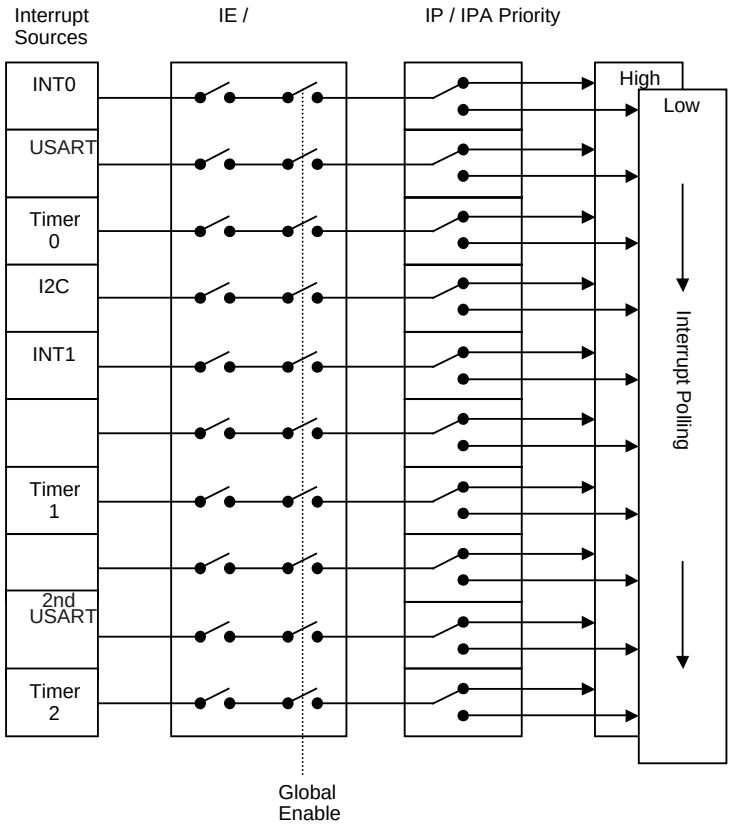
– The INT1 can be either level active or transition active depending on Bit IT1 in register TCON. The flag that actually generates this interrupt is Bit IE1 in TCON.

– When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition activated.

– If the interrupt was level activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

– The ADC can take over the External INT1 to generate an interrupt on conversion being completed

Figure 14. Interrupt System



USART Interrupt

– The USART Interrupt is generated by RI (Receive Interrupt) OR TI (Transmit Interrupt).
– When the USART Interrupt is generated, the corresponding request flag must be cleared by the software. The interrupt service routine will have to check the various USART registers to determine the source and clear the corresponding flag.
– Both USART's are identical, except for the additional interrupt controls in the Bit 4 of the additional interrupt control registers (A7H, B7H).

Interrupt Priority Structure

Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function register IP and IPA.

0 = low priority

1 = high priority

A low priority interrupt may be interrupted by a high priority interrupt level interrupt. A high priority interrupt routine cannot be interrupted by any other interrupt source. If two interrupts of different priority occur simultaneously, the high priority level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence.

Interrupts Enable Structure

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function register IE and IEA. All interrupt source can also be globally disabled by the clearing Bit EA in IE (see Table 19). Please see Tables 20, 21, 22, and 23 for individual bit descriptions.

Table 18. Priority Levels

Source	Priority with Level
Int0	0 (highest)
2nd USART	1
Timer 0	2
I²C	3
Int1	4
reserved	5
Timer 1	6
reserved	7
1st USART	8
Timer 2+EXF2	9 (lowest)

Table 19. SFR Register

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A7	IEA	—	—	—	ES2	—	—	EI^{2}C	—	00	Interrupt Enable (2nd)
A8	IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
B7	IPA	—	—	—	PS2	—	—	PI^{2}C	—	00	Interrupt Priority (2nd)
B8	IP	—	—	PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority

Table 20. Description of the IE Bits.

Bit	Symbol	Function
7	EA	Disable all interrupts: 0: no interrupt with be acknowledged 1: each interrupt source is individually enabled or disabled by setting or clearing its enable bit
6	—	Reserved
5	ET2	Enable Timer 2 Interrupt
4	ES	Enable USART Interrupt
3	ET1	Enable Timer 1 Interrupt
2	EX1	Enable External Interrupt (Int1)
1	ET0	Enable Timer 0 Interrupt
0	EX0	Enable External Interrupt (Int0)

Table 21. Description of the IEA Bits

Bit	Symbol	Function
7	—	Not used
6	—	Not used
5	—	Not used
4	ES2	Enable 2nd USART Interrupt
3	—	Not used
2	—	Not used
1	EI2C	Enable I²C Interrupt
0	—	Not used

Table 22. Description of the IP Bits

Bit	Symbol	Function
7	—	Reserved
6	—	Reserved
5	PT2	Timer 2 Interrupt priority level
4	PS	USART Interrupt priority level
3	PT1	Timer 1 Interrupt priority level
2	PX1	External Interrupt (Int1) priority level
1	PT0	Timer 0 Interrupt priority level
0	PX0	External Interrupt (Int0) priority level

Table 23. Description of the IPA Bits

Bit	Symbol	Function
7	—	Not used
6	—	Not used
5	—	Not used
4	PS2	2nd USART Interrupt priority level
3	—	Not used
2	—	Not used
1	PI2C	I²C Interrupt priority level
0	—	Not used

How Interrupts are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this H/W generated LCALL is not blocked by any of the following conditions:

– An interrupt of equal priority or higher priority level is already in progress.
– The current machine cycle is not the final cycle in the execution of the instruction in progress.
– The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle.

Note: If an interrupt flag is active but being responded to for one of the above mentioned conditions, if the flag is still inactive when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The processor acknowledges an interrupt request by executing a hardware generated LCALL to the appropriate service routine. The hardware generated LCALL pushes the contents of the Program Counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 24.

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note: A simple RET instruction would also return execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress, making future interrupts impossible.

Table 24. Vector Addresses

Source	Vector Address
Int0	0003h
2nd USART	004Bh
Timer 0	000Bh
I²C	0043h
Int1	0013h
Timer 1	001Bh
1st USART	0023h
Timer 2+EXF2	002Bh

POWER-SAVING MODE

Two software selectable modes of reduced power consumption are implemented (see Table 25).

Idle Mode

The following Functions are Switched Off.

– CPU (Halted)

The following Function Remain Active During Idle Mode.

– External Interrupts
– Timer 0, Timer 1, Timer 2
– USART
– 8-bit ADC
– I^2C Interface

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

Power-Down Mode

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

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

Power Control Register

The Idle and Power-down Modes are activated by software via the PCON register (see Tables 26 and Table 27, page 41).

Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle Mode is activated. Once in the Idle Mode, the CPU status is preserved in its entirety: Stack pointer, Program counter, Program status word, Accumulator, RAM and All other registers maintain their data during Idle Mode.

There are three ways to terminate the Idle Mode.

– Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic '1' to PCON.0.
– External hardware reset: the hardware reset is required to be active for two machine cycle to complete the RESET operation.
– Internal reset: the microcontroller restarts after 3 machine cycles in all cases.

Power-Down Mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down Mode. Once in Power-down Mode, the oscillator is stopped. The contents of the on-chip RAM and the Special Function Register are preserved.

The Power-down Mode can be terminated by an external RESET.

Table 25. Power-Saving Mode Power Consumption

Mode	Addr/Data	Ports1,3,4	I^2C
Idle	Maintain Data	Maintain Data	Active
Power-down	Maintain Data	Maintain Data	Disable

Table 26. Pin Status During Idle and Power-down Mode

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

Table 27. Description of the PCON Bits

Bit	Symbol	Function
7	SMOD	Double Baud Data Rate Bit UART
6	SMOD1	Double Baud Data Rate Bit 2nd UART
5	LVREN	LVR Disable Bit (active High)
4	ADSFINT	Enable ADC Interrupt
3	RCLK1(1)	Received Clock Flag (UART 2)
2	TCLK1(1)	Transmit Clock Flag (UART 2)
1	PD	Activate Power-down Mode (High enable)
0	IDL	Activate Idle Mode (High enable)

Note: 1. See the T2CON register for details of the flag description

I/O PORTS (MCU MODULE)

The MCU Module has three ports: Port1, Port3, and Port 4. (Refer to the PSD Module section on I/O ports B, C and D).

Port1 - Port3 are the same as in the standard 8032 micro-controllers, with the exception of the additional special peripheral functions (see Table 28). All ports are bi-directional. Pins of which the alternative function is not used may be used as normal bi-directional I/O.

The use of Port1- Port4 pins as alternative functions are carried out automatically by the µPSD3251F device provided the associated SFR Bit is set HIGH.

Table 28. I/O Port Functions

Port Name	Main Function	Alternate
Port 1	GPIO	Timer 2 - Bits 0,1 2nd UART - Bits 2,3 ADC - Bits 4..7
Port 3	GPIO	UART - Bits 0,1 Interrupt - Bits 2,3 Timers - Bits 4,5 I^2C - Bits 6,7
Port 4	GPIO	

The following SFR registers (Tables 29 and 30) are used to control the mapping of alternate functions onto the I/O port bits. Port 1 alternate functions are controlled using the P1SFS register, except for Timer 2 and the 2nd UART which are enabled by their configuration registers. P1.0 to P1.3 are default to GPIO after reset.

Port 3 pins 6 and 7 have been modified from the standard 8032. These pins that were used for READ and WRITE control signals are now GPIO or I^2C bus pins. The READ and WRITE pins are assigned to dedicated pins.

Port 3 (I^2C) alternate functions are controlled using the P3SFS Special Function Selection register. After a reset, the I/O pins default to GPIO. The alternate function is enabled if the corresponding bit in the P3SFS register is set to '1.' Other Port 3 alternative functions (UART, Interrupt, and Timer/Counter) are enabled by their configuration register and do not require setting of the bits in P3SFS.

Table 29. P1SFS (91H)

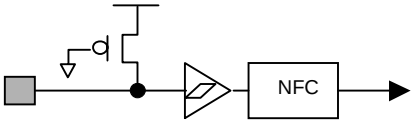
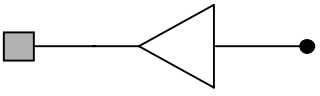
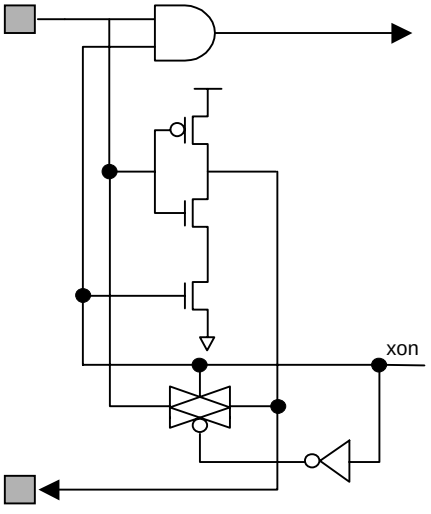
7	6	5	4	3	2	1	0
0=Port 1.7 1=ACH3	0=Port 1.6 1=ACH2	0=Port 1.5 1=ACH1	0=Port 1.4 1=ACH0	Bits Reserved		Bits Reserved	

Table 30. P3SFS (93H)

7	6	5	4	3	2	1	0
0 = Port 3.7 1 = SCL from I^2C unit	0 = Port 3.6 1 = SDA from I^2C unit	Bits are reserved.					

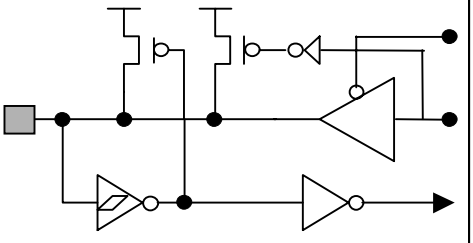
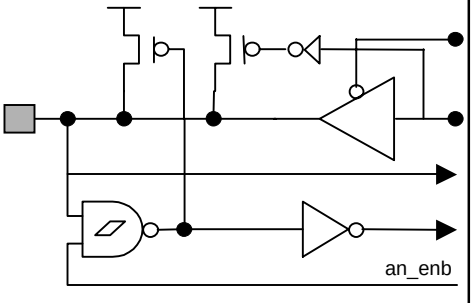
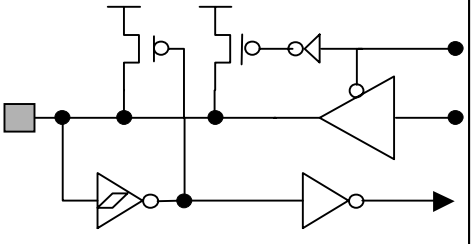
Port Type and Description

Figure 15. Port Type and Description (Part 1)

Symbol	In / Out	Circuit	Description
RESET	I		• Schmitt input with internal pull-up CMOS compatible interface NFC : 400ns
WR, RD,ALE, PSEN	O		Output only
XTAL1, XTAL2	I O		On-chip oscillator On-chip feedback resistor Stop in the power down mode External clock input available CMOS compatible interface

AI07862

Figure 16. PORT Type and Description (Part 2)

Symbol	In/ Out	Circuit	Function
PORT1 <3:0>, PORT3, PORT4<7:3,1:0>	I/O		Bidirectional I/O port with internal pull-ups Schmitt input CMOS compatible interface
PORT1 < 7:4 >	I/O		Bidirectional I/O port with internal pull-ups Schmitt input CMOS compatible interface Analog input option
PORT4.2	I/O		Bidirectional I/O port with internal pull-ups Schmitt input. TTL compatible interface

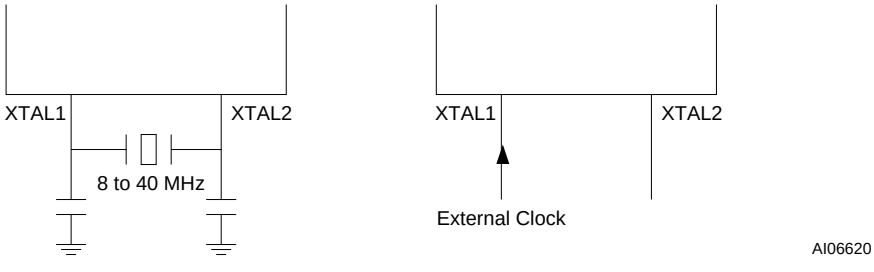
AI07863

OSCILLATOR

The oscillator circuit of the µPSD3251F device is a single stage inverting amplifier in a Pierce oscillator configuration (see Figure 17). The circuitry between XTAL1 and XTAL2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuit. Both are operated in parallel resonance.

XTAL1 is the high gain amplifier input, and XTAL2 is the output. To drive the µPSD3251F device externally, XTAL1 is driven from an external source and XTAL2 left open-circuit.

Figure 17. Oscillator



AI06620

SUPERVISORY

There are three ways to invoke a reset and initialize the µPSD3251F device:

- Via the external $\overline{\text{RESET}}$ pin
- Via the internal LVR Block.
- Via Watch Dog timer

The $\overline{\text{RESET}}$ mechanism is illustrated in Figure 18.

Each $\overline{\text{RESET}}$ source will cause an internal reset signal active. The CPU responds by executing an internal reset and puts the internal registers in a defined state. This internal reset is also routed as an active low reset input to the PSD Module.

External Reset

The $\overline{\text{RESET}}$ pin is connected to a Schmitt trigger for noise reduction. A $\overline{\text{RESET}}$ is accomplished by holding the $\overline{\text{RESET}}$ pin LOW for at least 1ms at power up while the oscillator is running. Refer to AC spec on other $\overline{\text{RESET}}$ timing requirements.

Low V_{DD} Voltage Reset

An internal reset is generated by the LVR circuit when the V_{DD} drops below the reset threshold. After V_{DD} reaching back up to the reset threshold, the $\overline{\text{RESET}}$ signal will remain asserted for 10ms before it is released. On initial power-up the LVR is enabled (default). After power-up the LVR can be disabled via the LVREN Bit in the PCON Register.

Note: The LVR logic is still functional in both the Idle and Power-down Modes.

The reset threshold:

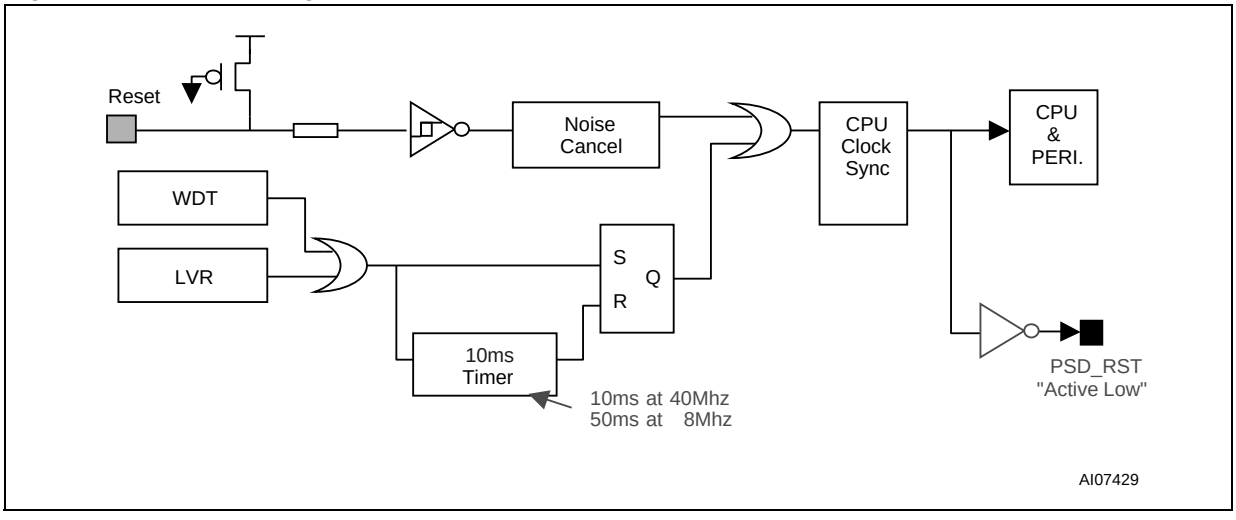
- 5V operation: 4V +/- 0.25V

This logic supports approximately 0.1V of hysteresis and 1µs noise-cancelling delay.

Watchdog Timer Overflow

The Watchdog Timer generates an internal reset when its 22-bit counter overflows. See WATCHDOG TIMER, page 46 for details.

Figure 18. $\overline{\text{RESET}}$ Configuration



WATCHDOG TIMER

The hardware Watchdog Timer (WDT) resets the µPSD3251F device when it overflows. The WDT is intended as a recovery method in situations where the CPU may be subjected to a software upset. To prevent a system reset the timer must be reloaded in time by the application software. If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will result in a reset upon overflow thus preventing the processor running out of control.

In the Idle Mode the watchdog timer and reset circuitry remain active. The WDT consists of a 22-bit counter, the Watchdog Timer RESET (WDRST) SFR and Watchdog Key Register (WDKEY).

Since the WDT is automatically enabled while the processor is running. the user only needs to be concerned with servicing it.

The 22-bit counter overflows when it reaches 4194304 (3FFFFFH). The WDT increments once every machine cycle.

This means the user must reset the WDT at least every 4194304 machine cycles (1.258 seconds at 40MHz). To reset the WDT the user must write a value between 00-7EH to the WDRST register. The value that is written to the WDRST is loaded to the 7MSB of the 22-bit counter. This allows the user to pre-loaded the counter to an initial value to generate a flexible Watchdog time out period. Writing a “00” to WDRST clears the counter.

The watchdog timer is controlled by the watchdog key register, WDKEY. Only pattern 01010101 (=55H), disables the watchdog timer. The rest of pattern combinations will keep the watchdog timer enabled. This security key will prevent the watchdog timer from being terminated abnormally when the function of the watchdog timer is needed.

In Idle Mode, the oscillator continues to run. To prevent the WDT from resetting the processor while in Idle, the user should always set up a timer that will periodically exit Idle, service the WDT, and re-enter Idle Mode.

Watchdog reset pulse width depends on the clock frequency (see Figure 19, page 47). The reset period is $Tf_{OSC} \times 12 \times 2^{22}$.

The $\overline{RESET}$ pulse width is $Tf_{OSC} \times 12 \times 2^{15}$.

Table 31. Watchdog Timer Key Register (WDKEY: 0AEH)

7	6	5	4	3	2	1	0
WDKEY7	WDKEY6	WDKEY5	WDKEY4	WDKEY3	WDKEY2	WDKEY1	WDKEY0

Table 32. Description of the WDKEY Bits

Bit	Symbol	Function
7 to 0	WDKEY7 to WDKEY0	Enable or disable Watchdog Timer. 01010101 (=55h): disable watchdog timer. Others: enable watchdog timer

Table 33. Watchdog Timer Clear Register (WDRST: 0A6H)

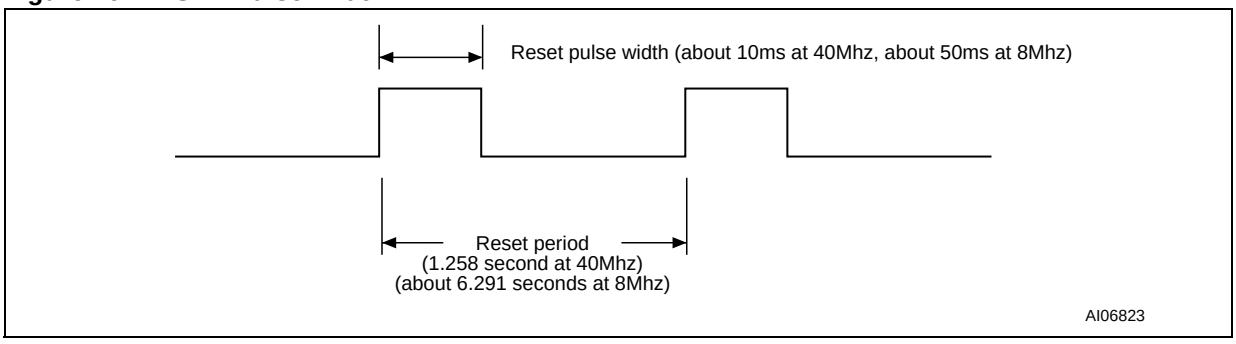
7	6	5	4	3	2	1	0
Reserved	WDRST6	WDRST5	WDRST4	WDRST3	WDRST2	WDRST1	WDRST0

Table 34. Description of the WDRST Bits

Bit	Symbol	Function
7	—	Reserved
6 to 0	WDRST6 to WDRST0	To reset Watchdog Timer, write any value beteen 00h and 7Eh to this register. This value is loaded to the 7 most significant bits of the 22-bit counter. For example: MOV WDRST,#1EH

Note: The Watchdog Timer (WDT) is enabled at power-up or reset and must be served or disabled.

Figure 19. $\overline{\text{RESET}}$ Pulse Width



TIMER/COUNTERS (TIMER 0, TIMER 1 AND TIMER 2)

The µPSD3251F device has three 16-bit Timer/ Counter registers: Timer 0, Timer 1 and Timer 2. All of them can be configured to operate either as timers or event counters and are compatible with standard 8032 architecture.

In the “Timer” function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 6 CPU clock periods, the count rate is 1/6 of the CPU clock frequency or 1/12 of Oscillator Frequency (f_{OSC}).

In the “Counter” function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 f_{OSC} clock periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the f_{OSC}. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full cycle. In addition to the “Timer” or “Counter” selection, Timer 0 and Timer 1 have four operating modes from which to select.

Timer 0 and Timer 1

The “Timer” or “Counter” function is selected by control bits C/T in the Special Function Register TMOD. These Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for Timers/ Counters. Mode 3 is different. The four operating modes are de-scribed in the following text.

Table 35. Control Register (TCON)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table 36. Description of the TCON Bits

Bit	Symbol	Function
7	TF1	Timer 1 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine
6	TR1	Timer 1 Run Control Bit. Set/cleared by software to turn Timer/Counter on or off
5	TF0	Timer 0 Overflow Flag. Set by hardier on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine
4	TR0	Timer 0 Run Control Bit. Set/cleared by software to turn Timer/Counter on or off
3	IE1	Interrupt 1 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed
2	IT1	Interrupt 1 Type Control Bit. Set/cleared by software to specify falling-edge/low-level triggered external interrupt
1	IE0	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed
0	IT0	Interrupt 0 Type Control Bit. Set/cleared by software to specify falling-edge/low-level triggered external interrupt

Table 37. TMOD Register (TMOD)

7	6	5	4	3	2	1	0
Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0

Table 38. Description of the TMOD Bits

Bit	Symbol	Timer	Function
7	Gate	Timer 1	Gating control when set. Timer/Counter 1 is enabled only while INT1 pin is High and TR1 control pin is set. When cleared, Timer 1 is enabled whenever TR1 control bit is set
6	C/$\overline{T}$		Timer or Counter selector, cleared for timer operation (input from internal system clock); set for counter operation (input from T1 input pin)
5	M1		(M1,M0)=(0,0): 13-bit Timer/Counter, TH1, with TL1 as 5-bit prescaler (M1,M0)=(0,1): 16-bit Timer/Counter. TH1 and TL1 are cascaded. There is no prescaler. (M1,M0)=(1,0): 8-bit auto-reload Timer/Counter. TH1 holds a value which is to be reloaded into TL1 each time it overflows (M1,M0)=(1,1): Timer/Counter 1 stopped
4	M0		
3	Gate	Timer 0	Gating control when set. Timer/Counter 0 is enabled only while INT0 pin is High and TR0 control pin is set. When cleared, Timer 0 is enabled whenever TR0 control bit is set
2	C/$\overline{T}$		Timer or Counter selector, cleared for timer operation (input from internal system clock); set for counter operation (input from T0 input pin)
1	M1		(M1,M0)=(0,0): 13-bit Timer/Counter, TH0, with TL0 as 5-bit prescaler (M1,M0)=(0,1): 16-bit Timer/Counter. TH0 and TL0 are cascaded. There is no prescaler. (M1,M0)=(1,0): 8-bit auto-reload Timer/Counter. TH0 holds a value which is to be reloaded into TL0 each time it overflows (M1,M0)=(1,1): TL0 is an 8-bit Timer/Counter controlled by the standard TImer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits
0	M0		

Mode 0. Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 20 shows the Mode 0 operation as it applies to Timer 1.

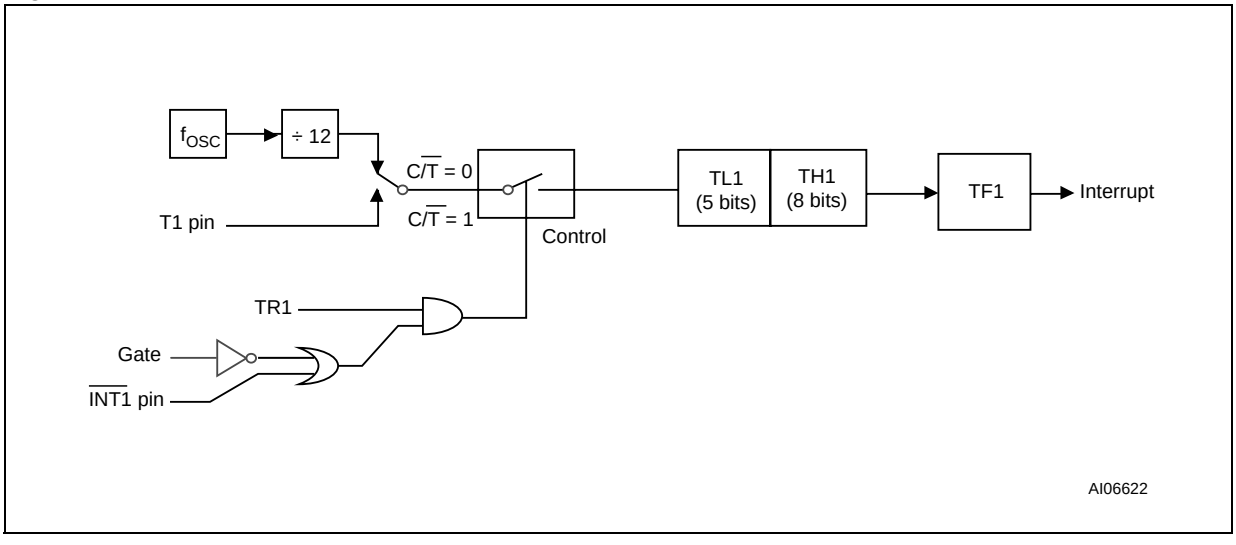
In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all '1s' to all '0s,' it sets the Timer Interrupt Flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or /INT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input /INT1, to facilitate pulse width measurements). TR1 is a control bit in the Special Function Register TCON (TCON Control Register). GATE is in 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 does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, and /INT0 for the corresponding Timer 1 signals in Figure 20. There are two different GATE Bits, one for Timer 1 and one for Timer 0.

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

Figure 20. Timer/Counter Mode 0: 13-bit Counter



Mode 2. Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 21. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

Timer 2

Like Timer 0 and 1, Timer 2 can operate as either an event timer or as an event counter. This is selected by Bit C/T2 in the special function register T2CON (see Table 39). It has three operating modes: Capture, Auto-reload, and Baud Rate Generator (see Table 40, page 51), which are selected by bits in the T2CON as shown in Table 41, page 51. In the Capture Mode there are two options which are selected by Bit EXEN2 in T2CON. if EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets Bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX 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 T2EX causes Bit EXF2 in T2CON to be set, and EXF2 like TF2 can generate an interrupt. The Capture Mode is illustrated in Figure 22, page 52.

In the Auto-reload Mode, there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The Auto-reload Mode is illustrated in Standard Serial Interface (UART) Figure 23, page 52. The Baud Rate Generation Mode is selected by (RCLK, RCLK1) = 1 and/or (TCLK, TCLK1) = 1. It will be described in conjunction with the serial port.

Figure 21. Timer/Counter Mode 2: 8-bit Auto-reload

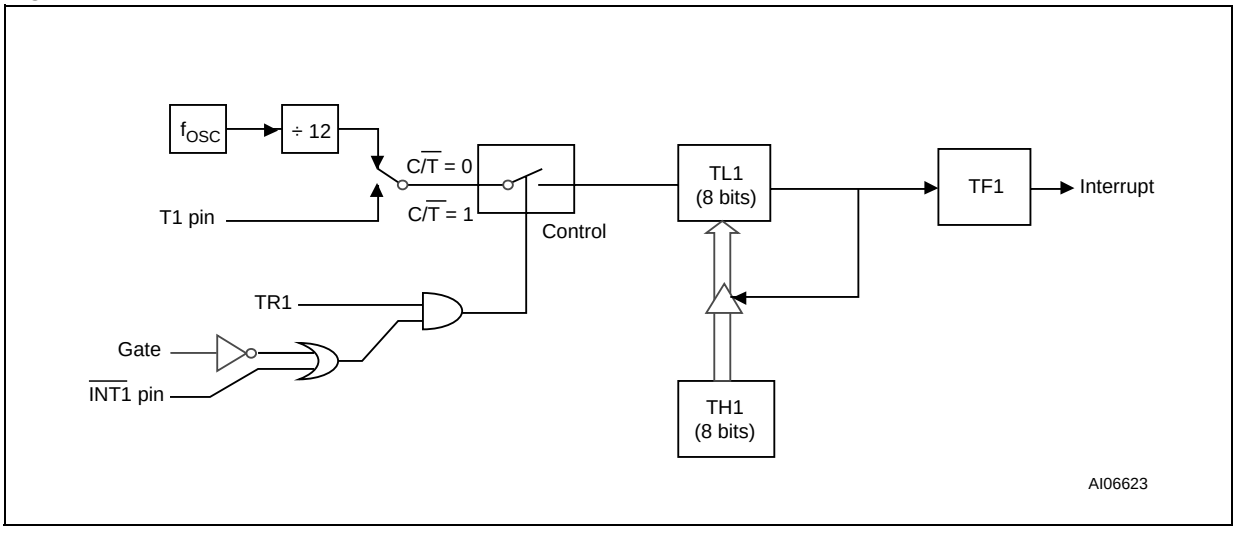


Table 39. Timer/Counter 2 Control Register (T2CON)

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Table 40. Timer/Counter 2 Operating Modes

Mode	T2CON			T2MOD DECN	T2CON EXEN	P1.1 T2EX	Remarks	Input Clock	
	RxCLK or TxCLK	CP/RL2	TR2					Internal	External (P1.0/T2)
16-bit Auto-reload	0	0	1	0	0	x	reload upon overflow	$f_{OSC}/12$	MAX $f_{OSC}/24$
	0	0	1	0	1	↓	reload trigger (falling edge)		
	0	0	1	1	x	0	Down counting		
	0	0	1	1	x	1	Up counting		
16-bit Capture	0	1	1	x	0	x	16-bit Timer/Counter (only up counting)	$f_{OSC}/12$	MAX $f_{OSC}/24$
	0	1	1	x	1	↓	Capture (TH1,TL2) → (RCAP2H,RCAP2L)		
Baud Rate Generator	1	x	1	x	0	x	No Overflow Interrupt Request (TF2)	$f_{OSC}/12$	MAX $f_{OSC}/24$
	1	x	1	x	1	↓	Extra External Interrupt (Timer 2)		
Off	x	x	0	x	x	x	Timer 2 stops	—	—

Note: ↓ = falling edge

Table 41. Description of the T2CON Bits

Bit	Symbol	Function
7	TF2	Timer 2 Overflow Flag. Set by a Timer 2 overflow, and must be cleared by software. TF2 will not be set when either (RCLK, RCLK1)=1 or (TCLK, TCLK)=1
6	EXF2	Timer 2 External Flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2=1. When Timer 2 Interrupt is enabled, EXF2=1 will cause the CPU to vector to the Timer 2 Interrupt routine. EXF2 must be cleared by software
5	RCLK(1)	Receive Clock Flag (UART 1). When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in Modes 1 and 3. TCLK=0 causes Timer 1 overflow to be used for the receive clock
4	TCLK(1)	Transmit Clock Flag (UART 1). When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in Modes 1 and 3. TCLK=0 causes Timer 1 overflow to be used for the transmit clock
3	EXEN2	Timer 2 External Enable Flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2=0 causes Time 2 to ignore events at T2EX
2	TR2	Start/stop control for Timer 2. A logic 1 starts the timer
1	C/$\overline{T}$2	Timer or Counter Select for Timer 2. Cleared for timer operation (input from internal system clock, t_{CPU}); set for external event counter operation (negative edge triggered)
0	CP/$\overline{RL}$2	Capture/Reload Flag. When set, capture will occur on negative transition of T2EX if EXEN2=1. When cleared, auto-reload will occur either with TImer 2 overflows, or negative transitions of T2EX when EXEN2=1. When either (RCLK, RCLK1)=1 or (TCLK, TCLK)=1, this bit is ignored, and timer is forced to auto-reload on Timer 2 overflow

Note: 1. The RCLK1 and TCLK1 Bits in the PCON Register control UART 2, and have the same function as RCLK and TCLK.

Figure 22. Timer 2 in Capture Mode

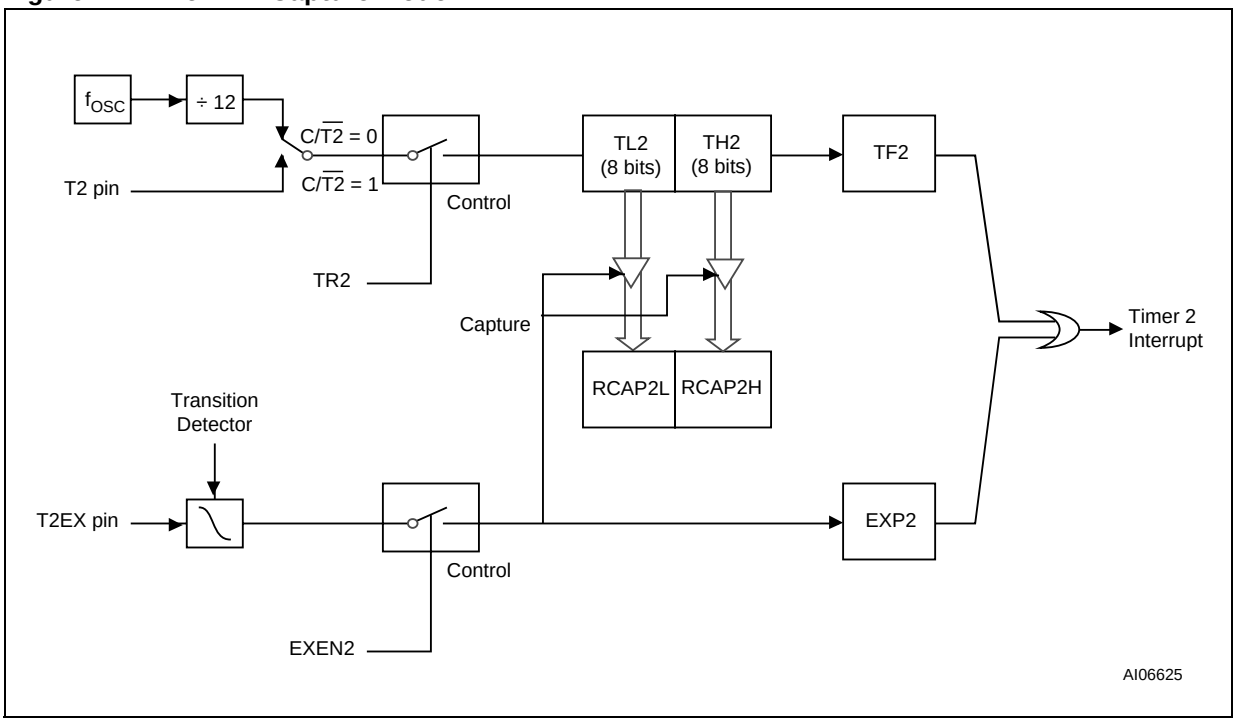
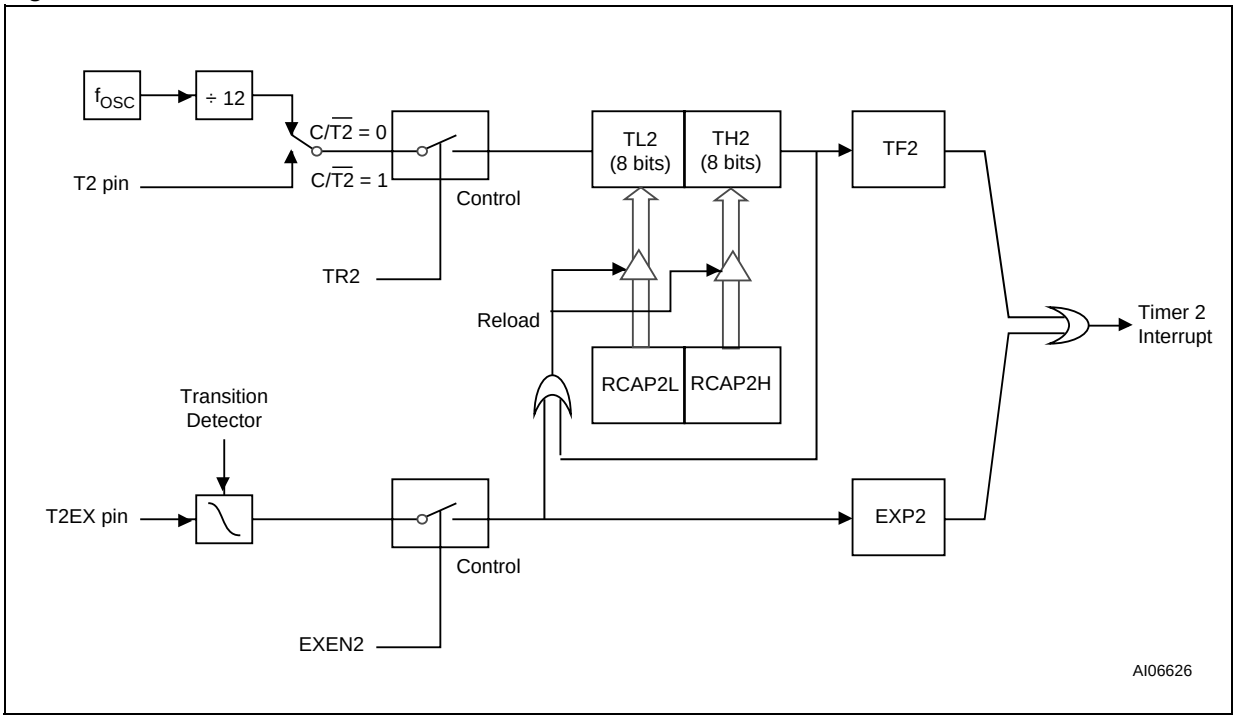


Figure 23. Timer 2 in Auto-Reload Mode

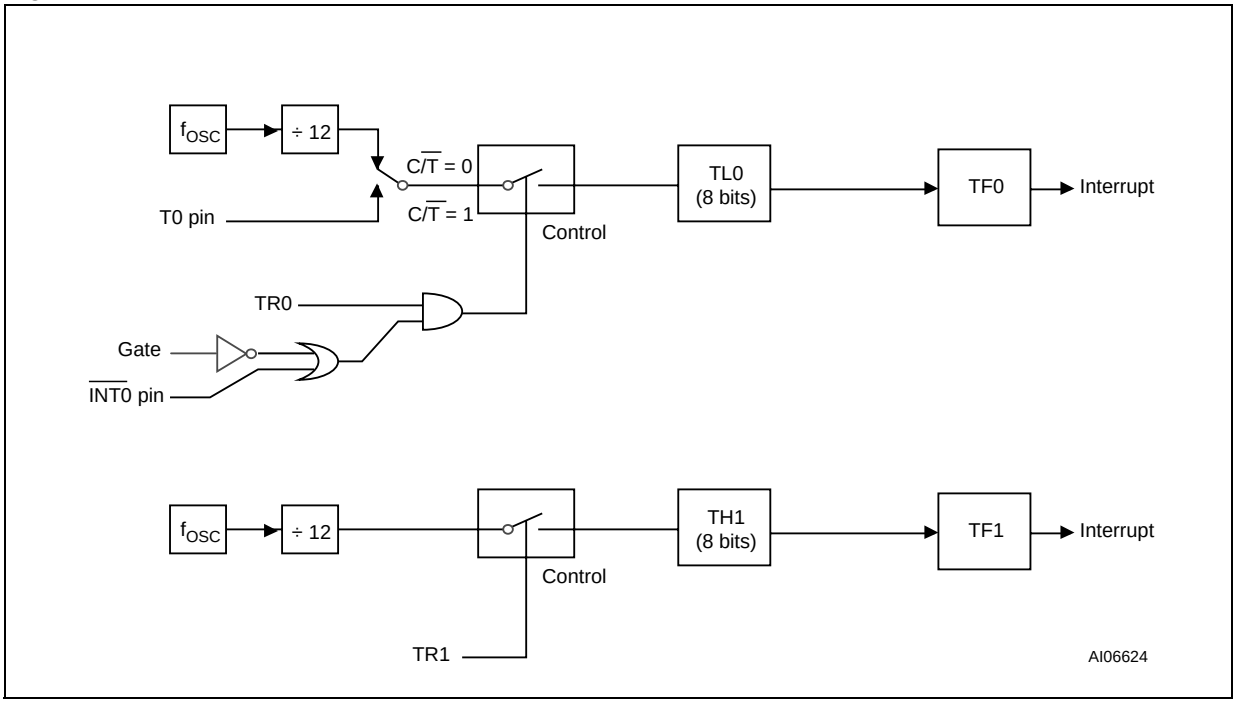


Mode 3. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 24. TL0 uses the Timer 0 control Bits: C/T, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the “Timer 1“Interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. With Timer 0 in Mode 3, an µPSD3251F device can look like it has three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

Figure 24. Timer/Counter Mode 3: Two 8-bit Counters



STANDARD SERIAL INTERFACE (UART)

The µPSD3251F device provides two standard 8032 UART serial ports. The first port is connected to pin P3.0 (RX) and P3.1 (TX). The second port is connected to pin P1.2 (RX) and P1.3(TX). The operation of the two serial ports are the same and are controlled by the SCON and SCON2 registers.

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still has not been read by the time reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF (or SBUF2 for the second serial port). Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

Mode 0. Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 the f_{OSC}.

Mode 1. 10 bits are transmitted (through TxD) or received (through RxD): a start Bit (0), 8 data bits (LSB first), and a Stop Bit (1). On receive, the Stop Bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2. 11 bits are transmitted (through TxD) or received (through RxD): start Bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a Stop Bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of '0' or '1.' Or, for example, the Parity Bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the Stop Bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3. 11 bits are transmitted (through TxD) or received (through RxD): a Start Bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a Stop Bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

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

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a Stop Bit. The port can be programmed such that when the Stop Bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting Bit SM2 in SCON. A way to use this feature in multi-processor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is '1' in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An ad-dress byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 Bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the Stop Bit. In a Mode 1 reception, if SM2 = 1, the Receive Interrupt will not be activated unless a valid Stop Bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON (SCON2 for the second port), shown in Figure 25. This register (see Tables 42 and 43) contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the Serial Port Interrupt Bits (TI and RI).

Figure 25. Serial Port Mode 0, Block Diagram

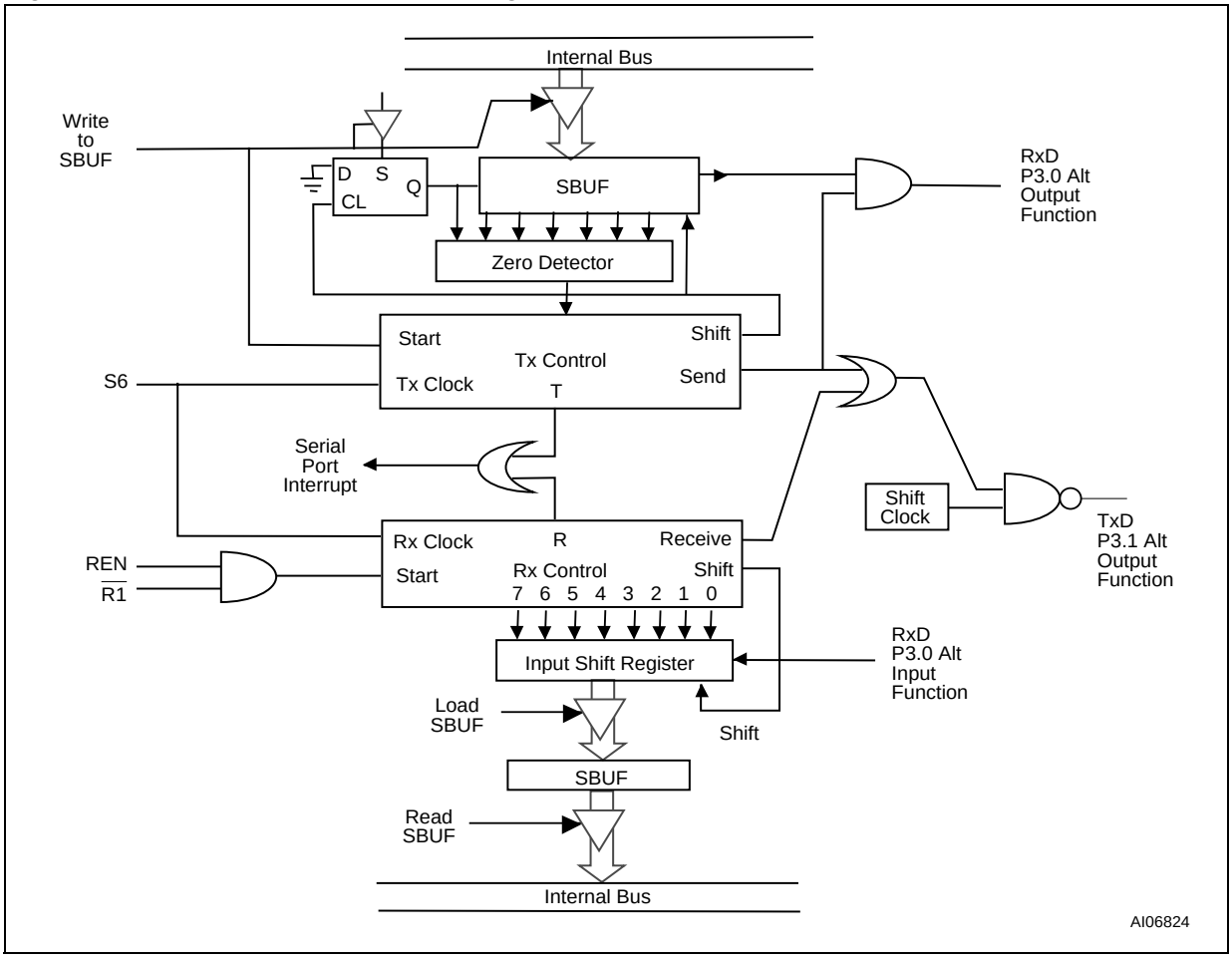


Table 42. Serial Port Control Register (SCON)

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Table 43. Description of the SCON Bits

Bit	Symbol	Function
7	SM0	(SM1,SM0)=(0,0): Shift Register. Baud rate = $f_{OSC}/12$ (SM1,SM0)=(1,0): 8-bit UART. Baud rate = variable (SM1,SM0)=(0,1): 8-bit UART. Baud rate = $f_{OSC}/64$ or $f_{OSC}/32$ (SM1,SM0)=(1,1): 8-bit UART. Baud rate = variable
6	SM1	
5	SM2	Enables the multiprocessor communication features in Mode 2 and 3. In Mode 2 or 3, if SM2 is set to '1,' RI will not be activated if its received 8th data bit (RB8) is '0.' In Mode 1, if SM2=1, RI will not be activated if a valid Stop Bit was not received. In Mode 0, SM2 should be '0'
4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception
3	TB8	The 8th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired
2	RB8	In Modes 2 and 3, this bit contains the 8th data bit that was received. In Mode 1, if SM2=0, RB8 is the Snap Bit that was received. In Mode 0, RB8 is not used
1	TI	Transmit Interrupt Flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the Stop Bit in the other modes, in any serial transmission. Must be cleared by software
0	RI	Receive Interrupt Flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the Stop Bit in the other modes, in any serial reception (except for SM2). Must be cleared by software

Baud Rates. The baud rate in Mode 0 is fixed:

Mode 0 Baud Rate = f_{OSC} / 12

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

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

In the µPSD3251F device, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

Using Timer 1 to Generate Baud Rates. When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows (see Table 44, page 58):

Mode 1,3 Baud Rate =

(2^{SMOD} / 32) x (Timer 1 overflow rate)

The Timer 1 Interrupt should be disabled in this application. The Timer itself can be configured for either “timer” or “counter” operation, and in any of its 3 running modes. In the most typical applications, it is configured for “timer” operation, in the Auto-reload Mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1,3 Baud Rate =

(2^{SMOD} / 32) x (f_{OSC} / (12 x [256 – (TH1)]))

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 Interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 Interrupt to do a 16-bit software reload. Figure 20 lists various commonly used baud rates and how they can be obtained from Timer 1.

Using Timer/Counter 2 to Generate Baud Rates. In the µPSD3251F device, Timer 2 selected as the baud rate generator by setting TCLK and/or RCLK (see Figure 20, page 49 Timer/ Counter 2 Control Register (T2CON)).

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

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

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

Now, the baud rates in Modes 1 and 3 are determined at Timer 2’s overflow rate as follows:

Mode 1,3 Baud Rate = Timer 2 Overflow Rate / 16

The timer can be configured for either “timer” or “counter” operation. In the most typical applications, it is configured for “timer” operation (C/T2 = 0). “Timer” operation is a little different for Timer 2 when it’s being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at the 1/6 the CPU clock frequency). In the case, the baud rate is given by the formula:

Mode 1,3 Baud Rate = f_{OSC} / (32 x [65536 - (RCAP2H, RCAP2L)]

where (RCAP2H, RCAP2L) is the content of RC2H and RC2L taken as a 16-bit unsigned integer.

Timer 2 also be used as the Baud Rate Generating Mode. This mode is valid only if RCLK + TCLK = 1 in T2CON or in PCON.

Note: A roll-over in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer interrupt does not have to be disabled when Timer 2 is in the Baud Rate Generator Mode.

Note: If EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in “timer” function in the Baud Rate Generator Mode, one should not try to READ or WRITE TH2 or TL2. Under these conditions the timer is being incremented every state time, and the results of a READ or WRITE may not be accurate. The RC registers may be read, but should not be written to, because a WRITE might overlap a reload and cause WRITE and/or reload errors. Turn the timer off (clear TR2) before accessing the Timer 2 or RC registers, in this case.

Table 44. Timer 1-Generated Commonly Used Baud Rates

Baud Rate	f_{OSC}	SMOD	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1MHz	12MHz	X	X	X	X
Mode 2 Max: 375K	12MHz	1	X	X	X
Modes 1, 3: 62.5K	12MHz	1	0	2	FFh
19.2K	11.059MHz	1	0	2	FDh
9.6K	11.059MHz	0	0	2	FDh
4.8K	11.059MHz	0	0	2	FAh
2.4K	11.059MHz	0	0	2	F4h
1.2K	11.059MHz	0	0	2	E8h
137.5	11.059MHz	0	0	2	1Dh
110	6MHz	0	0	2	72h
110	12MHz	0	0	1	FEEBh

More About Mode 0. Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the f_{OSC}.

Figure 25, page 55 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “WRITE to SBUF” signal at S6P2 also loads a '1' into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between “WRITE to SBUF” and activation of SEND.

SEND enables the output of the shift register to the alternate out-put function line of RxD and also enable SHIFT CLOCK to the alternate output function line of TxD. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

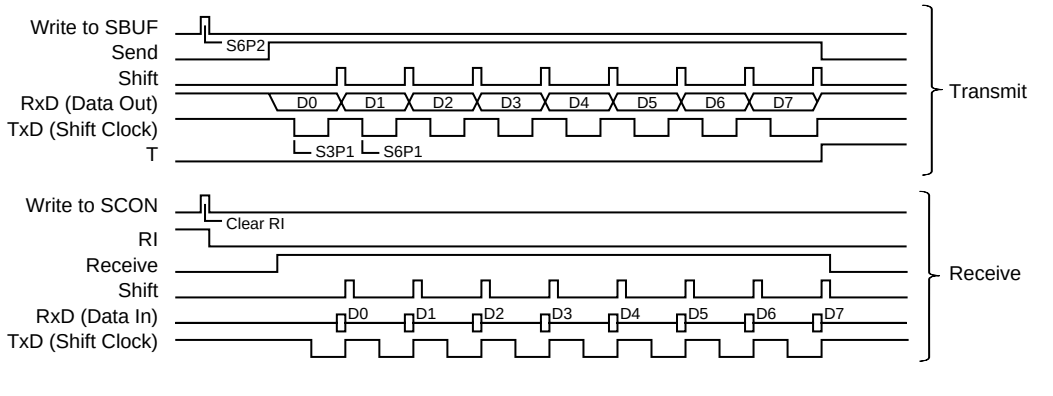
As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the '1' that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1. Both of these actions occur at S1P1 of the 10th machine cycle after “WRITE to SBUF.”

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of TxD. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the RxD pin at S5P2 of the same machine cycle.

As data bits come in from the right, '1s' shift out to the left. When the '0' that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the WRITE to SCON that cleared RI, RECEIVE is cleared as RI is set.

Figure 26. Serial Port Mode 0, Waveforms



More About Mode 1. Ten bits are transmitted (through TxD), or received (through RxD): a start Bit (0), 8 data bits (LSB first). and a Stop Bit (1). On receive, the Stop Bit goes into RB8 in SCON. In the µPSD3251F device the baud rate is determined by the Timer 1 or Timer 2 overflow rate.

Figure 27, page 60 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “WRITE to SBUF” signal also loads a '1' into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the “WRITE to SBUF” signal.)

The transmission begins with activation of SEND which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left (see Figure 28, page 60). When the MSB of the data byte is at the output position of the shift register, then the '1' that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after “WRITE to SBUF.”

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

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

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

1. R1 = 0, and

2. Either SM2 = 0, or the received Stop Bit = 1.

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

Figure 27. Serial Port Mode 1, Block Diagram

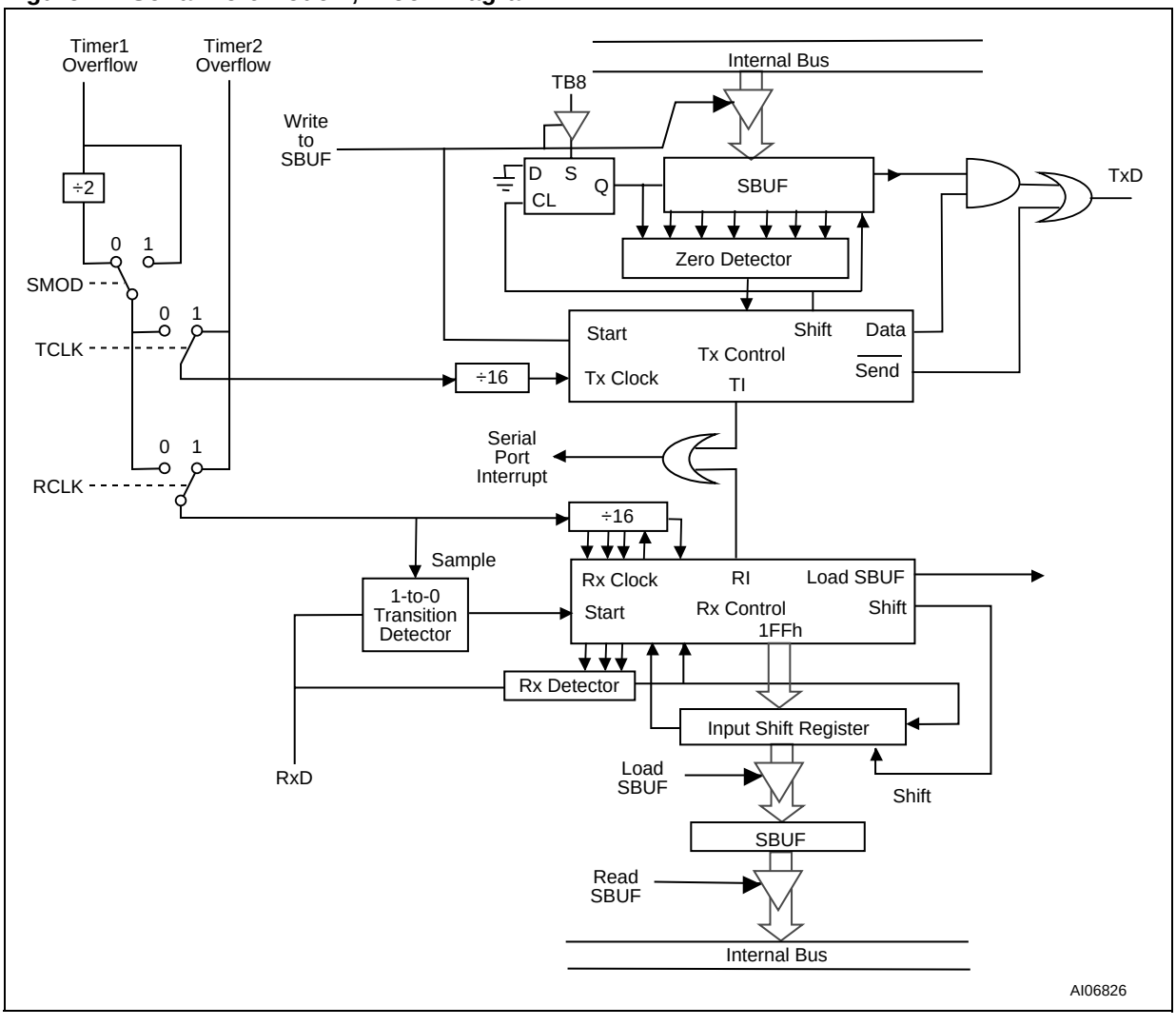
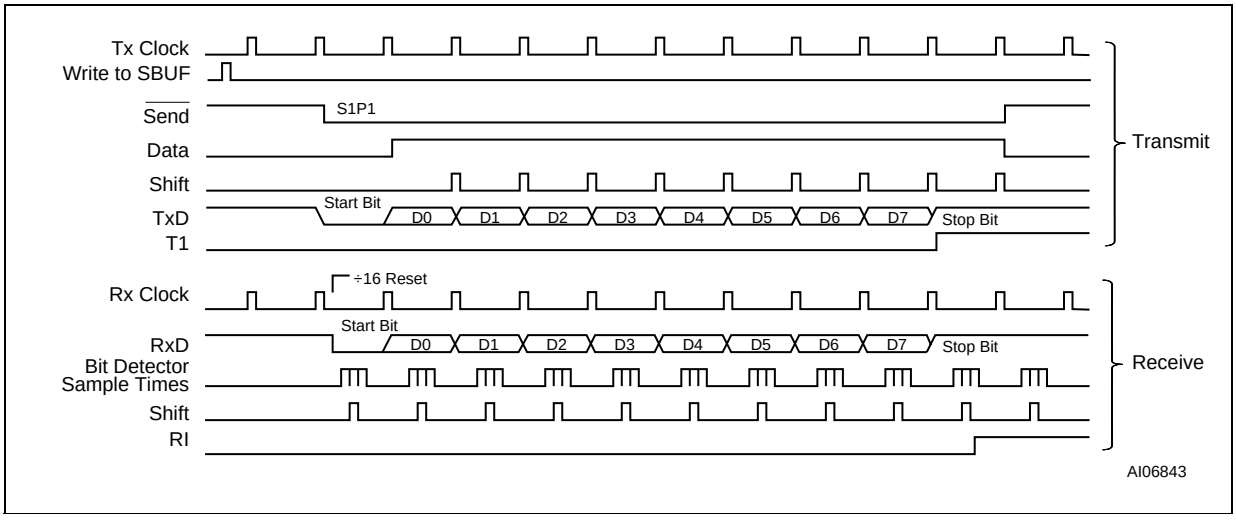


Figure 28. Serial Port Mode 1, Waveforms



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

Figure 29, page 62 and Figure 31, page 63 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

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

The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that (see Figure 30, page 62 and Figure 32, page 63). The first shift clocks a '1' (the Stop Bit) into the 9th bit position of the shift register. There-after, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the out-put position of the shift register, then the Stop Bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by 16 rollover after “WRITE to SUBF.”

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

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

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

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

1. RI = 0, and

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

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

Figure 29. Serial Port Mode 2, Block Diagram

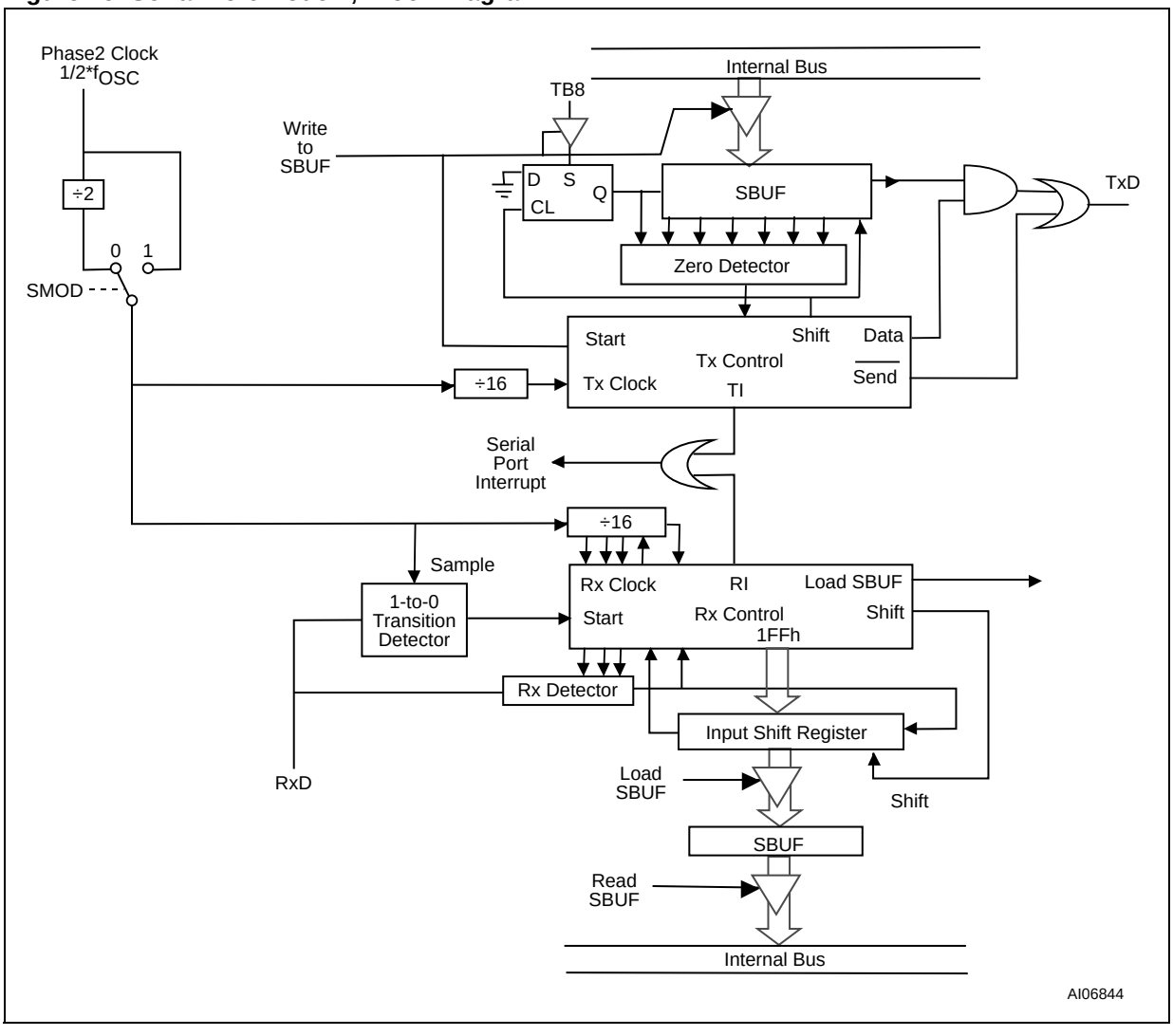


Figure 30. Serial Port Mode 2, Waveforms

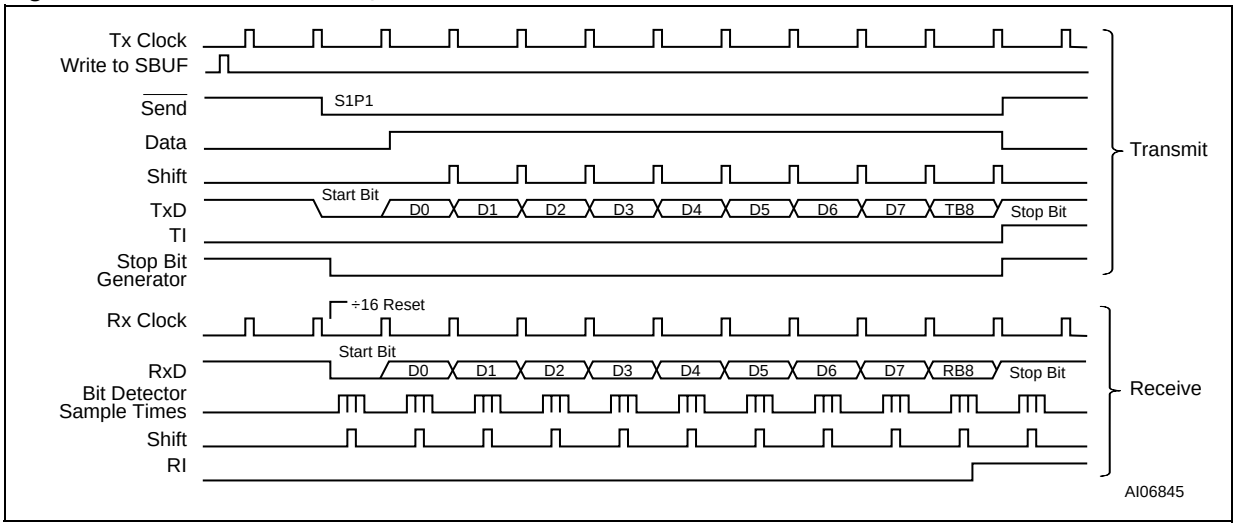


Figure 31. Serial Port Mode 3, Block Diagram

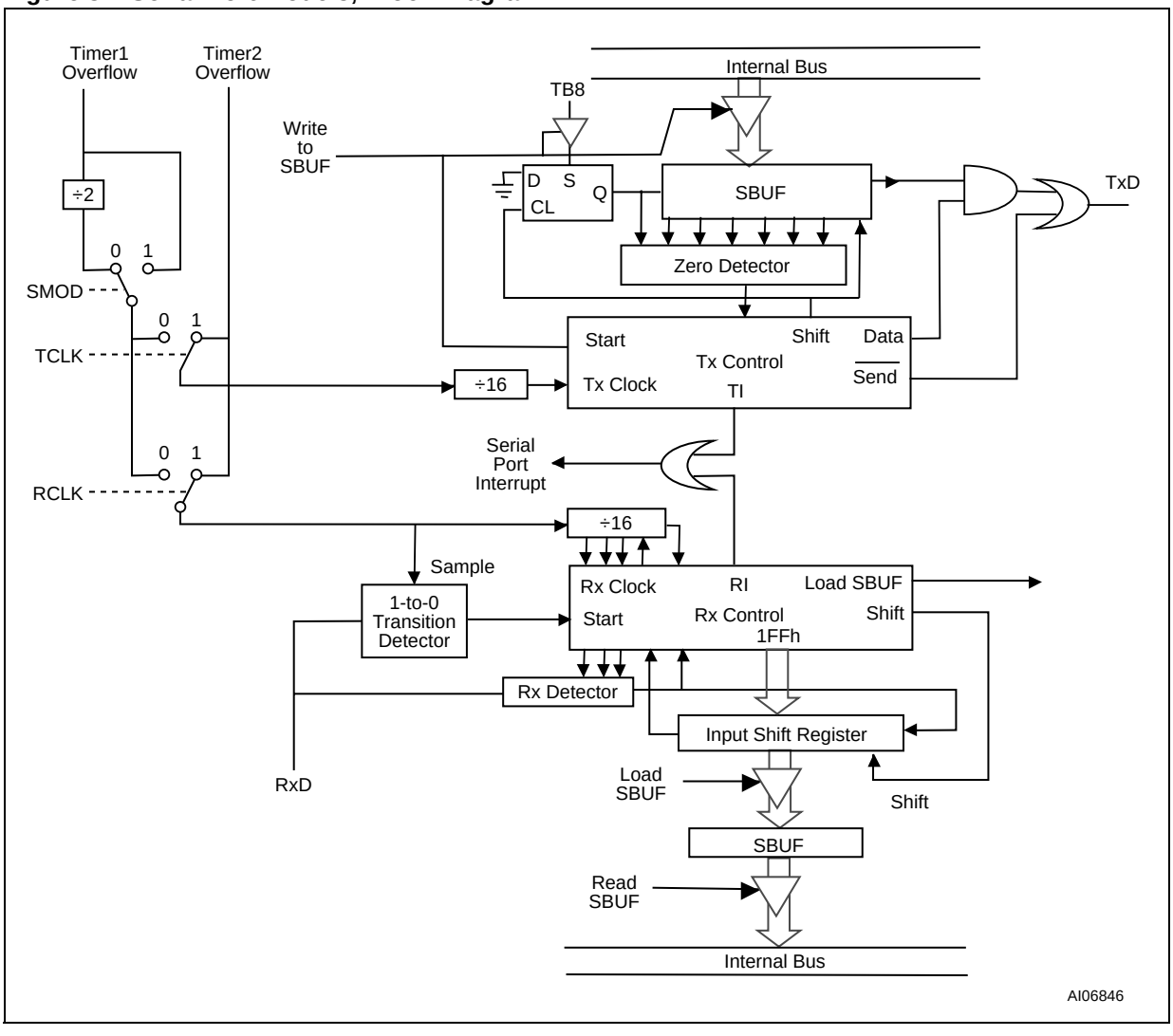
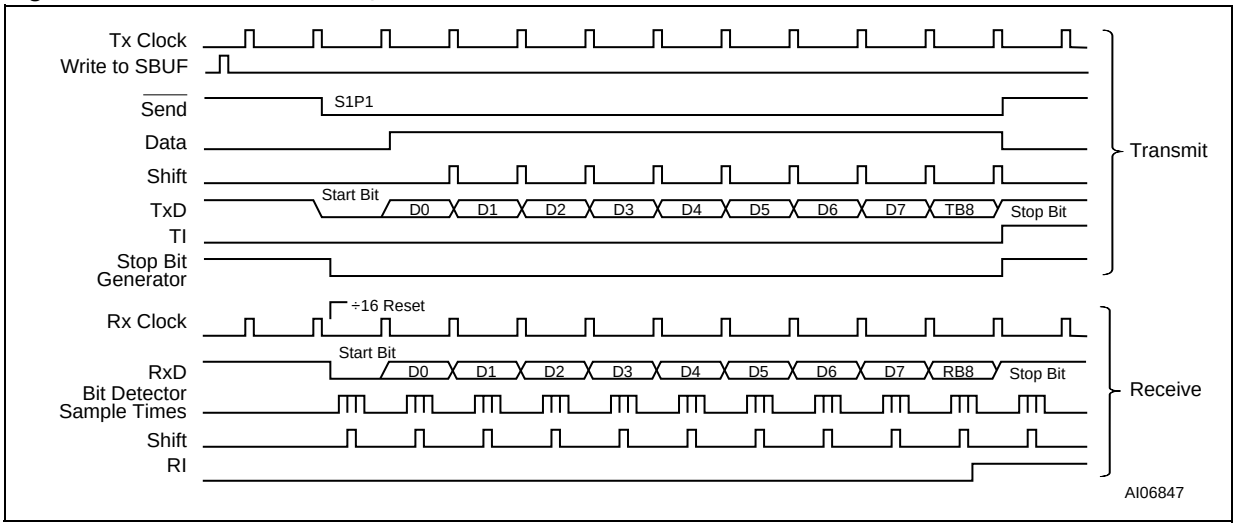


Figure 32. Serial Port Mode 3, Waveforms



ANALOG-TO-DIGITAL CONVERTOR (ADC)

The analog to digital (A/D) converter allows conversion of an analog input to a corresponding 8-bit digital value. The A/D module has four analog inputs, which are multiplexed into one sample and hold. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog supply voltage is connected to AVREF of ladder resistance of A/D module.

The A/D module has two registers which are the control register ACON and A/D result register ADAT. The register ACON, shown in Table 45 and Table 46, page 65, controls the operation of the A/D converter module. To use analog inputs, I/O is selected by P1SFS register. Also an 8-bit prescaler ASCL divides the main system clock input down to approximately 6MHz clock that is required for the ADC logic. Appropriate values need to be loaded into the prescaler based upon the main MCU clock frequency prior to use.

The processing of conversion starts when the Start Bit ADST is set to '1.' After one cycle, it is cleared by hardware. The register ADAT contains the results of the A/D conversion. When conversion is completed, the result is loaded into the ADAT the A/D Conversion Status Bit ADSF is set to '1.'

The block diagram of the A/D module is shown in Figure 33. The A/D Status Bit ADSF is set automatically when A/D conversion is completed, cleared when A/D conversion is in process.

The ASCL should be loaded with a value that results in a clock rate of approximately 6MHz for the ADC using the following formula (see Table 47, page 65):

ADC clock input = (f_{OSC} / 2) / (Prescaler register value +1)

Where f_{OSC} is the MCU clock input frequency

The conversion time for the ADC can be calculated as follows:

ADC Conversion Time = 8 clock * 8bits * (ADC Clock) ~= 10.67usec (at 6MHz)

ADC Interrupt

The ADSF Bit in the ACON register is set to '1' when the A/D conversion is complete. The status bit can be driven by the MCU, or it can be configured to generate a falling edge interrupt when the conversion is complete.

The ADSF Interrupt is enabled by setting the ADSFINT Bit in the PCON register. Once the bit is set, the external INT1 Interrupt is disabled and the ADSF Interrupt takes over as INT1. INT1 must be configured as if it is an edge interrupt input. The INP1 pin (p3.3) is available for general I/O functions, or Timer1 gate control.

Figure 33. A/D Block Diagram

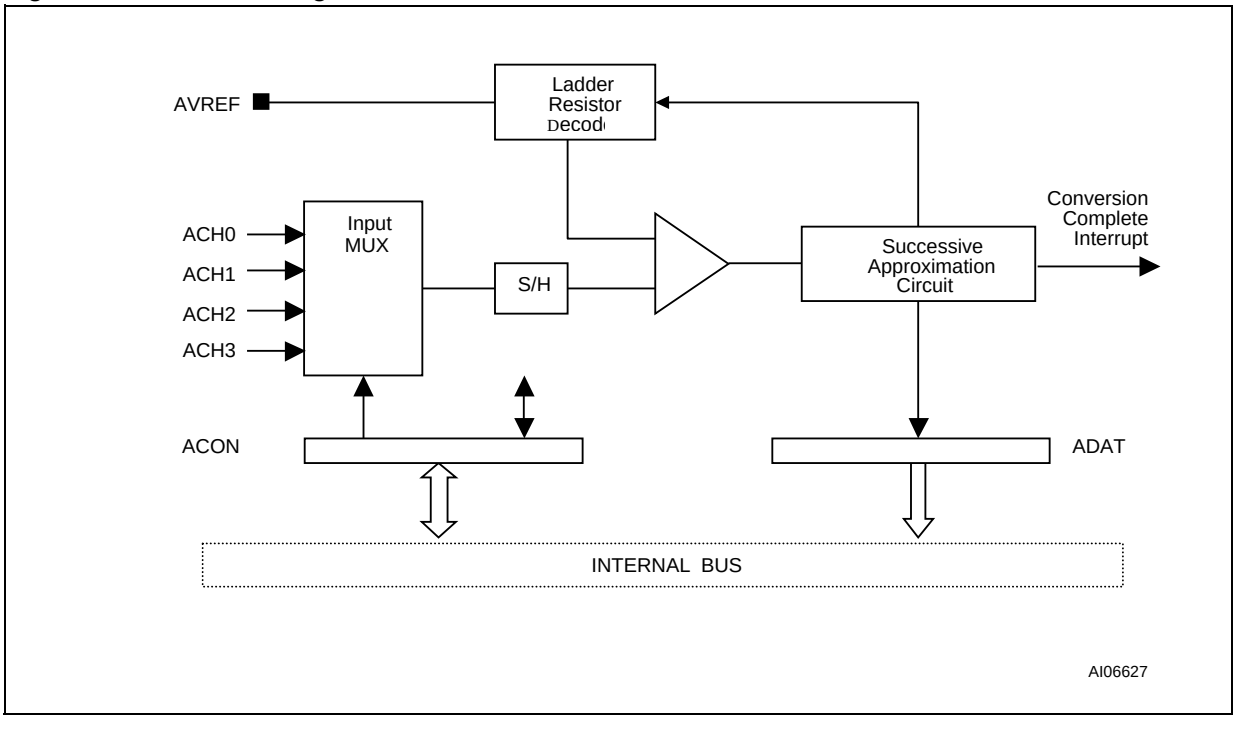


Table 45. ADC SFR Memory Map

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register

Table 46. Description of the ACON Bits

Bit	Symbol	Function
7 to 6	—	Reserved
5	ADEN	ADC Enable Bit: 0: ADC shut off and consumes no operating current 1: enable ADC
4	—	Reserved
3 to 2	ADS1, ADS0	Analog channel select
	0, 0	Channel0 (ACH0)
	0, 1	Channel1 (ACH1)
	1, 0	Channel2 (ACH2)
	1, 1	Channel3 (ACH3)
1	ADST	ADC Start Bit: 0: force to zero 1: start an ADC; after one cycle, bit is cleared to '0'
0	ADSF	ADC Status Bit: 0: A/D conversion is in process 1: A/D conversion is completed, not in process

Table 47. ADC Clock Input

MCU Clock Frequency	Prescaler Register Value	ADC Clock
40MHz	2	6.7MHz
36MHz	2	6MHz
24MHz	1	6MHz
12MHz	0	6MHz

I²C INTERFACE

The serial port supports the twin line I²C-bus, consisting of a data line (SDA1), and a clock line (SCL1) as shown in Figure 34. Depending on the configuration, the SDA1 and SCL1 lines may require pull-up resistors.

These lines also function as I/O port lines if the I²C bus is not enabled.

The system is unique because data transport, clock generation, address recognition, and bus control arbitration are all controlled by hardware.

The I²C serial I/O has complete autonomy in byte handling and operates in 4 modes.

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the SFRs (see Tables 48, 49, and Table 50, page 67):

– S2CON: the control of byte handling and the operation of 4 mode.

– S2STA: the contents of its register may also be used as a vector to various service routines.

– S2DAT: data shift register.

– S2ADR: slave address register. Slave address recognition is performed by On-Chip H/W.

Figure 34. Block Diagram of the I²C Bus Serial I/O

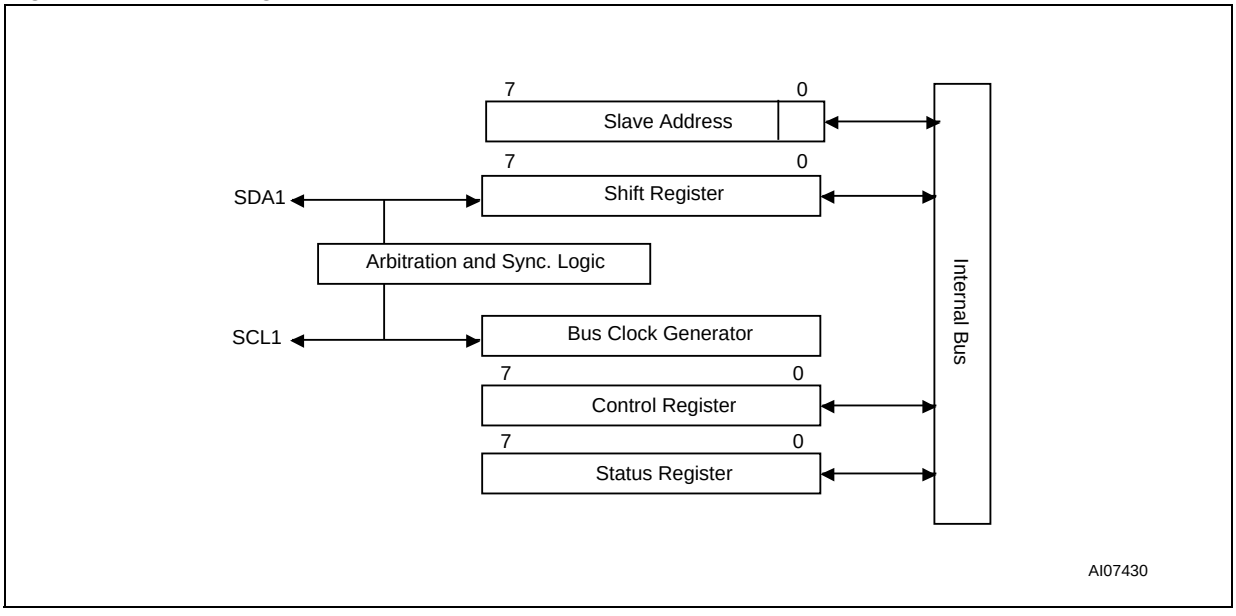


Table 48. Serial Control Register (S2CON)

7	6	5	4	3	2	1	0
CR2	ENII	STA	STO	ADDR	AA	CR1	CR0

Table 49. Description of the S2CON Bits

Bit	Symbol	Function
7	CR2	This bit along with Bits CR1and CR0 determines the serial clock frequency when SIO is in the Master Mode.
6	ENII	Enable IIC. When ENI1 = 0, the IIC is disabled. SDA and SCL outputs are in the high impedance state.
5	STA	START Flag. When this bit is set, the SIO H/W checks the status of the I^2C-bus and generates a START condition if the bus free. If the bus is busy, the SIO will generate a repeated START condition when this bit is set.
4	STO	STOP Flag. With this bit set while in Master Mode a STOP condition is generated. When a STOP condition is detected on the I^2C bus, the I^2C hardware clears the STO Flag. **Note:** This bit have to be set before 1 cycle interrupt period of STOP. That is, if this bit is set, STOP condition in Master Mode is generated after 1 cycle interrupt period.
3	ADDR	This bit is set when address byte was received. Must be cleared by software.
2	AA	Acknowledge enable signal. If this bit is set, an acknowledge (low level to SDA) is returned during the acknowledge clock pulse on the SCL line when: • Own slave address is received • A data byte is received while the device is programmed to be a Master Receiver • A data byte is received while the device is a selected Slave Receiver. When this bit is reset, no acknowledge is returned. SIO release SDA line as high during the acknowledge clock pulse.
1	CR1	These two bits along with the CR2 Bit determine the serial clock frequency when SIO is in the Master Mode.
0	CR0	

Table 50. Selection of the Serial Clock Frequency SCL in Master Mode

CR2	CR1	CR0	f_{OSC} Divisor	Bit Rate (kHz) at f_{OSC}			
				12MHz	24MHz	36MHz	40MHz
0	0	0	16	375	750	X	X
0	0	1	24	250	500	750	833
0	1	0	30	200	400	600	666
0	1	1	60	100	200	300	333
1	0	0	120	50	100	150	166
1	0	1	240	25	50	75	83
1	1	0	480	12.5	25	37.5	41
1	1	1	960	6.25	12.5	18.75	20

Serial Status Register (S2STA)

S2STA is a “Read-only” register. The contents of this register may be used as a vector to a service routine. This optimized the response time of the software and consequently that of the I^2C bus. The status codes for all possible modes of the I^2C bus interface are given Table 52.

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

1. Own slave address has been received during AA = 1: ack_int
2. The general call address has been received while GC(S2ADR.0) = 1 and AA = 1:
3. A data byte has been received or transmitted in Master Mode (even if arbitration is lost): ack_int
4. A data byte has been received or transmitted as selected slave: ack_int
5. A stop condition is received as selected slave receiver or transmitter: stop_int

Data Shift Register (S2DAT)

S2DAT contains the serial data to be transmitted or data which has just been received. The MSB (Bit 7) is transmitted or received first; that is, data shifted from right to left.

Table 51. Serial Status Register (S2STA)

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

Table 52. Description of the S2STA Bits

Bit	Symbol	Function
7	GC	General Call Flag
6	STOP	Stop Flag. This bit is set when a STOP condition is received
5	INTR(1,2)	Interrupt Flag. This bit is set when an I²C Interrupt condition is requested
4	TX_MODE	Transmission Mode Flag. This bit is set when the I²C is a transmitter; otherwise this bit is reset
3	BBUSY	Bus Busy Flag. This bit is set when the bus is being used by another master; otherwise, this bit is reset
2	BLOST	Bus Lost Flag. This bit is set when the master loses the bus contention; otherwise this bit is reset
1	/ACK_REP	Acknowledge Response Flag. This bit is set when the receiver transmits the not acknowledge signal This bit is reset when the receiver transmits the acknowledge signal
0	SLV	Slave Mode Flag. This bit is set when the I²C plays role in the Slave Mode; otherwise this bit is reset

Note: 1. Interrupt Flag Bit (INTR, S2STA Bit 5) is cleared by Hardware as reading S2STA register.
2. I^2C Interrupt Flag (INTR) can occur in below case.

Table 53. Data Shift Register (S2DAT)

7	6	5	4	3	2	1	0
S2DAT7	S2DAT6	S2DAT5	S2DAT4	S2DAT3	S2DAT2	S2DAT1	S2DAT0

Address Register (S2ADR)

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as a slave receive/transmitter.

The Start/Stop Hold Time Detection and System Clock registers (Tables 55 and 56) are included in the I^2C unit to specify the start/stop detection time to work with the large range of MCU frequency values supported. For example, with a system clock of 40MHz.

Table 54. Address Register (S2ADR)

7	6	5	4	3	2	1	0
SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	SLA0	—

Note: SLA6 to SLA0: Own slave address.

Table 55. Start /Stop Hold Time Detection Register (S2SETUP)

	Address	Register Name	Reset Value	Note
SFR	D2h	S2SETUP	00h	To control the start/stop hold time detection for the multi-master I²C module in Slave Mode

Table 56. System Cock of 40MHz

S1SETUP, S2SETUP Register Value	Number of Sample Clock (f_{OSC}/2 – > 50ns)	Required Start/ Stop Hold Time	Note
00h	1EA	50ns	When Bit 7 (enable bit) = 0, the number of sample clock is 1EA (ignore Bit 6 to Bit 0)
80h	1EA	50ns	
81h	2EA	100ns	
82h	3EA	150ns	
...	...	...	
8Bh	12EA	600ns	Fast Mode I²C Start/Stop hold time specification
...	...	...	
FFh	128EA	6000ns	

Table 57. System Clock Setup Examples

System Clock	S1SETUP, S2SETUP Register Value	Number of Sample Clock	Required Start/Stop Hold Time
40MHz (f_{OSC}/2 – > 50ns)	8Bh	12 EA	600ns
30MHz (f_{OSC}/2 – > 66.6ns)	89h	9 EA	600ns
20MHz (f_{OSC}/2 – > 100ns)	86h	6 EA	600ns
8MHz (f_{OSC}/2 – > 250ns)	83h	3 EA	750ns

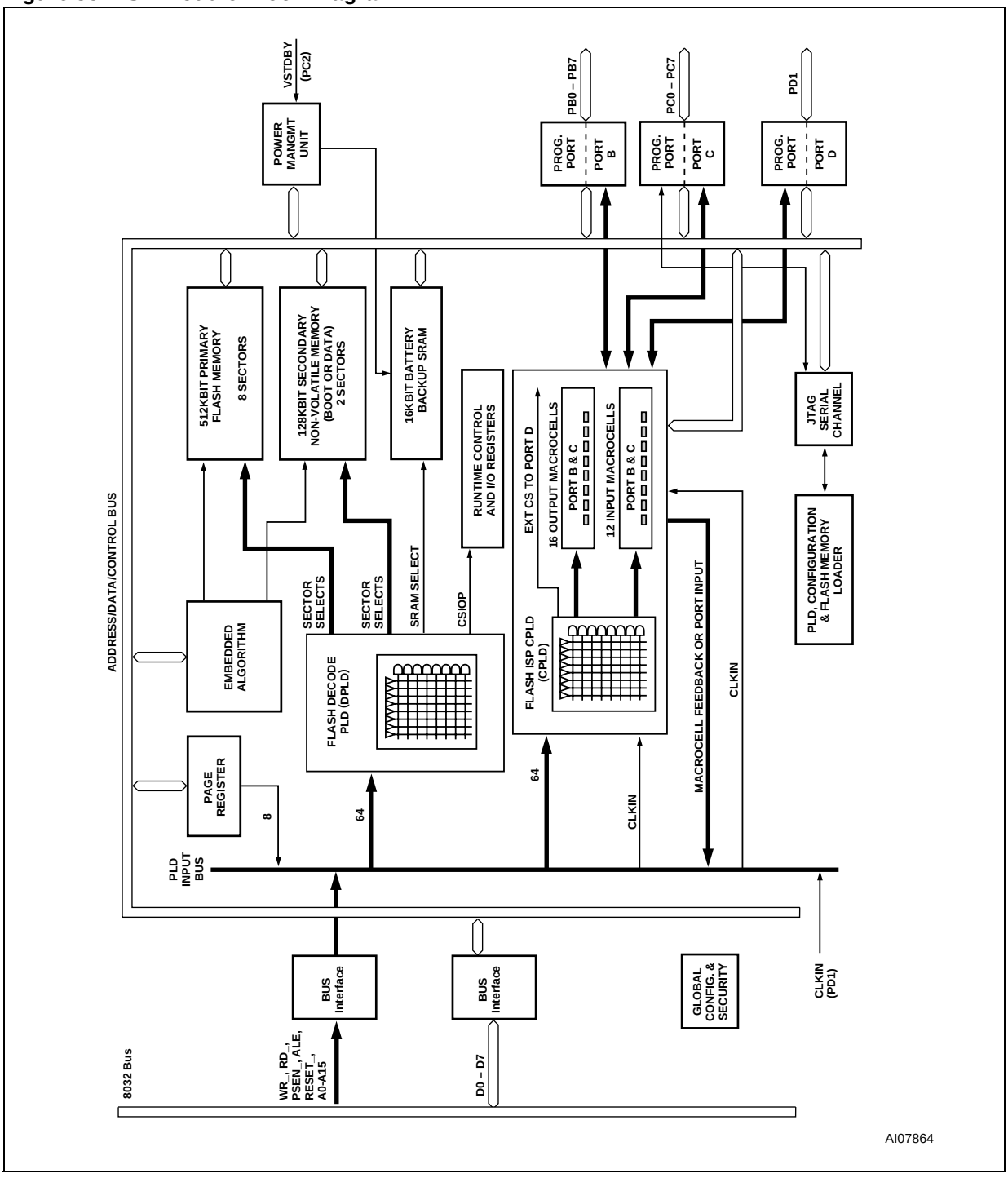
PSD MODULE

- The PSD Module provides configurable Program and Data memories to the 8032 CPU core (MCU). In addition, it has its own set of I/O ports and a PLD with 16 macrocells for general logic implementation.
- Ports B, C, and D are general purpose programmable I/O ports that have a port architecture which is different from the I/O ports in the MCU Module.
- The PSD Module communicates with the MCU Module through the internal address, data bus (A0-A15, D0-D7) and control signals (RD, WR, PSEN, ALE, RESET). The user defines the Decoding PLD in the PSDsoft Development Tool and can map the resources in the PSD Module to any program or data address space. Figure 35 shows the functional blocks in the PSD Module.

Functional Overview

- 512Kbit Flash memory. This is the main Flash memory. It is divided into 4 sectors (16KBytes each) that can be accessed with user-specified addresses.
- Secondary 128Kbit Flash boot memory. It is divided into 2 sectors (8KBytes each) that can be accessed with user-specified addresses. This secondary memory brings the ability to execute code and update the main Flash *concurrently.*
- 16Kbit SRAM. The SRAM's contents can be protected from a power failure by connecting an external battery.
- CPLD with 16 Output Micro Cells (OMCs) and up to 12 Input Micro Cells (IMCs). The CPLD may be used to efficiently implement a variety of logic functions for internal and external control. Examples include state machines, loadable shift registers, and loadable counters.
- Decode PLD (DPLD) that decodes address for selection of memory blocks in the PSD Module.
- Configurable I/O ports (Port B, C, and D) that can be used for the following functions:
  - MCU I/Os
  - PLD I/Os
  - Latched MCU address output
  - Special function I/Os.
  - I/O ports may be configured as open drain outputs.
- Built-in JTAG compliant serial port allows full-chip, In-System Programmability (ISP). With it, you can program a blank device or reprogram a device in the factory or the field.
- Internal page register that can be used to expand the 8032 MCU Module address space by a factor of 256.
- Internal programmable Power Management Unit (PMU) that supports a low-power mode called Power-down Mode. The PMU can automatically detect a lack of the 8032 CPU core activity and put the PSD Module into Power-down Mode.
- Erase/WRITE cycles:
  - Flash memory - 100,000 minimum
  - PLD - 1,000 minimum
  - Data Retention: 15 year minimum (for Main Flash memory, Boot, PLD and Configuration bits)

Figure 35. PSD Module Block Diagram



In-System Programming (ISP)

Using the JTAG signals on Port C, the entire PSD Module device can be programmed or erased without the use of the MCU. The primary Flash memory can also be programmed in-system by the MCU executing the programming algorithms out of the secondary memory, or SRAM. The secondary memory can be programmed the same way by executing out of the primary Flash memory. The PLD or other PSD Module Configuration blocks can be programmed through the JTAG port or a device programmer. Table 58 indicates which programming methods can program different functional blocks of the PSD Module.

Table 58. Methods of Programming Different Functional Blocks of the PSD Module

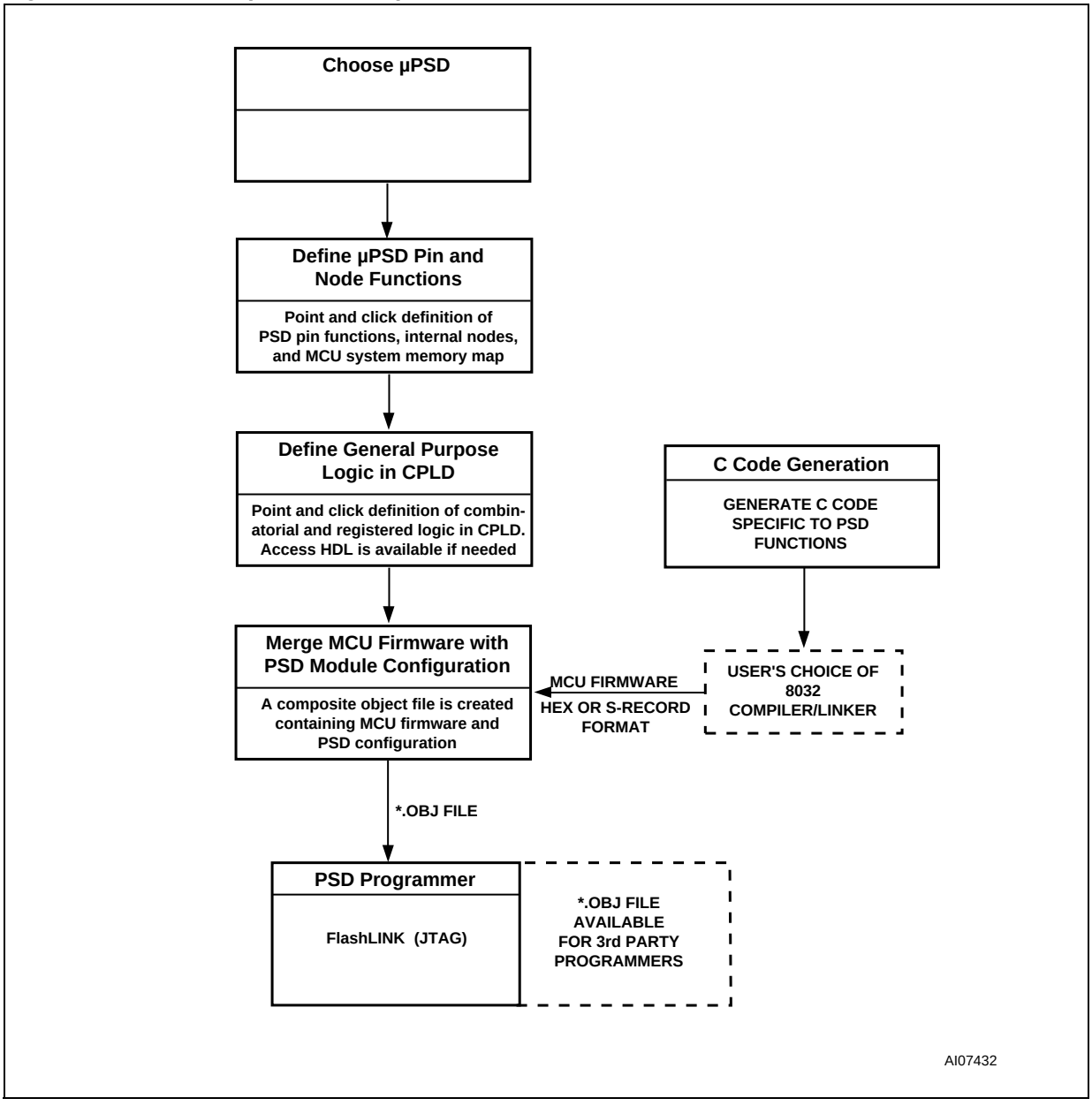
Functional Block	JTAG Programming	Device Programmer	IAP
Primary Flash Memory	Yes	Yes	Yes
Secondary Flash Memory	Yes	Yes	Yes
PLD Array (DPLD and CPLD)	Yes	Yes	No
PSD Module Configuration	Yes	Yes	No

DEVELOPMENT SYSTEM

The µPSD3200 is supported by PSDsoft, a Windows-based software development tool (Windows-95, Windows-98, Windows-NT). A PSD Module design is quickly and easily produced in a point and click environment. The designer does not need to enter Hardware Description Language (HDL) equations, unless desired, to define PSD Module pin functions and memory map information. The general design flow is shown in Figure 36. PSDsoft is available from our web site (the address is given on the back page of this data sheet) or other distribution channels.

PSDsoft directly supports a low cost device programmer from ST: FlashLINK (JTAG). The programmer may be purchased through your local distributor/representative. The µPSD3200 is also supported by third party device programmers. See our web site for the current list.

Figure 36. PSDsoft Express Development Tool



PSD MODULE REGISTER DESCRIPTION AND ADDRESS OFFSET

Table 59 shows the offset addresses to the PSD Module registers relative to the CSIOP base address. The CSIOP space is the 256 bytes of address that is allocated by the user to the internal PSD Module registers. Table 59 provides brief descriptions of the registers in CSIOP space. The following section gives a more detailed description.

Table 59. Register Address Offset

Register Name	Port B	Port C	Port D	Other[1]	Description
Data In	01	10	11		Reads Port pin as input, MCU I/O Input Mode
Control	03				Selects mode between MCU I/O or Address Out
Data Out	05	12	13		Stores data for output to Port pins, MCU I/O Output Mode
Direction	07	14	15		Configures Port pin as input or output
Drive Select	09	16	17		Configures Port pins as either CMOS or Open Drain on some pins, while selecting high slew rate on other pins.
Input Macrocell	0B	18			Reads Input Macrocells
Enable Out	0D	1A	1B		Reads the status of the output enable to the I/O Port driver
Output Macrocells AB	20				READ – reads output of macrocells AB WRITE – loads macrocell flip-flops
Output Macrocells BC	21	21			READ – reads output of macrocells BC WRITE – loads macrocell flip-flops
Mask Macrocells AB	22				Blocks writing to the Output Macrocells AB
Mask Macrocells BC	23	23			Blocks writing to the Output Macrocells BC
Primary Flash Protection				C0	Read-only – Primary Flash Sector Protection
Secondary Flash memory Protection				C2	Read-only – PSD Module Security and Secondary Flash memory Sector Protection
PMMR0				B0	Power Management Register 0
PMMR2				B4	Power Management Register 2
Page				E0	Page Register
VM				E2	Places PSD Module memory areas in Program and/or Data space on an individual basis.

Note: 1. Other registers that are not part of the I/O ports.

PSD MODULE DETAILED OPERATION

As shown in Figure 13, the PSD Module consists of five major types of functional blocks:

- Memory Block
- PLD Blocks
- I/O Ports
- Power Management Unit (PMU)
- JTAG Interface

The functions of each block are described in the following sections. Many of the blocks perform multiple functions, and are user configurable.

MEMORY BLOCKS

The PSD Module has the following memory blocks:

- Primary Flash memory
- Secondary Flash memory
- SRAM

The Memory Select signals for these blocks originate from the Decode PLD (DPLD) and are user-defined in PSDsoft Express.

Primary Flash Memory and Secondary Flash memory Description

The primary Flash memory is divided into 4 sectors (16KBytes each). The secondary Flash memory is divided into 2 sectors (8KBytes each). Each sector of either memory block can be separately protected from Program and Erase cycles.

Flash memory may be erased on a sector-by-sector basis. Flash sector erasure may be suspended while data is read from other sectors of the block and then resumed after reading.

During a Program or Erase cycle in Flash memory, the status can be output on Ready/Busy (PC3). This pin is set up using PSDsoft Express Configuration.

Memory Block Select Signals

The DPLD generates the Select signals for all the internal memory blocks (see the section entitled “PLDs,” page 88). Each of the eight sectors of the primary Flash memory has a Select signal (FS0-FS3) which can contain up to three product terms. Each of the 2 sectors of the secondary Flash memory has a Select signal (CSBOOT0-CSBOOT1) which can contain up to three product terms. Having three product terms for each Select signal allows a given sector to be mapped in Program or Data space.

Ready/Busy (PC3). This signal can be used to output the Ready/Busy status of the Flash memory. The output on Ready/Busy (PC3) is a 0 (Busy) when Flash memory is being written to, *or* when Flash memory is being erased. The output is a 1 (Ready) when no WRITE or Erase cycle is in progress.

Memory Operation. The primary Flash memory and secondary Flash memory are addressed through the MCU Bus. The MCU can access these memories in one of two ways:

– The MCU can execute a typical bus WRITE or READ *operation*.

– The MCU can execute a specific Flash memory instruction that consists of several WRITE and READ operations. This involves writing specific data patterns to special addresses within the Flash memory to invoke an embedded algorithm. These instructions are summarized in Table 60.

Typically, the MCU can read Flash memory using READ operations, just as it would read a ROM device. However, Flash memory can only be altered using specific Erase and Program instructions. For example, the MCU cannot write a single byte directly to Flash memory as it would write a byte to RAM. To program a byte into Flash memory, the MCU must execute a Program instruction, then test the status of the Program cycle. This status test is achieved by a READ operation or polling Ready/Busy (PC3).

Instructions

An instruction consists of a sequence of specific operations. Each received byte is sequentially decoded by the PSD Module and not executed as a standard WRITE operation. The instruction is executed when the correct number of bytes are properly received and the time between two consecutive bytes is shorter than the time-out period. Some instructions are structured to include READ operations after the initial WRITE operations.

The instruction must be followed exactly. Any invalid combination of instruction bytes or time-out between two consecutive bytes while addressing Flash memory resets the device logic into READ Mode (Flash memory is read like a ROM device).

The Flash memory supports the instructions summarized in Table 60:

Flash memory:

- Erase memory by chip or sector
- Suspend or resume sector erase
- Program a Byte
- RESET to READ Mode
- Read Sector Protection Status

These instructions are detailed in Table 60. For efficient decoding of the instructions, the first two bytes of an instruction are the coded cycles and are followed by an instruction byte or confirmation byte. The coded cycles consist of writing the data AAh to address X555h during the first cycle and data 55h to address XAAAh during the second cycle. Address signals A15-A12 are Don't Care during the instruction WRITE cycles. However, the appropriate Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) must be selected.

The primary and secondary Flash memories have the same instruction set. The Sector Select signals determine which Flash memory is to receive and execute the instruction. The primary Flash memory is selected if any one of Sector Select (FS0-FS3) is High, and the secondary Flash memory is selected if any one of Sector Select (CSBOOT0-CSBOOT1) is High.

Table 60. Instructions

Instruction	FS0-FS3 or CSBOOT0-CSBOOT1	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
READ(5)	1	"Read" RD @ RA						
READ Sector Protection(6,8,11)	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read status @ XX02h			
Program a Flash Byte(11)	1	AAh@ X555h	55h@ XAAAh	A0h@ X555h	PD@ PA			
Flash Sector Erase(7,11)	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ X555h	55h@ XAAAh	30h@ SA	30h(7)@ next SA
Flash Bulk Erase(11)	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ X555h	55h@ XAAAh	10h@ X555h	
Suspend Sector Erase(9)	1	B0h@ XXXXh						
Resume Sector Erase(10)	1	30h@ XXXXh						
RESET(6)	1	F0h@ XXXXh						

Note: 1. All bus cycles are WRITE bus cycles, except the ones with the "Read" label
2. All values are in hexadecimal:
X = Don't care. Addresses of the form XXXXh, in this table, must be even addresses
RA = Address of the memory location to be read
RD = Data READ from location RA during the READ cycle
PA = Address of the memory location to be programmed. Addresses are latched on the falling edge of WRITE Strobe ($\overline{WR}$, CNTL0). PA is an even address for PSD in Word Programming Mode.
PD = Data word to be programmed at location PA. Data is latched on the rising edge of WRITE Strobe ($\overline{WR}$, CNTL0)
SA = Address of the sector to be erased or verified. The Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) of the sector to be erased, or verified, must be Active (High).
3. Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) signals are active High, and are defined in PSDsoft Express.
4. Only address Bits A11-A0 are used in instruction decoding.
5. No Unlock or instruction cycles are required when the device is in the READ Mode
6. The RESET Instruction is required to return to the READ Mode after reading the Sector Protection Status, or if the Error Flag Bit (DQ5) goes High.
7. Additional sectors to be erased must be written at the end of the Sector Erase instruction within 80µs.
8. The data is 00h for an unprotected sector, and 01h for a protected sector. In the fourth cycle, the Sector Select is active, and (A1,A0)=(1,0)
9. The system may perform READ and Program cycles in non-erasing sectors, read the Sector Protection Status when in the Suspend Sector Erase Mode. The Suspend Sector Erase instruction is valid only during a Sector Erase cycle.
10. The Resume Sector Erase instruction is valid only during the Suspend Sector Erase Mode.
11. The MCU cannot invoke these instructions while executing code from the same Flash memory as that for which the instruction is intended. The MCU must retrieve, for example, the code from the secondary Flash memory when reading the Sector Protection Status of the primary Flash memory.

Power-down Instruction and Power-up Mode

Power-up Mode. The PSD Module internal logic is reset upon Power-up to the READ Mode. Sector Select (FS0-FS3 and CSBOOT0-CSBOOT1) must be held Low, and WRITE Strobe (WR, CNTL0) High, during Power-up for maximum security of the data contents and to remove the possibility of a byte being written on the first edge of WRITE Strobe (WR, CNTL0). Any WRITE cycle initiation is locked when V_{CC} is below V_{LKO}.

READ

Under typical conditions, the MCU may read the primary Flash memory or the secondary Flash memory using READ operations just as it would a ROM or RAM device. Alternately, the MCU may use READ operations to obtain status information about a Program or Erase cycle that is currently in progress. Lastly, the MCU may use instructions to read special data from these memory blocks. The following sections describe these READ functions.

READ Memory Contents. Primary Flash memory and secondary Flash memory are placed in the READ Mode after Power-up, chip reset, or a Reset Flash instruction (see Table 60, page 77). The MCU can read the memory contents of the primary Flash memory or the secondary Flash memory by using READ operations any time the READ operation is not part of an instruction.

READ Memory Sector Protection Status. The primary Flash memory Sector Protection Status is read with an instruction composed of 4 operations: 3 specific WRITE operations and a READ operation (see Table 60). During the READ operation, address Bits A6, A1, and A0 must be '0,' '1,' and '0,' respectively, while Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) designates the Flash memory sector whose protection has to be verified. The READ operation produces 01h if the Flash memory sector is protected, or 00h if the sector is not protected.

The sector protection status for all NVM blocks (primary Flash memory or secondary Flash memory) can also be read by the MCU accessing the Flash Protection registers in PSD I/O space. See the section entitled “Flash Memory Sector Protect,” page 83, for register definitions.

Reading the Erase/Program Status Bits. The Flash memory provides several status bits to be used by the MCU to confirm the completion of an Erase or Program cycle of Flash memory. These status bits minimize the time that the MCU spends performing these tasks and are defined in Table 61, page 79. The status bits can be read as many times as needed.

For Flash memory, the MCU can perform a READ operation to obtain these status bits while an Erase or Program instruction is being executed by the embedded algorithm. See the section entitled “Programming Flash Memory,” page 80, for details.

Data Polling Flag (DQ7). When erasing or programming in Flash memory, the Data Polling Flag Bit (DQ7) outputs the complement of the bit being entered for programming/writing on the DQ7 Bit. Once the Program instruction or the WRITE operation is completed, the true logic value is read on the Data Polling Flag Bit (DQ7) (in a READ operation).

- Data Polling is effective after the fourth WRITE pulse (for a Program instruction) or after the sixth WRITE pulse (for an Erase instruction). It must be performed at the address being programmed or at an address within the Flash memory sector being erased.
- During an Erase cycle, the Data Polling Flag Bit (DQ7) outputs a '0.' After completion of the cycle, the Data Polling Flag Bit (DQ7) outputs the last bit programmed (it is a '1' after erasing).
- If the byte to be programmed is in a protected Flash memory sector, the instruction is ignored.
- If all the Flash memory sectors to be erased are protected, the Data Polling Flag Bit (DQ7) is reset to '0' for about 100µs, and then returns to the previous addressed byte. No erasure is performed.

Toggle Flag (DQ6). The Flash memory offers another way for determining when the Program cycle is completed. During the internal WRITE operation and when either the FS0-FS3 or CSBOOT0-CSBOOT1 is true, the Toggle Flag Bit (DQ6) toggles from 0 to 1 and 1 to 0 on subsequent attempts to read any byte of the memory.

When the internal cycle is complete, the toggling stops and the data READ on the Data Bus D0-D7 is the addressed memory byte. The device is now accessible for a new READ or WRITE operation. The cycle is finished when two successive Reads yield the same output data.

– The Toggle Flag Bit (DQ6) is effective after the fourth WRITE pulse (for a Program instruction) or after the sixth WRITE pulse (for an Erase instruction).
– If the byte to be programmed belongs to a protected Flash memory sector, the instruction is ignored.
– If all the Flash memory sectors selected for erasure are protected, the Toggle Flag Bit (DQ6) toggles to '0' for about 100µs and then returns to the previous addressed byte.

Error Flag (DQ5). During a normal Program or Erase cycle, the Error Flag Bit (DQ5) is to 0. This bit is set to '1' when there is a failure during Flash memory Byte Program, Sector Erase, or Bulk Erase cycle.

In the case of Flash memory programming, the Error Flag Bit (DQ5) indicates the attempt to program a Flash memory bit from the programmed state, '0,' to the erased state, '1,' which is not valid. The Error Flag Bit (DQ5) may also indicate a Time-out condition while attempting to program a byte.

In case of an error in a Flash memory Sector Erase or Byte Program cycle, the Flash memory sector in which the error occurred or to which the programmed byte belongs must no longer be used. Other Flash memory sectors may still be used. The Error Flag Bit (DQ5) is reset after a Reset Flash instruction.

Erase Time-out Flag (DQ3). The Erase Time-out Flag Bit (DQ3) reflects the time-out period allowed between two consecutive Sector Erase instructions. The Erase Time-out Flag Bit (DQ3) is reset to 0 after a Sector Erase cycle for a time period of 100µs + 20% unless an additional Sector Erase instruction is decoded. After this time period, or when the additional Sector Erase instruction is decoded, the Erase Time-out Flag Bit (DQ3) is set to '1.'

Table 61. Status Bit

Functional Block	FS0-FS3/CSBOOT0-CSBOOT1	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Flash Memory	V_{IH}	Data Polling	Toggle Flag	Error Flag	X	Erase Time-out	X	X	X

Note: 1. X = Not guaranteed value, can be read either '1' or '0.'
2. DQ7-DQ0 represent the Data Bus bits, D7-D0.
3. FS0-FS3 and CSBOOT0-CSBOOT1 are active High.

Programming Flash Memory

Flash memory must be erased prior to being programmed. A byte of Flash memory is erased to all '1s' (FFh), and is programmed by setting selected bits to '0.' The MCU may erase Flash memory all at once or by-sector, but not byte-by-byte. However, the MCU may program Flash memory byte-by-byte.

The primary and secondary Flash memories require the MCU to send an instruction to program a byte or to erase sectors (see Table 60).

Once the MCU issues a Flash memory Program or Erase instruction, it must check for the status bits for completion. The embedded algorithms that are invoked support several means to provide status to the MCU. Status may be checked using any of three methods: Data Polling, Data Toggle, or Ready/Busy (PC3).

Data Polling. Polling on the Data Polling Flag Bit (DQ7) is a method of checking whether a Program or Erase cycle is in progress or has completed. Figure 37 shows the Data Polling algorithm.

When the MCU issues a Program instruction, the embedded algorithm begins. The MCU then reads the location of the byte to be programmed in Flash memory to check status. The Data Polling Flag Bit (DQ7) of this location becomes the complement of b7 of the original data byte to be programmed. The MCU continues to poll this location, comparing the Data Polling Flag Bit (DQ7) and monitoring the Error Flag Bit (DQ5). When the Data Polling Flag Bit (DQ7) matches b7 of the original data, and the Error Flag Bit (DQ5) remains '0,' the embedded algorithm is complete. If the Error Flag Bit (DQ5) is '1,' the MCU should test the Data Polling Flag Bit (DQ7) again since the Data Polling Flag Bit (DQ7) may have changed simultaneously with the Error Flag Bit (DQ5) (see Figure 37).

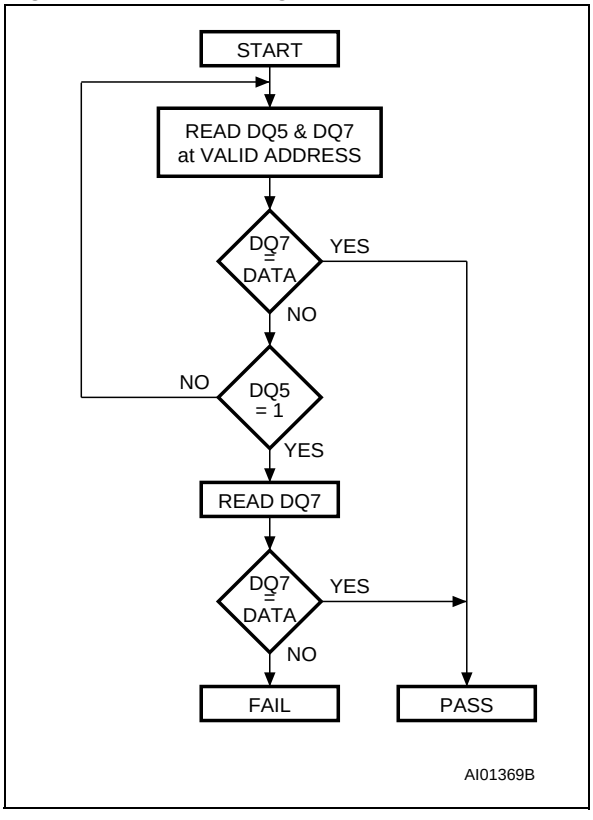
The Error Flag Bit (DQ5) is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic '0').

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte that was written to the Flash memory with the byte that was intended to be written.

When using the Data Polling method during an Erase cycle, Figure 37 still applies. However, the Data Polling Flag Bit (DQ7) is '0' until the Erase cycle is complete. A '1' on the Error Flag Bit (DQ5) indicates a time-out condition on the Erase cycle; a '0' indicates no error. The MCU can read any location within the sector being erased to get the Data Polling Flag Bit (DQ7) and the Error Flag Bit (DQ5).

PSDsoft Express generates ANSI C code functions which implement these Data Polling algorithms.

Figure 37. Data Polling Flowchart



Data Toggle. Checking the Toggle Flag Bit (DQ6) is a method of determining whether a Program or Erase cycle is in progress or has completed. Figure 38 shows the Data Toggle algorithm.

When the MCU issues a Program instruction, the embedded algorithm begins. The MCU then reads the location of the byte to be programmed in Flash memory to check status. The Toggle Flag Bit (DQ6) of this location toggles each time the MCU reads this location until the embedded algorithm is complete. The MCU continues to read this location, checking the Toggle Flag Bit (DQ6) and monitoring the Error Flag Bit (DQ5). When the Toggle Flag Bit (DQ6) stops toggling (two consecutive reads yield the same value), and the Error Flag Bit (DQ5) remains '0,' the embedded algorithm is complete. If the Error Flag Bit (DQ5) is '1,' the MCU should test the Toggle Flag Bit (DQ6) again, since the Toggle Flag Bit (DQ6) may have changed simultaneously with the Error Flag Bit (DQ5) (see Figure 38).

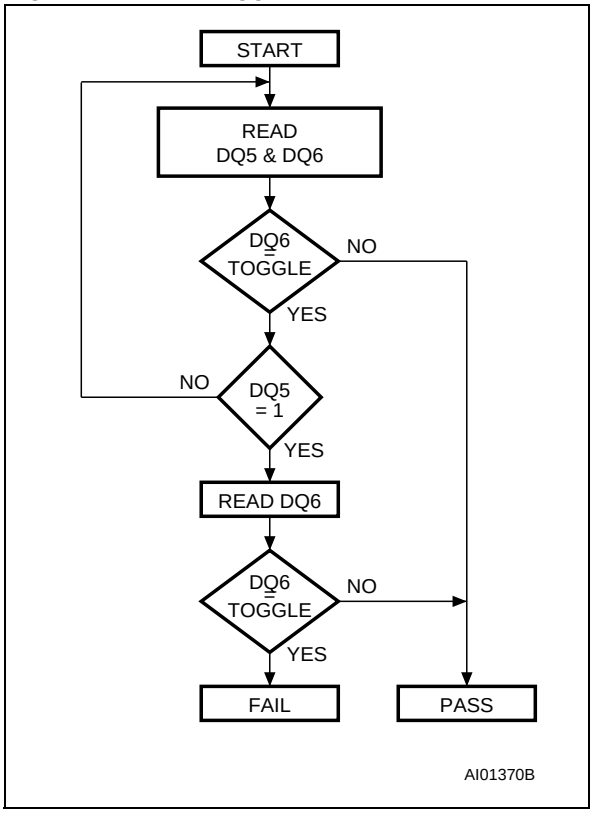
The Error Flag Bit(DQ5) is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte, or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic '0').

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte that was written to Flash memory with the byte that was intended to be written.

When using the Data Toggle method after an Erase cycle, Figure 38 still applies. the Toggle Flag Bit (DQ6) toggles until the Erase cycle is complete. A '1' on the Error Flag Bit (DQ5) indicates a time-out condition on the Erase cycle; a '0' indicates no error. The MCU can read any location within the sector being erased to get the Toggle Flag Bit (DQ6) and the Error Flag Bit (DQ5).

PSDsoft Express generates ANSI C code functions which implement these Data Toggling algorithms.

Figure 38. Data Toggle Flowchart



Erasing Flash Memory

Flash Bulk Erase. The Flash Bulk Erase instruction uses six WRITE operations followed by a READ operation of the status register, as described in Table 60. If any byte of the Bulk Erase instruction is wrong, the Bulk Erase instruction aborts and the device is reset to the READ Flash memory status.

During a Bulk Erase, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7), as detailed in the section entitled "Programming Flash Memory," page 80. The Error Flag Bit (DQ5) returns a '1' if there has been an Erase Failure (maximum number of Erase cycles have been executed).

It is not necessary to program the memory with 00h because the PSD Module automatically does this before erasing to 0FFh.

During execution of the Bulk Erase instruction, the Flash memory does not accept any instructions.

Flash Sector Erase. The Sector Erase instruction uses six WRITE operations, as described in Table 60. Additional Flash Sector Erase codes and Flash memory sector addresses can be written subsequently to erase other Flash memory sectors in parallel, without further coded cycles, if the additional bytes are transmitted in a shorter time than the time-out period of about 100µs. The input of a new Sector Erase code restarts the time-out period.

The status of the internal timer can be monitored through the level of the Erase Time-out Flag Bit (DQ3). If the Erase Time-out Flag Bit (DQ3) is '0,' the Sector Erase instruction has been received and the time-out period is counting. If the Erase Time-out Flag Bit (DQ3) is '1,' the time-out period has expired and the embedded algorithm is busy erasing the Flash memory sector(s). Before and during Erase time-out, any instruction other than Suspend Sector Erase and Resume Sector Erase instructions abort the cycle that is currently in progress, and reset the device to READ Mode.

During a Sector Erase, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7), as detailed in the section entitled "Programming Flash Memory," page 80.

During execution of the Erase cycle, the Flash memory accepts only RESET and Suspend Sector Erase instructions. Erasure of one Flash memory sector may be suspended, in order to read data from another Flash memory sector, and then resumed.

Suspend Sector Erase. When a Sector Erase cycle is in progress, the Suspend Sector Erase instruction can be used to suspend the cycle by writing 0B0h to any address when an appropriate Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) is High. (See Table 60). This allows reading of data from another Flash memory sector after the Erase cycle has been suspended. Suspend Sector Erase is accepted only during an Erase cycle and defaults to READ Mode. A Suspend Sector Erase instruction executed during an Erase time-out period, in addition to suspending the Erase cycle, terminates the time out period.

The Toggle Flag Bit (DQ6) stops toggling when the internal logic is suspended. The status of this bit must be monitored at an address within the Flash memory sector being erased. The Toggle Flag Bit (DQ6) stops toggling between 0.1µs and 15µs after the Suspend Sector Erase instruction has been executed. The Flash memory is then automatically set to READ Mode.

If an Suspend Sector Erase instruction was executed, the following rules apply:

– Attempting to read from a Flash memory sector that was being erased outputs invalid data.
– Reading from a Flash sector that was *not* being erased is valid.
– The Flash memory *cannot* be programmed, and only responds to Resume Sector Erase and Reset Flash instructions (READ is an operation and is allowed).
– If a Reset Flash instruction is received, data in the Flash memory sector that was being erased is invalid.

Resume Sector Erase. If a Suspend Sector Erase instruction was previously executed, the erase cycle may be resumed with this instruction. The Resume Sector Erase instruction consists of writing 030h to any address while an appropriate Sector Select (FS0-FS3 or CSBOOT0-CSBOOT1) is High. (See Table 60.)

Specific Features

Flash Memory Sector Protect. Each primary and secondary Flash memory sector can be separately protected against Program and Erase cycles. Sector Protection provides additional data security because it disables all Program or Erase cycles. This mode can be activated through the JTAG Port or a Device Programmer.

Sector protection can be selected for each sector using the PSDsoft Express Configuration program. This automatically protects selected sectors when the device is programmed through the JTAG Port or a Device Programmer. Flash memory sectors can be unprotected to allow updating of their contents using the JTAG Port or a Device Programmer. The MCU can read (but cannot change) the sector protection bits.

Any attempt to program or erase a protected Flash memory sector is ignored by the device. The Verify operation results in a READ of the protected data. This allows a guarantee of the retention of the Protection status.

The sector protection status can be read by the MCU through the Flash memory protection registers (in the CSIOP block). See Table 62 and Table 63.

Reset Flash. The Reset Flash instruction consists of one WRITE cycle (see Table 60). It can also be optionally preceded by the standard two WRITE decoding cycles (writing AAh to 555h and 55h to AAAh). It must be executed after:

– Reading the Flash Protection Status or Flash ID

– An Error condition has occurred (and the device has set the Error Flag Bit (DQ5) to '1' during a Flash memory Program or Erase cycle.

The Reset Flash instruction puts the Flash memory back into normal READ Mode. If an Error condition has occurred (and the device has set the Error Flag Bit (DQ5) to '1' the Flash memory is put back into normal READ Mode within a few milliseconds of the Reset Flash instruction having been issued. The Reset Flash instruction is ignored when it is issued during a Program or Bulk Erase cycle of the Flash memory. The Reset Flash instruction aborts any on-going Sector Erase cycle, and returns the Flash memory to the normal READ Mode within a few milliseconds.

Table 62. Sector Protection/Security Bit Definition – Flash Protection Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used	not used	not used	not used	Sec3_Prot	Sec2_Prot	Sec1_Prot	Sec0_Prot

Note: Bit Definitions:
Sec<i>_Prot 1 = Primary Flash memory or secondary Flash memory Sector <i> is write-protected.
Sec<i>_Prot 0 = Primary Flash memory or secondary Flash memory Sector <i> is not write-protected.

Table 63. Sector Protection/Security Bit Definition – Secondary Flash Protection Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Security_Bit	not used	not used	not used	not used	not used	Sec1_Prot	Sec0_Prot

Note: Bit Definitions:
Sec<i>_Prot 1 = Secondary Flash memory Sector <i> is write-protected.
Sec<i>_Prot 0 = Secondary Flash memory Sector <i> is not write-protected.
Security_Bit 0 = Security Bit in device has not been set; 1 = Security Bit in device has been set.

SRAM

The SRAM is enabled when SRAM Select (RS0) from the DPLD is High. SRAM Select (RS0) can contain up to two product terms, allowing flexible memory mapping.

The SRAM can be backed up using an external battery. The external battery should be connected to Voltage Standby (V_{STBY}, PC2). If you have an external battery connected to the µPSD3200, the contents of the SRAM are retained in the event of a power loss. The contents of the SRAM are retained so long as the battery voltage remains at 2V or greater. If the supply voltage falls below the battery voltage, an internal power switchover to the battery occurs.

PC4 can be configured as an output that indicates when power is being drawn from the external battery. Battery-on Indicator (V_{BATON}, PC4) is High with the supply voltage falls below the battery voltage and the battery on Voltage Standby (V_{STBY}, PC2) is supplying power to the internal SRAM.

SRAM Select (RS0), Voltage Standby (V_{STBY}, PC2) and Battery-on Indicator (V_{BATON}, PC4) are all configured using PSDsoft Express Configuration.

Sector Select and SRAM Select

Sector Select (FS0-FS3, CSBOOT0-CSBOOT1) and SRAM Select (RS0) are all outputs of the DPLD. They are setup by writing equations for them in PSDsoft Express. The following rules apply to the equations for these signals:

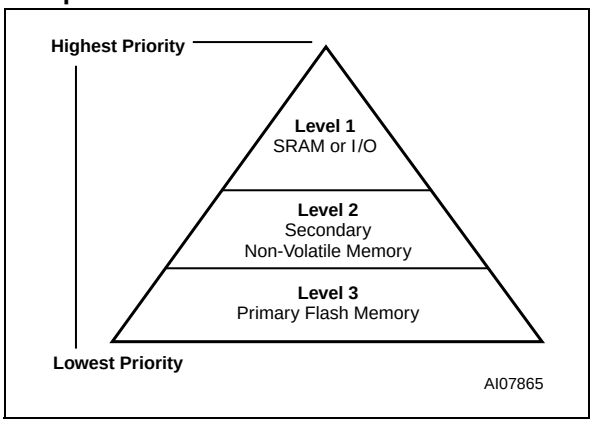
1. Primary Flash memory and secondary Flash memory Sector Select signals must *not* be larger than the physical sector size.
2. Any primary Flash memory sector must *not* be mapped in the same memory space as another Flash memory sector.
3. A secondary Flash memory sector must *not* be mapped in the same memory space as another secondary Flash memory sector.
4. SRAM and I/O must *not* overlap.
5. A secondary Flash memory sector *may* overlap a primary Flash memory sector. In case of overlap, priority is given to the secondary Flash memory sector.
6. SRAM and I/O *may* overlap any other memory sector. Priority is given to the SRAM or I/O.

Example. FS0 is valid when the address is in the range of 8000h to BFFFh, CSBOOT0 is valid from 8000h to 9FFFh, and RS0 is valid from 8000h to 87FFh. Any address in the range of RS0 always accesses the SRAM. Any address in the range of CSBOOT0 greater than 87FFh (and less than 9FFFh) automatically addresses secondary Flash memory segment 0. Any address greater than 9FFFh accesses the primary Flash memory segment 0. You can see that half of the primary Flash memory segment 0 and one-fourth of secondary Flash memory segment 0 cannot be accessed in this example.

Note: An equation that defined FS1 to anywhere in the range of 8000h to BFFFh would *not* be valid.

Figure 39 shows the priority levels for all memory components. Any component on a higher level can overlap and has priority over any component on a lower level. Components on the same level must *not* overlap. Level one has the highest priority and level 3 has the lowest.

Figure 39. Priority Level of Memory and I/O Components in the PSD Module



Memory Select Configuration in Program and Data Spaces. The MCU Core has separate address spaces for Program memory and Data memory. Any of the memories within the PSD Module can reside in either space or both spaces. This is controlled through manipulation of the VM Register that resides in the CSIOP space.

The VM Register is set using PSDsoft Express to have an initial value. It can subsequently be changed by the MCU so that memory mapping can be changed on-the-fly.

For example, you may wish to have SRAM and primary Flash memory in the Data space at Boot-up, and secondary Flash memory in the Program space at Boot-up, and later swap the primary and secondary Flash memories. This is easily done with the VM Register by using PSDsoft Express Configuration to configure it for Boot-up and having the MCU change it when desired. Table 64 describes the VM Register.

Table 64. VM Register

Bit 7	Bit 6	Bit 5	Bit 4 Primary FL_Data	Bit 3 Secondary Data	Bit 2 Primary FL_Code	Bit 1 Secondary Code	Bit 0 SRAM_Code
Not used(1)	Not used(1)	Not used(1)	0 = $\overline{RD}$ can't access Flash memory	0 = $\overline{RD}$ can't access Secondary Flash memory	0 = $\overline{PSEN}$ can't access Flash memory	0 = $\overline{PSEN}$ can't access Secondary Flash memory	0 = $\overline{PSEN}$ can't access SRAM
Not used(1)	Not used(1)	Not used(1)	1 = $\overline{RD}$ access Flash memory	1 = $\overline{RD}$ access Secondary Flash memory	1 = $\overline{PSEN}$ access Flash memory	1 = $\overline{PSEN}$ access Secondary Flash memory	1 = $\overline{PSEN}$ access SRAM

Note: 1. "Not used" bits should be set to '0.'

Separate Space Mode. Program space is separated from Data space. For example, Program Select Enable (PSEN) is used to access the program code from the primary Flash memory, while READ Strobe (RD) is used to access data from the secondary Flash memory, SRAM and I/O Port blocks. This configuration requires the VM Register to be set to 0Ch (see Figure 40).

Combined Space Modes. The Program and Data spaces are combined into one memory space that allows the primary Flash memory, secondary Flash memory, and SRAM to be accessed by either Program Select Enable (PSEN) or READ Strobe (RD). For example, to configure the primary Flash memory in Combined space, Bits b2 and b4 of the VM Register are set to '1' (see Figure 41).

Figure 40. Separate Space Mode

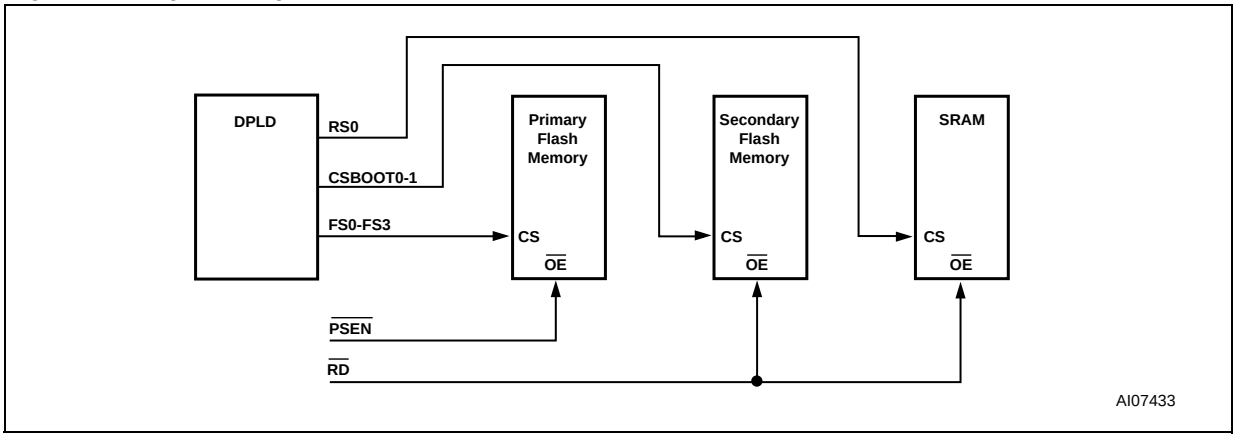
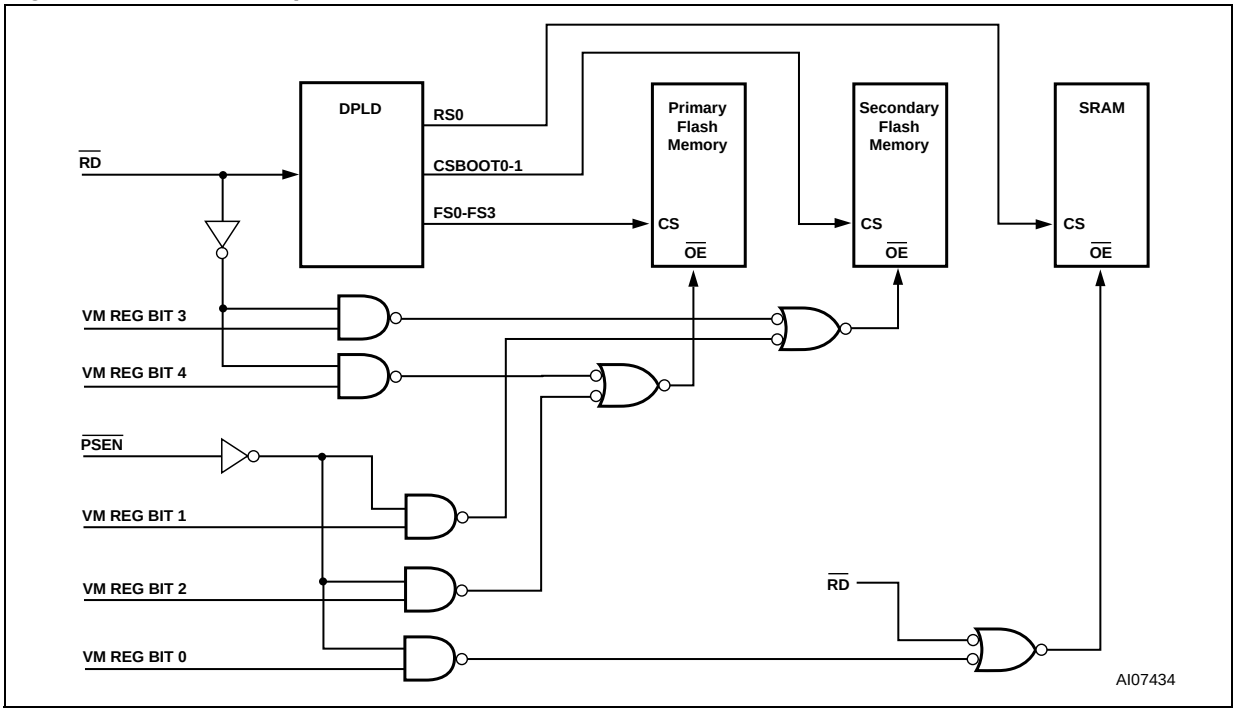


Figure 41. Combined Space Mode



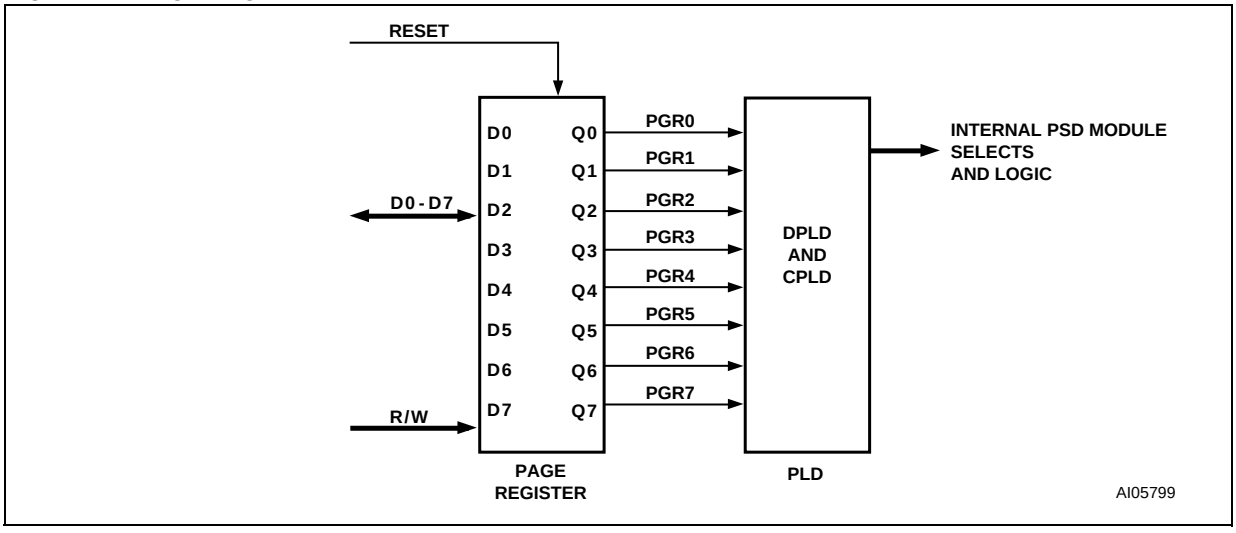
Page Register

The 8-bit Page Register increases the addressing capability of the MCU Core by a factor of up to 256. The contents of the register can also be read by the MCU. The outputs of the Page Register (PGR0-PGR7) are inputs to the DPLD decoder and can be included in the Sector Select (FS0-FS3, CSBOOT0-CSBOOT1), and SRAM Select (RS0) equations.

If memory paging is not needed, or if not all 8 page register bits are needed for memory paging, then these bits may be used in the CPLD for general logic.

Figure 42 shows the Page Register. The eight flip-flops in the register are connected to the internal data bus D0-D7. The MCU can write to or read from the Page Register. The Page Register can be accessed at address location CSIOP + E0h.

Figure 42. Page Register



PLDS

The PLDs bring programmable logic functionality to the µPSD. After specifying the logic for the PLDs in PSDsoft Express, the logic is programmed into the device and available upon Power-up.

Table 65. DPLD and CPLD Inputs

Input Source	Input Name	Number of Signals
MCU Address Bus	A15-A0	16
MCU Control Signals	$\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$, ALE	4
RESET	$\overline{RST}$	1
Power-down	PDN	1
Port B Input Macrocells	PB7-PB0	8
Port C Input Macrocells	PC7, PC4-PC2	4
Port D Inputs	PD1	1
Page Register	PGR7-PGR0	8
Macrocell AB Feedback	MCELLAB.FB7-FB0	8
Macrocell BC Feedback	MCELLBC.FB7-FB0	8
Flash memory Program Status Bit	Ready/$\overline{Busy}$	1

The PSD Module contains two PLDs: the Decode PLD (DPLD), and the Complex PLD (CPLD). The PLDs are briefly discussed in the next few paragraphs, and in more detail in the section entitled “Decode PLD (DPLD),” page 90, and the section entitled “Complex PLD (CPLD),” page 91. Figure 43 shows the configuration of the PLDs.

The DPLD performs address decoding for Select signals for PSD Module components, such as memory, registers, and I/O ports.

The CPLD can be used for logic functions, such as loadable counters and shift registers, state machines, and encoding and decoding logic. These logic functions can be constructed using the Output Macrocells (OMC), Input Macrocells (IMC), and the AND Array. The CPLD can also be used to generate External Chip Select (ECS1) signals.

The AND Array is used to form product terms. These product terms are specified using PSDsoft. The PLD input signals consist of internal MCU signals and external inputs from the I/O ports. The input signals are shown in Table 65.

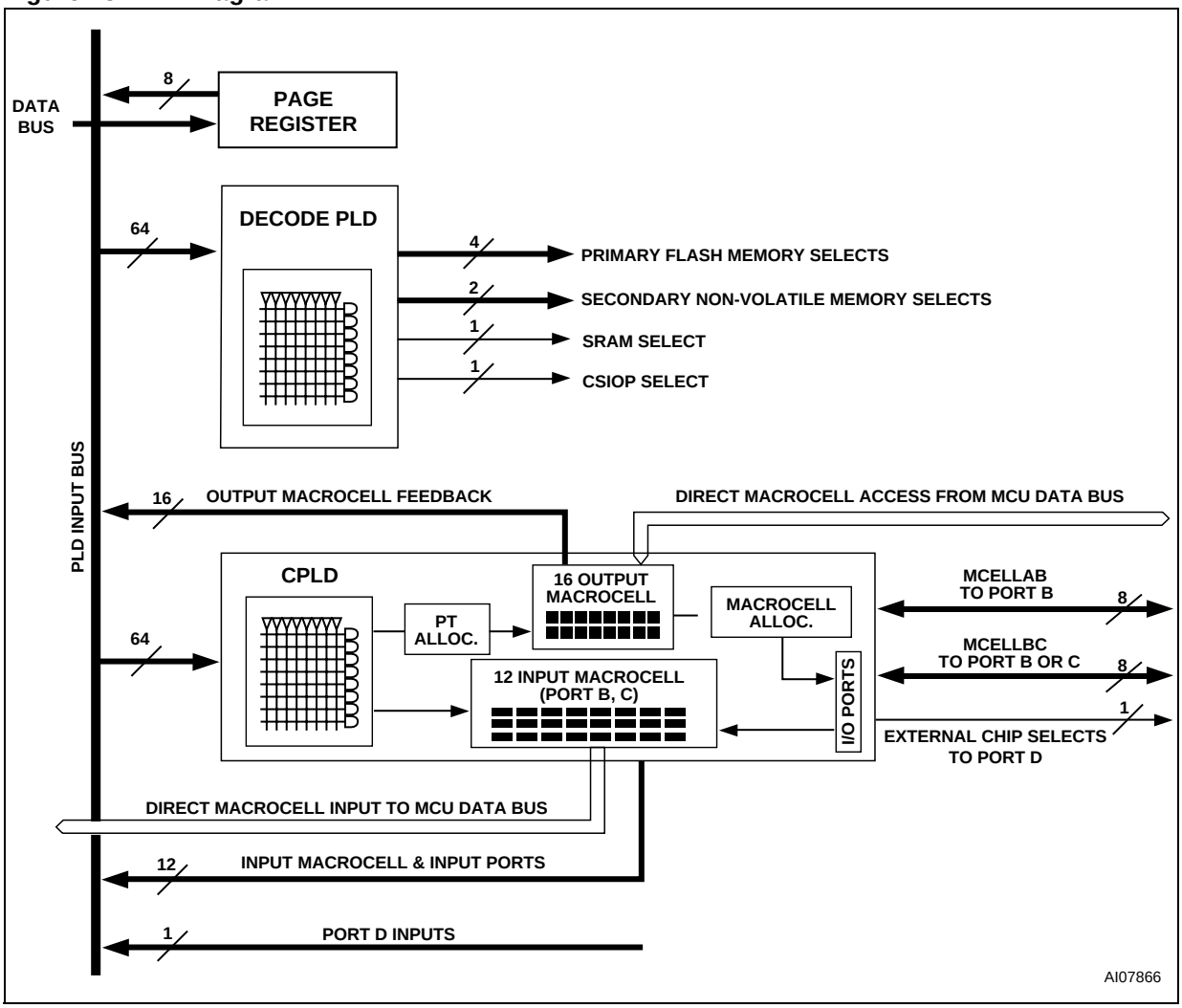
The Turbo Bit in PSD Module

The PLDs can minimize power consumption by switching off when inputs remain unchanged for an extended time of about 70ns. Resetting the Turbo Bit to '0' (Bit 3 of PMMR0) automatically places the PLDs into standby if no inputs are changing. Turning the Turbo Mode off increases propagation delays while reducing power consumption. See the section entitled “POWER MANAGEMENT,” page 104, on how to set the Turbo Bit.

Additionally, five bits are available in PMMR2 to block MCU control signals from entering the PLDs. This reduces power consumption and can be used only when these MCU control signals are not used in PLD logic equations.

Each of the two PLDs has unique characteristics suited for its applications. They are described in the following sections.

Figure 43. PLD Diagram

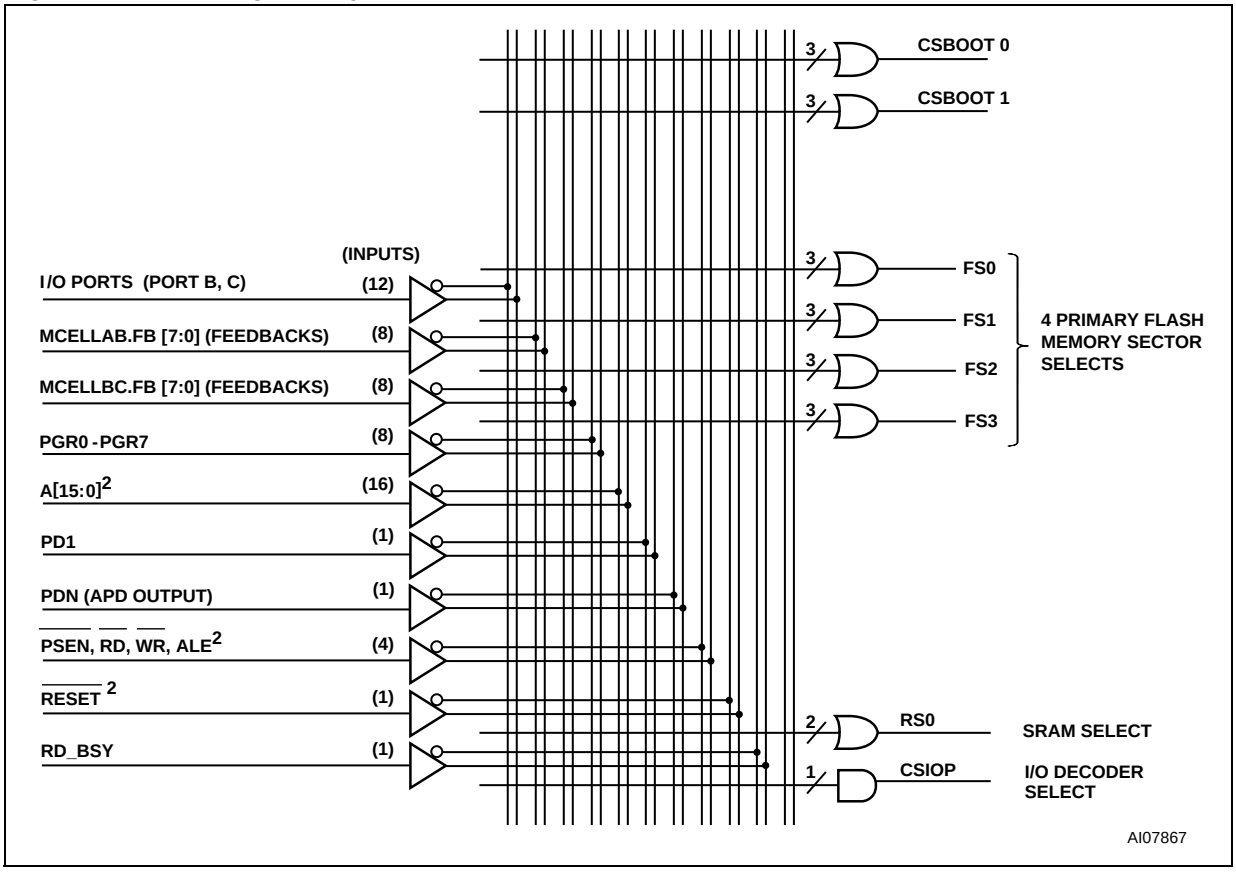


Decode PLD (DPLD)

The DPLD, shown in Figure 44, is used for decoding the address for PSD Module and external components. The DPLD can be used to generate the following decode signals:

– 4 Sector Select (FS0-FS3) signals for the primary Flash memory (three product terms each)
– 2 Sector Select (CSBOOT0-CSBOOT1) signals for the secondary Flash memory (three product terms each)
– 1 internal SRAM Select (RS0) signal (two product terms)
– 1 internal CSIOP Select signal (selects the PSD Module registers)

Figure 44. DPLD Logic Array



Note: 1. Inputs from the MCU module

Complex PLD (CPLD)

The CPLD can be used to implement system logic functions, such as loadable counters and shift registers, system mailboxes, handshaking protocols, state machines, and random logic. The CPLD can also be used to generate External Chip Select (ECS1), routed to Port D.

Although External Chip Select (ECS1) can be produced by any Output Macrocell (OMC), External Chip Select (ECS1) on Port D does not consume any Output Macrocells (OMC).

As shown in Figure 43, the CPLD has the following blocks:

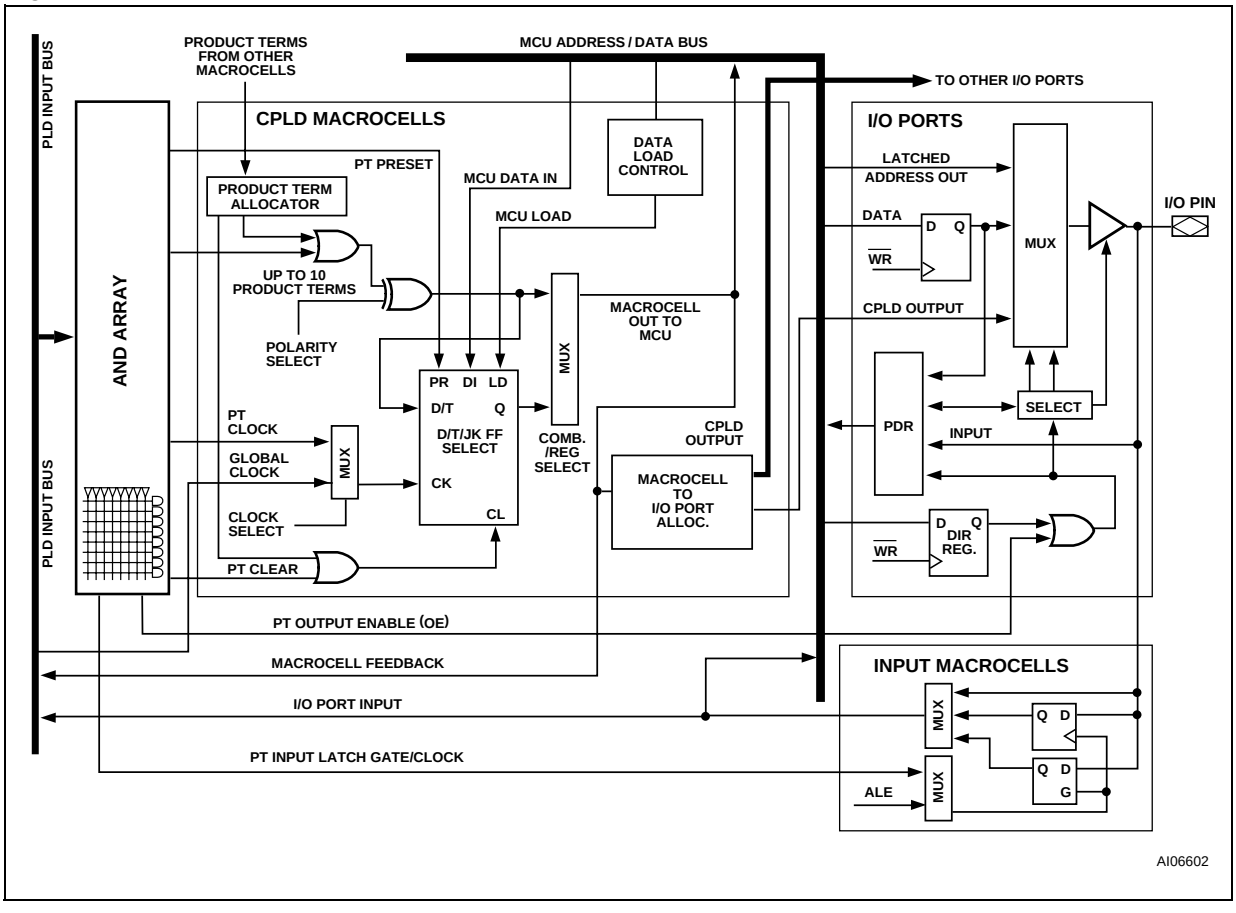
- 12 Input Macrocells (IMC)
- 16 Output Macrocells (OMC)
- Macrocell Allocator
- Product Term Allocator
- AND Array capable of generating up to 137 product terms
- Four I/O Ports.

Each of the blocks are described in the sections that follow.

The Input Macrocells (IMC) and Output Macrocells (OMC) are connected to the PSD Module internal data bus and can be directly accessed by the MCU. This enables the MCU software to load data into the Output Macrocells (OMC) or read data from both the Input and Output Macrocells (IMC and OMC).

This feature allows efficient implementation of system logic and eliminates the need to connect the data bus to the AND Array as required in most standard PLD macrocell architectures.

Figure 45. Macrocell and I/O Port



Output Macrocell (OMC)

Eight of the Output Macrocells (OMC) are connected to Ports B pin and are named as McellAB0-McellAB7. The other eight macrocells are connected to Ports B and C pins and are named as McellBC0-McellBC7. If an McellBC output is not assigned to a specific pin in PSDsoft, the Macrocell Allocator block assigns it to either Port B or C.

The Output Macrocell (OMC) architecture is shown in Figure 46. As shown in the figure, there are native product terms available from the AND Array, and borrowed product terms available (if unused) from other Output Macrocells (OMC). The polarity of the product term is controlled by the XOR gate. The Output Macrocell (OMC) can implement either sequential logic, using the flip-flop element, or combinatorial logic. The multiplexer selects between the sequential or combinatorial logic outputs. The multiplexer output can drive a port pin and has a feedback path to the AND Array inputs.

The flip-flop in the Output Macrocell (OMC) block can be configured as a D, T, JK, or SR type in PSDsoft. The flip-flop's clock, preset, and clear inputs may be driven from a product term of the AND Array. Alternatively, CLKIN (PD1) can be used for the clock input to the flip-flop. The flip-flop is clocked on the rising edge of CLKIN (PD1). The preset and clear are active High inputs. Each clear input can use up to two product terms.

Table 66. Output Macrocell Port and Data Bit Assignments

Output Macrocell	Port Assignment(1)	Native Product Terms	Maximum Borrowed Product Terms	Data Bit for Loading or Reading
McellAB0	Port B0	3	6	D0
McellAB1	Port B1	3	6	D1
McellAB2	Port B2	3	6	D2
McellAB3	Port B3	3	6	D3
McellAB4	Port B4	3	6	D4
McellAB5	Port B5	3	6	D5
McellAB6	Port B6	3	6	D6
McellAB7	Port B7	3	6	D7
McellBC0	Port B0(2)	4	5	D0
McellBC1	Port B1(2)	4	5	D1
McellBC2	Port B2, C2	4	5	D2
McellBC3	Port B3, C3	4	5	D3
McellBC4	Port B4, C4	4	6	D4
McellBC5	Port B5(2)	4	6	D5
McellBC6	Port B6(2)	4	6	D6
McellBC7	Port B7, C7	4	6	D7

Note: 1. Port PC0, PC1, PC5 and PC6 are assigned to JTAG pins, and are not available as macrocell outputs

Product Term Allocator

The CPLD has a Product Term Allocator. PSDsoft uses the Product Term Allocator to borrow and place product terms from one macrocell to another. The following list summarizes how product terms are allocated:

- McellAB0-McellAB7 all have three native product terms and may borrow up to six more
- McellBC0-McellBC3 all have four native product terms and may borrow up to five more
- McellBC4-McellBC7 all have four native product terms and may borrow up to six more.

Each macrocell may only borrow product terms from certain other macrocells. Product terms already in use by one macrocell are not available for another macrocell.

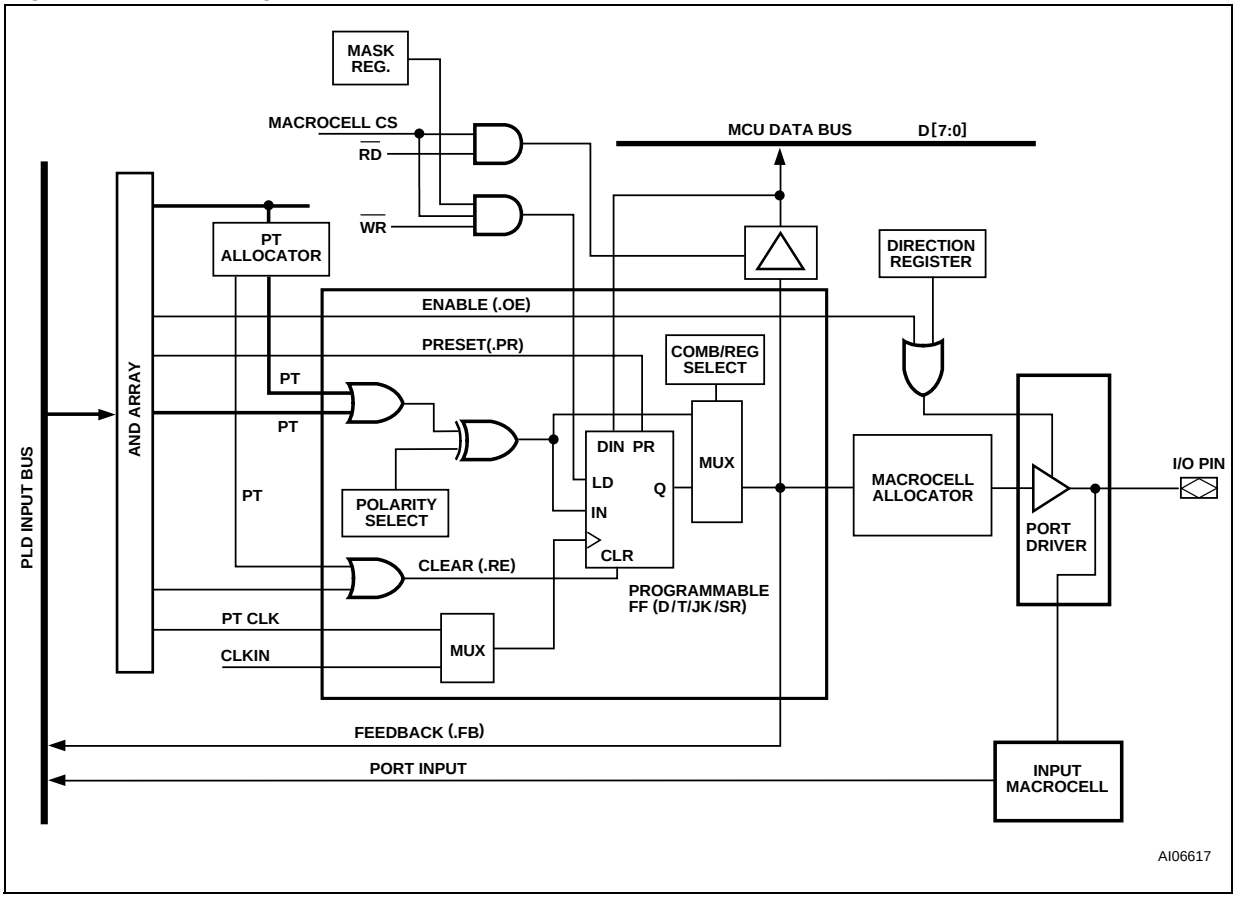
If an equation requires more product terms than are available to it, then "external" product terms are required, which consume other Output Macrocells (OMC). If external product terms are used, extra delay is added for the equation that required the extra product terms.

This is called product term expansion. PSDsoft Express performs this expansion as needed.

Loading and Reading the Output Macrocells (OMC). The Output Macrocells (OMC) block occupies a memory location in the MCU address space, as defined by the CSIOP block (see the section entitled "I/O PORTS (PSD MODULE)," on page 95). The flip-flops in each of the 16 Output Macrocells (OMC) can be loaded from the data bus by a MCU. Loading the Output Macrocells (OMC) with data from the MCU takes priority over internal functions. As such, the preset, clear, and clock inputs to the flip-flop can be overridden by the MCU. The ability to load the flip-flops and read them back is useful in such applications as loadable counters and shift registers, mailboxes, and handshaking protocols.

Data can be loaded to the Output Macrocells (OMC) on the trailing edge of WRITE Strobe (WR, edge loading) or during the time that WRITE Strobe (WR) is active (level loading). The method of loading is specified in PSDsoft Express Configuration.

Figure 46. CPLD Output Macrocell



The OMC Mask Register. There is one Mask Register for each of the two groups of eight Output Macrocells (OMC). The Mask Registers can be used to block the loading of data to individual Output Macrocells (OMC). The default value for the Mask Registers is 00h, which allows loading of the Output Macrocells (OMC). When a given bit in a Mask Register is set to a '1,' the MCU is blocked from writing to the associated Output Macrocells (OMC). For example, suppose McellAB0-McellAB3 are being used for a state machine. You would not want a MCU write to McellAB to overwrite the state machine registers. Therefore, you would want to load the Mask Register for McellAB (Mask Macrocell AB) with the value 0Fh.

The Output Enable of the OMC. The Output Macrocells (OMC) block can be connected to an I/O port pin as a PLD output. The output enable of each port pin driver is controlled by a single product term from the AND Array, ORed with the Direction Register output. The pin is enabled upon Power-up if no output enable equation is defined and if the pin is declared as a PLD output in PSDsoft Express.

If the Output Macrocell (OMC) output is declared as an internal node and not as a port pin output in the PSDabel file, the port pin can be used for other I/O functions. The internal node feedback can be routed as an input to the AND Array.

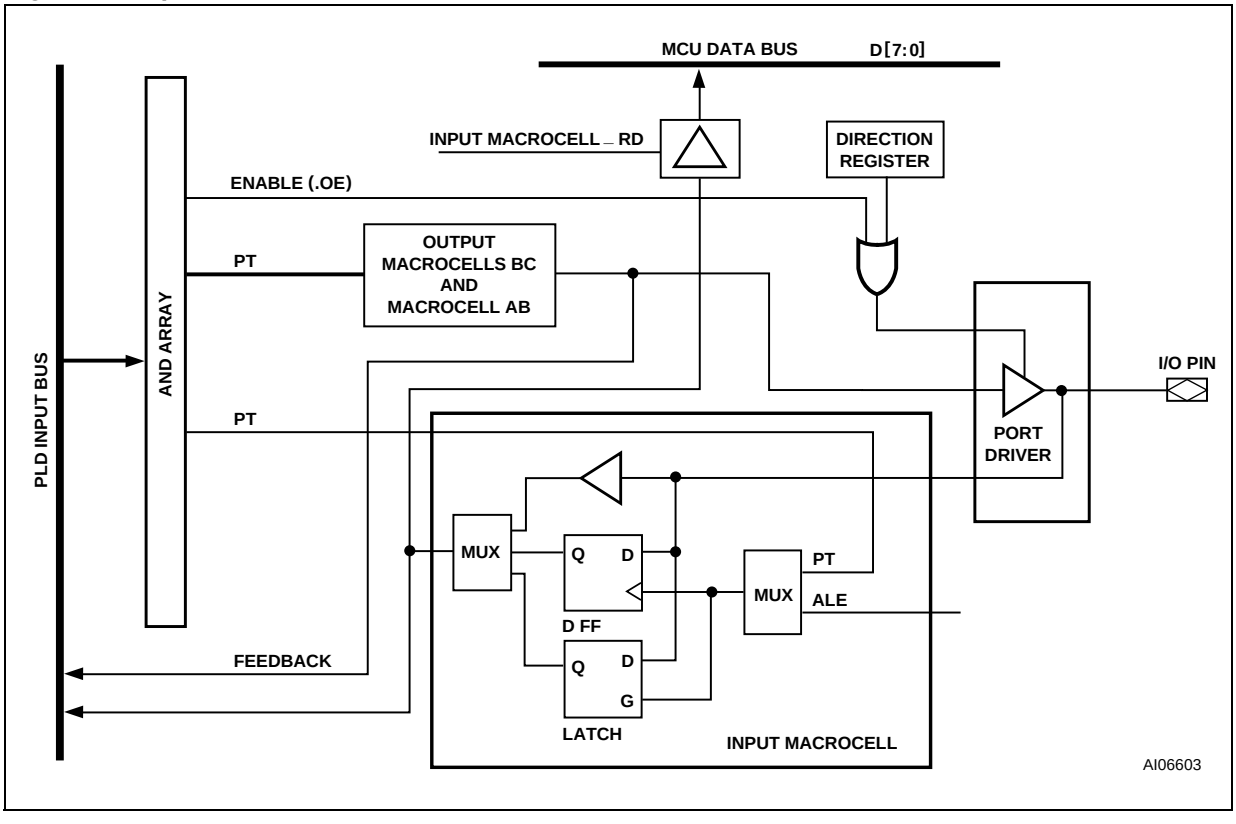
Input Macrocells (IMC)

The CPLD has 12 Input Macrocells (IMC), one for each pin on Port B, and four on Port C. The architecture of the Input Macrocells (IMC) is shown in Figure 47. The Input Macrocells (IMC) are individually configurable, and can be used as a latch, register, or to pass incoming Port signals prior to driving them onto the PLD input bus. The outputs of the Input Macrocells (IMC) can be read by the MCU through the internal data bus.

The enable for the latch and clock for the register are driven by a multiplexer whose inputs are a product term from the CPLD AND Array or the MCU Address Strobe (ALE). Each product term output is used to latch or clock four Input Macrocells (IMC). Port inputs 3-0 can be controlled by one product term and 7-4 by another.

Configurations for the Input Macrocells (IMC) are specified by equations written in PSDsoft (see Application Note *AN1171*). Outputs of the Input Macrocells (IMC) can be read by the MCU via the IMC buffer. See the section entitled “I/O PORTS (PSD MODULE),” page 95.

Figure 47. Input Macrocell



I/O PORTS (PSD MODULE)

There are three programmable I/O ports: Ports B, C, and D in the PSD Module. Each of the ports is eight bits except Port D, which is 1 bit. Each port pin is individually user configurable, thus allowing multiple functions per port. The ports are configured using PSDsoft Express Configuration or by the MCU writing to on-chip registers in the CSIOP space.

The topics discussed in this section are:

- General Port architecture
- Port operating modes
- Port Configuration Registers (PCR)
- Port Data Registers
- Individual Port functionality.

General Port Architecture

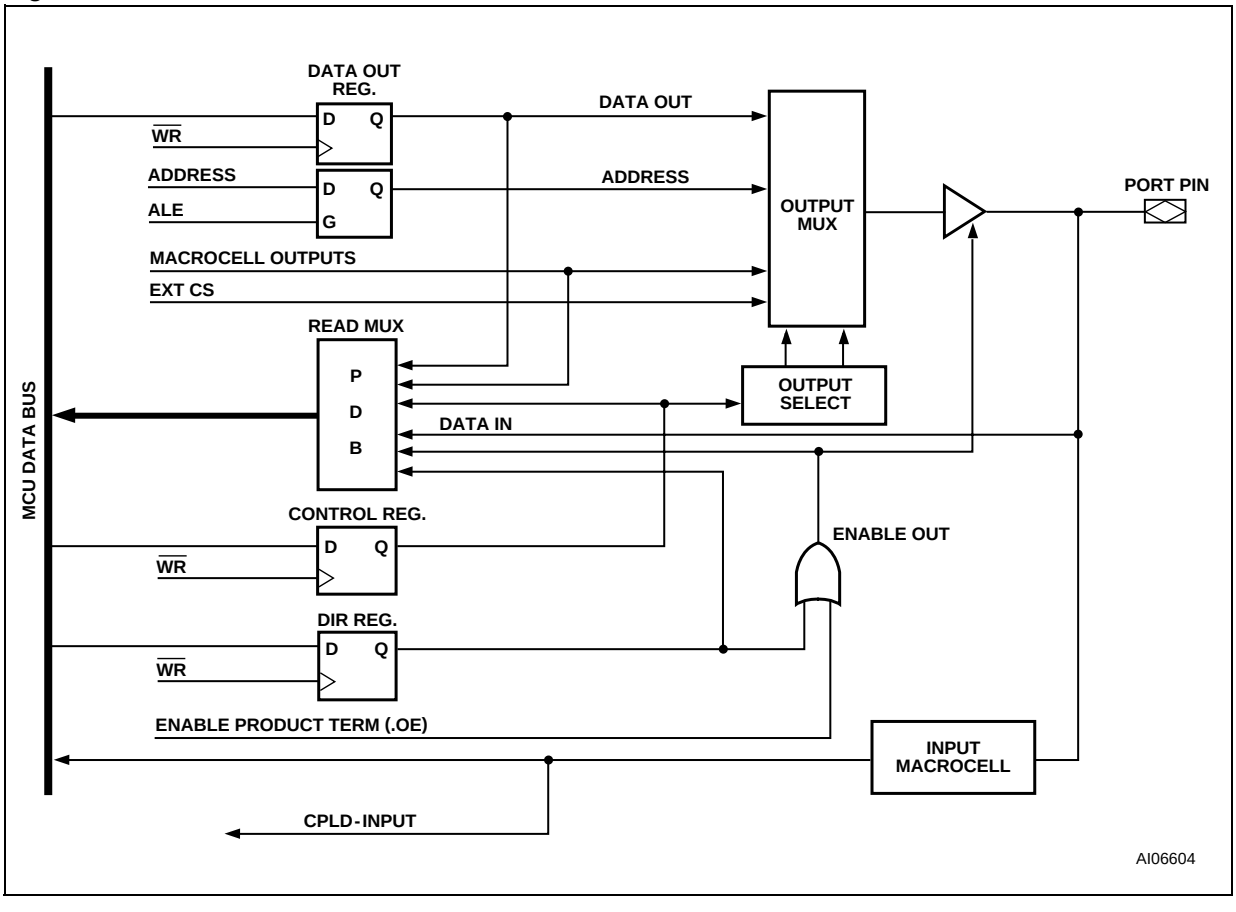
The general architecture of the I/O Port block is shown in Figure 48. Individual Port architectures are shown in Figure 49 to Figure 52. In general, once the purpose for a port pin has been defined, that pin is no longer available for other purposes. Exceptions are noted.

As shown in Figure 48, the ports contain an output multiplexer whose select signals are driven by the configuration bits in the Control Registers (Port B only) and PSDsoft Express Configuration. Inputs to the multiplexer include the following:

- Output data from the Data Out register
- Latched address outputs
- CPLD macrocell output
- External Chip Select (ECS1) from the CPLD.

The Port Data Buffer (PDB) is a tri-state buffer that allows only one source at a time to be read. The Port Data Buffer (PDB) is connected to the Internal Data Bus for feedback and can be read by the MCU. The Data Out and macrocell outputs, Direction and Control Registers, and port pin input are all connected to the Port Data Buffer (PDB).

Figure 48. General I/O Port Architecture



The Port pin's tri-state output driver enable is controlled by a two input OR gate whose inputs come from the CPLD AND Array enable product term and the Direction Register. If the enable product term of any of the Array outputs are not defined and that port pin is not defined as a CPLD output in the PSDsoft, then the Direction Register has sole control of the buffer that drives the port pin.

The contents of these registers can be altered by the MCU. The Port Data Buffer (PDB) feedback path allows the MCU to check the contents of the registers.

Ports B and C have embedded Input Macrocells (IMC). The Input Macrocells (IMC) can be configured as latches, registers, or direct inputs to the PLDs. The latches and registers are clocked by Address Strobe (ALE) or a product term from the PLD AND Array. The outputs from the Input Macrocells (IMC) drive the PLD input bus and can be read by the MCU. See the section entitled "Input Macrocell," page 94.

Port Operating Modes

The I/O Ports have several modes of operation. Some modes can be defined using PSDsoft, some by the MCU writing to the Control Registers in CSIOP space, and some by both. The modes that can only be defined using PSDsoft must be programmed into the device and cannot be changed unless the device is reprogrammed. The modes that can be changed by the MCU can be done so dynamically at run-time. The PLD I/O, Data Port, and Address Input Modes are the only modes that must be defined before programming the device. All other modes can be changed by the MCU at run-time. See Application Note *AN1171* for more detail.

Table 67 summarizes which modes are available on each port. Table 70 shows how and where the different modes are configured. Each of the port operating modes are described in the following sections.

MCU I/O Mode

In the MCU I/O Mode, the MCU uses the I/O Ports block to expand its own I/O ports. By setting up the CSIOP space, the ports on the PSD Module are mapped into the MCU address space. The addresses of the ports are listed in Table 59.

A port pin can be put into MCU I/O Mode by writing a '0' to the corresponding bit in the Control Register. The MCU I/O direction may be changed by writing to the corresponding bit in the Direction Register, or by the output enable product term. When the pin is configured as an output, the content of the Data Out Register drives the pin. When configured as an input, the MCU can read the port input through the Data In buffer. See Figure 48, page 95.

Ports C and D do not have Control Registers, and are in MCU I/O Mode by default. They can be used for PLD I/O if equations are written for them in PSDabel.

PLD I/O Mode

The PLD I/O Mode uses a port as an input to the CPLD's Input Macrocells (IMC), and/or as an output from the CPLD's Output Macrocells (OMC). The output can be tri-stated with a control signal. This output enable control signal can be defined by a product term from the PLD, or by resetting the corresponding bit in the Direction Register to '0.' The corresponding bit in the Direction Register must not be set to '1' if the pin is defined for a PLD input signal in PSDsoft. The PLD I/O Mode is specified in PSDsoft by declaring the port pins, and then writing an equation assigning the PLD I/O to a port.

Address Out Mode

Address Out Mode can be used to drive latched MCU addresses on to the port pins. These port pins can, in turn, drive external devices. Either the output enable or the corresponding bits of both the Direction Register and Control Register must be set to a '1' for pins to use Address Out Mode. This must be done by the MCU at run-time. See Table 69 for the address output pin assignments on Port B.

JTAG In-System Programming (ISP)

Port C is JTAG compliant, and can be used for In-System Programming (ISP). For more information on the JTAG Port, see the section entitled "PROGRAMMING IN-CIRCUIT USING THE JTAG SERIAL INTERFACE," page 109.

Table 67. Port Operating Modes

Port Mode	Port B	Port C	Port D
MCU I/O	Yes	Yes	Yes
PLD I/O McellAB Outputs McellBC Outputs Additional Ext. CS Outputs PLD Inputs	 Yes Yes No Yes	 No Yes(3) No Yes	 No No Yes Yes
Address Out	Yes (A7 – 0)	No	No
JTAG ISP	No	Yes(1)	No

Note: 1. JTAG pins (TMS, TCK, TDI, TDO) are dedicated pins.
2. On pins PC2, PC3, PC4 and PC7 only.

Table 68. Port Operating Mode Settings

Mode	Defined in PSDsoft	Control Register Setting	Direction Register Setting	VM Register Setting
MCU I/O	Declare pins only	0	1 = output, 0 = input (Note 1)	N/A
PLD I/O	Logic equations	N/A	(Note 1)	N/A
Address Out (Port B)	Declare pins only	1	1 (Note 1)	N/A

Note: N/A = Not Applicable
1. The direction of the Port B, C, and D pins are controlled by the Direction Register ORed with the individual output enable product term (.oe) from the CPLD AND Array.

Table 69. I/O Port Latched Address Output Assignments

Port B (PB3-PB0)	Port B (PB7-PB4)
Address a3-a0	Address a7-a4

Port Configuration Registers (PCR)

Each Port has a set of Port Configuration Registers (PCR) used for configuration. The contents of the registers can be accessed by the MCU through normal READ/WRITE bus cycles at the addresses given in Table 59. The addresses in Table 59 are the offsets in hexadecimal from the base of the CSIOP register.

The pins of a port are individually configurable and each bit in the register controls its respective pin. For example, Bit 0 in a register refers to Bit 0 of its port. The three Port Configuration Registers (PCR), shown in Table 70, are used for setting the Port configurations. The default Power-up state for each register in Table 70 is 00h.

Control Register. Any bit reset to '0' in the Control Register sets the corresponding port pin to MCU I/O Mode, and a '1' sets it to Address Out Mode. The default mode is MCU I/O. Only Port B has an associated Control Register.

Direction Register. The Direction Register, in conjunction with the output enable (except for Port D), controls the direction of data flow in the I/O Ports. Any bit set to '1' in the Direction Register causes the corresponding pin to be an output, and any bit set to '0' causes it to be an input. The default mode for all port pins is input.

Figure 49, page 100 and Figure 50, page 101 show the Port Architecture diagrams for Ports B and C, respectively. The direction of data flow for Ports B, and C are controlled not only by the direction register, but also by the output enable product term from the PLD AND Array. If the output enable product term is not active, the Direction Register has sole control of a given pin's direction.

An example of a configuration for a Port with the three least significant bits set to output and the remainder set to input is shown in Table 73. Since Port D only contains one pin (shown in Figure 52), the Direction Register for Port D has only one bit active.

Drive Select Register. The Drive Select Register configures the pin driver as Open Drain or CMOS for some port pins, and controls the slew rate for the other port pins. An external pull-up resistor should be used for pins configured as Open Drain.

A pin can be configured as Open Drain if its corresponding bit in the Drive Select Register is set to a '1.' The default pin drive is CMOS.

Note: The slew rate is a measurement of the rise and fall times of an output. A higher slew rate means a faster output response and may create more electrical noise. A pin operates in a high slew rate when the corresponding bit in the Drive Register is set to '1.' The default rate is slow slew.

Table 74, page 99 shows the Drive Register for Ports B, C, and D. It summarizes which pins can be configured as Open Drain outputs and which pins the slew rate can be set for.

Table 70. Port Configuration Registers (PCR)

Register Name	Port	MCU Access
Control	B	WRITE/READ
Direction	B, C, D	WRITE/READ
Drive Select(1)	B, C, D	WRITE/READ

Note: 1. See Table 74 for Drive Register Bit definition.

Table 71. Port Pin Direction Control, Output Enable P.T. Not Defined

Direction Register Bit	Port Pin Mode
0	Input
1	Output

Table 72. Port Pin Direction Control, Output Enable P.T. Defined

Direction Register Bit	Output Enable P.T.	Port Pin Mode
0	0	Input
0	1	Output
1	0	Output
1	1	Output

Table 73. Port Direction Assignment Example

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

Port Data Registers

The Port Data Registers, shown in Table 75, are used by the MCU to write data to or read data from the ports. Table 75 shows the register name, the ports having each register type, and MCU access for each register type. The registers are described below.

Data In. Port pins are connected directly to the Data In buffer. In MCU I/O Input Mode, the pin input is read through the Data In buffer.

Data Out Register. Stores output data written by the MCU in the MCU I/O Output Mode. The contents of the Register are driven out to the pins if the Direction Register or the output enable product term is set to '1.' The contents of the register can also be read back by the MCU.

Output Macrocells (OMC). The CPLD Output Macrocells (OMC) occupy a location in the MCU's address space. The MCU can read the output of the Output Macrocells (OMC). If the OMC Mask Register Bits are not set, writing to the macrocell loads data to the macrocell flip-flops. See the section entitled "PLDs," page 88.

OMC Mask Register. Each OMC Mask Register Bit corresponds to an Output Macrocell (OMC) flip-flop. When the OMC Mask Register Bit is set to a '1,' loading data into the Output Macrocell (OMC) flip-flop is blocked. The default value is '0' or unblocked.

Input Macrocells (IMC). The Input Macrocells (IMC) can be used to latch or store external inputs. The outputs of the Input Macrocells (IMC) are routed to the PLD input bus, and can be read by the MCU. See the section entitled "PLDs," page 88.

Enable Out. The Enable Out register can be read by the MCU. It contains the output enable values for a given port. A '1' indicates the driver is in output mode. A '0' indicates the driver is in tri-state and the pin is in input mode.

Table 74. Drive Register Pin Assignment

Drive Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Port B	Open Drain	Open Drain	Open Drain	Open Drain	Slew Rate	Slew Rate	Slew Rate	Slew Rate
Port C	Open Drain	NA(1)	NA(1)	Open Drain	Open Drain	Open Drain	NA(1)	NA(1)
Port D	NA(1)	NA(1)	NA(1)	NA(1)	NA(1)	Slew Rate	Slew Rate	NA(1)

Note: 1. NA = Not Applicable.

Table 75. Port Data Registers

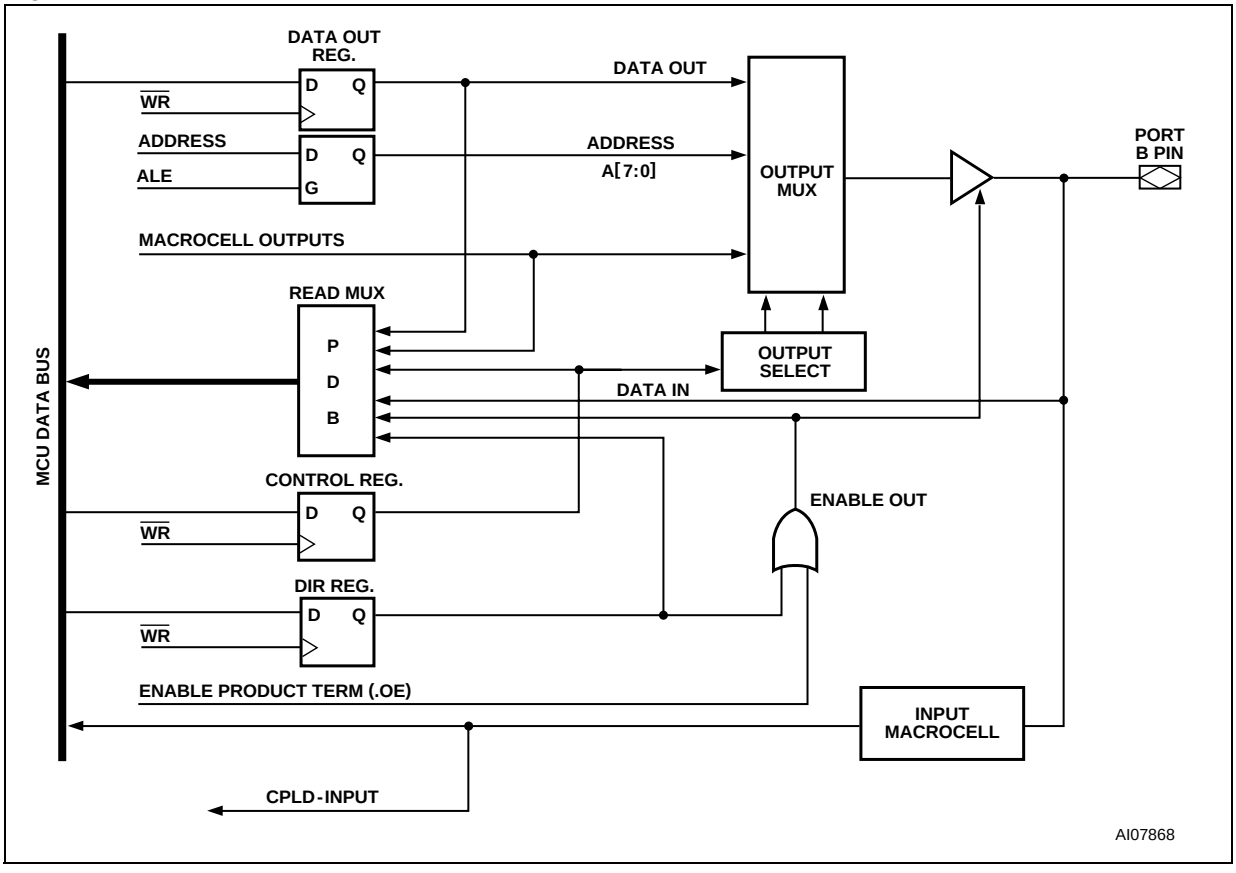
Register Name	Port	MCU Access
Data In	B, C, D	READ – input on pin
Data Out	B, C, D	WRITE/READ
Output Macrocell	B, C	READ – outputs of macrocells WRITE – loading macrocells flip-flop
Mask Macrocell	B, C	WRITE/READ – prevents loading into a given macrocell
Input Macrocell	B, C	READ – outputs of the Input Macrocells
Enable Out	B, C	READ – the output enable control of the port driver

Port B – Functionality and Structure

Port B can be configured to perform one or more of the following functions (see Figure 49):

- MCU I/O Mode
- CPLD Output – Macrocells McellAB7-McellAB0 can be connected to Port B. McellBC7-McellBC0 can be connected to Port B or Port C.
- CPLD Input – Via the Input Macrocells (IMC).
- Latched Address output – Provide latched address output as per Table 69.
- Open Drain/Slew Rate – pins PB3-PB0 can be configured to fast slew rate, pins PB7-PB4 can be configured to Open Drain Mode.

Figure 49. Port B Structure



Port C – Functionality and Structure

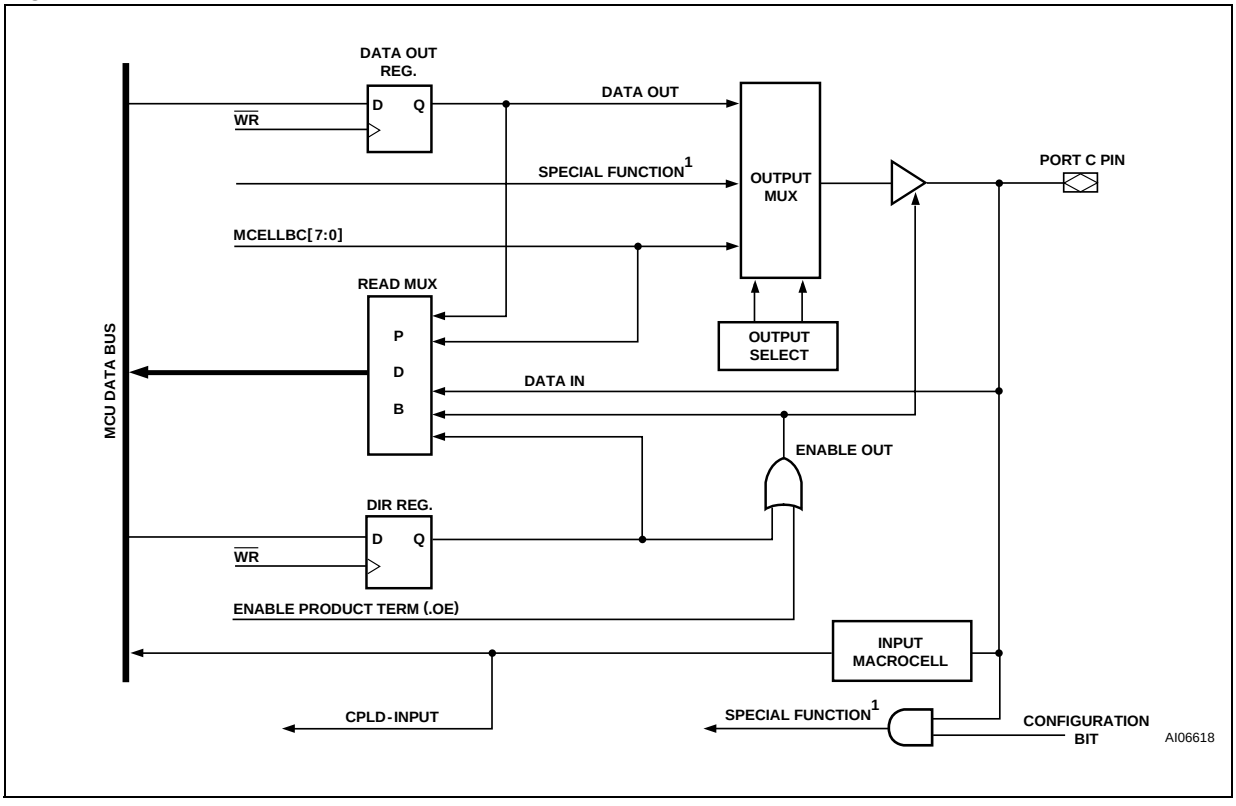
Port C can be configured to perform one or more of the following functions (see Figure 50):

- MCU I/O Mode
- CPLD Output – McellBC7-McellBC0 outputs can be connected to Port B or Port C.
- CPLD Input – via the Input Macrocells (IMC)
- In-System Programming (ISP) – JTAG pins (TMS, TCK, TDI, TDO) are dedicated pins for device programming. (See the section entitled "PROGRAMMING IN-CIRCUIT USING THE JTAG SERIAL INTERFACE," page 109, for more information on JTAG programming.)
- Open Drain – Port C pins can be configured in Open Drain Mode
- Battery Backup features – PC2 can be configured for a battery input supply, Voltage Standby (V_{STBY}).

  PC4 can be configured as a Battery-on Indicator (V_{BATON}), indicating when V_{CC} is less than V_{BAT}.

Port C does not support Address Out Mode, and therefore no Control Register is required.

Figure 50. Port C Structure



Note: 1. ISP or battery back-up

Port D – Functionality and Structure

Port D has only one pin, PD1. See Figure 51 and Figure 52. This port does not support Address Out Mode, and therefore no Control Register is required. Of the eight bits in the Port D registers, only Bit 1 is used to configure pin PD1.

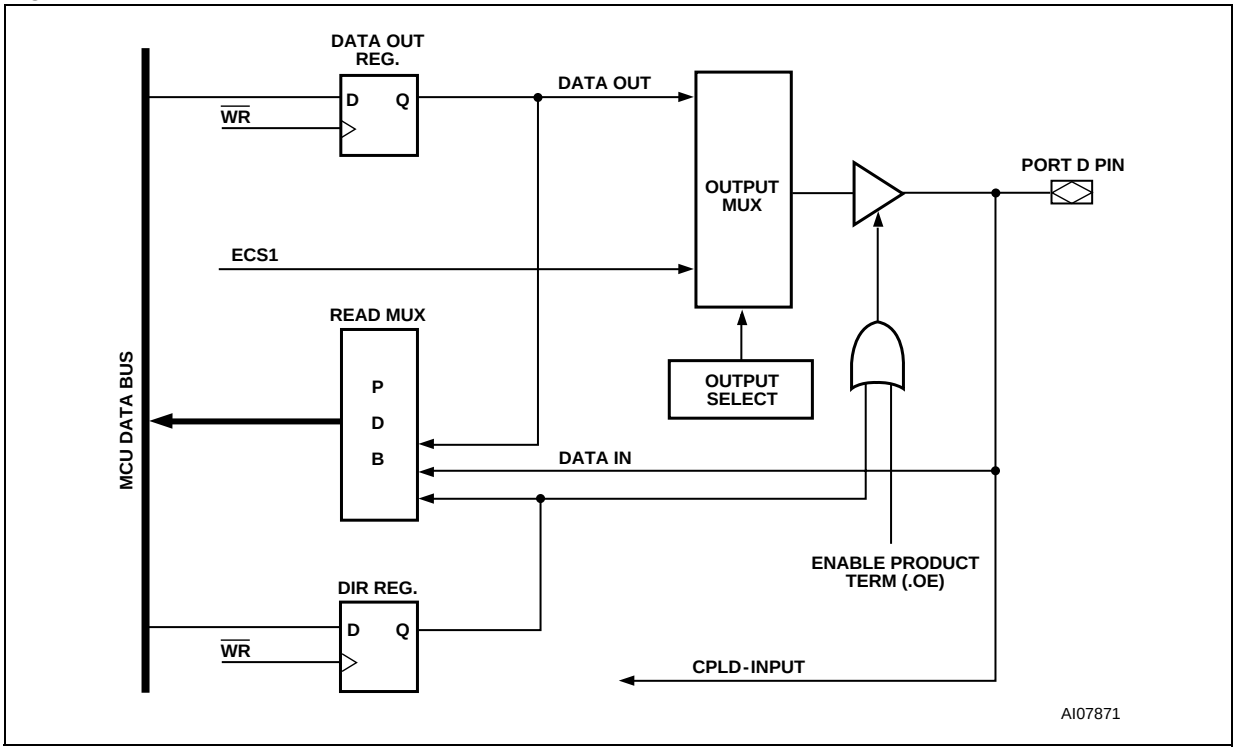
Port D can be configured to perform one or more of the following functions:

- MCU I/O Mode
- CPLD Output – External Chip Select (ECS1)
- CPLD Input – direct input to the CPLD, no Input Macrocells (IMC)
- Slew rate – pins can be set up for fast slew rate

Port D pins can be configured in PSDsoft Express as input pins for other dedicated functions:

- CLKIN (PD1) as input to the macrocells flip-flops and APD counter

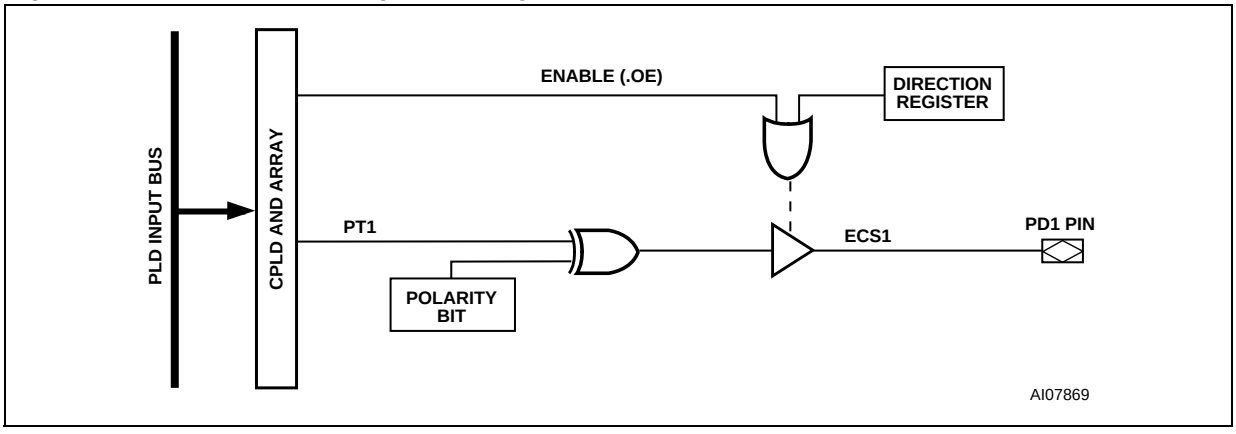
Figure 51. Port D Structure



External Chip Select

The CPLD also provides one External Chip Select (ECS1) output on the Port D pin that can be used to select external devices. External Chip Select (ECS1) consists of one product term that can be configured active High or Low. The output enable of the pin is controlled by either the output enable product term or the Direction Register. (See Figure 52.)

Figure 52. Port D External Chip Select Signals



POWER MANAGEMENT

All PSD Module offers configurable power saving options. These options may be used individually or in combinations, as follows:

- The primary and secondary Flash memory, and SRAM blocks are built with power management technology. In addition to using special silicon design methodology, power management technology puts the memories into Standby Mode when address/data inputs are not changing (zero DC current). As soon as a transition occurs on an input, the affected memory "wakes up," changes and latches its outputs, then goes back to standby. The designer does *not* have to do anything special to achieve Memory Standby Mode when no inputs are changing—it happens automatically.

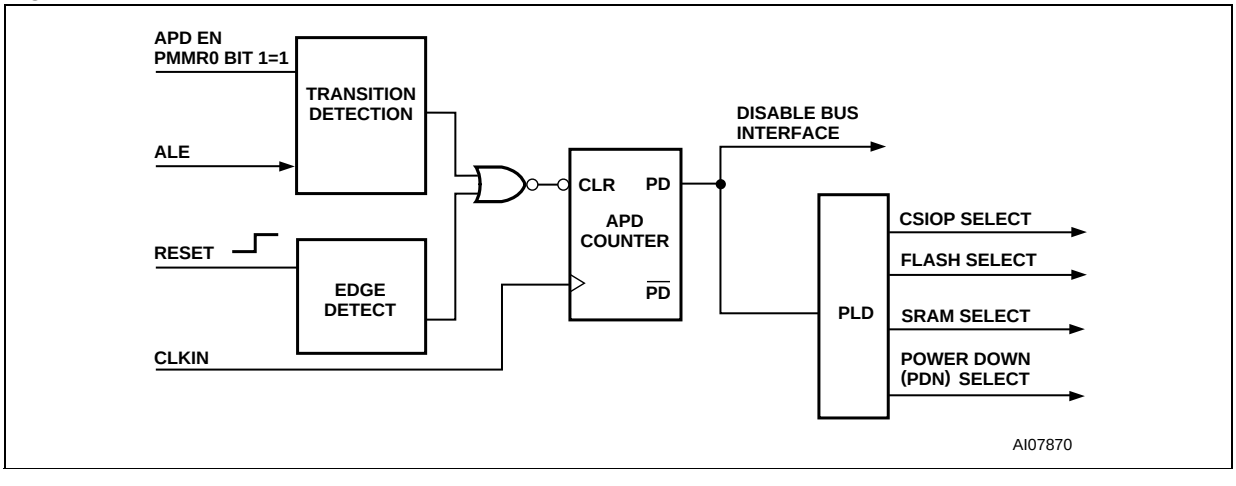
  The PLD sections can also achieve Standby Mode when its inputs are not changing, as described in the sections on the Power Management Mode Registers (PMMR).

- As with the Power Management Mode, the Automatic Power Down (APD) block allows the PSD Module to reduce to standby current automatically. The APD Unit can also block MCU address/data signals from reaching the memories and PLDs. The APD Unit is described in more detail in the sections entitled "POWER MANAGEMENT" page 104.

  Built in logic monitors the Address Strobe of the MCU for activity. If there is no activity for a certain time period (MCU is asleep), the APD Unit initiates Power-down Mode (if enabled). Once in Power-down Mode, all address/data signals are blocked from reaching memory and PLDs, and the memories are deselected internally. This allows the memory and PLDs to remain in Standby Mode even if the address/data signals are changing state externally (noise, other devices on the MCU bus, etc.). Keep in mind that any unblocked PLD input signals that are changing states keeps the PLD out of Standby Mode, but not the memories.

- The PMMRs can be written by the MCU at runtime to manage power. The PSD Module supports "blocking bits" in these registers that are set to block designated signals from reaching both PLDs. Current consumption of the PLDs is directly related to the composite frequency of the changes on their inputs (see Figure 56). Significant power savings can be achieved by blocking signals that are not used in DPLD or CPLD logic equations.

Figure 53. APD Unit



The PSD Module has a Turbo Bit in PMMR0. This bit can be set to turn the Turbo Mode off (the default is with Turbo Mode turned on). While Turbo Mode is off, the PLDs can achieve standby current when no PLD inputs are changing (zero DC current). Even when inputs do change, significant power can be saved at lower frequencies (AC current), compared to when Turbo Mode is on. When the Turbo Mode is on, there is a significant DC current component and the AC component is higher.

Automatic Power-down (APD) Unit and Power-down Mode. The APD Unit, shown in Figure 53, puts the PSD Module into Power-down Mode by monitoring the activity of Address Strobe (ALE). If the APD Unit is enabled, as soon as activity on Address Strobe (ALE) stops, a four-bit counter starts counting. If Address Strobe (ALE/AS) remains inactive for fifteen clock periods of CLKIN (PD1), Power-down (PDN) goes High, and the PSD Module enters Power-down Mode, as discussed next.

Power-down Mode. By default, if you enable the APD Unit, Power-down Mode is automatically enabled. The device enters Power-down Mode if Address Strobe (ALE) remains inactive for fifteen periods of CLKIN (PD1).

The following should be kept in mind when the PSD Module is in Power-down Mode:

– If Address Strobe (ALE) starts pulsing again, the PSD Module returns to normal Operating mode. The PSD Module also returns to normal Operating mode if the RESET input is High.
– The MCU address/data bus is blocked from all memory and PLDs.
– Various signals can be blocked (prior to Power-down Mode) from entering the PLDs by setting the appropriate bits in the PMMR registers. The blocked signals include MCU control signals and the common CLKIN (PD1).
– **Note:** Blocking CLKIN (PD1) from the PLDs does not block CLKIN (PD1) from the APD Unit.
– All memories enter Standby Mode and are drawing standby current. However, the PLD and I/O ports blocks do *not* go into Standby Mode because you don't want to have to wait for the logic and I/O to "wake-up" before their outputs can change. See Table 76 for Power-down Mode effects on PSD Module ports.
– Typical standby current is of the order of micro-amperes. These standby current values assume that there are no transitions on any PLD input.

Other Power Saving Options. The PSD Module offers other reduced power saving options that are independent of the Power-down Mode. Except for the SRAM Standby, they are enabled by setting bits in PMMR0 and PMMR2.

Figure 54. Enable Power-down Flow Chart

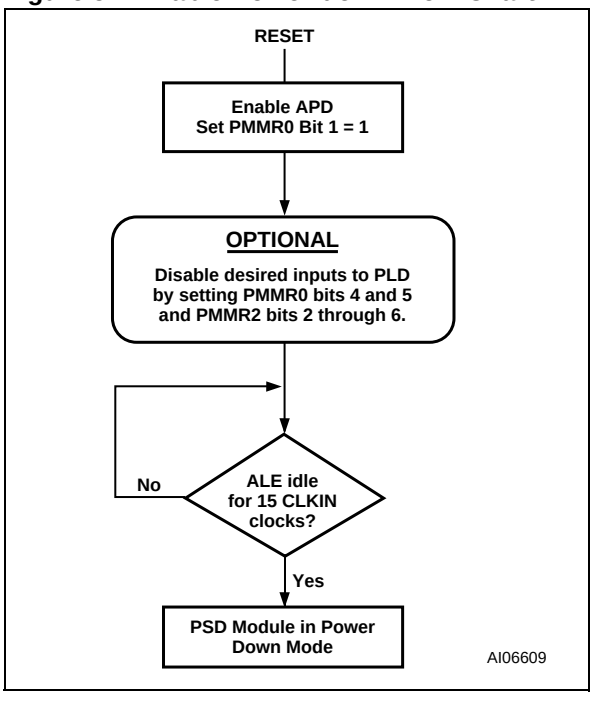


Table 76. Power-down Mode's Effect on Ports

Port Function	Pin Level
MCU I/O	No Change
PLD Out	No Change
Address Out	Undefined

PLD Power Management

The power and speed of the PLDs are controlled by the Turbo Bit (Bit 3) in PMMR0 (see Table 77). By setting the bit to '1,' the Turbo Mode is off and the PLDs consume the specified standby current when the inputs are not switching for an extended time of 70ns. The propagation delay time is increased by 10ns (for a 5V device) after the Turbo Bit is set to '1' (turned off) when the inputs change at a composite frequency of less than 15MHz. When the Turbo Bit is reset to '0' (turned on), the PLDs run at full power and speed. The Turbo Bit affects the PLD's DC power, AC power, and propagation delay. When the Turbo Mode is off, the µPSD3200 input clock frequency is reduced by 5MHz from the maximum rated clock frequency.

Blocking MCU control signals with the bits of PMMR2 (see Table 78, page 107) can further reduce PLD AC power consumption.

SRAM Standby Mode (Battery Backup). The SRAM in the PSD Module supports a battery backup mode in which the contents are retained in the event of a power loss. The SRAM has Voltage Standby (V_{STBY}, PC2) that can be connected to an external battery. When V_{CC} becomes lower than V_{STBY} then the SRAM automatically connects to Voltage Standby (V_{STBY}, PC2) as a power source. The SRAM Standby Current (I_{STBY}) is typically 0.5 µA. The SRAM data retention voltage is 2V minimum. The Battery-on Indicator (V_{BATON}) can be routed to PC4. This signal indicates when the V_{CC} has dropped below V_{STBY}.

Input Clock

CLKIN (PD1) can be turned off, to the PLD to save AC power consumption. CLKIN (PD1) is an input to the PLD AND Array and the Output Macrocells (OMC).

During Power-down Mode, or, if CLKIN (PD1) is not being used as part of the PLD logic equation, the clock should be disabled to save AC power. CLKIN (PD1) is disconnected from the PLD AND Array or the Macrocells block by setting Bits 4 or 5 to a '1' in PMMR0.

Input Control Signals

The PSD Module provides the option to turn off the MCU signals (WR, RD, PSEN, and Address Strobe (ALE)) to the PLD to save AC power consumption (see Table 79, page 107). These control signals are inputs to the PLD AND Array. During Power-down Mode, or, if any of them are not being used as part of the PLD logic equation, these control signals should be disabled to save AC power. They are disconnected from the PLD AND Array by setting Bits 2, 3, 4, 5, and 6 to a '1' in PMMR2.

Table 77. Power Management Mode Registers PMMR0

Bit 0	X	0	Not used, and should be set to zero.
Bit 1	APD Enable	0 = off	Automatic Power-down (APD) is disabled.
		1 = on	Automatic Power-down (APD) is enabled.
Bit 2	X	0	Not used, and should be set to zero.
Bit 3	PLD Turbo	0 = on	PLD Turbo Mode is on
		1 = off	PLD Turbo Mode is off, saving power. µPSD3200 operates at 5MHz below the maximum rated clock frequency
Bit 4	PLD Array clk	0 = on	CLKIN (PD1) input to the PLD AND Array is connected. Every change of CLKIN (PD1) Powers-up the PLD when Turbo Bit is '0.'
		1 = off	CLKIN (PD1) input to PLD AND Array is disconnected, saving power.
Bit 5	PLD MCell clk	0 = on	CLKIN (PD1) input to the PLD macrocells is connected.
		1 = off	CLKIN (PD1) input to PLD macrocells is disconnected, saving power.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

Table 78. Power Management Mode Registers PMMR2

Bit 0	X	0	Not used, and should be set to zero.
Bit 1	X	0	Not used, and should be set to zero.
Bit 2	PLD Array $\overline{WR}$	0 = on	$\overline{WR}$ input to the PLD AND Array is connected.
		1 = off	$\overline{WR}$ input to PLD AND Array is disconnected, saving power.
Bit 3	PLD Array $\overline{RD}$	0 = on	$\overline{RD}$ input to the PLD AND Array is connected.
		1 = off	$\overline{RD}$ input to PLD AND Array is disconnected, saving power.
Bit 4	PLD Array $\overline{PSEN}$	0 = on	$\overline{PSEN}$ input to the PLD AND Array is connected.
		1 = off	$\overline{PSEN}$ input to PLD AND Array is disconnected, saving power.
Bit 5	PLD Array ALE	0 = on	ALE input to the PLD AND Array is connected.
		1 = off	ALE input to PLD AND Array is disconnected, saving power.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

Note: The bits of this register are cleared to zero following Power-up. Subsequent $\overline{RESET}$ pulses do not clear the registers.

Table 79. APD Counter Operation

APD Enable Bit	ALE Level	APD Counter
0	X	Not Counting
1	Pulsing	Not Counting
1	0 or 1	Counting (Generates PDN after 15 Clocks)

RESET TIMING AND DEVICE STATUS AT RESET

Upon Power-up, the PSD Module requires a Reset (RESET) pulse of duration $t_{NLNH-PO}$ after V_{CC} is steady. During this period, the device loads internal configurations, clears some of the registers and sets the Flash memory into operating mode. After the rising edge of Reset (RESET), the PSD Module remains in the Reset Mode for an additional period, t_{OPR}, before the first memory access is allowed.

The Flash memory is reset to the READ Mode upon Power-up. Sector Select (FS0-FS3 and CSBOOT0-CSBOOT1) must all be Low, WRITE Strobe (WR, CNTL0) High, during Power-on RESET for maximum security of the data contents and to remove the possibility of a byte being written on the first edge of WRITE Strobe (WR). Any Flash memory WRITE cycle initiation is prevented automatically when V_{CC} is below V_{LKO}.

Warm RESET

Once the device is up and running, the PSD Module can be reset with a pulse of a much shorter duration, t_{NLNH}. The same t_{OPR} period is needed before the device is operational after a Warm RESET. Figure 55 shows the timing of the Power-up and Warm RESET.

I/O Pin, Register and PLD Status at RESET

Table 80 shows the I/O pin, register and PLD status during Power-on RESET, Warm RESET, and Power-down Mode. PLD outputs are always valid during Warm RESET, and they are valid in Power-on RESET once the internal Configuration bits are loaded. This loading is completed typically long before the V_{CC} ramps up to operating level. Once the PLD is active, the state of the outputs are determined by the PLD equations.

Figure 55. Reset (RESET) Timing

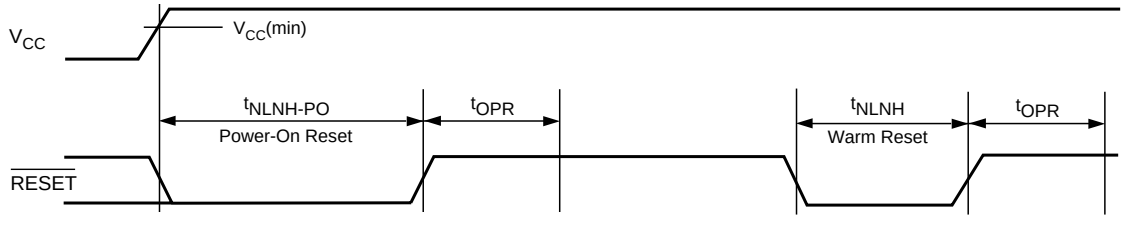


Table 80. Status During Power-on RESET, Warm RESET and Power-down Mode

Port Configuration	Power-on RESET	Warm RESET	Power-down Mode
MCU I/O	Input mode	Input mode	Unchanged
PLD Output	Valid after internal PSD configuration bits are loaded	Valid	Depends on inputs to PLD (addresses are blocked in PD Mode)
Address Out	Tri-stated	Tri-stated	Not defined

Register	Power-on RESET	Warm RESET	Power-down Mode
PMMR0 and PMMR2	Cleared to '0'	Unchanged	Unchanged
Macrocells flip-flop status	Cleared to '0' by internal Power-on RESET	Depends on .re and .pr equations	Depends on .re and .pr equations
VM Register(1)	Initialized, based on the selection in PSDsoft Configuration menu	Initialized, based on the selection in PSDsoft Configuration menu	Unchanged
All other registers	Cleared to '0'	Cleared to '0'	Unchanged

Note: 1. The SR_cod and PeriphMode Bits in the VM Register are always cleared to '0' on Power-on RESET or Warm RESET.

PROGRAMMING IN-CIRCUIT USING THE JTAG SERIAL INTERFACE

The JTAG Serial Interface pins (TMS, TCK, TDI, TDO) are dedicated pins on Port C (see Table 81). All memory blocks (primary and secondary Flash memory), PLD logic, and PSD Module Configuration Register Bits may be programmed through the JTAG Serial Interface block. A blank device can be mounted on a printed circuit board and programmed using JTAG.

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

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

Standard JTAG Signals

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

The RESET input to the µPS3200 should be active during JTAG programming. The active RESET puts the MCU module into RESET Mode while the PSD Module is being programmed. See Application Note AN1153 for more details on JTAG In-System Programming (ISP).

The µPSD3251F device supports JTAG In-System-Configuration (ISC) commands, but not Boundary Scan. The PSDsoft Express software tool and FlashLINK JTAG programming cable implement the JTAG In-System-Configuration (ISC) commands.

Table 81. JTAG Port Signals

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

JTAG Extensions

TSTAT and TERR are two JTAG extension signals enabled by an “ISC_ENABLE” command received over the four standard JTAG signals (TMS, TCK, TDI, and TDO). They are used to speed Program and Erase cycles by indicating status on µPDS signals instead of having to scan the status out serially using the standard JTAG channel. See Application Note *AN1153*.

TERR indicates if an error has occurred when erasing a sector or programming a byte in Flash memory. This signal goes Low (active) when an Error condition occurs, and stays Low until an “ISC_CLEAR” command is executed or a chip Reset (RESET) pulse is received after an “ISC_DISABLE” command.

TSTAT behaves the same as Ready/Busy described in the section entitled “Ready/Busy (PC3),” page 75. TSTAT is High when the PSD Module device is in READ Mode (primary and secondary Flash memory contents can be read). TSTAT is Low when Flash memory Program or Erase cycles are in progress, and also when data is being written to the secondary Flash memory.

TSTAT and TERR can be configured as “open drain” type signals during an “ISC_ENABLE” command.

Security and Flash memory Protection

When the Security Bit is set, the device cannot be read on a Device Programmer or through the JTAG Port. When using the JTAG Port, only a Full Chip Erase command is allowed.

All other Program, Erase and Verify commands are blocked. Full Chip Erase returns the part to a non-secured blank state. The Security Bit can be set in PSDsoft Express Configuration.

All primary and secondary Flash memory sectors can individually be sector protected against erasures. The sector protect bits can be set in PSDsoft Express Configuration.

INITIAL DELIVERY STATE

When delivered from ST, the µPSD3251F device have all bits in the memory and PLDs set to '1.' The code, configuration, and PLD logic are loaded using the programming procedure. Information for programming the device is available directly from ST. Please contact your local sales representative.

AC/DC PARAMETERS

These tables describe the AD and DC parameters of the µPSD3251F device:

➜ DC Electrical Specification

➜ AC Timing Specification

- PLD Timing
  - Combinatorial Timing
  - Synchronous Clock Mode
  - Asynchronous Clock Mode
  - Input Macrocell Timing
- MCU Module Timing
  - READ Timing
  - WRITE Timing
  - Power-down and $\overline{\text{RESET}}$ Timing

The following are issues concerning the parameters presented:

- In the DC specification the supply current is given for different modes of operation.
- The AC power component gives the PLD, Flash memory, and SRAM mA/MHz specification. Figure 56 shows the PLD mA/MHz as a function of the number of Product Terms (PT) used.
- In the PLD timing parameters, add the required delay when Turbo Bit is '0.'

Figure 56. PLD I_{CC} /Frequency Consumption (5V range)

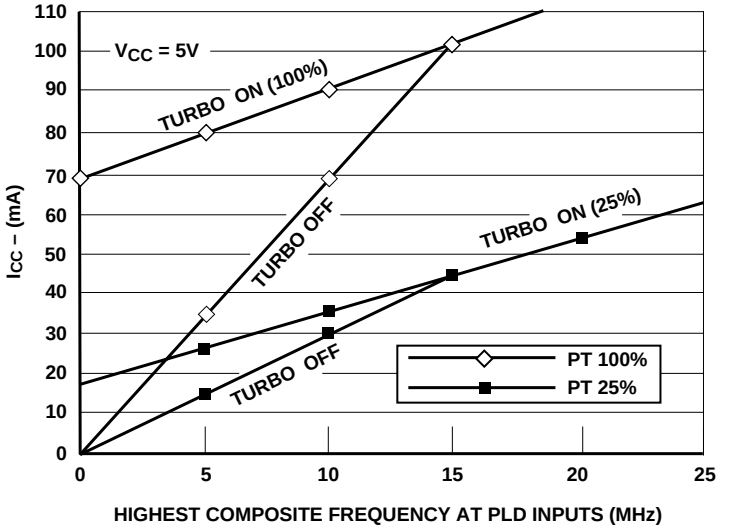


Table 82. PSD Module Example, Typ. Power Calculation at V_{CC} = 5.0V (Turbo Mode Off)

Conditions		
MCU Clock Frequency		= 12MHz
Highest Composite PLD input frequency		
	(Freq PLD)	= 8MHz
MCU ALE frequency (Freq ALE)		= 2MHz
	% Flash memory Access	= 80%
	% SRAM access	= 15%
	% I/O access	= 5% (no additional power above base)
Operational Modes		
	% Normal	= 40%
	% Power-down Mode	= 60%
Number of product terms used		
	(from fitter report)	= 45 PT
	% of total product terms	= 45/182 = 24.7%
Turbo Mode		= Off
Calculation (using typical values)		
I_{CC} total	= I_{CC}(MCUactive) x %MCUactive + I_{CC}(PSDactive) x %PSDactive + I_{PD}(pwrdown) x %pwrdown	
	I_{CC}(MCUactive)	= 20mA
	I_{PD}(pwrdown)	= 250µA
	I_{CC}(PSDactive)	= I_{CC}(ac) + I_{CC}(dc)
		= %flash x 2.5 mA/MHz x Freq ALE
		+ %SRAM x 1.5 mA/MHz x Freq ALE
		+ % PLD x (from graph using Freq PLD)
		= 0.8 x 2.5 mA/MHz x 2MHz + 0.15 x 1.5 mA/MHz x 2MHz + 24 mA
		= (4 + 0.45 + 24) mA
		= 28.45mA
I_{CC} total	= 20mA x 40% + 28.45mA x 40% + 250µA x 60%	
		= 8mA + 11.38mA + 150µA
		= 19.53mA
This is the operating power with no Flash memory Erase or Program cycles in progress. Calculation is based on all I/O pins being disconnected and I_{OUT} = 0 mA.		

MAXIMUM RATING

Stressing the device above the rating listed in the Absolute Maximum Ratings" table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 83. Absolute Maximum Ratings

Symbol	Parameter	Min.	Max.	Unit
T_{STG}	Storage Temperature	–65	125	°C
T_{LEAD}	Lead Temperature during Soldering (20 seconds max.)(1)		235	°C
V_{IO}	Input and Output Voltage (Q = V_{OH} or Hi-Z)	–0.5	6.5	V
V_{CC}	Supply Voltage	–0.5	6.5	V
V_{PP}	Device Programmer Supply Voltage	–0.5	14.0	V
V_{ESD}	Electrostatic Discharge Voltage (Human Body Model) 2	–2000	2000	V

Note: 1. IPC/JEDEC J-STD-020A
2. JEDEC Std JESD22-A114A (C1=100pF, R1=1500 Ω, R2=500 Ω)

DC AND AC PARAMETERS

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC Characteristic tables that follow are derived from tests performed under the Measurement Conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 84. Operating Conditions (5V Device)

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply Voltage	4.5	5.5	V
T_A	Ambient Operating Temperature (industrial)	–40	85	°C
	Ambient Operating Temperature (commercial)	0	70	°C

Table 85. AC Symbols for Timing

Signal Letters	
A	Address
C	Clock
D	Input Data
I	Instruction
L	ALE
N	$\overline{\text{RESET}}$ Input or Output
P	$\overline{\text{PSEN}}$ signal
Q	Output Data
R	RD signal
W	WR signal
B	V_{STBY} Output
M	Output Macrocell

Signal Behavior	
t	Time
L	Logic Level Low or ALE
H	Logic Level High
V	Valid
X	No Longer a Valid Logic Level
Z	Float
PW	Pulse Width

Example: t_{AVLX} – Time from Address Valid to ALE Invalid.

Figure 57. Switching Waveforms – Key

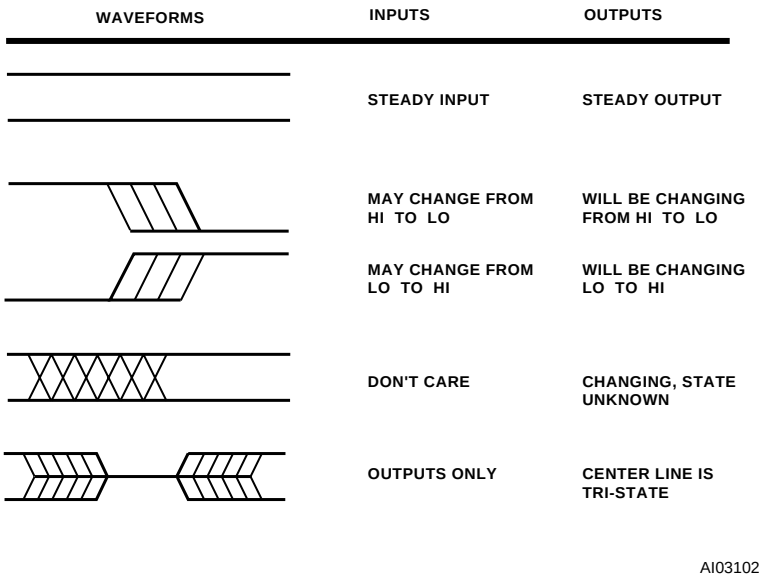


Table 86. DC Characteristics (5V Device)

Symbol	Parameter	Test Condition (in addition to those in Table 84, page 112)	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage (Ports 1, 3, 4[Bits 7,6,5,4,3,1,0], XTAL1, RESET)	4.5V < V_{CC} < 5.5V	0.7V_{CC}		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage (Ports B, C, D, 4[Bit 2])	4.5V < V_{CC} < 5.5V	2.0		V_{CC} + 0.5	V
V_{IL}	Input Low Voltage (Ports 1, 3, 4[Bits 7,6,5,4,3,1,0], XTAL1, RESET)	4.5V < V_{CC} < 5.5V	V_{SS} – 0.5		0.3V_{CC}	V
V_{IL1}	Input Low Voltage (Ports B, C, D)	4.5V < V_{CC} < 5.5V	–0.5		0.8	V
	Input Low Voltage (Port 4[Bit 2])	4.5V < V_{CC} < 5.5V	V_{SS} – 0.5		0.8	V
V_{OL}	Output Low Voltage (Ports B, C, D)	I_{OL} = 20µA V_{CC} = 4.5V		0.01	0.1	V
		I_{OL} = 8mA V_{CC} = 4.5V		0.25	0.45	V
V_{OL1}	Output Low Voltage (Ports 1,3,4)	I_{OL} = 1.6mA			0.45	V
V_{OH}	Output High Voltage (Ports B, C, D)	I_{OH} = –20µA V_{CC} = 4.5V	4.4	4.49		V
		I_{OH} = –2mA V_{CC} = 4.5V	2.4	3.9		V
V_{OH1}	Output High Voltage (Ports 1,3,4)	I_{OH} = –80µA	2.4			V
		I_{OH} = –10µA	4.05			V
V_{OH3}	Output High Voltage V_{STBYON}	I_{OH} = –1µA	V_{STBY} – 0.8			V
V_{LVR}	Low Voltage $\overline{RESET}$	0.1V hysteresis	3.75	4.0	4.25	V
V_{OP}	XTAL Open Bias Voltage (XTAL1, XTAL2)	I_{OL} = 3.2mA	2.0		3.0	V
V_{LKO}	V_{CC}(min) for Flash Erase and Program		2.5		4.2	V
V_{STBY}	SRAM (PSD) Standby Voltage		2.0		V_{CC} – 0.2	V
V_{DF}	SRAM (PSD) Data Retention Voltage	Only on V_{STBY}	2			V
I_{IL}	Logic '0' Input Current (Ports 1,3,4)	V_{IN} = 0.45V (0V for Port 4[pin 2])	–10		–50	µA
I_{TL}	Logic 1-to-0 Transition Current (Ports 1,3,4)	V_{IN} = 3.5V (2.5V for Port 4[pin 2])	–65		–650	µA
I_{STBY}	SRAM (PSD) Standby Current (V_{STBY} input)	V_{CC} = 0V		0.5	1	µA
I_{IDLE}	SRAM (PSD) Idle Current (V_{STBY} input)	V_{CC} > V_{STBY}	–0.1		0.1	µA

Symbol	Parameter		Test Condition (in addition to those in Table 84, page 112)	Min.	Typ.	Max.	Unit
I_{RST}	Reset Pin Pull-up Current (RESET)		$V_{IN} = V_{SS}$	–10		–55	µA
I_{FR}	XTAL Feedback Resistor Current (XTAL1)		XTAL1 = V_{CC} XTAL2 = V_{SS}	–20		–50	µA
I_{LI}	Input Leakage Current		$V_{SS} < V_{IN} < V_{CC}$	–1		1	µA
I_{LO}	Output Leakage Current		$0.45 < V_{OUT} < V_{CC}$	–10		10	µA
I_{PD}(1)	Power-down Mode		V_{CC} = 5.5V LVD logic disabled			250	µA
			LVD logic enabled			380	µA
I_{CC_CPU}(2,3,5)	Active (12MHz)		V_{CC} = 5V		20	30	mA
	Idle (12MHz)				8	10	mA
	Active (24MHz)		V_{CC} = 5V		30	38	mA
	Idle (24MHz)				15	20	mA
	Active (40MHz)		V_{CC} = 5V		40	62	mA
	Idle (40MHz)				20	30	mA
I_{CC_PSD} (DC)(5)	Operating Supply Current	PLD Only	PLD_TURBO = Off, f = 0MHz(4)		0		µA/PT(5)
			PLD_TURBO = On, f = 0MHz		400	700	µA/PT
		Flash memory	During Flash memory WRITE/Erase Only		15	30	mA
			Read only, f = 0MHz		0	0	mA
		SRAM	f = 0MHz		0	0	mA
I_{CC_PSD} (AC)(5)	PLD AC Base			Note 4			
	Flash memory AC Adder				2.5	3.5	mA/ MHz
	SRAM AC Adder				1.5	3.0	mA/ MHz

Note: 1. I_{PD} (Power-down Mode) is measured with:
XTAL1=V_{SS}; XTAL2=not connected; $\overline{RESET}$=V_{CC}; all other pins are disconnected. PLD not in Turbo Mode.
2. I_{CC_CPU} (active mode) is measured with:
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, V_{IL} = V_{SS}+0.5V, V_{IH} = Vcc – 0.5V, XTAL2 = not connected; $\overline{RESET}$=V_{SS}; all other pins are disconnected. I_{CC} would be slightly higher if a crystal oscillator is used (approximately 1mA).
3. I_{CC_CPU} (Idle Mode) is measured with:
XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, V_{IL} = V_{SS}+0.5V, V_{IH} = V_{CC}– 0.5V, XTAL2 = not connected; $\overline{RESET}$=V_{CC}; all other pins are disconnected.
4. See Figure 56 for the PLD current calculation.
5. I/O current = 0 mA, all I/O pins are disconnected.

Table 87. External Clock Drive (with the 5V MCU Module)

Symbol	Parameter[1]	40MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 24 to 40MHz		Unit
		Min	Max	Min	Max	
t_{RLRH}	Oscillator period			25	41.7	ns
t_{WLWH}	High time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{LLAX2}	Low time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{RHDX}	Rise time				10	ns
t_{RHDX}	Fall time				10	ns

Note: 1. Conditions (in addition to those in Table 84, V_{CC} = 4.5 to 5.5V): V_{SS} = 0V; C_L for ALE and PSEN output is 100pF; C_L for other outputs is 80pF

Table 88. A/D Analog Specification

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
AV_{REF}	Analog Power Supply Input Voltage Range		V_{SS}		V_{CC}	V
V_{AN}	Analog Input Voltage Range		$V_{SS} - 0.3$		$AV_{REF} + 0.3$	V
I_{AVDD}	Current Following between V_{CC} and V_{SS}				200	µA
CA_{IN}	Overall Accuracy				±2	l.s.b.
N_{NLE}	Non-Linearity Error				±2	l.s.b.
N_{DNLE}	Differential Non-Linearity Error				±2	l.s.b.
N_{ZOE}	Zero-Offset Error				±2	l.s.b.
N_{FSE}	Full Scale Error				±2	l.s.b.
N_{GE}	Gain Error				±2	l.s.b.
T_{CONV}	Conversion Time	at 8MHz clock			20	µs

Figure 58. Input to Output Disable / Enable

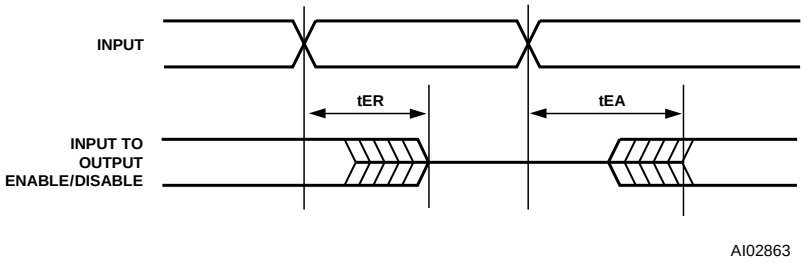


Table 89. CPLD Combinatorial Timing (5V Device)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate(1)	Unit
t_{PD}(2)	CPLD Input Pin/Feedback to CPLD Combinatorial Output			20	+ 2	+ 10	– 2	ns
t_{EA}	CPLD Input to CPLD Output Enable			21		+ 10	– 2	ns
t_{ER}	CPLD Input to CPLD Output Disable			21		+ 10	– 2	ns
t_{ARP}	CPLD Register Clear or Preset Delay			21		+ 10	– 2	ns
t_{ARPW}	CPLD Register Clear or Preset Pulse Width		10			+ 10		ns
t_{ARD}	CPLD Array Delay	Any macrocell		11	+ 2			ns

Note: 1. Fast Slew Rate output available on PB3-PB0 and PD1. Decrement times by given amount.
2. t_{PD} for MCU address and control signals refers to delay from pins on MCU Bus to CPLD combinatorial output.

Figure 59. Synchronous Clock Mode Timing – PLD

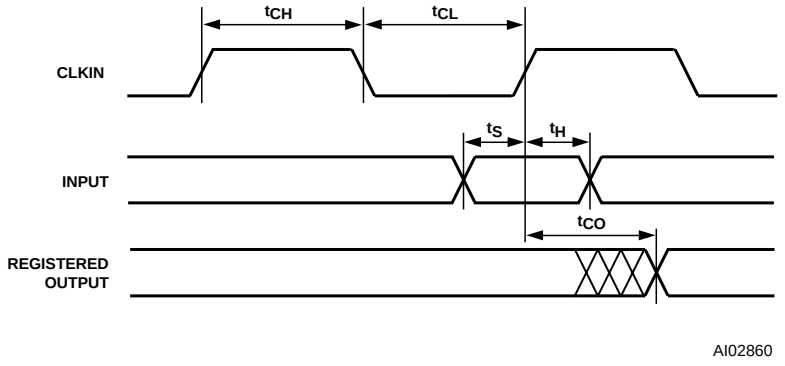


Table 90. CPLD Macrocell Synchronous Clock Mode Timing (5V Device)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate(1)	Unit
f_{MAX}	Maximum Frequency External Feedback	$1/(t_S+t_{CO})$		40.0				MHz
	Maximum Frequency Internal Feedback (f_{CNT})	$1/(t_S+t_{CO}–10)$		66.6				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CH}+t_{CL})$		83.3				MHz
t_S	Input Setup Time		12		+ 2	+ 10		ns
t_H	Input Hold Time		0					ns
t_{CH}	Clock High Time	Clock Input	6					ns
t_{CL}	Clock Low Time	Clock Input	6					ns
t_{CO}	Clock to Output Delay	Clock Input		13			– 2	ns
t_{ARD}	CPLD Array Delay	Any macrocell		11	+ 2			ns
t_{MIN}	Minimum Clock Period(2)	$t_{CH}+t_{CL}$	12					ns

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

Figure 60. Asynchronous $\overline{\text{RESET}}$ / Preset

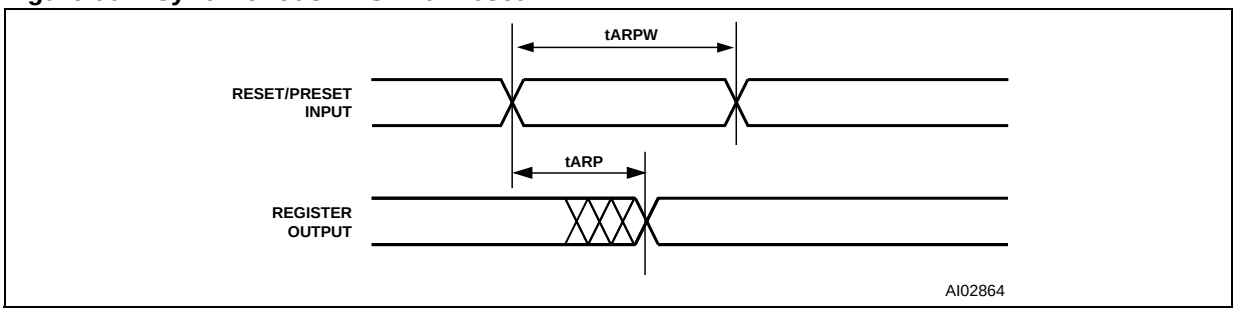


Figure 61. Asynchronous Clock Mode Timing (product term clock)

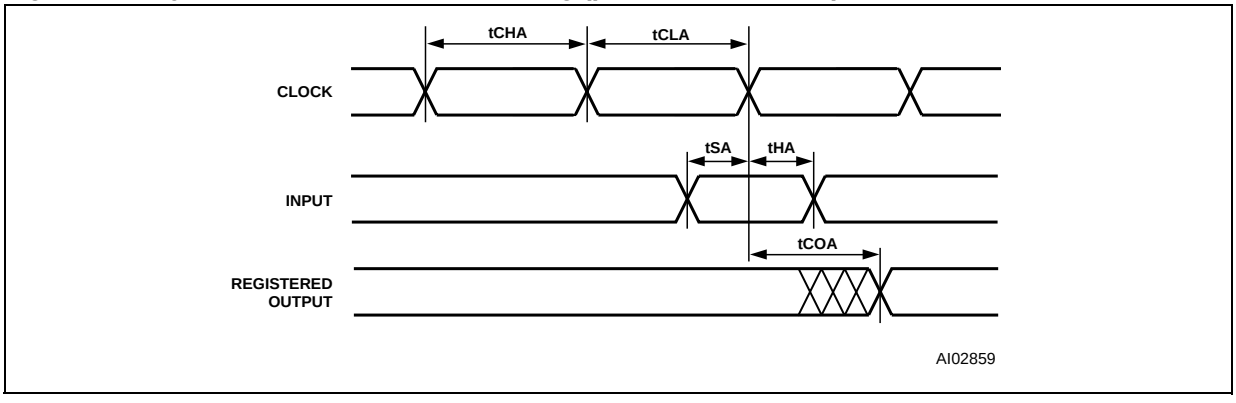


Table 91. CPLD Macrocell Asynchronous Clock Mode Timing (5V Device)

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

Figure 62. Input Macrocell Timing (product term clock)

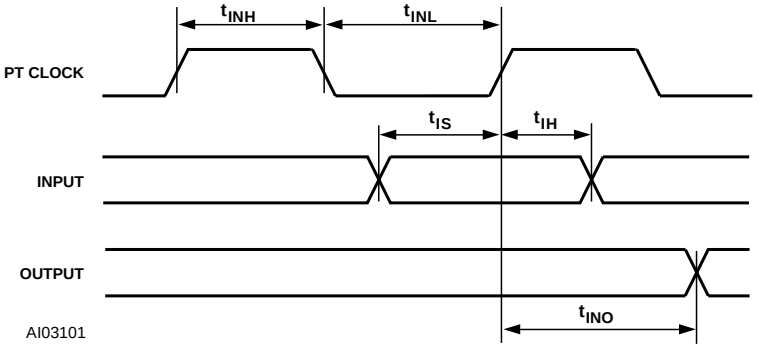


Table 92. Input Macrocell Timing (5V Device)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Unit
t_{IS}	Input Setup Time	(Note 1)	0				ns
t_{IH}	Input Hold Time	(Note 1)	15			+ 10	ns
t_{INH}	NIB Input High Time	(Note 1)	9				ns
t_{INL}	NIB Input Low Time	(Note 1)	9				ns
t_{INO}	NIB Input to Combinatorial Delay	(Note 1)		34	+ 2	+ 10	ns

Note: 1. Inputs from Port B and C relative to register/ latch clock from the PLD. ALE/AS latch timings refer to t_{AVLX} and t_{LXAX}.

Table 93. Program, WRITE and Erase Times (5V Device)

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Flash Program		8.5		s
	Flash Bulk Erase(1) (pre-programmed)		3	30	s
	Flash Bulk Erase (not pre-programmed)		5		s
t_{WHQV3}	Sector Erase (pre-programmed)		1	30	s
t_{WHQV2}	Sector Erase (not pre-programmed)		2.2		s
t_{WHQV1}	Byte Program		14	150	µs
	Program / Erase Cycles (per Sector)	100,000			cycles
t_{WHWLO}	Sector Erase Time-out		100		µs
t_{Q7VQV}	DQ7 Valid to Output (DQ7-DQ0) Valid (Data Polling)(2)			30	ns

Note: 1. Programmed to all zero before erase.
2. The polling status, DQ7, is valid t_{Q7VQV} time units before the data byte, DQ0-DQ7, is valid for reading.

Figure 63. Reset ($\overline{\text{RESET}}$) Timing

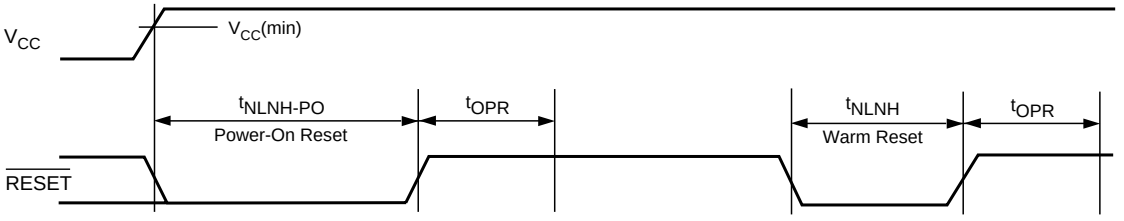


Table 94. Reset ($\overline{\text{RESET}}$) Timing (5V Device)

Symbol	Parameter	Conditions	Min	Max	Unit
t_{NLNH}	$\overline{\text{RESET}}$ Active Low Time(1)		150		ns
$t_{NLNH-PO}$	Power-on Reset Active Low Time		1		ms
t_{OPR}	$\overline{\text{RESET}}$ High to Operational Device			120	ns

Note: 1. Reset ($\overline{\text{RESET}}$) does not reset Flash memory Program or Erase cycles.

Table 95. V_{STBYON} Definitions Timing (5V Device)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{BVBH}	V_{STBY} Detection to V_{STBYON} Output High	(Note 1)		20		µs
t_{BXBL}	V_{STBY} Off Detection to V_{STBYON} Output Low	(Note 1)		20		µs

Note: 1. V_{STBYON} timing is measured at V_{CC} ramp rate of 2ms.

Figure 64. ISC Timing

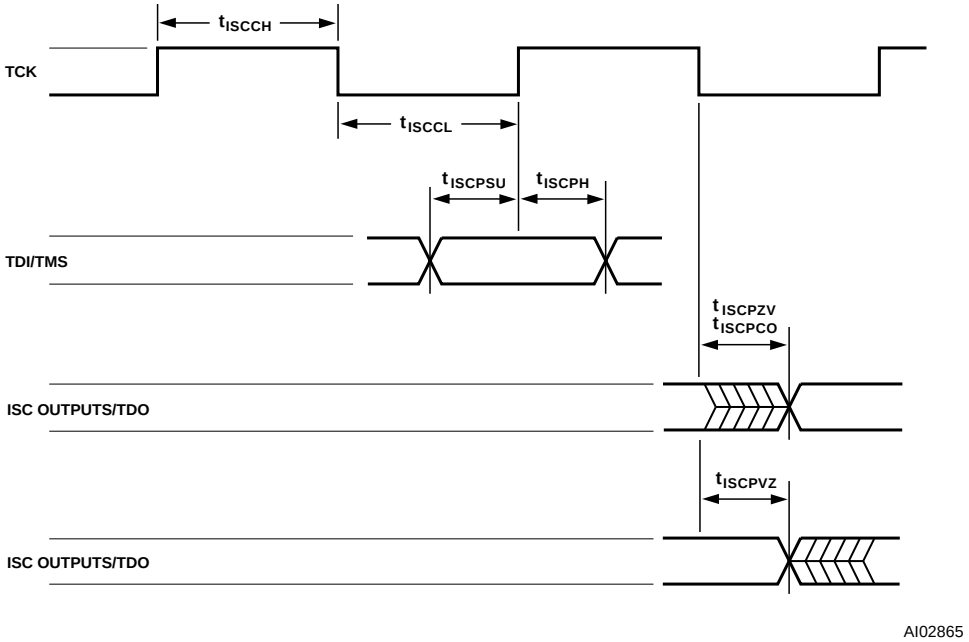
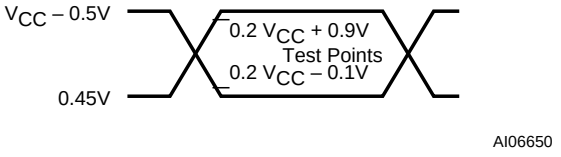


Table 96. ISC Timing (5V Device)

Symbol	Parameter	Conditions	Min	Max	Unit
t_{ISCCF}	Clock (TCK, PC1) Frequency (except for PLD)	(Note 1)		20	MHz
t_{ISCCH}	Clock (TCK, PC1) High Time (except for PLD)	(Note 1)	23		ns
t_{ISCCL}	Clock (TCK, PC1) Low Time (except for PLD)	(Note 1)	23		ns
t_{ISCCFP}	Clock (TCK, PC1) Frequency (PLD only)	(Note 2)		2	MHz
t_{ISCCHP}	Clock (TCK, PC1) High Time (PLD only)	(Note 2)	240		ns
t_{ISCCLP}	Clock (TCK, PC1) Low Time (PLD only)	(Note 2)	240		ns
t_{ISCPSU}	ISC Port Set Up Time		7		ns
t_{ISCPH}	ISC Port Hold Up Time		5		ns
t_{ISCPCO}	ISC Port Clock to Output			21	ns
t_{ISCPZV}	ISC Port High-Impedance to Valid Output			21	ns
t_{ISCPVZ}	ISC Port Valid Output to High-Impedance			21	ns

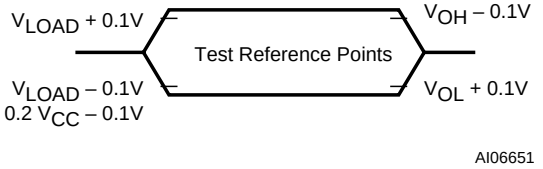
Note: 1. For non-PLD Programming, Erase or in ISC By-pass Mode.
2. For Program or Erase PLD only.

Figure 65. MCU Module AC Measurement I/O Waveform



Note: AC inputs during testing are driven at V_{CC}–0.5V for a logic '1,' and 0.45V for a logic '0.'
Timing measurements are made at V_{IH}(min) for a logic '1,' and V_{IL}(max) for a logic '0'

Figure 66. PSD Module AC Float I/O Waveform



Note: For timing purposes, a Port pin is considered to be no longer floating when a 100mV change from load voltage occurs, and begins to float when a 100mV change from the loaded V_{OH} or V_{OL} level occurs
I_{OL} and I_{OH} ≥ 20mA

Figure 67. External Clock Cycle

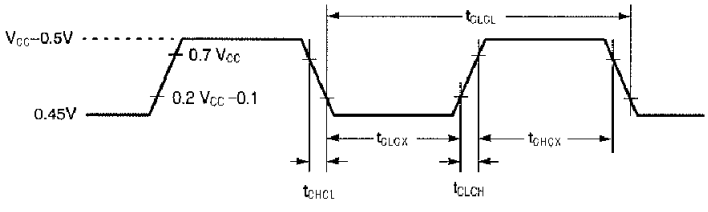
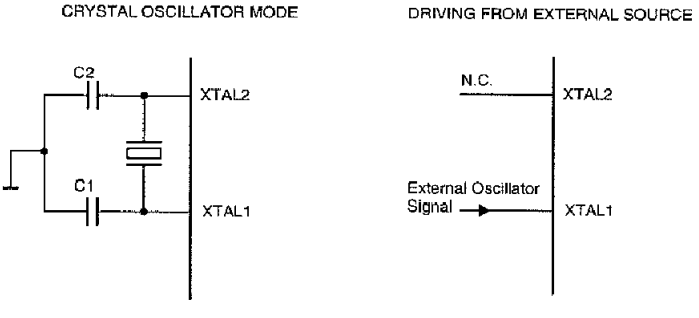


Figure 68. Recommended Oscillator Circuits



Note: C1, C2 = 30pF ± 10pF for crystals
For ceramic resonators, contact resonator manufacturer
Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Since each crystal and ceramic resonator have their own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

Figure 69. PSD Module AC Measurement I/O Waveform

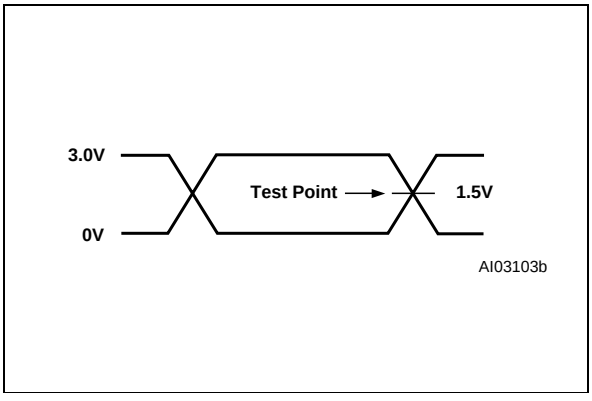


Figure 70. PSD Module AC Measurement Load Circuit

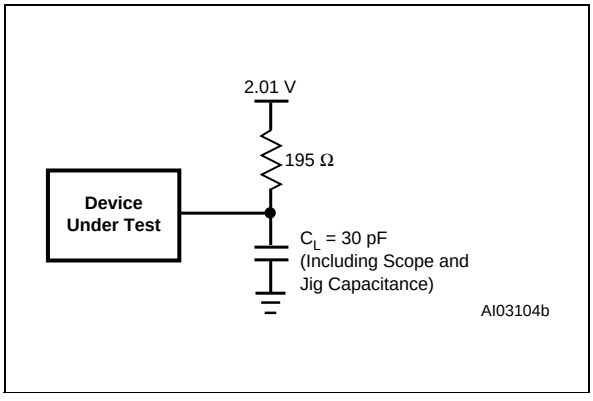


Table 97. Capacitance

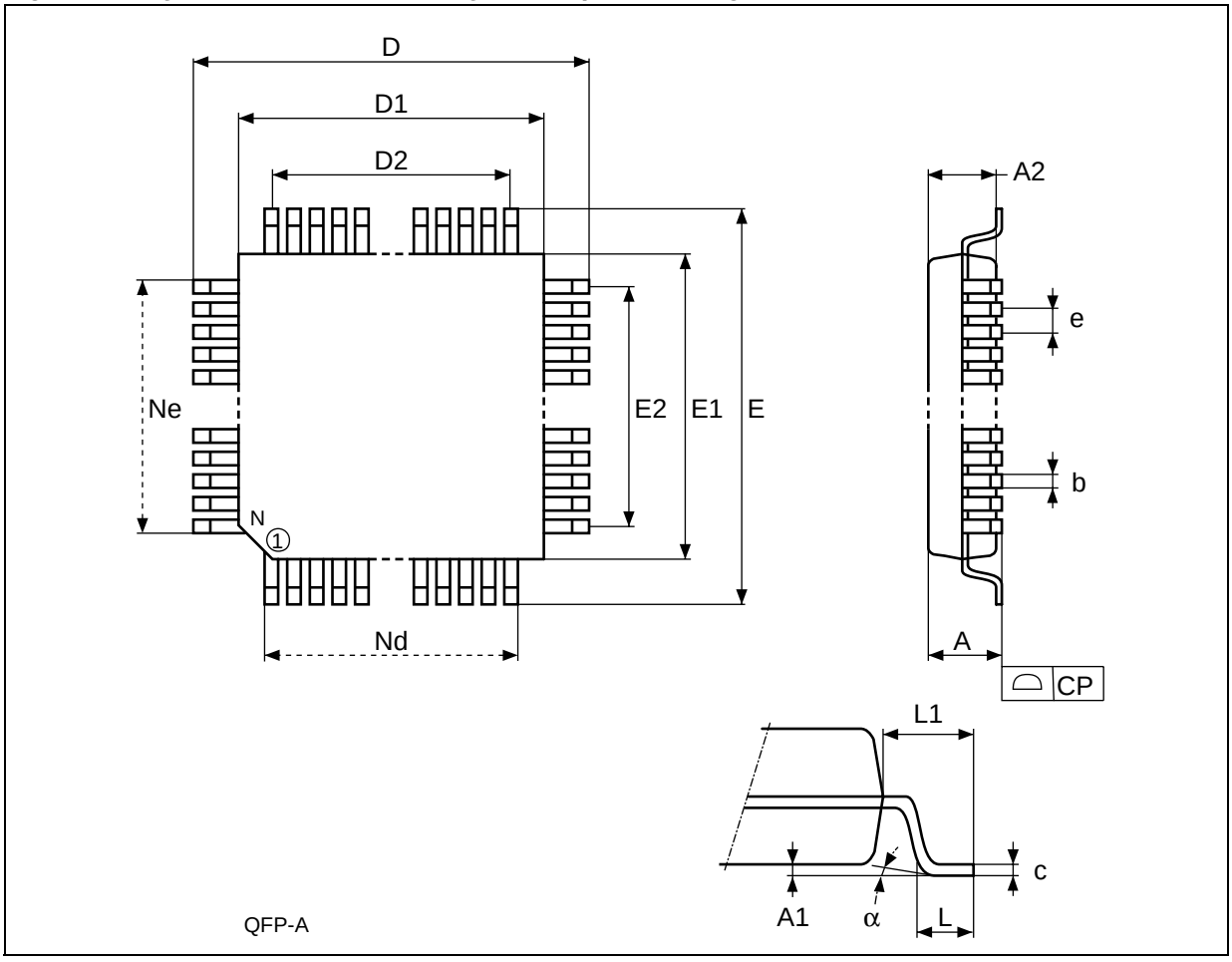
Symbol	Parameter	Test Condition	Typ.(1)	Max.	Unit
C_{IN}	Input Capacitance (for input pins)	V_{IN} = 0V	4	6	pF
C_{OUT}	Output Capacitance (for input/ output pins)	V_{OUT} = 0V	8	12	pF

Note: Sampled only, not 100% tested.

1. Typical values are for T_A = 25°C and nominal supply voltages.

PACKAGE MECHANICAL INFORMATION

Figure 71. TQFP52 – 52-lead Plastic Quad Flatpack Package Outline



Note: Drawing is not to scale.

Table 98. TQFP52 – 52-lead Plastic Quad Flatpack Package Mechanical Data

Symb	mm			inches		
	Typ	Min	Max	Typ	Min	Max
A	–	–	1.75	–	–	0.069
A1	–	0.05	0.020	–	0.002	0.008
A2	–	1.25	1.55	–	0.049	0.061
b	–	0.02	0.04	–	0.007	0.016
c	–	0.07	0.23	–	0.002	0.009
D	12.00	–	–	0.473	–	–
D1	10.00	–	–	0.394	–	–
D2						
E	12.00	–	–	0.473	–	–
E1	10.00	–	–	0.394	–	–
E2						
e	0.65	–	–	0.026	–	–
L	–	0.45	0.75	–	0.018	0.030
L1	1.00	–	–	0.039	–	–
α	–	0°	7°	–	0°	7°
n	52			52		
Nd	13			13		
Ne	13			13		
CP	–	–	0.10	–	–	0.004

REVISION HISTORY

Table 99. Document Revision History

Date	Rev. #	Revision Details
June 2003	1.0	First Issue