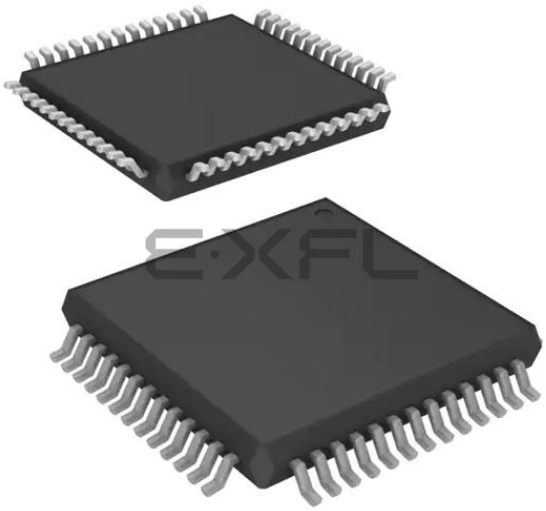


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8032 |
| Core Size | 8-Bit |
| Speed | 40MHz |
| Connectivity | I²C, UART/USART |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 37 |
| Program Memory Size | 160KB (160K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 32K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 4x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 52-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3253b-40t6 |

**Table 2. 80-pin package pin description (continued)**

| Port pin | Signal name | Pin no. | In/ out | Function | |
|---|---|---|---|---|---|
| | | | | Basic | Alternate |
| P4.7 | PWM4 | 18 | I/O | General I/O port pin | Programmable 8-bit Pulse Width modulation output 4 |
| | USB– | 8 | I/O | Pull-up resistor required (2 kΩ for 3 V devices, 7.5 kΩ for 5 V devices) | |
| | $V_{REF}$ | 70 | O | Reference Voltage input for ADC | |
| | RD_ | 65 | O | READ signal, external bus | |
| | WR_ | 62 | O | WRITE signal, external bus | |
| | PSEN_ | 63 | O | $\overline{PSEN}$ signal, external bus | |
| | ALE | 4 | O | Address Latch signal, external bus | |
| | RESET_ | 68 | I | Active low $\overline{RESET}$ input | |
| | XTAL1 | 48 | I | Oscillator input pin for system clock | |
| | XTAL2 | 49 | O | Oscillator output pin for system clock | |
| PA0 | | 35 | I/O | General I/O port pin | PLD macrocell outputs<br>PLD inputs<br>Latched address out (A0-A7)<br>Peripheral I/O mode |
| PA1 | | 34 | I/O | General I/O port pin | |
| PA2 | | 32 | I/O | General I/O port pin | |
| PA3 | | 28 | I/O | General I/O port pin | |
| PA4 | | 26 | I/O | General I/O port pin | |
| PA5 | | 24 | I/O | General I/O port pin | |
| PA6 | | 22 | I/O | General I/O port pin | |
| PA7 | | 21 | I/O | General I/O port pin | |
| PB0 | | 80 | I/O | General I/O port pin | PLD macrocell outputs<br>PLD inputs<br>Latched address out (A0-A7) |
| PB1 | | 78 | I/O | General I/O port pin | |
| PB2 | | 76 | I/O | General I/O port pin | |
| PB3 | | 74 | I/O | General I/O port pin | |
| PB4 | | 73 | I/O | General I/O port pin | |
| PB5 | | 72 | I/O | General I/O port pin | |
| PB6 | | 67 | I/O | General I/O port pin | |
| PB7 | | 66 | I/O | General I/O port pin | |

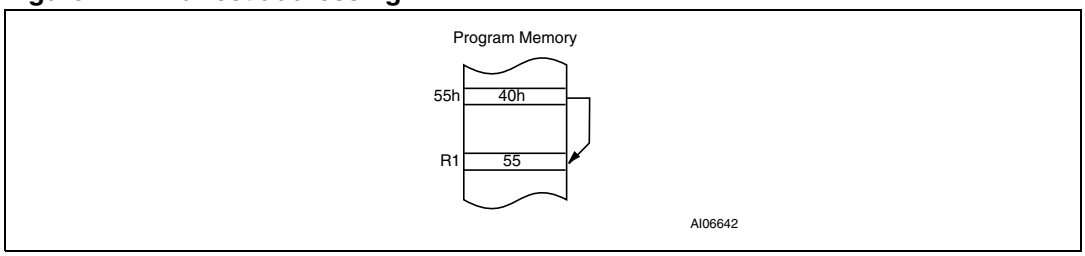
### 2.9.2 Indirect addressing

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 11. Indirect addressing**




### 2.9.3 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
```

### 2.9.4 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.

### 2.9.5 Immediate constants addressing

The value of a constant can follow the opcode in Program memory.

Example:

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

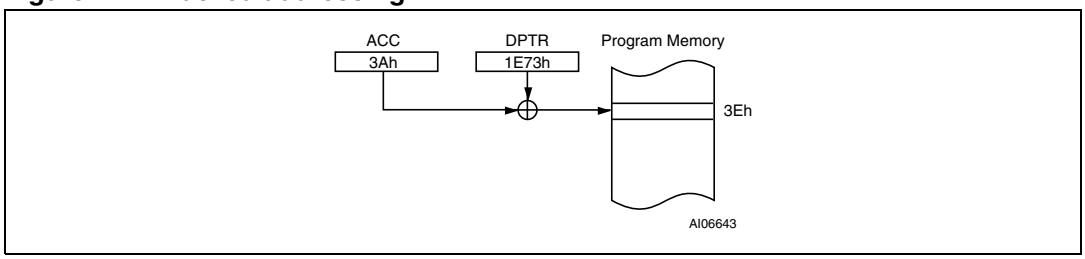
### 2.9.6 Indexed addressing

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.

Example:

movc A, @A+DPTR

**Figure 12. Indexed addressing**




## 2.10 Arithmetic instructions

The arithmetic instructions is listed in *Table 4*. 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 operation 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 |

*Note: In UPSD325xx devices, 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 | | |

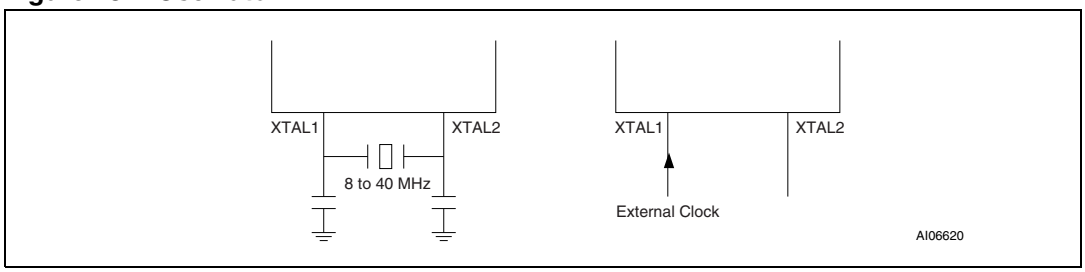
First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (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.

# 8 Oscillator

The oscillator circuit of the UPSD325xx devices is a single stage inverting amplifier in a Pierce oscillator configuration. 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 UPSD325xx devices externally, XTAL1 is driven from an external source and XTAL2 left open-circuit.

**Figure 18. Oscillator**



# 10 Watchdog timer

The hardware watchdog timer (WDT) resets the UPSD325xx devices 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 $\overline{\text{RESET}}$ (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. The reset period is $tf_{OSC}$ x 12 x $2^{22}$.

The $\overline{\text{RESET}}$ pulse width is $tf_{OSC}$ x 12 x $2^{15}$.

**Table 32. Watchdog timer key register (WDKEY: 0AEh)**

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

**Table 33. Description of the WDKEY Bits**

| Bit | Symbol | Function |
|---|---|---|
| 7 to 0 | WDKEY7 to WDKEY0 | Enable or disable watchdog timer.<br>01010101 (=55h): disable watchdog timer. Others: enable watchdog timer |

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

Modes 1 and 3 Baud Rate = Timer 2 Overflow Rate / 16

**Table 45. 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 |

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:

Modes 1 and 3 Baud 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.

### 12.2.4 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 $f_{OSC}$.

# 22 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.

## 22.1 Primary Flash memory and secondary Flash memory description

The primary Flash memory is divided evenly into eight equal sectors. The secondary Flash memory is divided into four equal sectors. 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/$\overline{\text{Busy}}$ (PC3). This pin is set up using PSDsoft Express Configuration.

## 22.2 Memory block select signals

The DPLD generates the Select signals for all the internal memory blocks (see *Section 23: PLDs*). Each of the eight sectors of the primary Flash memory has a Select signal (FS0-FS7) which can contain up to three product terms. Each of the four sectors of the secondary Flash memory has a Select signal (CSBOOT0-CSBOOT3) 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.

### 22.2.1 Ready/$\overline{\text{Busy}}$ (PC3)

This signal can be used to output the Ready/$\overline{\text{Busy}}$ status of the Flash memory. The output on Ready/$\overline{\text{Busy}}$ (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.

### 22.2.2 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 85*.

**Table 87. Sector protection/security bit definition – Flash protection register**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Sec7_Prot | Sec6_Prot | Sec5_Prot | Sec4_Prot | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

1. 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 88. 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 | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

1. 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.

### 22.8.2 Reset Flash

The Reset Flash instruction consists of one WRITE cycle (see *Table 85*). 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 25µs 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 25µs.

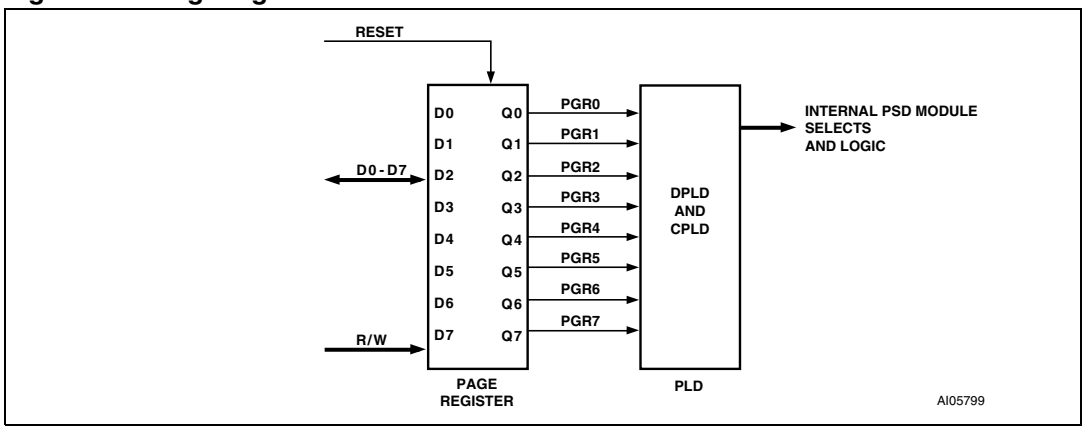
### 22.8.3 Reset ($\overline{\text{RESET}}$) signal

A pulse on Reset ($\overline{\text{RESET}}$) aborts any cycle that is in progress, and resets the Flash memory to the READ mode. When the reset occurs during a Program or Erase cycle, the Flash memory takes up to 25µs to return to the READ mode. It is recommended that the Reset ($\overline{\text{RESET}}$) pulse (except for Power-on $\overline{\text{RESET}}$, as described in *Section 26: RESET timing and device status at reset*) be at least 25µs so that the Flash memory is always ready for the MCU to retreive the bootstrap instructions after the reset cycle is complete.

## 22.9 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.

**Figure 56. Page register**

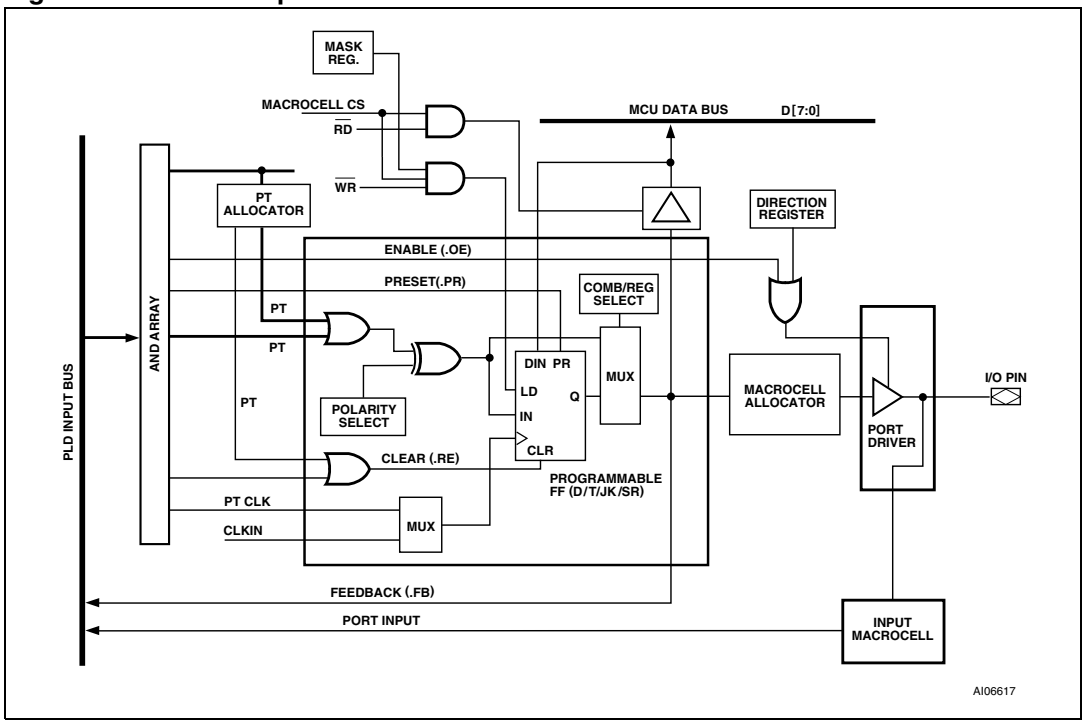


### 23.5.1 Loading and Reading the Output Macrocells (OMC)

The Output Macrocells (OMC) block occupies a memory location in the MCU address space, as defined by the CSIOP block (see *Section 24: I/O ports (PSD module)*). 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 ($\overline{WR}$, edge loading) or during the time that WRITE Strobe ($\overline{WR}$) is active (level loading). The method of loading is specified in PSDsoft Express Configuration.

**Figure 59. CPLD output macrocell**




### 23.5.2 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.

### 23.5.3 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

# 24 I/O ports (PSD module)

There are four programmable I/O ports: Ports A, B, C, and D in the PSD module. Each of the ports is eight bits except Port D, which is 3 bits. 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. Port A is not available in the 52-pin package.

The topics discussed in this section are:

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

## 24.1 General port architecture

The general architecture of the I/O Port block is shown in *Figure 61*. Individual Port architectures are shown in *Figure 63* to *Figure 66*. 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 61*, the ports contain an output multiplexer whose select signals are driven by the configuration bits in the Control Registers (Ports A and 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-ECS2) 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).

# 27 Programming in-circuit using the JTAG serial interface

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

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

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

## 27.1 Standard JTAG Signals

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

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

UPSD325xx devices support JTAG In-System-Configuration (ISC) commands, but not Boundary Scan. The PSDsoft Express software tool and FlashLINK JTAG programming cable implement the JTAG In-System-Configuration (ISC) commands.

**Table 107. JTAG port signals**

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

## 27.2 JTAG extensions

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

scan the status out serially using the standard JTAG channel. See Application Note *AN1153*.

$\overline{\text{TERR}}$ 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 ($\overline{\text{RESET}}$) pulse is received after an “ISC_DISABLE” command.

$\overline{\text{TSTAT}}$ behaves the same as Ready/$\overline{\text{Busy}}$ described in *Section 22.2.1: Ready/Busy (PC3)*. $\overline{\text{TSTAT}}$ is High when the PSD module device is in READ mode (primary and secondary Flash memory contents can be read). $\overline{\text{TSTAT}}$ is Low when Flash memory Program or Erase cycles are in progress, and also when data is being written to the secondary Flash memory.

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ can be configured as open-drain type signals during an “ISC_ENABLE” command.

## 27.3 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.

# 29 AC/DC parameters

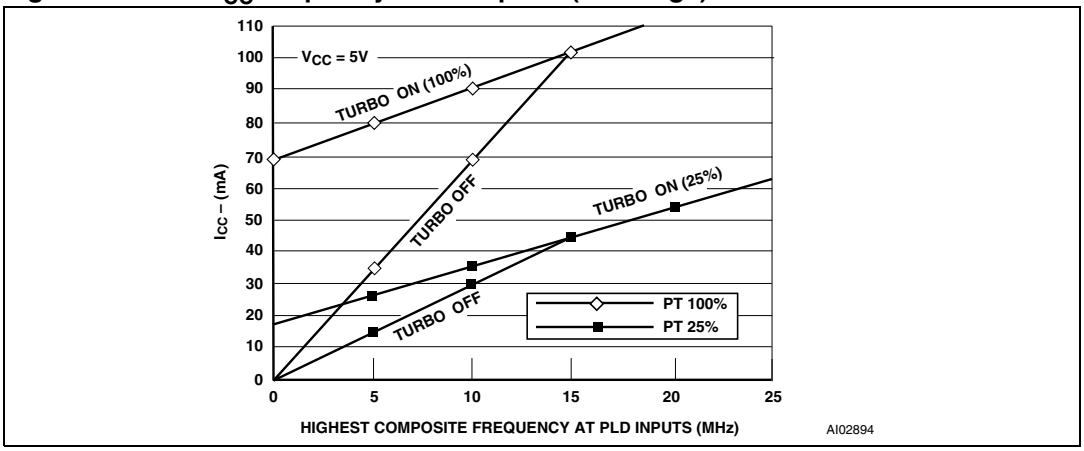
These tables describe the AD and DC parameters of the UPSD325xx devices:

- 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 70* and *Figure 71* show 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 70. PLD $I_{CC}$/frequency consumption (5 V range)**



# 32 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 113. Operating conditions (5 V devices)**

| 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 114. Operating conditions (3 V devices)**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | 3.0 | 3.6 | V |
| $T_A$ | Ambient operating temperature (industrial) | –40 | 85 | °C |
| | Ambient operating temperature (commercial) | 0 | 70 | °C |

**Table 115. AC signal letters for timing**

| | |
|---|---|
| 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 |
| M | Output Macrocell |

1. Example: $t_{AVLX}$ = Time from Address Valid to ALE Invalid.

**Table 125. External data memory AC characteristics (with the 3 V MCU module)**

| Symbol | Parameter[1] | 24 MHz oscillator | | Variable oscillator $1/t_{CLCL}$ = 8 to 24 MHz | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| $t_{RLRH}$ | $\overline{RD}$ pulse width | 180 | | 6 $t_{CLCL}$ – 70 | | ns |
| $t_{WLWH}$ | $\overline{WR}$ pulse width | 180 | | 6 $t_{CLCL}$ – 70 | | ns |
| $t_{LLAX2}$ | Address hold after ALE | 56 | | 2 $t_{CLCL}$ – 27 | | ns |
| $t_{RHDX}$ | $\overline{RD}$ to valid data in | | 118 | | 5 $t_{CLCL}$ – 90 | ns |
| $t_{RHDX}$ | Data hold after $\overline{RD}$ | 0 | | 0 | | ns |
| $t_{RHDZ}$ | Data float after $\overline{RD}$ | | 63 | | 2 $t_{CLCL}$ – 20 | ns |
| $t_{LLDV}$ | ALE to valid data in | | 200 | | 8 $t_{CLCL}$ – 133 | ns |
| $t_{AVDV}$ | Address to valid data in | | 220 | | 9 $t_{CLCL}$ – 155 | ns |
| $t_{LLWL}$ | ALE to $\overline{WR}$ or $\overline{RD}$ | 75 | 175 | 3 $t_{CLCL}$ – 50 | $t_{CLCL}$ + 50 | ns |
| $t_{AVWL}$ | Address valid to $\overline{WR}$ or $\overline{RD}$ | 67 | | 4 $t_{CLCL}$ – 97 | | ns |
| $t_{WHLH}$ | $\overline{WR}$ or $\overline{RD}$ High to ALE High | 17 | 67 | $t_{CLCL}$ – 25 | $t_{CLCL}$ + 25 | ns |
| $t_{QVWX}$ | Data valid to $\overline{WR}$ transition | 5 | | $t_{CLCL}$ – 37 | | ns |
| $t_{QVWH}$ | Data set-up before $\overline{WR}$ | 170 | | 7 $t_{CLCL}$ – 122 | | ns |
| $t_{WHQX}$ | Data hold after $\overline{WR}$ | 15 | | $t_{CLCL}$ – 27 | | ns |
| $t_{RLAZ}$ | Address float after $\overline{RD}$ | | 0 | | 0 | ns |

1. Conditions (in addition to those in *Table 114*, $V_{CC}$ = 3.0 to 3.6 V): $V_{SS}$ = 0 V; $C_L$ for Port 0, ALE and PSEN output is 100 pF, for 5 V devices, and 50 pF for 3 V devices; $C_L$ for other outputs is 80 pF, for 5 V devices, and 50 pF for 3 V devices)

**Table 126. 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 8 MHz clock | | | 20 | µs |

**Figure 77. Synchronous clock mode timing – PLD**

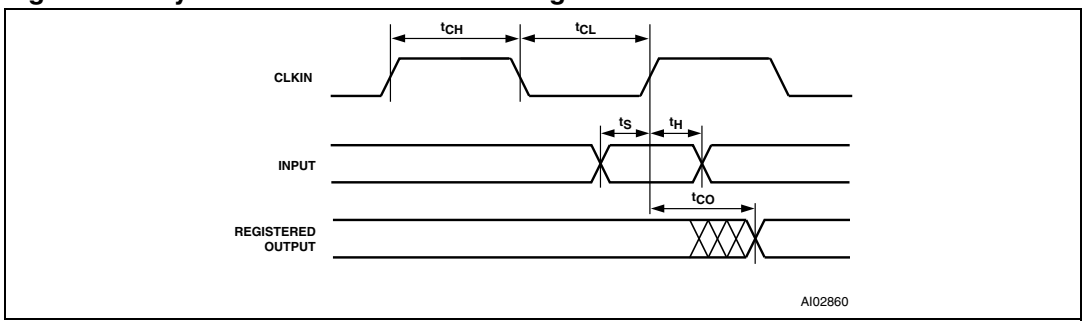



**Table 129. CPLD macrocell synchronous clock mode timing (5 V devices)**

| 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 |

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

**Table 141. Reset ($\overline{\text{RESET}}$) timing (5 V devices)**

| 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 |
| $t_{NLNH-A}$ | Warm $\overline{\text{RESET}}$ (2) | | 25 | | µs |
| $t_{OPR}$ | $\overline{\text{RESET}}$ high to operational device | | | 120 | ns |

1. Reset ($\overline{\text{RESET}}$) does not reset Flash memory Program or Erase cycles.
2. Warm $\overline{\text{RESET}}$ aborts Flash memory Program or Erase cycles, and puts the device in READ mode.

**Table 142. Reset ($\overline{\text{RESET}}$) timing (3 V devices)**

| Symbol | Parameter | Conditions | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $t_{NLNH}$ | $\overline{\text{RESET}}$ active low time(1) | | 300 | | ns |
| $t_{NLNH-PO}$ | Power-on reset active low time | | 1 | | ms |
| $t_{NLNH-A}$ | Warm $\overline{\text{RESET}}$ (2) | | 25 | | µs |
| $t_{OPR}$ | $\overline{\text{RESET}}$ high to operational device | | | 300 | ns |

1. Reset ($\overline{\text{RESET}}$) does not reset Flash memory Program or Erase cycles.
2. Warm $\overline{\text{RESET}}$ aborts Flash memory Program or Erase cycles, and puts the device in READ mode.

**Figure 84. ISC timing**

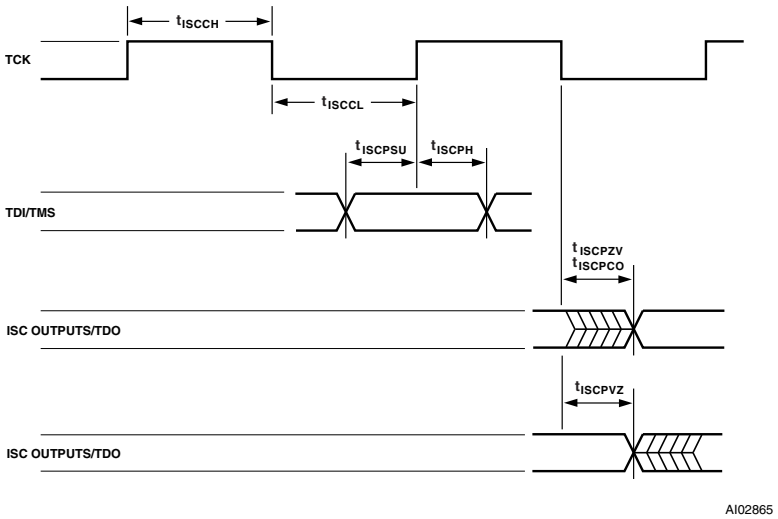
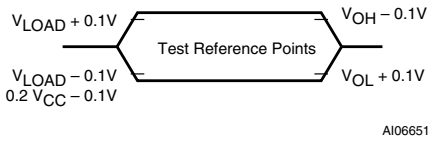


**Figure 86. PSD module AC float I/O waveform**




1. 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
2. $I_{OL}$ and $I_{OH} \geq 20mA$

**Figure 87. External clock cycle**

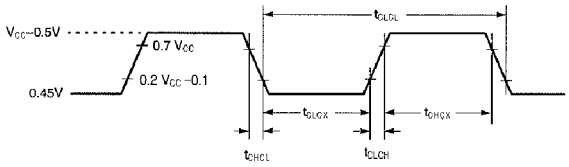
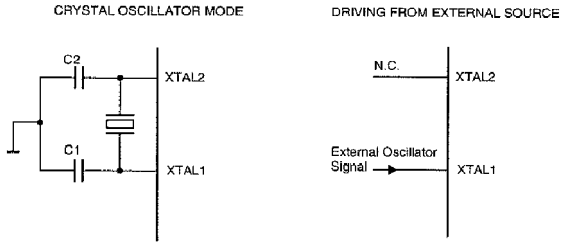



**Figure 88. Recommended oscillator circuits**




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

**Figure 89. PSD module AC measurement I/O waveform**