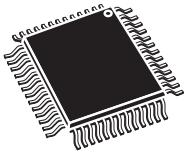
DSM2190F4V

DSM (Digital Signal Processor System Memory) For Analog Devices ADSP-2191 DSPs (3.3V Supply)

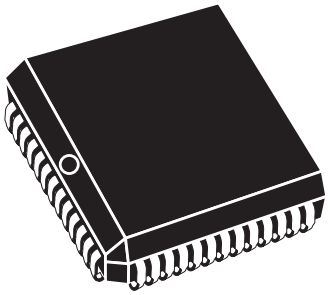
FEATURES SUMMARY

- **Glueless Connection to DSP**
  - Easily add memory, logic, and I/O to the External Port of ADSP-2191 DSP
- **Dual Flash Memories**
  - Two independent Flash memory arrays for storing DSP code and data. DSP may access the two arrays concurrently (read from one while erasing or writing the other)
  - 256K x 8 Main Flash memory divided into 8 sectors (32KByte each)
    - Ample storage for booting DSP code/data upon reset and subsequent code swaps
    - Large capacity for data recording
  - 32K x 8 Secondary Flash memory divided into 4 sectors (8 KByte each). Multiple uses:
    - Small sector size ideal for small data sets, and calibration or configuration constants
    - Store custom start-up code in one or more sectors and configure DSP to run from external memory upon reset (no boot)
    - Concatenate Secondary Flash with Main Flash for total of 288 KBytes
  - Each Flash sector can be write protected.
  - Built-in programmable address decoding logic allows mapping individual Flash sectors to any address boundary
- **Up to 16 Multifunction I/O Pins**
  - Increase total DSP system I/O capability
  - I/O controlled by DSP software or PLD logic
- **General purpose PLD**
  - Over 3,000 Gates of PLD with 16 macro cells
  - Use for peripheral glue logic to keypads, control panel, displays, LCDs, and other devices
  - Eliminate PLDs and external logic devices
  - Create state machines, chip selects, simple shifters and counters, clock dividers, delays
  - Simple PSDsoft Express™ software...Free
- **Operating Range**
  - V_{CC}: 3.3V±10%; Temperature: –40°C to +85°C

Figure 1. Packages



PQFP52 (T)



PLCC52 (K)

- **In-System Programming (ISP) with JTAG**
  - Program entire chip in 10-25 seconds with no involvement of the DSP
  - Links with ADSP-2191 JTAG debug port
  - Eliminate sockets for pre-programmed memory and logic devices
  - ISP allows efficient manufacturing and product testing supporting Just-In-Time inventory
  - Use low-cost FlashLINK™ cable with PC
- **Content Security**
  - Programmable Security Bit blocks access of device programmers and readers
- **Zero-Power Technology**
  - As low as 25µA standby current
- **Packaging**
  - 52-pin PQFP or 52-pin PLCC
- **Flash Memory Speed, Endurance, Retention**
  - 150 ns, 100K cycles, 15 year retention

TABLE OF CONTENTS

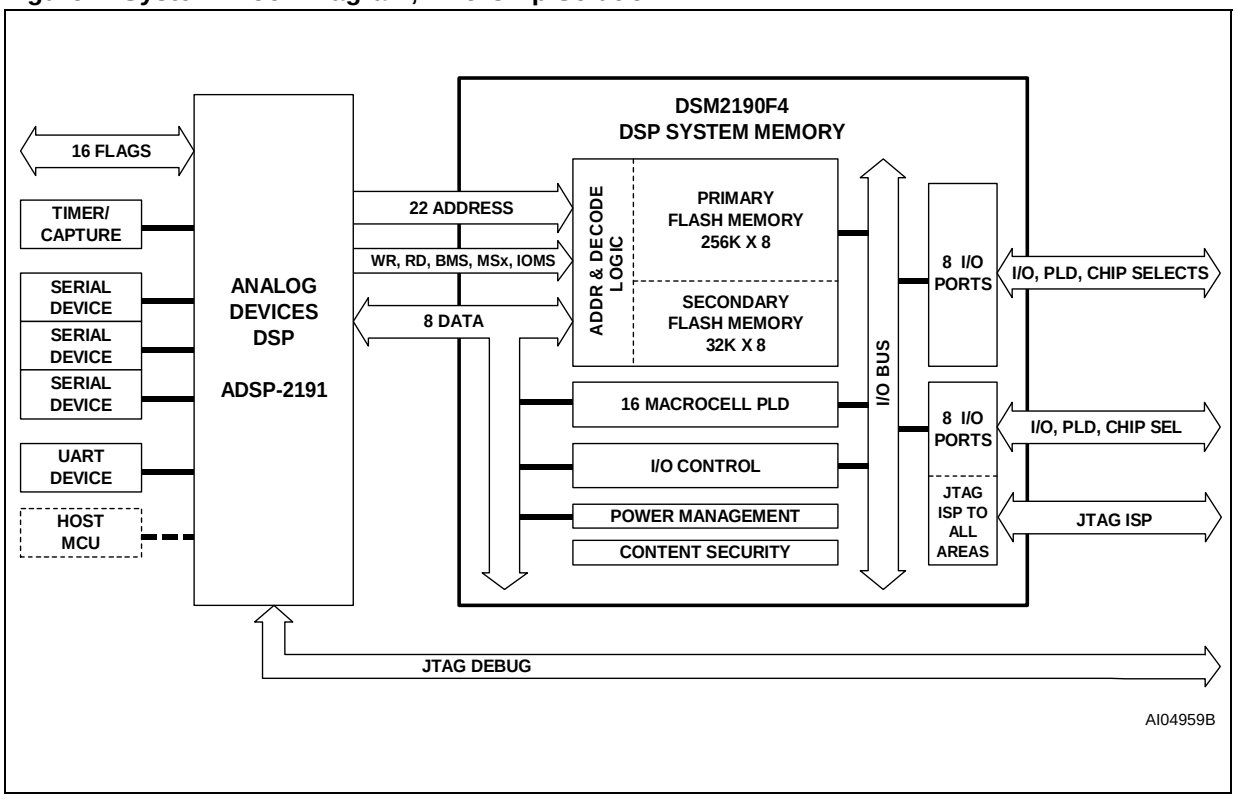
SUMMARY DESCRIPTION

The DSM2190F4 is a system memory device for use with the Analog Devices ADSP-2191 DSP. DSM means Digital signal processor System Memory. A DSM device brings In-System Programmable (ISP) Flash memory, parameter storage, programmable logic, and additional I/O to DSP systems. The result is a simple and flexible two-chip solution for DSP designs. DSM devices provide the flexibility of Flash memory and smart JTAG programming techniques for both manufacturing and the field. On-chip integrated memory decode logic makes it easy to map dual banks of Flash memory to the ADSP-2191 in a variety of ways for bootloading, code execution, data recording, code swapping, and parameter storage.

JTAG ISP reduces development time, simplifies manufacturing flow, and lowers the cost of field upgrades. The JTAG ISP interface eliminates the need for sockets and pre-programmed memory and logic devices. For manufacturing, end products may be assembled with a blank DSM device soldered to the circuit board and programmed at the end of the manufacturing line in 10 to 25 seconds with no involvement of the DSP. This allows efficient means to test product and manage inventory by rapidly programming test code, then application code as determined by inventory requirements (Just-In Time inventory). Additionally, JTAG ISP reduces development time by turning fast iterations of DSP code in the lab. Code updates in the field require no disassembly of product. The FlashLINK™ JTAG programming cable costs $59 USD and plugs into any PC or notebook parallel port.

In addition to ISP Flash memory, DSM devices add programmable logic (PLD) and up to 16 configurable I/O pins to the DSP system. The state of each I/O pin can be driven by DSP software or PLD logic. PLD and I/O configuration are programmable by JTAG ISP, just like the Flash memory. The PLD consists of more than 3000 gates and has 16 macro cell registers. Common uses for the PLD include chip selects for external devices, state-machines, simple shifters and counters, keypad and control panel interfaces, clock dividers, handshake delay, multiplexers, etc. This eliminates the need for small external PLDs and logic devices. Configuration of PLD, I/O, and Flash memory mapping are easily entered in a point-and-click environment using the software development tool, PSDsoft Express™. This software is available at no charge from *www.st.com/psm*.

Figure 2. System Block Diagram, Two-Chip Solution



The two-chip combination of a DSP and a DSM device is ideal for systems which have limitations on size, EMI levels, and power consumption. DSM memory and logic are “zero-power”, meaning they automatically go to standby between memory accesses or logic input changes, producing low active and standby current consumption, which is ideal for battery powered products.

A programmable security bit in the DSM protects its contents from unauthorized viewing and copying. When set, the security bit will block access of programming devices (JTAG or others) to the DSM Flash memories and PLD configuration. The only way to defeat the security bit is to erase the entire DSM device, after which the device is blank and may be used again. The DSP will always have access to Flash memory contents through the 8-bit data port even while the security bit is set.

Table 1. DSM2190F4V DSP Memory System Devices

Part Number	Main Flash Memory	Secondary Flash Memory	PLD	I/O Ports	V_{CC} and I/O	Mem Speed	Package	Operating Temp
DSM2190F4VV-15T6	256KBytes = 8 sectors x 32KByte	32KBytes = 4 sectors x 8KByte	16 macro -cells	Up to 16	3.3V ±10%	150 ns	52-pin PQFP	–40°C to +85°C
DSM2190F4VV-15K6	256KBytes = 8 sectors x 32KByte	32KBytes = 4 sectors x 8KByte	16 macro -cells	Up to 16	3.3V ±10%	150 ns	52-pin PLCC	–40°C to +85°C

Table 2. Compatible Analog Devices DSP

DSP Part Number	Operating Voltage, V_{CC}	I/O Capability
ADSP-2191M	2.5V	2.5 - 3.6V

Figure 3. PLCC Connections

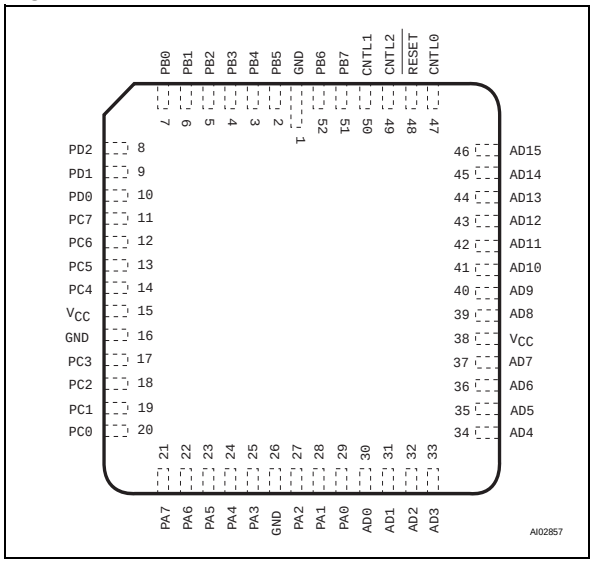
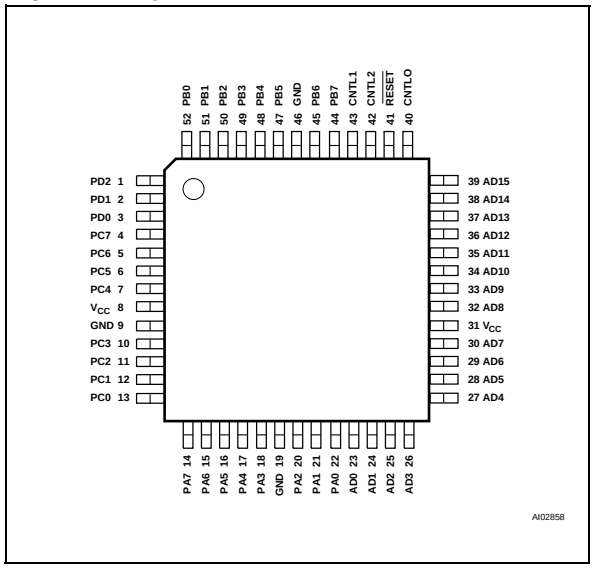


Figure 4. PQFP Connections



ARCHITECTURAL OVERVIEW

Major functional blocks are shown in Figure 5.

DSP Address/Data/Control Interface

These DSP signals attach directly to the DSM for a glueless connection. An 8-bit data connection is formed and all 22 DSP address lines can be decoded as well as DSP memory strobes; $\overline{BMS}$, $\overline{IOMS}$, and $\overline{MSx}$. There are many different ways the DSM2190F4 can be configured and used depending on system requirements. One convenient way is to combine the function of the $\overline{MSx}$ signals into the $\overline{BMS}$ signal. Doing this allows the DSP core to access DSM memory at runtime even after the boot process is complete using only the $\overline{BMS}$ signal. Combining $\overline{MSx}$ and $\overline{BMS}$ consumes less I/O pin(s) on the DMS device. See Analog Devices ADSP-2191 DSP Hardware Reference Manual, Chapter 7, Code Example: BMS Runtime Access. Alternatively, any of the $\overline{MSx}$ signals may also be used to decode any of the sectors of DSM Main Flash or Secondary flash memories.

Main Flash Memory

The 2M bit (256K x 8) Flash memory is divided into eight equally-sized 32K byte sectors that are individually selectable through the Decode PLD. Each Flash memory sector can be located at any address as defined by the user with PSDsoft Express. DSP code and data is easily placed in flash memory using the PSDsoft Express software development tool.

Secondary Flash Memory

The 256K bit (32K x 8) Flash memory is divided into eight equally-sized 8K byte sectors that are individually selectable through the Decode PLD. Each Flash memory sector can be located at any address as defined by the user with PSDsoft Express. DSP code and data can also be placed Secondary Flash memory using the PSDsoft Express development tool.

Secondary flash memory is good for storing data because of its small sectors. Additionally, software EEPROM emulation techniques can be used for small data sets that change frequently on a byte-by-byte basis.

Secondary flash may also be used to store custom start-up code for applications that do not “boot” using DMA, but instead start executing code from external memory upon reset. Storing code here can keep the entire Main Flash free of initialization code for clean software partitioning. If only one or more 8K byte sectors are needed for start-up code, the remaining sectors of Secondary Flash may be used for data storage.

Secondary Flash may also be used as an extension to Main Flash memory producing a total of 288K bytes

Miscellaneous: Main and Secondary Flash memories are totally independent, allowing concurrent operation if needed. The DSP can read from one memory while erasing or programming the other. The DSP can erase Flash memories by individual sectors or the entire Flash memory array may be erased at one time. Each sector in either Flash memory may be individually write protected, blocking any writes from the DSP (good for boot and start-up code protection). The Flash memories automatically go to standby between DSP read or write accesses to conserve power. Maximum access times include sector decoding time. Maximum erase cycles is 100K and data retention is 15 years minimum. Flash memory, as well as the entire DSM device may be programmed with the JTAG ISP interface with no DSP involvement.

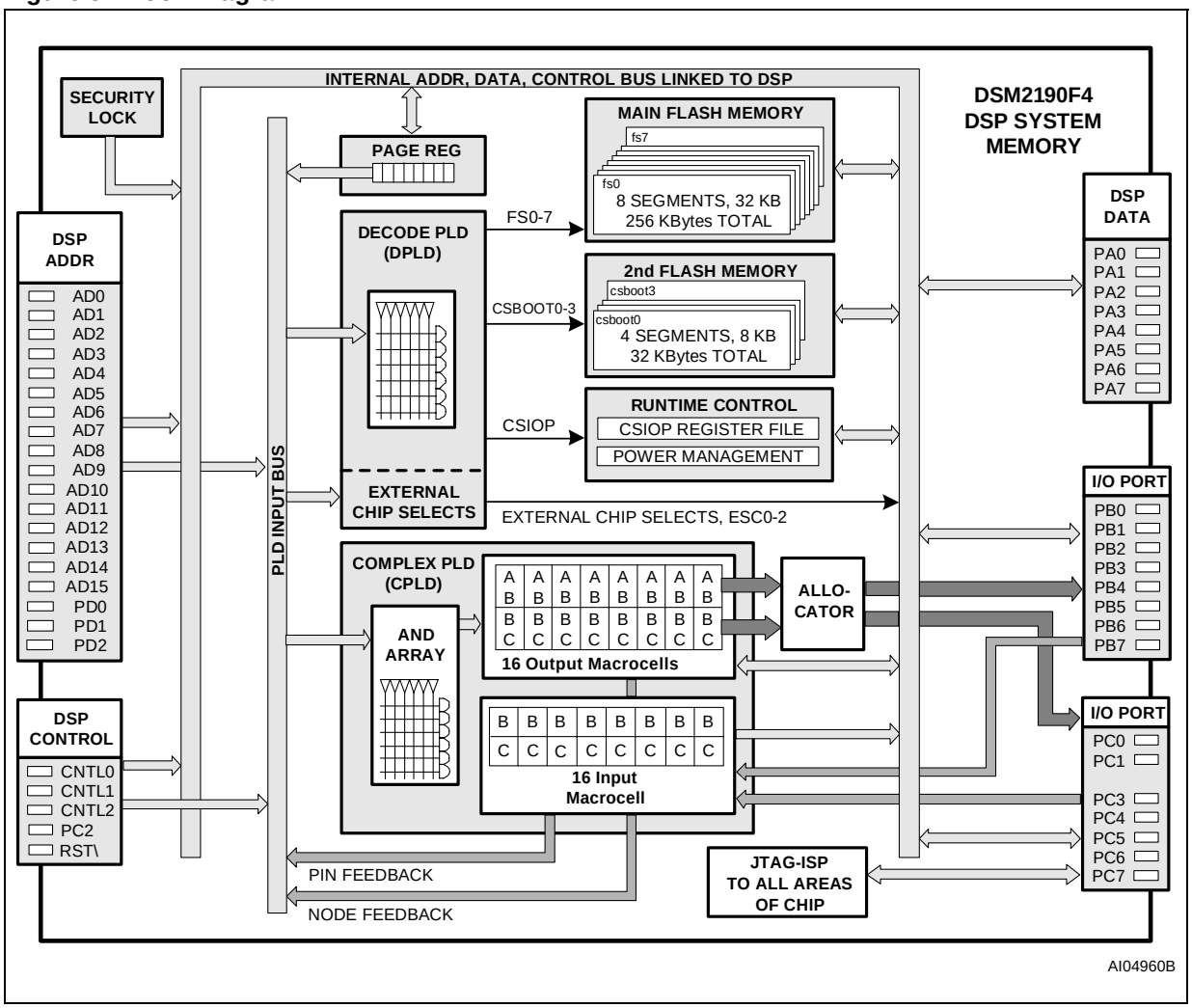
Programmable Logic (PLDs)

The DSM family contains two PLDS that may optionally run in Turbo or Non-Turbo mode. PLDs operate faster (less propagation delay) while in Turbo mode but consume more power than Non-Turbo mode. Non-Turbo mode allows the PLDs to automatically go to standby when no inputs are change to conserve power. The Turbo mode setting is controlled at runtime by DSP software.

Decode PLD (DPLD). This is programmable logic used to select one of the eight individual Main Flash memory segments, one of four individual Secondary Flash memory segments, or the group of control registers within the DSM device. The DPLD can also optionally drive external chip select signals on Port D pins. DPLD input signals include: DSP address and control signals, Page Register outputs, DSM Port Pins, CPLD logic feedback.

Complex PLD (CPLD). This programmable logic is used to create both combinatorial and sequential general purpose logic. The CPLD contains 16 Output Macrocells (OMCs) and 16 Input Macrocells (IMCs). PSD Macrocell registers are unique in that that have direct connection to the DSP data bus allowing them to be loaded and read directly by the DSP at runtime. This direct access is good for making small peripheral devices (shifters, counters, state machines, etc.) that are accessed directly by the DSP with little overhead. DPLD inputs include DSP address and control signals, Page Register outputs, DSM Port Pins, and CPLD feedback.

Figure 5. Block Diagram



OMCs: The general structure of the CPLD is similar in nature to a 22V10 PLD device with the familiar sum-of-products (AND-OR) construct. True and compliment versions of 64 input signals are available to a large AND array. AND array outputs feed into a multiple product-term OR gate within each OMC (up to 10 product-terms for each OMC). Logic output of the OR gate can be passed on as combinatorial logic or combined with a flip-flop within in each OMC to realize sequential logic. OMCs can be used as a buried nodes with feedback to the AND array or OMC output can be routed to pins on Port B or PortC.

IMCs: Inputs from pins on Port B or Port C are routed to IMCs for conditioning (clocking or latching) as they enter the chip, which is good for sampling and debouncing inputs. Alternatively, IMCs can pass Port input signals directly to PLD inputs without clocking or latching. The DSP may read the IMCs at any time.

Runtime Control Registers

A block of 256 bytes is decoded inside the DSM device as DSM control and status registers. 27 registers are used in the block of 256 locations to control the output state of I/O pins, to read I/O pins, to control power management, to read/write macrocells, and other functions at runtime. See Table 4 for description. The base address of these 256 locations is referred to in this data sheet as *csiop* (Chip Select I/O Port). Individual registers within this block are accessed with an offset from the base address. The DSP accesses *csiop* registers using I/O memory with the $\overline{\text{IOMS}}$ strobe. *csiop* registers are accessed as bytes.

Memory Page Register

This 8-bit register can be loaded and read by the DSP at runtime as one of the *csiop* registers. Its outputs feed directly into the PLDs. The page register can be used for special memory mapping requirements and also for general logic.

I/O Ports

The DSM has 19 individually configurable I/O pins distributed over the three ports (Ports B, C, and D). Each I/O pin can be individually configured for different functions such as standard MCU I/O ports or PLD I/O on a pin by pin basis. (MCU I/O means that for each pin, its output state can be controlled or its input value can be read by the DSP at runtime using the *csiop* registers like an MCU would do.)

Port C hosts the JTAG ISP signals. Since JTAG-ISP does not occur frequently during the life of a product, those Port C pins are under-utilized. In applications that need every I/O pin, JTAG signals can be multiplexed with general I/O signals to use them for I/O when not performing ISP. See section titled “Programming In-Circuit using JTAG ISP” on page 40 for muxing JTAG pins on Port C, and Application Note *AN1153*.

The static configuration of all Port pins is defined with the PSDsoft Express™ software development tool. The dynamic action of the Ports pins is controlled by DSP runtime software.

JTAG ISP Port

In-System Programming (ISP) can be performed through the JTAG signals on Port C. This serial interface allows programming of the entire DSM device or subsections (that is, only Flash memory but not the PLDs) without the participation of the DSP. A blank DSM device soldered to a circuit board can be completely programmed in 10 to 25 seconds. The basic JTAG signals; TMS, TCK, TDI, and TDO form the IEEE-1149.1 interface. The DSM device does not implement the IEEE-1149.1 Boundary Scan functions. The DSM uses the JTAG interface for ISP only. However, the DSM device can reside in a standard JTAG chain with other JTAG devices (including the ADSP-2191) and it will remain in BYPASS mode while other devices perform Boundary Scan.

ISP programming time can be reduced as much as 30% by using two more signals on Port C, TSTAT and $\overline{\text{TERR}}$ in addition to TMS, TCK, TDI and TDO.

The FlashLINK™ JTAG programming cable is available from STMicroelectronics for $59USD and PSDsoft Express software is available at no charge from *www.st.com/psm*. That is all that is needed to program a DSM device using the parallel port on any PC or note-book. See section titled “Programming In-Circuit using JTAG ISP” on page 40.

Power Management

The DSM has bits in *csiop* control registers that are configured at run-time by the DSP to reduce power consumption of the CPLD. The Turbo bit in the PMMR0 register can be set to logic 1 and the CPLD will go to Non-Turbo mode, meaning it will latch its outputs and go to sleep until the next transition on its inputs. There is a slight penalty in PLD performance (longer propagation delay), but significant power savings are realized.

Additionally, bits in two *csiop* registers can be set by the DSP to selectively block signals from entering the CPLD which reduces power consumption. See section titled “Power Management” on page 37.

Security and NVM Sector Protection

A programmable security bit in the DSM protects its contents from unauthorized viewing and copying. When set, the security bit will block access of programming devices (JTAG or others) to the DSM Flash memory and PLD configuration. The only way to defeat the security bit is to erase the entire DSM device, after which the device is blank and may be used again.

Additionally, the contents of each individual Flash memory sector can be write protected (sector protection) by configuration with PSDsoft Express™. This is typically used to protect DSP boot code from being corrupted by inadvertent writes to Flash memory from the DSP.

Pin Assignments

Pin assignment are shown for the 52-pin PLCC package in Figure 3, and the 52-pin PQFP package in Figure 4.

Table 3. Pin Description

Pin Name	Type	Description
ADIO0-15	In	Sixteen address inputs from the DSP.
CNTL0	In	Active low write strobe input ($\overline{WR}$) from the DSP
CNTL1	In	Active low read strobe input ($\overline{RD}$) from the DSP.
CNTL2	In	Active low Byte Memory Select ($\overline{BMS}$) signal from the DSP.
$\overline{Reset}$	In	Active low reset input from system. Resets DSM I/O Ports, Page Register contents, and other DSM configuration registers. Must be logic Low at Power-up.
PA0-7	I/O	Eight data bus signals connected to DSP pins D8 - D15.
PB0-7	I/O	Eight configurable Port B signals with the following functions: 1. MCU I/O – DSP may write or read pins directly at runtime with csiop registers. 2. CPLD Output Macrocell (McellAB0-7 or McellBC0-7) outputs. 3. Inputs to the PLDs (Input Macrocells). Note: Each of the four Port B signals PB0-PB3 may be configured at run-time as either standard CMOS or for high slew rate. Each of the four Port B signals PB3-PB7 may be configured at run-time as either standard CMOS or Open Drain Outputs.
PC0-7	I/O	Eight configurable Port C signals with the following functions: 1. MCU I/O – DSP may write or read pins directly at runtime with csiop registers. 2. CPLD Output Macrocell (McellBC0-7) output. 3. Input to the PLDs (Input Macrocells). 4. Pins PC0, PC1, PC5, and PC6 can optionally form the JTAG IEEE-1149.1 ISP serial interface as signals TMS, TCK, TDI, and TDO respectively. 5. Pins PC3 and PC4 can optionally form the enhanced JTAG signals TSTAT and $\overline{TERR}$ respectively. Reduces ISP programming time by up to 30% when used in addition to the standard four JTAG signals: TDI, TDO, TMS, TCK. 6. Pin PC3 can optionally be configured as the Ready/Busy output to indicate Flash memory programming status during parallel programming. May be polled by DSP or used as DSP interrupt to indicate when Flash memory byte programming or erase operations are complete. Note 1: Port C pin PC2 input (or any PLD input pin) can be connected to the DSP $\overline{IOMS}$ output. See Figure 6. Note 2: When used as general I/O, each of the eight Port C signals may be configured at run-time as either standard CMOS or Open Drain Outputs. Note 3: The JTAG ISP pins may be multiplexed with other I/O functions.
PD0-2	I/O	Three configurable Port D signals with the following functions: 1. MCU I/O – DSP may write or read pins directly at runtime with csiop registers. 2. Input to the PLDs (no associated Input Macrocells, routes directly into PLDs). 3. CPLD output (External Chip Select). Does not consume Output Macrocells. 4. Pin PD1 can optionally be configured as CLKIN, a common clock input to PLD. 5. Pin PD2 can optionally be configured as $\overline{CSI}$, an active low Chip Select Input to select Flash memory. Flash memory is disabled to conserve more power when $\overline{CSI}$ is logic high. Can connect $\overline{CSI}$ to ADSP-218X PWDACK output signal. Note 1: Port D pin PD0 (or any PLD input pin) can be connected to the DSP A16 output. See Figure 6 Note 2: Port D pin PD1 (or any PLD input pin) can be connected to the DSP A17 output. See Figure 6. Note 3: Port D pin PD2 (or any PLD input pin) can be connected to the DSP A18 output. See Figure 6
V_{CC}		Supply Voltage
GND		Ground pins

TYPICAL CONNECTIONS

Figure 6 shows a typical connection scheme. Many connection possibilities exist since many DSM pins are multipurpose. This scheme illustrates the use of a combined function $\overline{BSM}$ signal (functions as $\overline{BMS}$ and $\overline{MSx}$), and many I/O pins. It also illustrates how to chain the DSM and DSP devices together on the JTAG bus. The JTAG connector definition depends on development and production environment requirements. A specially defined connector can be devised to combine the signals of the FlashLINK and the Analog Devices emulator. Alternatively, two separate JTAG connectors can be used, one matching the pinout of FlashLINK and the other matching the emulator pinout.

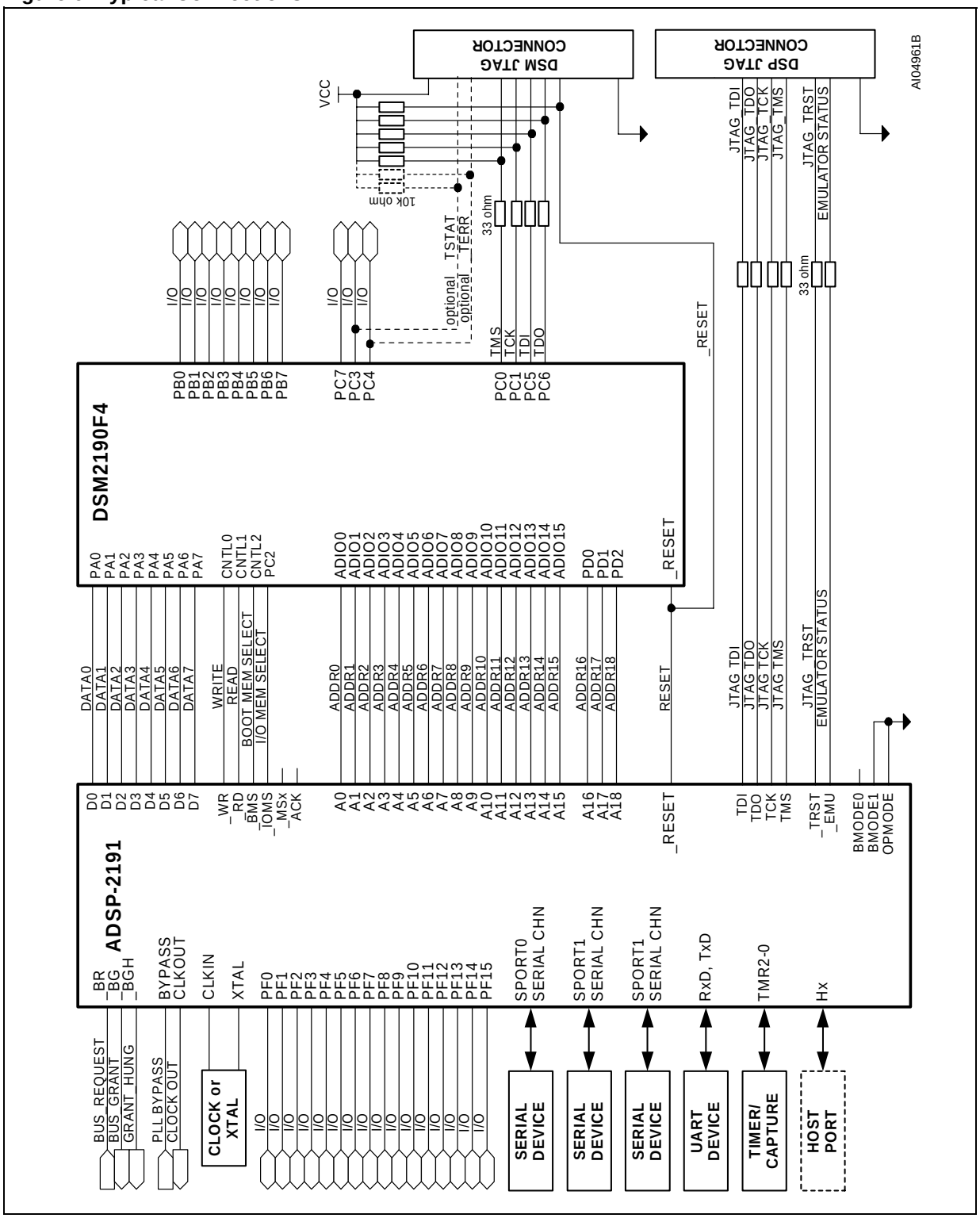
Keep in mind that signals $\overline{BMS}$, $\overline{IOMS}$, $\overline{MSx}$, ADDR16, ADDR17, ADDR18 can be connected to any DSM pin that is a PLD input. I/O pins on Port B and Port C are more capable (more PLD functions) than Port D pins. It is recommended to use Port D pins primarily for decode inputs first, leaving pins on Port B and Port C available for general logic. Figure 6 illustrates a common way to make connections.

Following are connection options to consider:

Port C JTAG: Figure 6 shows four JTAG signals (TMS, TCK, TDI, TDO) connected to the DSM. Alternatively, using six-pin JTAG (two more signals, TSTAT and $\overline{TERR}$) can reduce ISP time by as much as 30% compared to four-pin JTAG. Other JTAG options include multiplexing JTAG pins with general I/O (see “Programming In-Circuit using JTAG ISP” on page 40 and Application Note *AN1153*), or not using JTAG at all. If no JTAG is used, the DSM device has to be programmed on a conventional programmer before it is installed on the circuit board. Using no JTAG makes more DSM I/O available.

Pins PC2 and PD2. If not all 288K address locations need to be decoded in the DSM, then ADDR18 on pin PD2 is not needed. In this case, the $\overline{IOMS}$ signal can be connected to pin PD2, freeing pin PC2 for general I/O usage.

Figure 6. Typical Connections



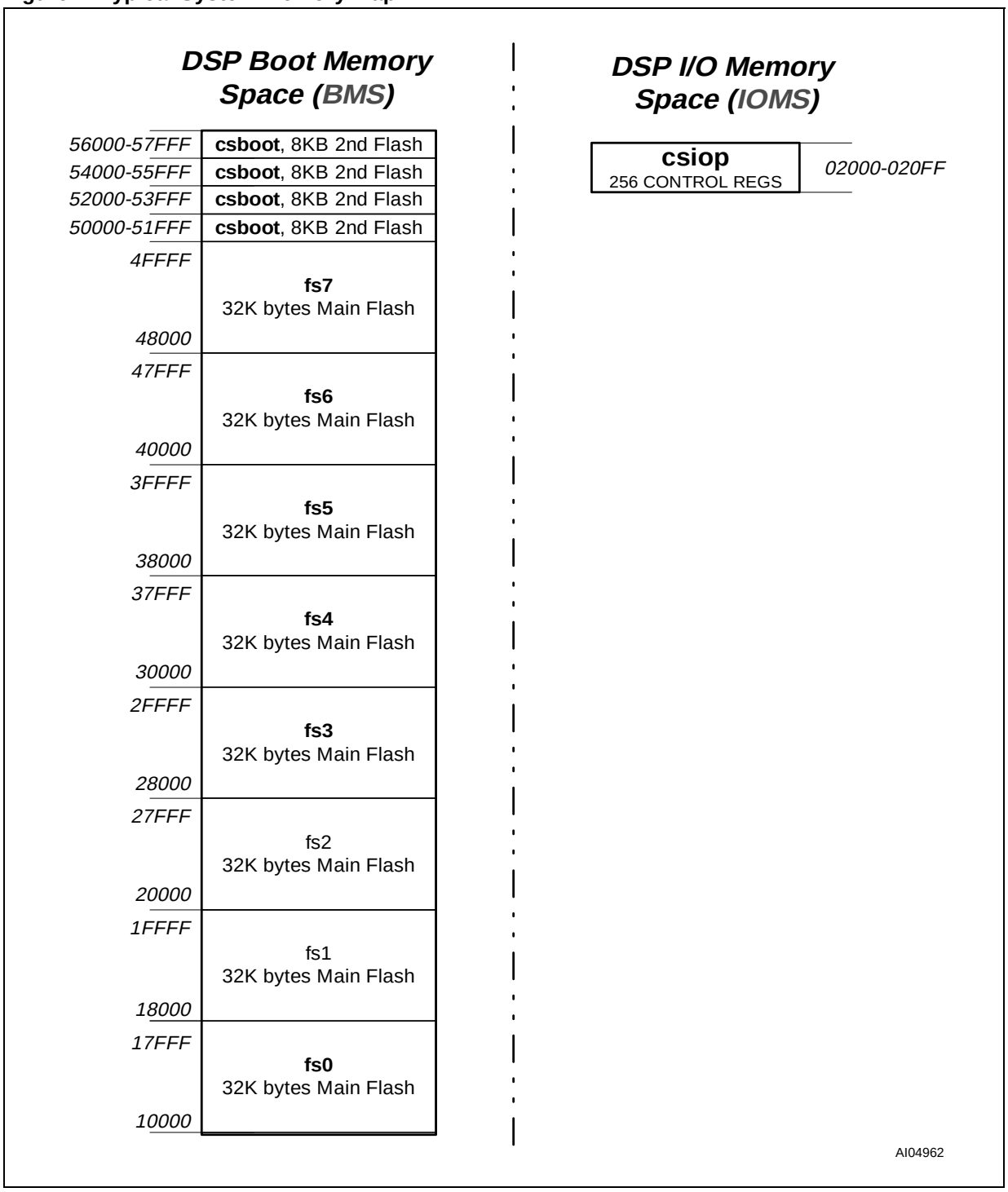
TYPICAL MEMORY MAP

There many different ways to place (or map) the addresses of DSM memory and I/O depending on system requirements. The DPLD allows complete mapping flexibility. Figure 7 shows one possible system memory map. In this case, the DSP will bootload (via DMA) the contents of Main Flash memory upon reset. The Secondary Flash memory can be used for parameter storage or additional code storage. $\overline{BMS}$ and $\overline{MSx}$ are configured in the DSP to be combined into the $\overline{BMS}$ signal, allowing the DSP to access both Flash memories at runtime (after DMA boot). The DSP may execute code directly from the DSM and well as erase and write new code or data to DSM Flash.

The nomenclature *fs0..fs7* are designators for the individual sectors of Main Flash memory, 32K bytes each. *csboot0..csboot3* are designators for the individual Secondary Flash memory segments, 8K bytes each. *csiop* designates the DSM control register block.

The designer may easily specify memory mapping in a point-and-click software environment using PSDsoft Express™.

Figure 7. Typical System Memory Map



SPECIFYING THE MEMORY MAP WITH PSDSOFT EXPRESS™

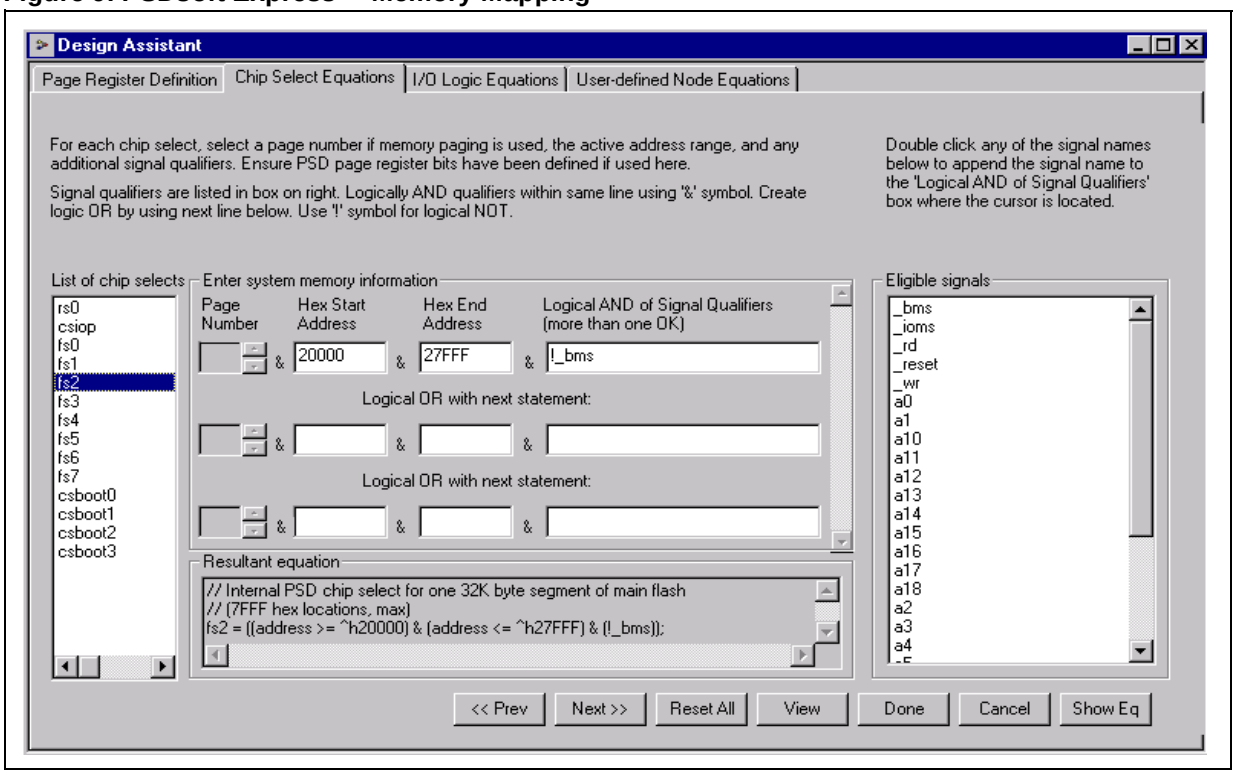
The memory map shown in Figure 7 can be easily implemented using PSDsoft Express™ in a point-and-click environment. PSDsoft Express™ will generate Hardware Definition Language (HDL) statements of the ABEL language. Figure 8 shows the resulting equations generated by PSDsoft Express™.

Figure 8. HDL Statements Generated from PSDsoft Express to Implement Memory Map

```
csiop = ((address >= ^h2000) & (address <= ^h20FF) & (!_ioms));
fs0 = ((address >= ^h10000) & (address <= ^h17FFF) & (!_bms));
fs1 = ((address >= ^h18000) & (address <= ^h1FFFF) & (!_bms));
fs2 = ((address >= ^h20000) & (address <= ^h27FFF) & (!_bms));
fs3 = ((address >= ^h28000) & (address <= ^h2FFFF) & (!_bms));
fs4 = ((address >= ^h30000) & (address <= ^h37FFF) & (!_bms));
fs5 = ((address >= ^h38000) & (address <= ^h3FFFF) & (!_bms));
fs6 = ((address >= ^h40000) & (address <= ^h47FFF) & (!_bms));
fs7 = ((address >= ^h48000) & (address <= ^h4FFFF) & (!_bms));
csboot0 = ((address >= ^h50000) & (address <= ^h51FFF) & (!_bms));
csboot1 = ((address >= ^h52000) & (address <= ^h53FFF) & (!_bms));
csboot2 = ((address >= ^h54000) & (address <= ^h55FFF) & (!_bms));
csboot3 = ((address >= ^h56000) & (address <= ^h57FFF) & (!_bms));
```

Specifying these equations using PSDsoft Express™ is very simple. Figure 9 shows how to specify the equation for the 32K Byte Flash memory segment, *fs2*. Notice *fs2* is qualified with the signals $\overline{\text{BMS}}$. This specification process is repeated for all other Flash memory segments, the *csiop* register block, and any external chip select signals that may be needed (ADC, etc.).

Figure 9. PSDsoft Express™ Memory Mapping



RUNTIME CONTROL REGISTER DEFINITION

There are up to 256 addresses decoded inside the DSM device for control and status information. 27 of these locations contain registers that the DSP can access at runtime. The base address of this block of 256 locations is referred to in this manual as *csiop* (Chip Select I/O Port). Table 4 lists the 27 registers and their offsets (in hexadecimal) from the *csiop* base address needed to access individual DSM control and status registers. The DSP will access these registers in I/O memory space using its $\overline{\text{IOMS}}$ strobe. These registers are accesses in bytes, so the DSP should ignore the upper byte of its 16-bit I/O access.

Note1: All *csiop* registers are cleared to logic 0 at reset.

Note2: Do not write to unused locations within the *csiop* block of 256 registers. They should remain logic zero.

Table 4. *CSIOP* Registers and their Offsets (in hexadecimal)

Register Name	Port B	Port C	Port D	Other	Description
Data In	01	10	11		MCUI/O input mode. Read to obtain current logic level of Port pins. No writes.
Data Out	05	12	13		MCU I/O output mode. Write to set logic level on Port pins. Read to check status.
Direction	07	14	15		MCU I/O mode. Configures Port pin as input or output. Write to set direction of Port pins. Logic 1 = out, Logic 0 = in. Read to check status.
Drive Select	09	16	17		Write to configure Port pins as either standard CMOS or Open Drain on some pins, while selecting high slew rate on other pins. Read to check status.
Input Macrocells	0B	18			Read to obtain state of IMCs. No writes.
Enable Out	0D	1A	1B		Read to obtain the status of the output enable logic on each I/O Port driver. No writes.
Output Macrocells AB				20	Read to get logic state of output of OMC bank AB. Write to load registers of OMC bank AB.
Output Macrocells BC				21	Read to get logic state of output of OMC bank BC. Write to load registers of OMC bank BC.
Mask Macrocells AB				22	Write to set mask for loading OMCs in bank AB. A logic 1 in a bit position will block reads/writes of the corresponding OMC. A logic 0 will pass OMC value. Read to check status.
Mask Macrocells BC				23	Write to set mask for loading OMCs in bank BC. A logic 1 in a bit position will block reads/writes of the corresponding OMC. A logic 0 will pass OMC value. Read to check status.
Main Flash Sector Protection				C0	Read to determine Main Flash Sector Protection Setting. No writes.
Security Bit and Secondary Flash Sector Protection				C2	Read to determine if DSM devices Security Bit is active. Logic 1 = device secured. Also read to determine Secondary Flash Protection Setting status. No Writes.
JTAG Enable				C7	Write to enable JTAG Pins (optional feature). Read to check status.
PMMR0				B0	Power Management Register 0. Write and read.
PMMR2				B4	Power Management Register 2. Write and read.
Page				E0	Memory Page Register. Write and read.

DETAILED OPERATION

Figure 5 shows major functional areas of the device:

- Flash Memories
- PLDs (DPLD, CPLD, Page Register)
- DSP Bus Interface (Address, Data, Control)
- I/O Ports
- Runtime Control Registers
- JTAG ISP Interface

The following describes these functions in more detail.

Flash Memories

The Main Flash memory array is divided into eight equal 32K byte sectors. The Secondary Flash memory array is divided into four equal 8K byte sectors. Each sector is selected by the DPLD can be separately protected from program and erase cycles. This configuration is specified by using PSDsoft Express™.

Memory Sector Select Signals. The DPLD generates the Select signals for all the internal memory blocks (see Figure 14). Each of the twelve sectors of the Flash memories has a select signal (*FS0-FS7, or CSBOOT0-CSBOOT3*) which contains up to three product terms. Having three product terms for each select signal allows a given sector to be mapped into multiple areas of system memory if needed.

Ready/$\overline{\text{Busy}}$ (PC3). This signal can be used to output the Ready/$\overline{\text{Busy}}$ status of the device. The output on Ready/$\overline{\text{Busy}}$ is a 0 (Busy) when either Flash memory array is being written, *or* when either Flash memory array is being erased. The output is a 1 (Ready) when no Write or Erase cycle is in progress. This signal may be polled by the DSP or used as a DSP interrupt to indicate when an erase or program cycle is complete.

Memory Operation. The Flash memories are accessed through the DSP Address, Data, and Control Bus Interface.

DSPs and MCUs cannot write to Flash memory as it would an SRAM device. Flash memory must first be “unlocked” with a special sequence of byte write operations to invoke an internal algorithm, then a single data byte is written to the Flash memory array, then programming status is checked by a byte read operation or by checking the Ready/$\overline{\text{Busy}}$ pin (PC3). Table 5 lists all of the special instruction sequences to program (write) data to the Flash memory arrays, erase the arrays, and check for different types of status from the arrays. These instruction sequences are different combinations of individual byte write and byte read operations. IMPORTANT: The DSP may not read and execute code from the same Flash memory array for which it is directing an instruction sequence. Or more simply stated, the DSP may not read code the same Flash array that is writing or erasing. Instead, the DSP must execute code from an alternate memory (like its own internal SRAM or a different Flash array) while sending instructions to a given Flash array. Since the two Flash memory arrays inside the DSM device are completely independent, the DSP may read code from one array while sending instructions to the other.

After a Flash memory array is programmed (written) it will go to “Read Array” mode, then the DSP can read from Flash memory just as if would from any 8-bit ROM or SRAM device.

Table 5. Instruction Sequences[1,2,3,4]

Instruction Sequence	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
Read Memory Contents[5]	Read byte from any valid Flash memory addr						
Read Flash Identifier (Main Flash only)[6,7]	Write AAh to XX555h	Write 55h to XXAAAh	Write 90h to XX555h	Read identifier with addr lines A6,A1,A0 = 0,0,1			
Read Memory Sector Protection Status[6,7,8]	Write AAh to XX555h	Write 55h to XXAAAh	Write 90h to XX555h	Read identifier with addr lines A6,A1,A0 = 0,1,0			
Program a Flash Byte	Write AAh to XX555h	Write 55h to XXAAAh	Write A0h to XX555h	Write (program) data to addr			
Flash Bulk Erase[9]	Write AAh to XX555h	Write 55h to XXAAAh	Write 80h to XX555h	Write AAh to XX555h	Write 55h to XXAAAh	Write 10h to XX555h	
Flash Sector Erase[10]	Write AAh to XX555h	Write 55h to XXAAAh	Write 80h to XX555h	Write AAh to XX555h	Write 55h to XXAAAh	Write 30h to another Sector	Write 30h to another Sector
Suspend Sector Erase[11]	Write B0h to address that activates any of FS0 - FS7						
Resume Sector Erase[12]	Write 30h to addr that activates any of FS0 - FS7						
Reset Flash[6]	Write F0h to address that activates any of FS0 - FS7						

Note: 1. All values are in hexadecimal, X = Don't Care
2. A desired internal Flash memory sector select signal (FS0 - FS7 or CSBOOT0 - CSBOOT3) must be active for each write or read cycle. Only one of these sector select signals will be active at any given time depending on the address presented by the DSP and the memory mapping defined in PSDsoft Express. FS0 - FS7 and CSBOOT0-CSBOOT3 are active high logic internally.
3. DSP addresses A18 through A12 are Don't Care during the instruction sequence decoding. Only address bits A11-A0 are used during Flash memory instruction sequence decoding bus cycles. The individual sector select signal (FS0 - FS7 or CSBOOT0-CSBOOT3) which is active during the instruction sequences determines the complete address.
4. For write operations, addresses are latched on the falling edge of Write Strobe ($\overline{WR}$, CNTL0), Data is latched on the rising edge of Write Strobe ($\overline{WR}$, CNTL0)
5. No Unlock or Instruction cycles are required when the device is in the Read Array mode. Operation is like reading a ROM device.
6. The Reset Flash instruction is required to return to the normal Read Array mode if the Error Flag (DQ5) bit goes High, or after reading the Flash Identifier or after reading the Sector Protection Status.
7. The DSP cannot invoke this instruction sequence while executing code from the same Flash memory as that for which the instruction sequence is intended. The DSP must fetch, for example, the code from the DSP SRAM when reading the Flash memory Identifier or Sector Protection Status.
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. Directing this command to any individual active Flash memory segment (FS0 - FS7) will invoke the bulk erase of all eight Flash memory sectors.
10. DSP writes command sequence to initial segment to be erased, then writes the byte 30h to additional sectors to be erased. The byte 30h must be addressed to one of the other Flash memory segments (FS0 - FS7) for each additional segment (write 30h to any address within a desired sector). No more than 80uS can elapse between subsequent additional sector erase commands.
11. The system may perform Read and Program cycles in non-erasing sectors, read the Flash ID or read the Sector Protect Status, when in the Suspend Sector Erase mode. The Suspend Sector Erase instruction sequence is valid only during a Sector Erase cycle.
12. The Resume Sector Erase instruction sequence is valid only during the Suspend Sector Erase mode.

Instruction Sequences

An instruction sequence consists of a sequence of specific write or read operations. Each byte written to the device is received and sequentially decoded and not executed as a standard write operation to the memory array. The instruction sequence 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 instruction sequences are structured to include read operations after the initial write operations.

The instruction sequence 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 Array mode (Flash memory is read like a ROM device). The device supports the instruction sequences summarized in Table 5:

Flash memory:

- Erase memory by chip or sector
- Suspend or resume sector erase
- Program a Byte
- Reset to Read Array mode
- Read primary Flash Identifier value
- Read Sector Protection Status

These instruction sequences are detailed in Table 5. For efficient decoding of the instruction sequences, the first two bytes of an instruction sequence 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 XX555h during the first cycle and data 55h to address XXAAAh during the second cycle. Address signals A18-A12 are Don't Care during the instruction sequence Write cycles. However, the appropriate internal Sector Select (*FS0-FS7 or CSBOOT0-CSBOOT3*) must be selected internally (active, which is logic 1).

Reading Flash Memory

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

Read Memory Contents. Flash memory is placed in the Read Array mode after Power-up, chip reset, or a Reset Flash memory instruction sequence (see Table 5). The DSP can read the memory contents of the Flash memory by using read operations any time the read operation is not part of an instruction sequence.

Read Main Flash Identifier. The Main Flash memory identifier is read with an instruction sequence composed of 4 operations: 3 specific write operations and a read operation (see Table 5). During the read operation, address bits A6, A1, and A0 must be 0,0,1, respectively, and the appropriate internal Sector Select (*FS0-FS7*) must be active. The identifier is 0xE7. Not Applicable to Secondary Flash.

Read Memory Sector Protection Status. The Flash memory Sector Protection Status is read with an instruction sequence composed of 4 operations: 3 specific write operations and a read operation (see Table 5). During the read operation, address bits A6, A1, and A0 must be 0,1,0, respectively, while internal Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) 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 can also be read by the DSP accessing the Flash memory Protection registers in *csiop* space. See the section entitled "Flash Memory Sector Protect" for register definitions.

Table 6. Status Bit Definition

Functional Block	FS0-FS7, or CSBOOT0-CSBOOT3	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Flash Memory	Active (the desired segment is selected)	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.

Reading the Erase/Program Status Bits. The device provides several status bits to be used by the DSP to confirm the completion of an Erase or Program cycle of Flash memory. These status bits minimize the time that the DSP spends performing these tasks and are defined in Table 6. The status bits can be read as many times as needed.

For Flash memory, the DSP can perform a read operation to obtain these status bits while an Erase or Program instruction sequence is being executed by the embedded algorithm. See the section entitled “Programming Flash Memory”, on page 19, for details.

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

Flash memory instruction features.

- Data Polling is effective after the fourth Write pulse (for a Program instruction sequence) or after the sixth Write pulse (for an Erase instruction sequence). 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 (DQ7) bit outputs a 0. After completion of the cycle, the Data Polling Flag (DQ7) bit 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 sequence is ignored.
- If all the Flash memory sectors to be erased are protected, the Data Polling Flag (DQ7) bit is reset to 0 for about 100 µs, and then returns to the previous addressed byte. No erasure is performed.

Toggle Flag (DQ6). The device offers another way for determining when the Flash memory Program cycle is completed. During the internal write operation and when the Sector Select FS0-FS7 (or CSBOOT0-CSBOOT3) is true, the Toggle Flag (DQ6) bit 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 D0-7 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. Flash memory specific features:

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

Error Flag (DQ5). During a normal Program or Erase cycle, the Error Flag (DQ5) bit 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 (DQ5) bit 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 (DQ5) bit 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 (DQ5) bit is reset after a Reset Flash instruction sequence.

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

Programming Flash Memory

When a byte of Flash memory is programmed, individual bits are programmed to logic 0. You cannot program a bit in Flash memory to a logic 1 once it has been programmed to a logic 0. A bit must be erased to logic 1, and programmed to logic 0. That means Flash memory must be erased prior to being programmed. A byte of Flash memory is erased to all 1s (FFh). The DSP may erase the entire Flash memory array all at once or individual sector-by-sector, but not byte-by-byte. However, the DSP may program Flash memory byte-by-byte.

The Flash memory requires the DSP to send an instruction sequence to program a byte or to erase sectors (see Table 5).

Once the DSP issues a Flash memory Program or Erase instruction sequence, it must check for the status bits for completion. The embedded algorithms that are invoked inside the device provide several ways give status to the DSP. Status may

be checked using any of three methods: Data Polling, Data Toggle, or Ready/Busy (pin PC3).

Data Polling. Polling on the Data Polling Flag (DQ7) bit is a method of checking whether a Program or Erase cycle is in progress or has completed. Figure 10 shows the Data Polling algorithm.

When the DSP issues a Program instruction sequence, the embedded algorithm within the device begins. The DSP then reads the location of the byte to be programmed in Flash memory to check status. The Data Polling Flag (DQ7) bit of this location becomes the compliment of bit 7 of the original data byte to be programmed. The DSP continues to poll this location, comparing the Data Polling Flag (DQ7) bit and monitoring the Error Flag (DQ5) bit. When the Data Polling Flag (DQ7) bit matches bit7 of the original data, and the Error Flag (DQ5) bit remains 0, then the embedded algorithm is complete. If the Error Flag (DQ5) bit is 1, the DSP should test the Data Polling Flag (DQ7) bit again since the Data Polling Flag (DQ7) bit may have changed simultaneously with the Error Flag (DQ5) bit (see Figure 10).

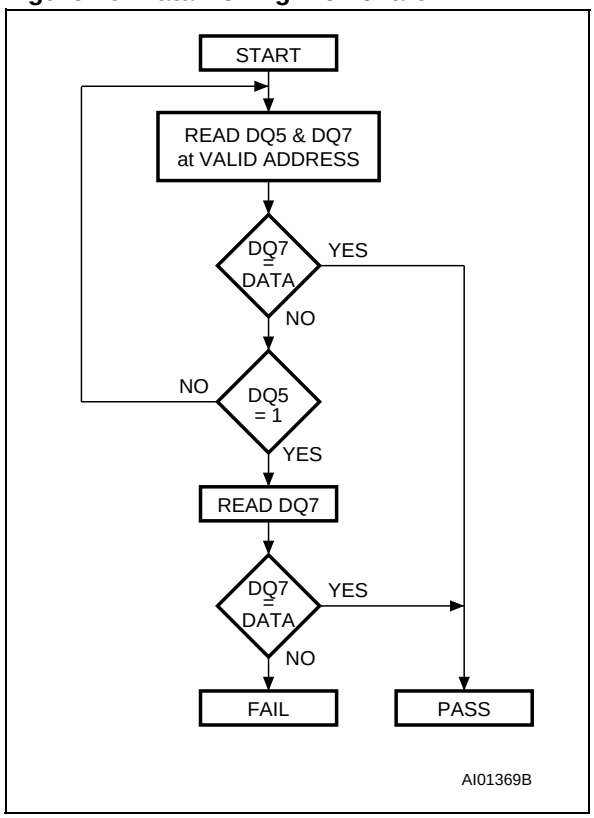
The Error Flag (DQ5) bit is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte or if the DSP attempted to program a 1 to a bit that was not erased (not erased is logic 0).

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 10 still applies. However, the Data Polling Flag (DQ7) bit is 0 until the Erase cycle is complete. A 1 on the Error Flag (DQ5) bit indicates a time-out condition on the Erase cycle, a 0 indicates no error. The DSP can read any location within the sector being erased to get the Data Polling Flag (DQ7) bit and the Error Flag (DQ5) bit.

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

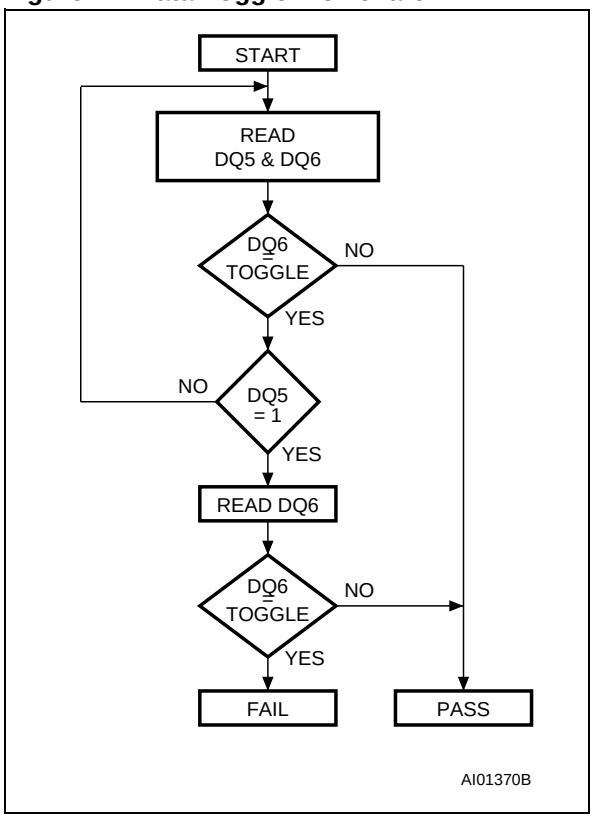
Figure 10. Data Polling Flowchart



Data Toggle. Checking the Toggle Flag (DQ6) bit is a method of determining whether a Program or Erase cycle is in progress or has completed. Figure 11 shows the Data Toggle algorithm.

When the DSP issues a Program instruction sequence, the embedded algorithm within the device begins. The DSP then reads the location of the byte to be programmed in Flash memory to check status. The Toggle Flag (DQ6) bit of this location toggles each time the DSP reads this location until the embedded algorithm is complete. The DSP continues to read this location, checking the Toggle Flag (DQ6) bit and monitoring the Error Flag (DQ5) bit. When the Toggle Flag (DQ6) bit stops toggling (two consecutive reads yield the same value), and the Error Flag (DQ5) bit remains 0, then the embedded algorithm is complete. If the Error Flag (DQ5) bit is 1, the DSP should test the Toggle Flag (DQ6) bit again, since the Toggle Flag (DQ6) bit may have changed simultaneously with the Error Flag (DQ5) bit (see Figure 11).

Figure 11. Data Toggle Flowchart



The Error Flag (DQ5) bit is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte, or if the DSP 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 11 still applies. the Toggle Flag (DQ6) bit toggles until the Erase cycle is complete. A 1 on the Error Flag (DQ5) bit indicates a time-out condition on the Erase cycle, a 0 indicates no error. The DSP can read any location within the sector being erased to get the Toggle Flag (DQ6) bit and the Error Flag (DQ5) bit.

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

Erasing Flash Memory

Flash Bulk Erase. The Flash Bulk Erase instruction sequence uses six write operations followed by a read operation of the status register, as described in Table 5. If any byte of the Bulk Erase instruction sequence is wrong, the Bulk Erase instruction sequence aborts and the device is reset to the Read Flash memory status. The Bulk Erase command may be addresses to any one individual valid Flash memory segment (*FS0-FS7 or CSBOOT0-CSBOOT3*) and the entire array (all segments in one array) will be erased.

During a Bulk Erase, the memory status may be checked by reading the Error Flag (DQ5) bit, the Toggle Flag (DQ6) bit, and the Data Polling Flag (DQ7) bit, as detailed in the section entitled “Programming Flash Memory”, on page 19. The Error Flag (DQ5) bit 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 device automatically does this before erasing to 0FFh.

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

The address provided with the Flash Bulk Erase command sequence (Table 5) may select any one of the eight internal Flash memory Sector Select signals FS0 - FS7 or one of the four signals CSBOOT0-CSBOOT3. An erase of that entire Flash memory array will occur even though the command was sent to just one Flash memory sector.

Flash Sector Erase. The Sector Erase instruction sequence uses six write operations, as described in Table 5. 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 100 µs. 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 (DQ3) bit. If the Erase Time-out Flag (DQ3) bit is 0, the Sector Erase instruction sequence has been received and the time-out period is counting. If the Erase Time-out Flag (DQ3) bit is 1, the time-out period has expired and the device is busy erasing the Flash memory sector(s). Before and during Erase time-out, any instruction sequence other than Suspend Sector Erase and Resume Sector Erase instruction sequences abort the cycle that is currently in progress, and reset the device to Read Array mode. It is not necessary to program the Flash memory sector with 00h as the device does this automatically before erasing (byte=FFh).

During a Sector Erase, the memory status may be checked by reading the Error Flag (DQ5) bit, the Toggle Flag (DQ6) bit, and the Data Polling Flag (DQ7) bit, as detailed in the section entitled “Programming Flash Memory”, on page 19.

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

The address provided with the initial Flash Sector Erase command sequence (Table 5) must select the first desired sector (FS0 - FS7 or CSBOOT0-CSBOOT3) to erase. Subsequent sector erase commands that are appended on within the time-out period must be addressed to other desired segments (FS0 - FS7 or CSBOOT0-CSBOOT3).

Suspend Sector Erase. When a Sector Erase cycle is in progress, the Suspend Sector Erase instruction sequence can be used to suspend the cycle by writing 0B0h to any address when an appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) is selected (See Table 5). 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 sequence executed during an Erase time-out period, in addition to suspending the Erase cycle, terminates the time out period.

The Toggle Flag (DQ6) bit stops toggling when the device 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 (DQ6) bit stops toggling between 0.1 µs and 15 µs after the Suspend Sector Erase instruction sequence has been executed. The device is then automatically set to Read mode.

If an Suspend Sector Erase instruction sequence 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 memory sector that was *not* being erased is valid.

– The Flash memory *cannot* be programmed, and only responds to Resume Sector Erase and Reset Flash instruction sequences (Read is an operation and is allowed).

– If a Reset Flash instruction sequence is received, data in the Flash memory sector that was being erased is invalid.

Resume Sector Erase. If a Suspend Sector Erase instruction sequence was previously executed, the erase cycle may be resumed with this instruction sequence. The Resume Sector Erase instruction sequence consists of writing 030h to any address while an appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) is active. (See Table 5.)

Flash Memory Sector Protect.

Each 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 PSDsoft Express.

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 DSP 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 DSP through the Main Flash memory protection register (in the *csiop* block) as defined in Table 7, and Secondary Flash memory protection register in Table 8.

Table 7. Main Flash Memory Protection Register Definition

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

Note: Bit Definitions:
Sec<i>_Prot 1 = Flash memory sector <i> is write protected.
Sec<i>_Prot 0 = Flash memory sector <i> is not write protected.

Table 8. Secondary Flash Memory Protection/Security Bit Register Definition

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

Note: Security_Bit = 1, device is secured.

Note: **Sec<i>_Prot** 1 = Flash memory sector <i> is write protected.
Sec<i>_Prot 0 = Flash memory sector <i> is not write protected.

DSM Security Bit

A programmable security bit in the DSM protects its contents from unauthorized viewing and copying. When set, the security bit will block access of programming devices (JTAG or others) to the DSM Flash memory and PLD configuration. The only way to defeat the security bit is to erase the entire DSM device, after which the device is blank and may be used again. The DSP will always have access to Flash memory contents through the 8-bit data port even while the security bit is set. The DSP can read the status of the security bit (but it cannot change it) by reading the Device Security register in the *csiop* block as defined in Table 8.

Reset Flash

The Reset Flash instruction sequence resets the internal memory logic state machine and puts Flash memory into Read Array mode. It consists of one write cycle (see Table 5). 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 (DQ5) bit to 1) during a Flash memory Program or Erase cycle.

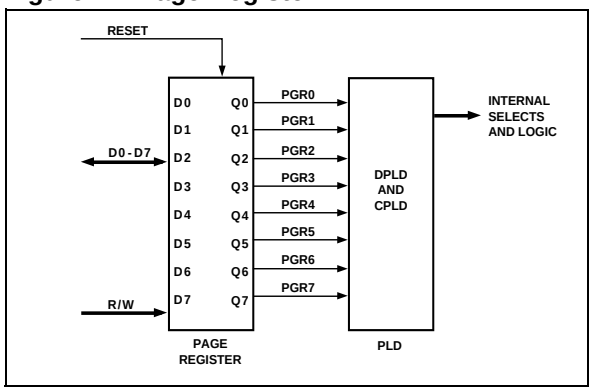
The Reset Flash instruction sequence puts the Flash memory back into normal Read Array mode. It may take the Flash memory up to a few milliseconds to complete the Reset cycle. The Reset Flash instruction sequence is ignored when it is issued during a Program or Bulk Erase cycle of the Flash memory. The Reset Flash instruction sequence aborts any on-going Sector Erase cycle, and returns the Flash memory to the normal Read Array mode within a few milliseconds.

Page Register

The 8-bit Page Register increases the addressing capability of the DSP by a factor of up to 256. The contents of the register can also be read by the DSP. The outputs of the Page Register (PG0-PG7) are inputs to the DPLD decoder and can be included in the Sector Select (*FS0-FS7 or CSBOOT0-CSBOOT3*) equations. See Figure 12.

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. The eight flip-flops in the register are connected to the internal data bus D0-D7. The DSP can write to or read from the Page Register. The Page Register can be accessed at address location *csiop* + E0h. Page Register outputs are cleared to logic 0 at reset.

Figure 12. Page Register



PLDs

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

The PLDs have selectable levels of performance and power consumption.

The device contains two PLDs: the Decode PLD (DPLD), and the Complex PLD (CPLD), as shown in Figure 13.

Table 9. DPLD and CPLD Inputs

Input Source	Input Name	Number of Signals
DSP Address Bus[1]	A15-A0	16
DSP Control Signals[2]	CNTL2-CNTL0	3
Reset	$\overline{\text{RST}}$	1
PortB Input Macrocells	PB7-PB0	8
PortC Input Macrocells	PC7-PC0	8
Port D Inputs	PD2-PD0	3
Page Register	PG7-PG0	8
Macrocell AB Feedback	MCELLAB FB7-0	8
Macrocell BC Feedback	MCELLBC FB7-0	8
Flash memory Program Status Bit	Ready/$\overline{\text{Busy}}$	1

Note: 1. DSP address lines A16, A17, and others may enter the DSM device on any pin on ports B, C, or D. See Figure 6 for recommended connections.

2. Additional DSP control signals may enter the DMS device on any pin on Ports B, C, or D. See Figure 6 for recommended connections.

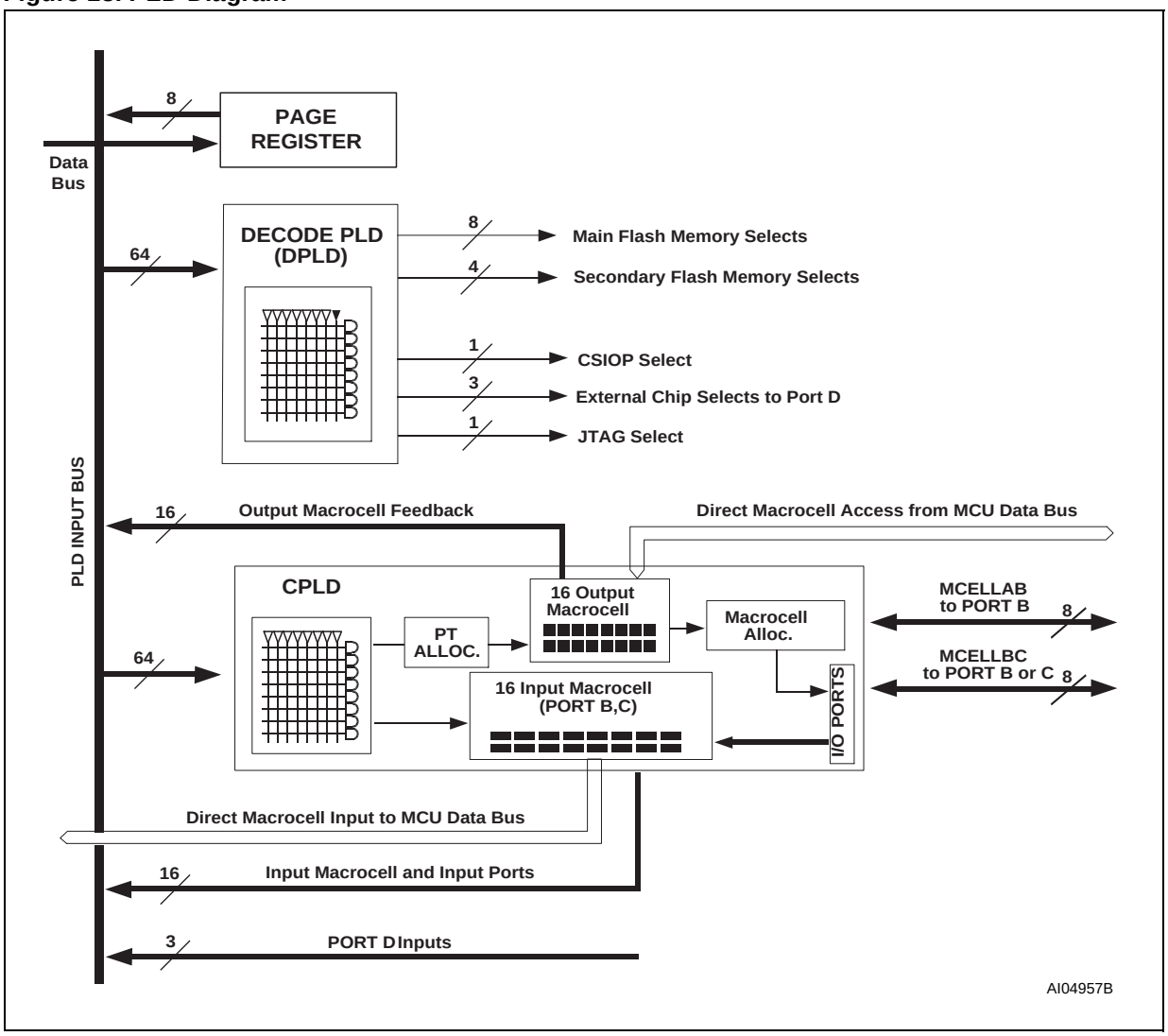
The DPLD performs address decoding, and generates select signals for internal and external components, such as memory, registers, and I/O ports. The DPLD can generates External Chip Select (ECS0-ECS2) signals on Port D.

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 16 Output Macrocells (OMC), 16 Input Macrocells (IMC), and the AND Array.

The AND Array is used to form product terms. These product terms are configured from the logic definition entered in PSDsoft Express. An Input Bus consisting of 64 signals is connected to the PLDs. Input signals are shown in Table 9.

Turbo Bit. The PLDs in the device can minimize power consumption by switching off when inputs remain unchanged for an extended time of about 70 ns. Resetting the Turbo bit to 0 (Bit 3 of the PMMR0 register) automatically places the PLDs into standby if no inputs are changing. Turning the Turbo mode off increases propagation delays while reducing power consumption. Additionally, five bits are available in the PMMR registers in *csiop* to block DSP control signals from entering the PLDs. This reduces power consumption and can be used only when these DSP 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 13. PLD Diagram

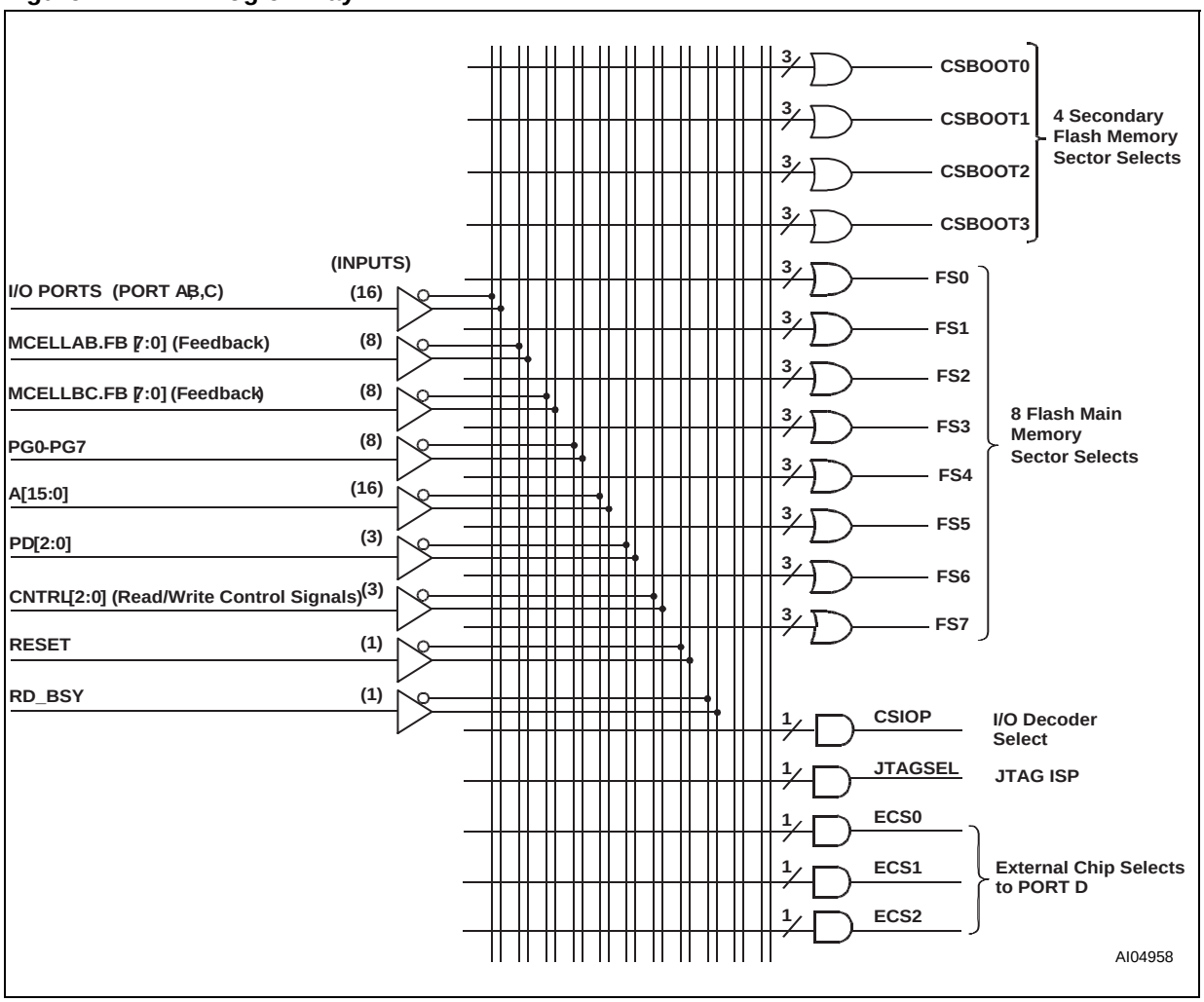


DECODE PLD (DPLD)

The DPLD, shown in Figure 14, is used for decoding the address for internal and external components. The DPLD can be used to generate the following decode signals:

- 8 Main Flash memory Sector Select (*FS0-FS7*) signals with three product terms each
- 4 Secondary Flash memory Sector Select (*CSBOOT0-CSBOOT3*) signals with three product terms each
- 1 internal *csiop* select for DSM device control and status registers (*csiop* is the base address of the block of 256 byte locations)
- 1 JTAG Select signal (enables JTAG operations on Port C when multiplexing JTAG signals with general I/O signals)
- 3 external chip select output signals for Port D pins, each with one product term.

Figure 14. DPLD Logic Array



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. See application note *AN1171* for details on how to specify logic using PSDsoft Express.

As shown in Figure 15, the CPLD has the following blocks:

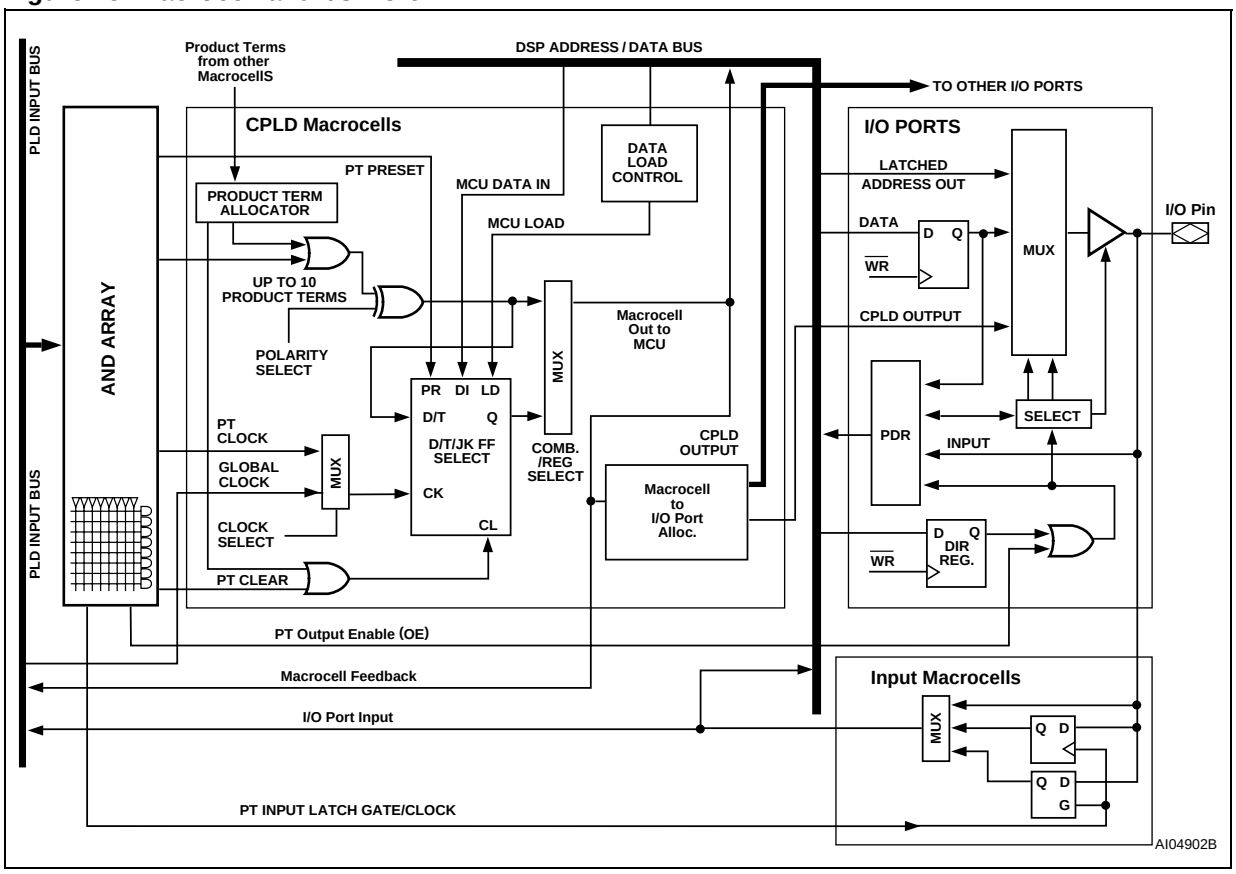
- 16 Input Macrocells (IMC)
- 16 Output Macrocells (OMC)
- Macrocell Allocator
- Product Term Allocator
- AND Array capable of generating up to 130 product terms
- Two I/O Ports.

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

The Input Macrocells (IMC) and Output Macrocells (OMC) are connected to the device internal data bus and can be directly accessed by the DSP. This enables the DSP 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 macro cell architectures.

Figure 15. Macrocell and I/O Port



Output Macrocell (OMC). Eight of the Output Macrocells (OMC) are connected to Port B pins and are named as McellAB0-McellAB7. The other eight Macrocells are connected to Ports B or C pins and are named as McellBC0-McellBC7. OMCs may be used for internal feedback only (buried registers), or their outputs may be routed to external Port pins.

The Output Macrocell (OMC) architecture is shown in Figure 17. 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 Express™. 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.

Output Macrocell Allocator. Outputs of the 16 OMCs can be routed to a combination of pins on Port B or Port D as shown in Figure 16. The OMC output pin is automatically determined by choosing pin functions in PSDsoft Express™. Routing can occur on a bit-by-bit basis, spitting assignment between the Ports. However, one OMC can be routed to one Port pin only, not both.

Figure 16. OMC Allocator

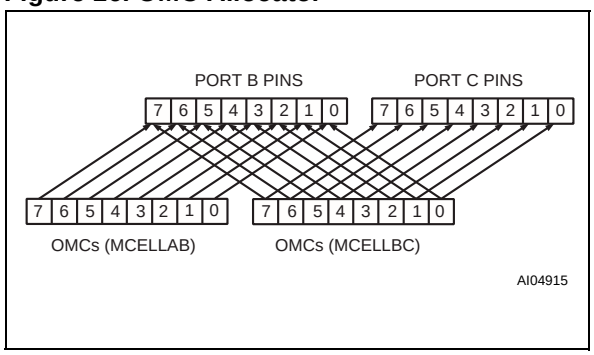


Table 10. Output Macrocell Port and Data Bit Assignments

Output Macrocell	Port Assignment	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 or C0	4	5	D0
McellBC1	Port B1 or C1	4	5	D1
McellBC2	Port B or, C2	4	5	D2
McellBC3	Port B3 orC3	4	5	D3
McellBC4	Port B4 orC4	4	6	D4
McellBC5	Port B5 or C5	4	6	D5
McellBC6	Port B6 orC6	4	6	D6
McellBC7	Port B7 orC7	4	6	D7

Product Term Allocator. The CPLD has a Product Term Allocator. PSDsoft Express™ uses the Product Term Allocator to borrow and place product terms from one Macrocell to another. This happens automatically in PSDsoft Express™, but understanding how allocation works will help you if your logic design does not "fit", in which case you may try selecting a different pin or different OMC where the allocation resources may differ and the design will then fit. 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. Product term allocation does not add any propagation delay to the logic.

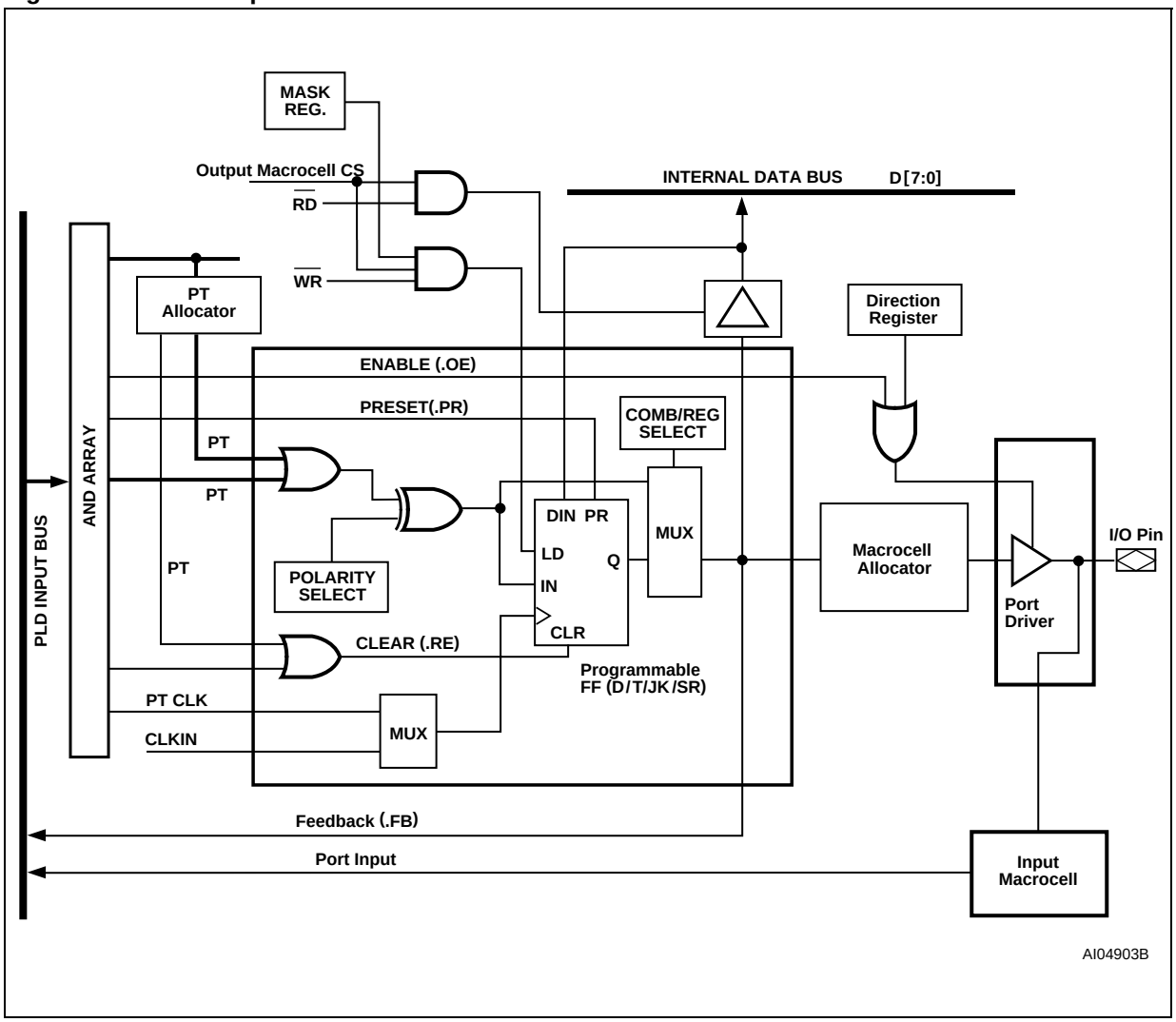
If an equation requires more product terms than are available to it through product term allocation, then “external” product terms are required, which consumes other Output Macrocells (OMC). This is called product term expansion and also happens automatically in PSDsoft Express™ as needed. Product tern expansion causes additional propagation delay because an OMC is consumed by the expansion and it’s output is rerouted (or fed back) into the AND array.

You can examine the fitter report generated by PSDsoft Express to see resulting product term allocation and product term expansion.

Loading and Reading the Output Macrocells (OMCs). Each of the two OMC blocks (8 OMCs each) occupies a memory location in the DSP address space, as defined in the *csiop* block MCELLAB0-7 and MCELLBC0-7 (see Table 4). The flip-flops in each of the 16 OMCs can be loaded from the data bus by a DSP. Loading the OMCs with data from the DSP takes priority over internal functions. As such, the preset, clear, and clock inputs to the flip-flop can be overridden by the DSP. 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 is loaded into the Output Macrocells (OMC) on the trailing edge of Write Strobe ($\overline{WR}$, CNTL0).

Figure 17. CPLD Output Macrocell

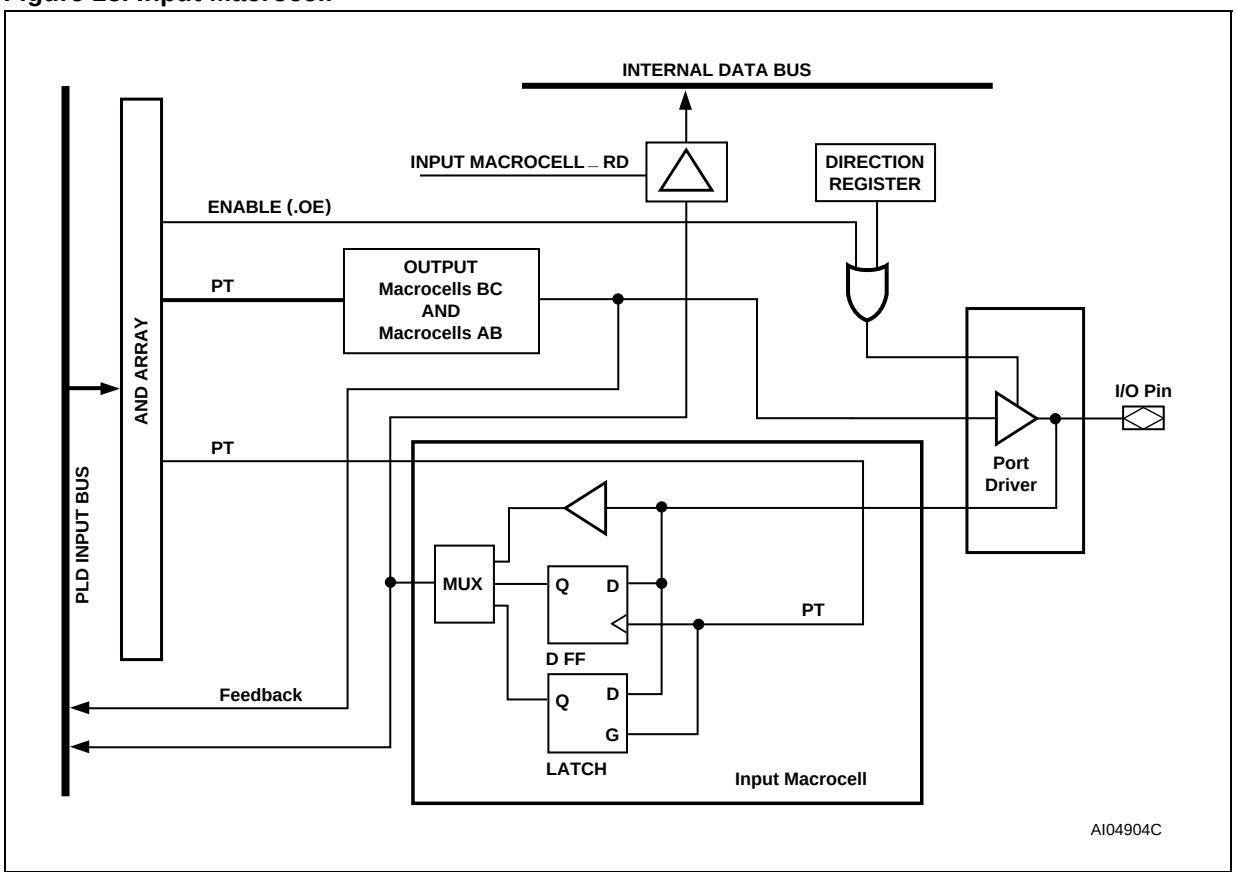


The OMC Mask Register. There is one Mask Register for each of the two groups of eight Output Macrocells (OMC). The Mask Registers can be used to block the loading of data to individual Output Macrocells (OMC). The default value for the Mask Registers is 00h, which allows loading of the Output Macrocells (OMC). When a given bit in a Mask Register is set to a 1, the DSP is blocked from writing to the associated Output Macrocells (OMC). For example, suppose McellAB0-3 are being used for a state machine. You would not want a DSP 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 specified as an internal node and not as a port pin output in the PSDsoft Express, then 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.

Figure 18. Input Macrocell



Input Macrocells (IMC). The CPLD has 16 Input Macrocells (IMC), one for each pin on Ports B and C. The architecture of the IMCs is shown in Figure 18. The IMCs are individually configurable, and can be used as a latch, a register, or to pass incoming Port signals prior to driving them onto the PLD input bus. This is useful for sampling and debouncing inputs to the AND array (keypad inputs, etc.). Additionally, the outputs of the IMCs can be read by the DSP asynchronously at any time through the internal data bus using the csiop register block (see Table 4).

The enable for the latch and clock for the register are driven by a product term from the CPLD. Each product term output is used to latch or clock four IMCs. Port inputs 3-0 can be controlled by one product term and 7-4 by another.

Configurations for the IMCs are specified by equations specified in PSDsoft Express. See Application note *AN1171*.

DSP Bus Interface

The “no-glue logic” DSP Bus Interface allows direct connection. DSP address, data, and control signals connect directly to the DSM device. See Figure 6 for typical connections.

DSP address, data and control signals are routed to Flash memory, I/O control (*csiop*), OMCs, and IMCs within the DMS. The DSP address range for each of these components is specified in PSDsoft Express™.

I/O Ports

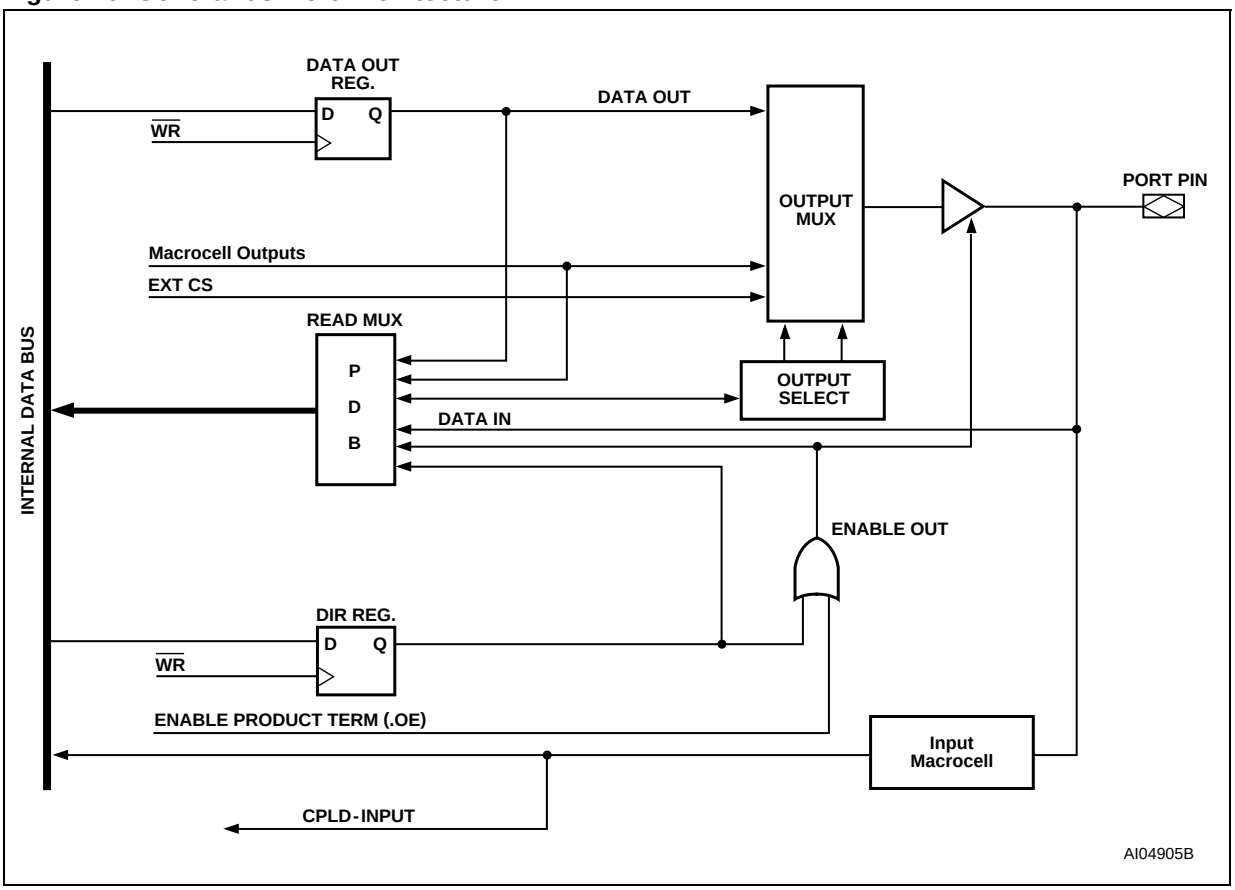
There are three programmable I/O ports: Ports B, C, and D. Each of the ports is eight bits except Port D, which is 3 bits. Each port pin is individually user configurable, thus allowing multiple functions per port. The ports are configured using PSDsoft Express™ or by the DSP writing to on-chip registers in the *csiop* block.

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 19. Individual Port architectures are shown in Figure 20 to Figure 23. In general, once the purpose for a port pin has been defined in PSDsoft Express™, that pin is no longer available for other purposes. Exceptions are noted.

Figure 19. General I/O Port Architecture



As shown in Figure 19, the ports contain an output multiplexer whose select signals are driven by the configuration bits determined by PSDsoft Express. Inputs to the multiplexer include the following:

- Output data from the Data Out register (for MCU I/O mode)
- CPLD Macrocell output (OMC)
- External Chip Selects ESC0-2 from the DPLD to Port D pins only.

The Port Data Buffer (PDB) is a tri-state buffer that allows only one source at a time to be read by the DSP. The Port Data Buffer (PDB) is connected to the Internal Data Bus for feedback and can be read by the DSP. The Data Out and Macrocell out-

puts, Direction Registers, and port pin input are all connected to the Port Data Buffer (PDB).

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 PSDsoft Express™, 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 DSP. The Port Data Buffer (PDB) feedback path allows the DSP to check the contents of the registers.

Ports B, and C have embedded IMCs. The IMCs can be configured as registers (for sampling or debouncing), as transparent latches, or direct inputs to the PLDs. The registers and latches are clocked by a product term from the PLD AND Array. The outputs from the IMCs drive the PLD input bus and can be read by the DSP. See the section entitled "Input Macrocell", on page 29.

Port Operating Modes

The I/O Ports have several modes of operation. Modes are defined using PSDsoft Express™, and then runtime control from the DSP can occur using the registers in the *csiop* block. See Application Note *AN1171* for more detail.

Table 11 summarizes which modes are available on each port. Each of the port operating modes are described in the following sections.

Table 11. Port Operating Modes

Port Mode	Port B	Port C	Port D
MCU I/O	Yes	Yes	Yes
PLD I/O			
McellAB Outputs	Yes	No	No
McellBC Outputs	Yes	Yes	No
Additional External CS Outputs	No	No	Yes
PLD Inputs	Yes	Yes	Yes
JTAG ISP	No	Yes[1]	No

Note: 1. Can be multiplexed with other I/O functions.

MCU I/O Mode. In the MCU I/O mode, the DSP uses the I/O Ports block to expand its own I/O ports. The DSP can read I/O pins, set the direction of I/O pins, and change the state of I/O pins by accessing the registers in the *csiop* block. The *csiop* register definition and their addresses may be found in Table 4.

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 DSP can read the port input through the Data In buffer. See Figure 19.

PLD I/O Mode. Inputs from Ports B and C to either PLD (DPLD or CPLD) come through IMCs. Inputs from Port D to either PLDs are routed directly in and do not use IMCs. Outputs from the CPLD to Port B come from the OMC group MCELLAB0-7. Outputs from the CPLD to Port C come from OMC group MCELLBC0-7. Outputs from the DPLD to Port D come from the external chip select logic block ECS0-2.

All PLD outputs may be tri-stated at the Port pins 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 logic 1 by the DSP if the pin is defined for a PLD input signal in PSDsoft Express. The PLD I/O mode is defined in PSDsoft Express by specifying PLD equations.

JTAG In-System Programming (ISP). Some of the pins on Port C are based on the IEEE 1194.1 JTAG specification and is used for In-System Programming (ISP). You can multiplex the function of these Port C JTAG pins with other functions. ISP is not performed very frequently in the life of the product, so multiplexing these pin's functions with general purpose I/O functions gives more utility from Port C. See the section entitled "Programming In-Circuit Using JTAG ISP", and Application Note *AN1153*.

Port Configuration Registers (PCR). Each Port has a set of Port Configuration Registers (PCR) used for configuration of the pins. The contents of the registers can be accessed by the DSP through normal read/write bus cycles of the *csiop* registers listed in Table 4.

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), are shown in Table 12. Default is logic 0.

Table 12. Port Configuration Registers (PCR)

Register Name	Port	DSP Access
Data In	B,C,D	Read
Data Out	B,C,D	Write/Read
Direction	B,C,D	Write/Read
Drive Select[1]	B,C,D	Write/Read

Note: 1. See Table 16 for Drive Register bit definition.

Data In Register. The DSP may read the Data In registers in the *csiop* block at any time to determine the logic state of a Port pin. This will be the state at the pin regardless of whether it is driven by a source external to the DSM or driven internally from the DSM device. Reading a logic zero for a bit in a Data In register means the corresponding Port pin is also at logic zero. Reading logic one means the pin is logic one. Each bit in a Data In register corresponds to an individual Port pin. For a given Port, bit 0 in a Data In register corresponds to pin 0 of the Port. Example, bit 0 of the Data In register for Port B corresponds to Port B pin PB0.

Data Out Register. The DSP may write (or read) the Data Out register in the *csiop* block at any time. Writing the Data Out register will change the logic state of a Port pin only if it is not driven or controlled by the CPLD. Writing a logic zero to a bit in a Data Out register will force the corresponding Port pin to be logic zero. Writing logic one will drive the pin to logic one. Each bit in the Data Out registers correspond to Port pins the same way as the Data In registers described above. When some pins of a Port are driven by the CPLD, writing to the corresponding bit in a Data Out register will have no effect as the CPLD overrides the Data Out 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.

Table 13. Port Pin Direction Control, Output Enable P.T. Not Defined

Direction Register Bit	Port Pin Mode
0	Input
1	Output

Table 14. 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 15. 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

Figure 20 and Figure 21 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 15. Since Port D only contains three pins (shown in Figure 23), the Direction Register for Port D has only the three least significant bits active.

Drive Select Register. The Drive Select Register configures the pin driver as Open Drain or CMOS (standard push/pull) 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. Open Drain outputs are diode clamped, thus the maximum voltage on an pin configured as Open Drain is Vcc + 0.7V.

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 that 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 standard slew.

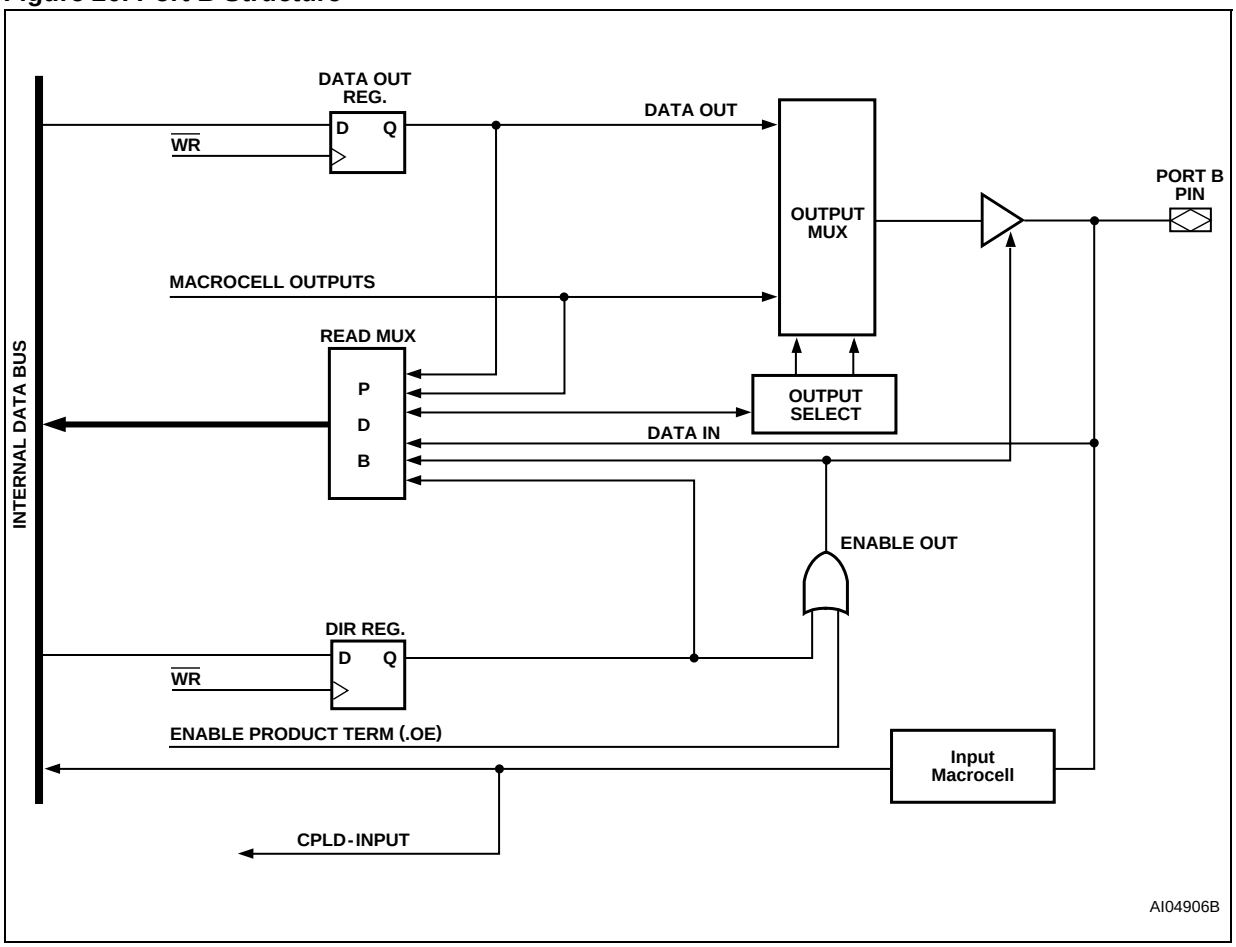
Table 16 shows the Drive Register for Ports B, C, and D. It summarizes which pins can be configured as Open Drain outputs and which pins the slew rate can be set for.

Table 16. 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	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain
Port D	NA[1]	NA[1]	NA[1]	NA[1]	NA[1]	Slew Rate	Slew Rate	Slew Rate

Note: 1. NA = Not Applicable.

Figure 20. Port B Structure

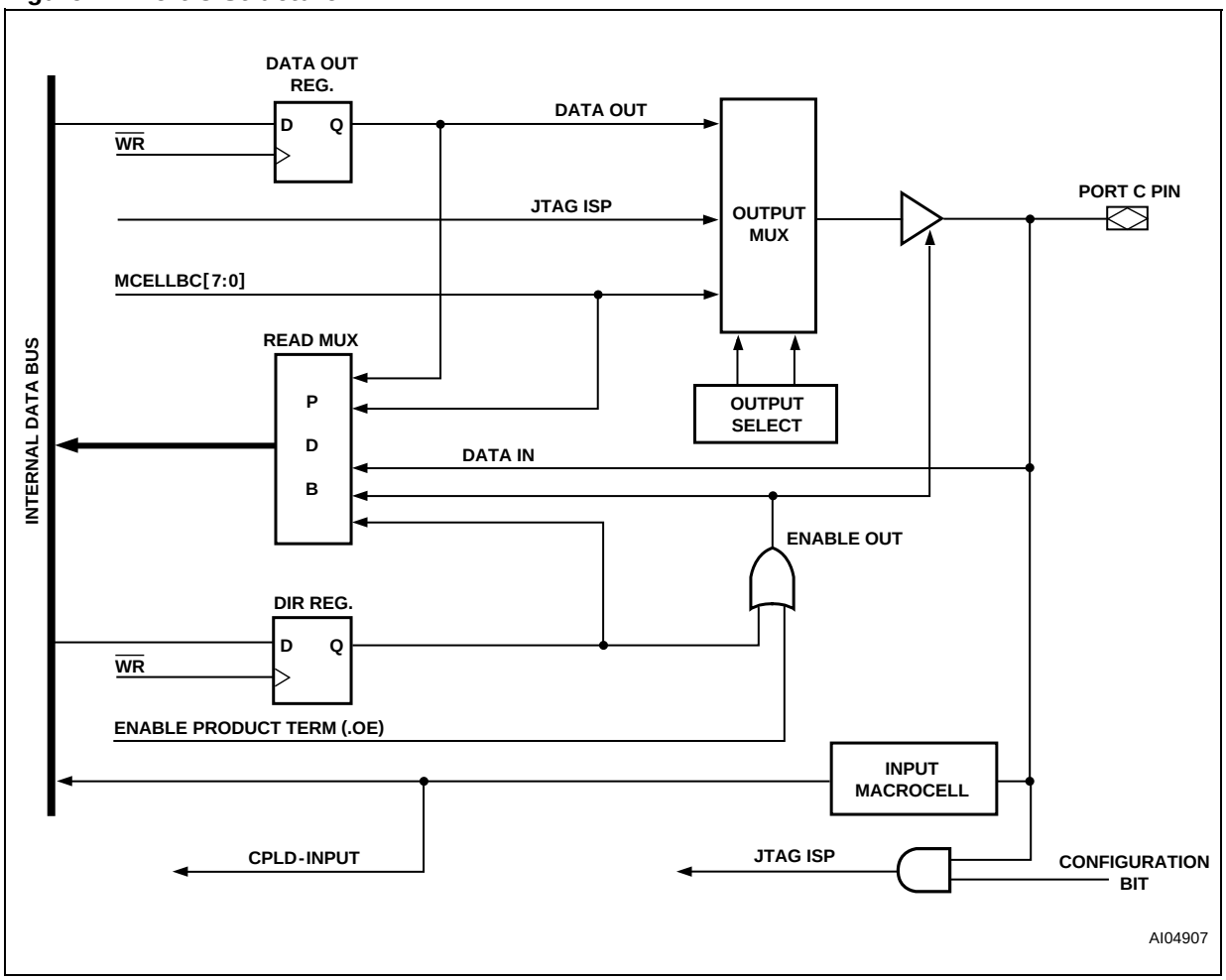


Port B – Functionality and Structure

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

- 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).
- 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 21. Port C Structure

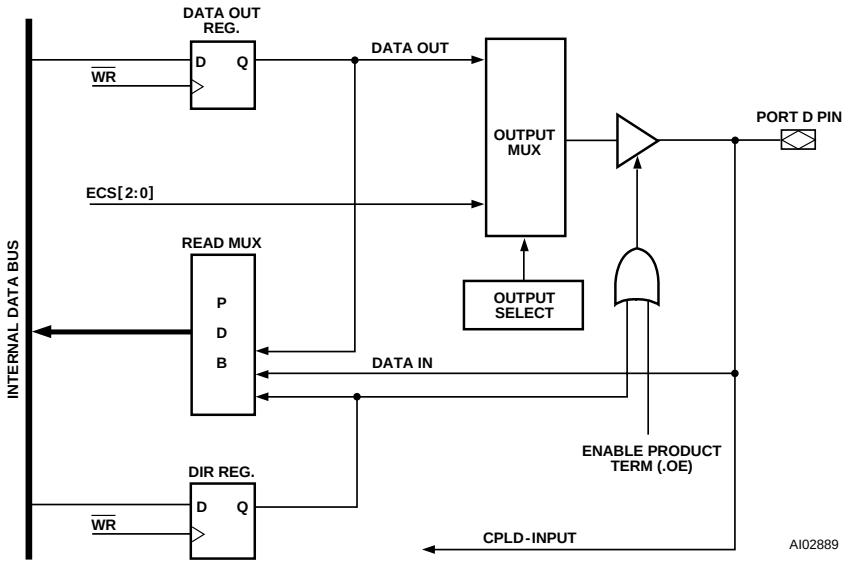


Port C – Functionality and Structure

Port C can be configured to perform one or more of the following functions (see Figure 21):

- 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 port can be enabled for programming/erase of the device. (See the section entitled “Programming In-Circuit Using JTAG ISP”, and Application Note *AN1153*, for more information on JTAG programming.)
- Open Drain – Port C pins can be configured in Open Drain Mode

Figure 22. Port D Structure



Port D – Functionality and Structure

Port D has three I/O pins. See Figure 22 and Figure 23. Port D can be configured to perform one or more of the following functions:

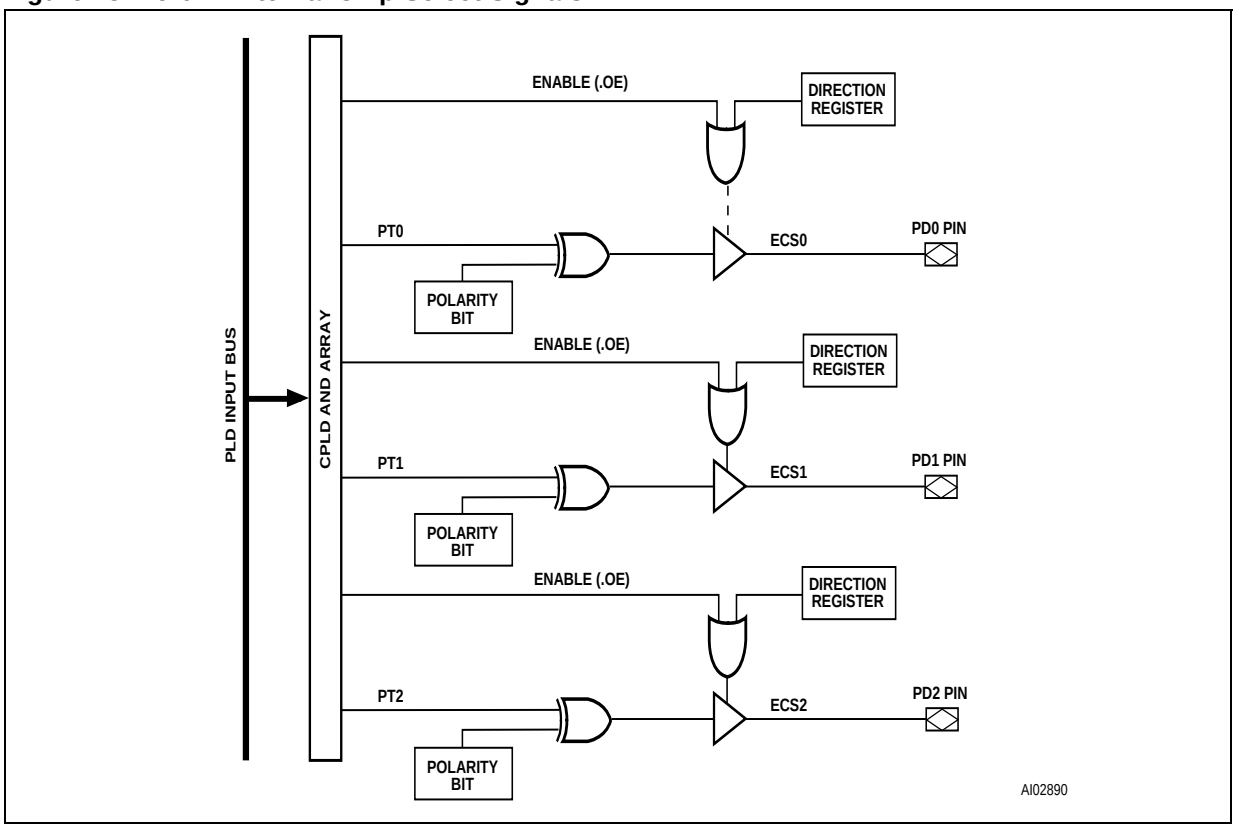
- MCU I/O Mode
- DPLD Output – External Chip Selects, ECS0-2 does not consume OMCs
- CPLD Input – direct input to the CPLD, does not use IMCs
- Slew rate – pins can be set up for fast slew rate

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

- CLKIN (PD1) as input to the OMCs Flip-flops
- PSD Chip Select Input ($\overline{CSI}$, PD2). Driving this signal logic High disables the Flash memory, putting it in standby mode.

External Chip Select. The DPLD also provides three External Chip Select outputs (ESC0-2) on Port D pins that can be used to select external devices as defined in PSDsoft Express. Each External Chip Select 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 23.) External Chip Selects for Port D pins do not consume OMCs. External chip select outputs can also come from the CPLD if chip select equations are specified in PSDsoft Express for Ports B or C.

Figure 23. Port D External Chip Select Signals



POWER MANAGEMENT

The device offers configurable power saving options. These options may be used individually or in combinations, as follows:

- All memory blocks in the device are built with zero-power management technology. Zero-power 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.

  Both PLDs (DPLD and CPLD) are also Zero-power, but this is not the default operation. The DSP must set a bit at run-time to achieve Zero-power as described next.
- The PMMR registers can be written by the DSP at run-time to manage power. The device has a Turbo bit in the PMMR0 register. 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.

  Further significant power savings can be achieved by blocking signals that are not used in DPLD or CPLD logic equations. The “blocking bits” in PMMR registers can be set to logic 1 by the DSP 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 25), so blocking unused PLD inputs can significantly lower PLD operating frequency and power consumption. The DSP also has the option of blocking certain PLD input when not needed, then letting them pass for when needed for specific logic operations. Table 17 and Table 18 define the PMMR registers.
- PSD Chip Select Input ($\overline{CSI}$, PD2) can be used to disable the internal memories and *csiop* registers, placing them in standby mode even if inputs are changing. This feature does not block any internal signals or disable the PLDs. There is a slight penalty in memory access time when PSD Chip Select Input ($\overline{CSI}$, PD2) makes its initial transition from deselected to selected.

Table 17. Power Management Mode Registers PMMR0[1]

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	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.
Bit 4	PLD Array clk	0 = on	CLKIN (PD1) input to the PLD AND Array is passed onto PLDs. Every change of CLKIN (PD1) Powers-up the PLD when Turbo bit is 0.
		1 = off	CLKIN (PD1) input to PLD AND Array is blocked, saving power.
Bit 5	PLD MCell clk	0 = on	CLKIN (PD1) input to the PLD Macrocells is passed onto PLDs.
		1 = off	CLKIN (PD1) input to PLD Macrocells is blocked, 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: 1. The bits of this register are cleared to zero following Power-up. Subsequent Reset ($\overline{Reset}$) pulses do not clear the registers.

PLD Power Management

The power and speed of the PLDs are controlled by the Turbo bit (bit 3) in the PMMR0. By setting the bit to 1, the Turbo mode is off and the PLDs consume the specified stand-by current when the inputs are not switching for an extended time of 70 ns (100 ns for 3.3 V devices). The propagation delay time is increased by 10 ns after the Turbo bit is set to 1 (turned off) when the inputs change at a composite frequency of less than 15 MHz (10 MHz for 3.3 V devices). 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.

Blocking MCU control signals with the bits of the PMMR registers can further reduce PLD AC power consumption by lowering the effective composite frequency of inputs to the PLDs.

Table 18. Power Management Mode Registers PMMR2[1]

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 CNTL0	0 = on	Cntl0 input to the PLD AND Array is passed onto PLDs.
		1 = off	Cntl0 input to PLD AND Array is blocked, saving power.
Bit 3	PLD Array CNTL1	0 = on	Cntl1 input to the PLD AND Array is passed onto PLDs.
		1 = off	Cntl1 input to PLD AND Array is blocked, saving power.
Bit 4	PLD Array CNTL2	0 = on	Cntl2 input to the PLD AND Array is passed onto PLDs.
		1 = off	Cntl2 input to PLD AND Array is blocked, saving power.
Bit 5	PLD Array PD0	0 = on	PD0 input to the PLD AND Array is passed onto PLDs.
		1 = off	PD0 input to PLD AND Array is blocked, saving power.
Bit 6	PLD Array PC7	0 = on	PC7 input to the PLD AND Array is passed onto PLDs.
		1 = off	PC7 input to PLD AND Array is blocked, saving power.
Bit 7	X	0	Not used, and should be set to zero.

Note: 1. The bits of this register are cleared to zero following Power-up. Subsequent Reset ($\overline{\text{Reset}}$) pulses do not clear the registers.

PSD Chip Select Input ($\overline{\text{CSI}}$, PD2)

PD2 of Port D can be configured in PSDsoft Express as PSD Chip Select Input ($\overline{\text{CSI}}$). When Low, the signal selects and enables the internal Flash memory and I/O blocks for Read or Write operations involving the device. A High on PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) disables the Flash memory and reduces the device power consumption. However, the PLD and I/O signals remain operational when PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) is High.

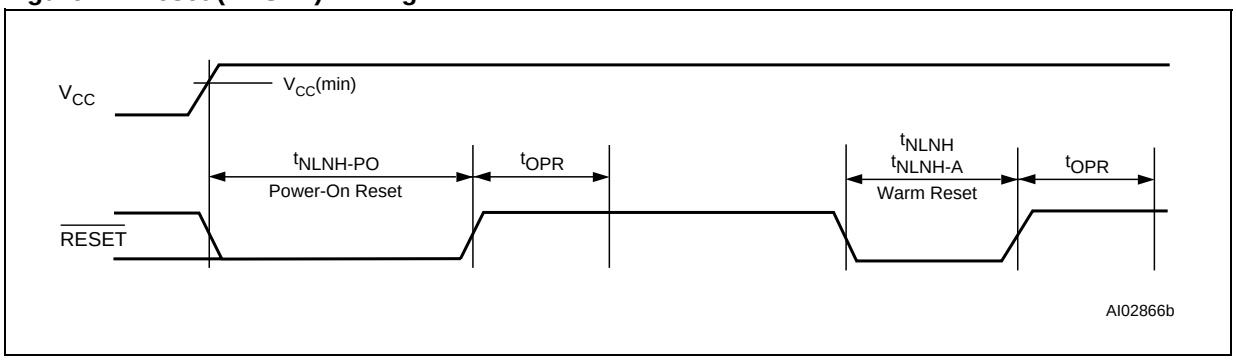
There may be a timing penalty when using PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) depending on the speed grade of the device that you are using. See the timing parameter t_{SLQV} in Table 31.

Input Clock. The device provides the option to block CLKIN (PD1) from reaching the PLDs to save AC power consumption. CLKIN (PD1) is an input to the PLD AND Array and the OMCs.

If CLKIN (PD1) is not being used as part of the PLD logic equation, the clock should be blocked 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 device provides the option to block the input control signals (CNTL0, CNTL1, CNTL2, PD0, and PC7) from reaching the PLDs to save AC power consumption. These control signals are inputs to the PLD AND Array. If any of these 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 the PMMR2 register. Note: CNTL0 and CNTL1 (DSP $\overline{\text{WR}}$ and DSP $\overline{\text{RD}}$) are permanently routed to the Flash memory array and cannot be blocked from the array by the PMMR registers (that's why WR and RD signals do not have to be specified in PSDsoft Express for Flash memory segment chip-select equations for FS0 - FS7). CNTL0 and CNTL1 are blocked from the PLDs with PMMR registers bits when these signals are specifically used in logic equations specified in PSDsoft Express.

Figure 24. Reset ($\overline{RESET}$) Timing



Power On Reset, Warm Reset, Power-down

Power On Reset. Upon Power-up, the device requires a Reset ($\overline{RESET}$) pulse of duration $t_{NLNH-PO}$ after V_{CC} is steady. During this time 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 ($\overline{RESET}$), the device 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 Array mode upon Power-up. Sector Select FS0-FS7 must all be Low, Write Strobe ($\overline{WR}$, 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 ($\overline{WR}$, CNTL0). Any Flash memory Write cycle initiation is prevented automatically when V_{CC} is below V_{LKO}.

Table 19. 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)

Register	Power-On Reset	Warm Reset	Power-down Mode
PMMR0 and PMMR2	Cleared to 0	Unchanged	Unchanged
OMC Flip-flop status	Cleared to 0 by internal Power-On Reset	Depends on .re and .pr equations	Depends on .re and .pr equations
All other registers	Cleared to 0	Cleared to 0	Unchanged

Warm Reset. Once the device is up and running, the device 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 warm reset. Figure 24 shows the timing of the Power-up and warm reset.

I/O Pin, Register and PLD Status at Reset. Table 19 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 device Configuration bits are loaded. This loading of the device 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 PSDsoft Express equations.

PROGRAMMING IN-CIRCUIT USING JTAG ISP

In-System Programming (ISP) can be performed through the JTAG signals on Port C. This serial interface allows programming of the entire DSM device or subsections (i.e. only Flash memory but not the PLDs) without and participation of the DSP. A blank DSM device soldered to a circuit board can be completely programmed in 10 to 25 seconds. The basic JTAG signals; TMS, TCK, TDI, and TDO form the IEEE-1149.1 interface. The DSM device does not implement the IEEE-1149.1 Boundary Scan functions. The DSM uses the JTAG interface for ISP only. However, the DSM device can reside in a standard JTAG chain with other JTAG devices as it will remain in BYPASS mode while other devices perform Boundary Scan.

ISP programming time can be reduced as much as 30% by using two more signals on Port C, TSTAT and $\overline{\text{TERR}}$ in addition to TMS, TCK, TDI and TDO. See Table 20. The FlashLINK™ JTAG programming cable available from STMicroelectronics for $59USD and PSDsoft Express software that is available at no charge from *www.st.com/psm* is all that is needed to program a DSM device using the parallel port on any PC or laptop.

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

See Application Note *AN1153* for more details on JTAG In-System Programming (ISP).

Standard JTAG Signals. The standard JTAG signals (TMS, TCK, TDI, and TDO) can be enabled by any of three different conditions that are logically ORed.

The following symbolic logic equation specifies the conditions enabling the four basic JTAG signals (TMS, TCK, TDI, and TDO) on their respective Port C pins. For purposes of discussion, the logic label JTAG_ON is used. When JTAG_ON is true, the four pins are enabled for JTAG operation. When JTAG_ON is false, the four pins can be used for general device I/O as specified in PSDsoft Express. JTAG_ON can become true by any of three different ways as shown:

JTAG_ON =

1. PSDsoft Express Pin Configuration -OR-
2. PSDsoft Express PLD equation -OR-
3. DSP writes to register in *csiop* block

Method 1 is most common. This is when the JTAG pins are selected in PSDsoft Express to be “dedicated” JTAG pins. They can always transmit and receive JTAG information because they are “full-time” JTAG pins.

Method 2 is used only when the JTAG pins are multiplexed with general I/O functions. For designs that need every I/O pin, the JTAG pins may be used for general I/O when they are not used for ISP. However, when JTAG pins are multiplexed with general I/O functions, the designer must include a way to get the pins back into JTAG mode when it is time for JTAG operations again. In this case, a single PLD input from Ports B, C, or D must be dedicated to switch the Port C pins from I/O mode back to ISP mode at any time. It is recommended to physically connect this dedicated PLD input pin to the JEN\ output signal from the Flashlink cable when multiplexing JTAG signals. See Application Note *AN1153* for details.

Method 3 is rarely used to control JTAG pin operation. The DSP can set the port C pins to function as JTAG ISP by setting the JTAG Enable bit in a register of the *csiop* block, but as soon as the DSM chip is reset, the csiop block registers are cleared, which turns off the JTAG-ISP function. Controlling JTAG pins using this method is not recommended.

Table 20. JTAG Port Signals

Port C Pin	JTAG Signals	Description
PC0	TMS	Mode Select
PC1	TCK	Clock
PC3	$\overline{\text{TSTAT}}$	Status
PC4	$\overline{\text{TERR}}$	Error Flag
PC5	TDI	Serial Data In
PC6	TDO	Serial Data Out

JTAG Extensions. $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ are two JTAG extension signals (must be used as a pair) enabled by a command received over the four standard JTAG signals (TMS, TCK, TDI, and TDO) by PSDsoft Express. They are used to speed Program and Erase cycles by indicating status on device pins 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.

$\overline{\text{TSTAT}}$ behaves the same as Ready/$\overline{\text{Busy}}$ described previously. $\overline{\text{TSTAT}}$ is inactive logic 1 when the device is in Read mode (Flash memory contents can be read). $\overline{\text{TSTAT}}$ is logic 0 when Flash memory Program or Erase cycles are in progress.

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ can be configured as open-drain type signals with PSDsoft Express. This facilitates a wired-OR connection of $\overline{\text{TSTAT}}$ signals

from multiple DSM2190F4V devices and a wired-OR connection of TERR signals from those same devices. This is useful when several devices are "chained" together in a JTAG environment. PSDsoft Express puts TSTAT and TERR signals to open-drain by default. Click on 'Properties' in the JTAG-ISP window of PSDsoft Express to change to standard CMOS push-pull. It is recommended to use 10 kΩ pull-up resistors to V_{CC} on all JTAG-ISP signals on your circuit board.

Initial Delivery State

When delivered from ST, the device has all bits in the memory and PLDs erased to logic 1. The DSM Configuration Register bits are set to 0. The code, configuration, and PLD logic are loaded using the programming procedure. The four basic JTAG ISP signals (TCK, TMS, TDI, TDO) are ready for ISP function.

AC/DC PARAMETERS

These tables describe the AC and DC parameters of the device:

❏ DC Electrical Specification

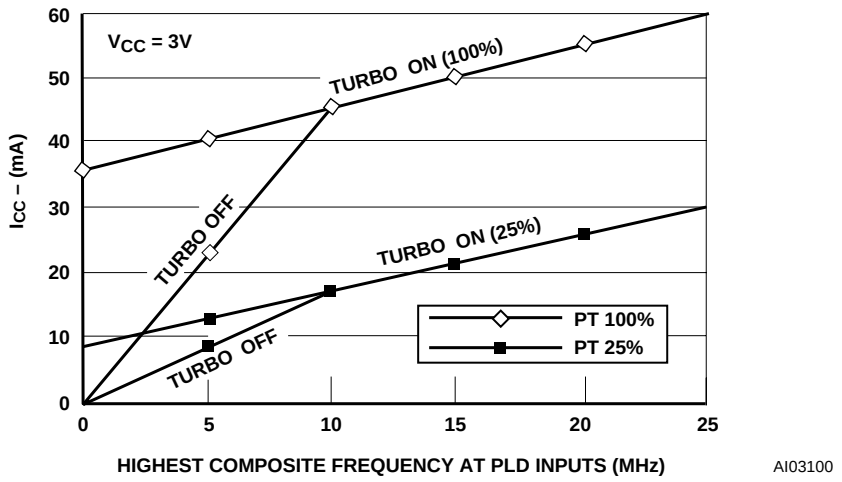
❏ AC Timing Specification

- PLD Timing
  - Combinatorial Timing
  - Synchronous Clock Mode
  - Asynchronous Clock Mode
  - Input Macrocell Timing
- DSP Timing
  - Read Timing
  - Write Timing
  - Reset Timing

The following are issues concerning the parameters presented:

- In the DC specification the supply current is given for different modes of operation. Before calculating the total power consumption, determine the percentage of time that the device is in each mode. Also, the supply power is considerably different if the Turbo bit is 0.
- The AC power component gives the PLD and Flash memory a mA/MHz specification. Figure 25 shows the PLD mA/MHz as a function of the number of Product Terms (PT) used.
- The fitter report of PSDsoft Express indicates the number of Product Terms (PTs) used for a given design. This number may be used to estimate PLD power consumption using Figure 25.
- In the PLD timing parameters, add the required delay when Turbo bit is 0.

Figure 25. PLD I_{CC} /Frequency Consumption (3.3 V)



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 21. 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)	–0.6	7.0	V
V_{CC}	Supply Voltage	–0.6	7.0	V
V_{PP}	Device Programmer Supply Voltage	–0.6	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=100 pF, 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 22. Operating Conditions

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply Voltage	3.0	3.6	V
T_A	Ambient Operating Temperature (industrial)	–40	85	°C

Table 23. AC Measurement Conditions

Symbol	Parameter	Min.	Max.	Unit
C_L	Load Capacitance	30		pF
	Input Rise and Fall Times		5	ns
	Input Pulse Voltages	1.5		V
	Input and Output Timing Reference Voltages	1.5		V

Note: 1. Output Hi-Z is defined as the point where data out is no longer driven.

Figure 26. AC Measurement I/O Waveform

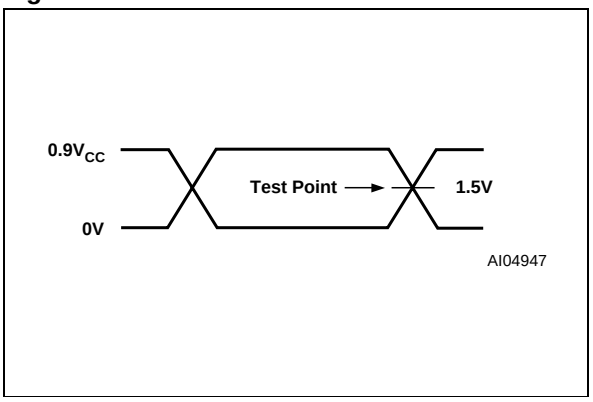


Figure 27. AC Measurement Load Circuit

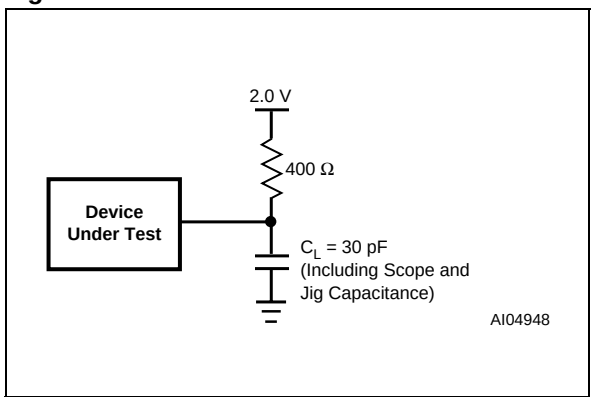


Table 24. Capacitance

Symbol	Parameter	Test Condition	Typ.[2]	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
C_{VPP}	Capacitance (for CNTL2/V_{PP})	V_{PP} = 0V	18	25	pF

Note: 1. Sampled only, not 100% tested.
2. Typical values are for T_A = 25°C and nominal supply voltages.

Table 25. AC Symbols for PLD Timing

Signal Letters	
A	Address Input
C	CEout Output
D	Input Data
E	E Input
N	Reset Input or Output
P	Port Signal Output
Q	Output Data
R	$\overline{RD}$ Input (read)
S	Chip Select Input, $\overline{BMS}$, $\overline{DMS}$, $\overline{IOMS}$, or FSx
W	$\overline{WR}$ Input (write)
B	V_{STBY} Output
M	Output Macrocell

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

Example: t_{AVWL} – Time from Address Valid to Write input Low.

Figure 28. Switching Waveforms – Key

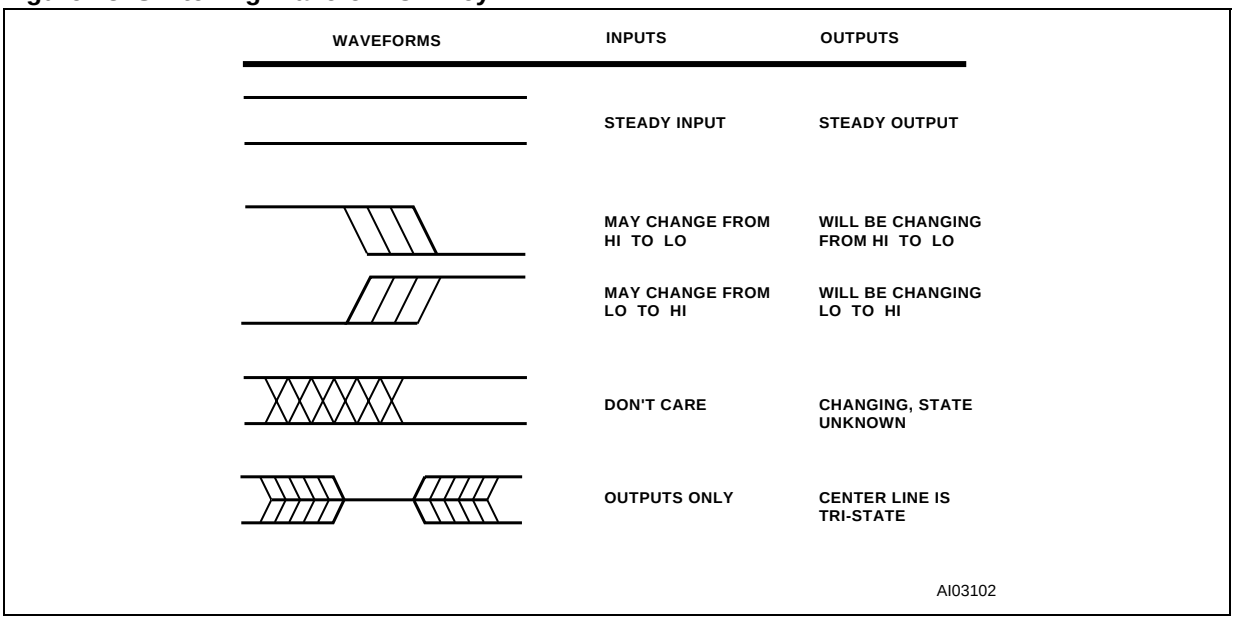


Table 26. DC Characteristics

Symbol	Parameter		Conditions	Min.	Typ.	Max.	Unit
V_{IH}	High Level Input Voltage		3.0 V < V_{CC} < 3.6 V	0.7V_{CC}		V_{CC} +0.5	V
V_{IL}	Low Level Input Voltage		3.0 V < V_{CC} < 3.6 V	–0.5		0.8	V
V_{IH1}	Reset High Level Input Voltage		(Note 1)	0.8V_{CC}		V_{CC} +0.5	V
V_{IL1}	Reset Low Level Input Voltage		(Note 1)	–0.5		0.2V_{CC} –0.1	V
V_{HYS}	Reset Pin Hysteresis			0.3			V
V_{LKO}	V_{CC} (min) for Flash Erase and Program			1.5		2.2	V
V_{OL}	Output Low Voltage		I_{OL} = 20 µA, V_{CC} = 3.0V		0.01	0.1	V
			I_{OL} = 4 mA, V_{CC} = 3.0 V		0.15	0.45	V
V_{OH}	Output High Voltage Except V_{STBY} On		I_{OH} = –20 µA, V_{CC} = 3.0 V	2.9	2.99		V
			I_{OH} = –2 mA, V_{CC} = 3.0 V	2.7	2.6		V
V_{OH1}	Output High Voltage V_{STBY} On		I_{OH1} = 1 µA	V_{STBY} – 0.8			V
I_{IDLE}	Idle Current (VSTBY input)		V_{CC} > V_{STBY}	–0.1		0.1	µA
V_{DF}	SRAM Data Retention Voltage		Only on V_{STBY}	2			V
I_{SB}	Stand-by Supply Current for Power-down Mode		$\overline{CSI}$ >V_{CC} –0.3 V (Notes 2,3)		25	100	µA
I_{LI}	Input Leakage Current		V_{SS} < V_{IN} < V_{CC}	–1	±.1	1	µA
I_{LO}	Output Leakage Current		0.45 < V_{IN} < V_{CC}	–10	±5	10	µA
I_{CC} (DC) (Note 5)	Operating Supply Current	PLD Only	PLD_TURBO = Off, f = 0 MHz (Note 3)		0		µA/PT
			PLD_TURBO = On, f = 0 MHz		200	400	µA/PT
		Flash memory	During Flash memory Write/ Erase Only		10	25	mA
			Read Only, f = 0 MHz		0	0	mA
I_{CC} (AC) (Note 5)	PLD AC Adder			(see note 4)			
	Flash memory AC Adder				1.5	2.0	mA/ MHz

Note: 1. Reset ($\overline{Reset}$) has hysteresis. V_{IL1} is valid at or below 0.2V_{CC} –0.1. V_{IH1} is valid at or above 0.8V_{CC} .
2. $\overline{CSI}$ deselected.
3. PLD is in non-Turbo mode, and none of the inputs are switching.
4. Please see Figure 25 for the PLD current calculation.
5. I_{OUT} = 0 mA

Table 27. CPLD Combinatorial Timing

Symbol	Parameter	Conditions	-15		PT Aloc	Turbo Off	Slew Rate[1]	Unit
			Min	Max				
t_{PD}	CPLD Input Pin/Feedback to CPLD Combinatorial Output			45	Add 4	Add 20	Sub 6	ns
t_{EA}	CPLD Input to CPLD Output Enable			45		Add 20	Sub 6	ns
t_{ER}	CPLD Input to CPLD Output Disable			45		Add 20	Sub 6	ns
t_{ARP}	CPLD Register Clear or Preset Delay			43		Add 20	Sub 6	ns
t_{ARPW}	CPLD Register Clear or Preset Pulse Width		30			Add 20		ns
t_{ARD}	CPLD Array Delay	Any Macrocell		29	Add 4			ns

Note: 1. Fast Slew Rate output available on PB3-PB0, and PD2-PD0.

Table 28. CPLD Macrocell Synchronous Clock Mode Timing

Symbol	Parameter	Conditions	-15		PT Aloc	Turbo Off	Slew Rate[1]	Unit
			Min	Max				
f_{MAX}	Maximum Frequency External Feedback	$1/(t_S+t_{CO})$		18.8				MHz
	Maximum Frequency Internal Feedback (f_{CNT})	$1/(t_S+t_{CO}-10)$		23.2				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CH}+t_{CL})$		33.3				MHz
t_S	Input Setup Time		25		Add 4	Add 20		ns
t_H	Input Hold Time		0					ns
t_{CH}	Clock High Time	Clock Input	15					ns
t_{CL}	Clock Low Time	Clock Input	15					ns
t_{CO}	Clock to Output Delay	Clock Input		28			Sub 6	ns
t_{ARD}	CPLD Array Delay	Any Macrocell		29	Add 4			ns
t_{MIN}	Minimum Clock Period[2]	$t_{CH}+t_{CL}$	29					ns

Note: 1. Fast Slew Rate output available on PB3-PB0, and PD2-PD0.
2. CLKIN (PD1) $t_{CLCL} = t_{CH} + t_{CL}$.

Table 29. CPLD Macrocell Asynchronous Clock Mode Timing

Symbol	Parameter	Conditions	-15		PT Aloc	Turbo Off	Slew Rate	Unit
			Min	Max				
f_{MAXA}	Maximum Frequency External Feedback	$1/(t_{SA}+t_{COA})$		19.2				MHz
	Maximum Frequency Internal Feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		23.8				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CHA}+t_{CLA})$		27				MHz
t_{SA}	Input Setup Time		12		Add 4	Add 20		ns
t_{HA}	Input Hold Time		15					ns
t_{CHA}	Clock High Time		22			Add 20		ns
t_{CLA}	Clock Low Time		15			Add 20		ns
t_{COA}	Clock to Output Delay			40		Add 20	Sub 6	ns
t_{ARD}	CPLD Array Delay	Any Macrocell		29	Add 4			ns
t_{MINA}	Minimum Clock Period	$1/f_{CNTA}$	42					ns

Figure 29. Input to Output Disable / Enable

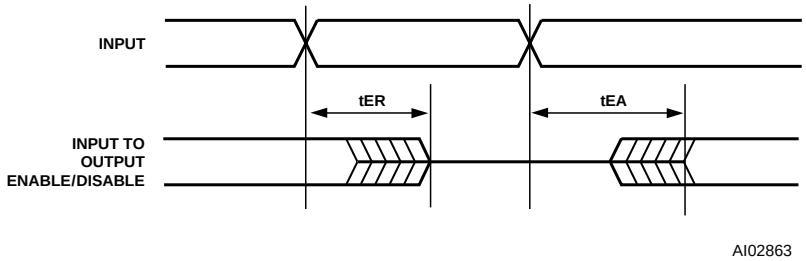


Figure 30. Asynchronous Reset / Preset

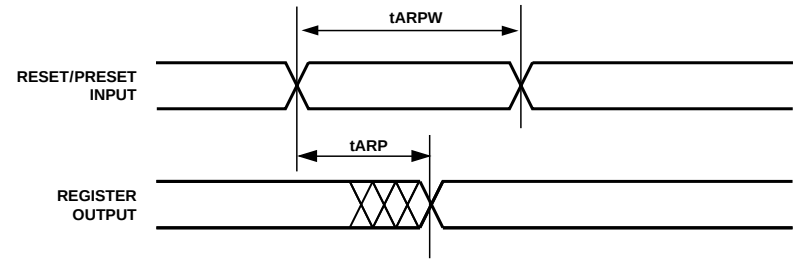


Figure 31. Synchronous Clock Mode Timing – PLD

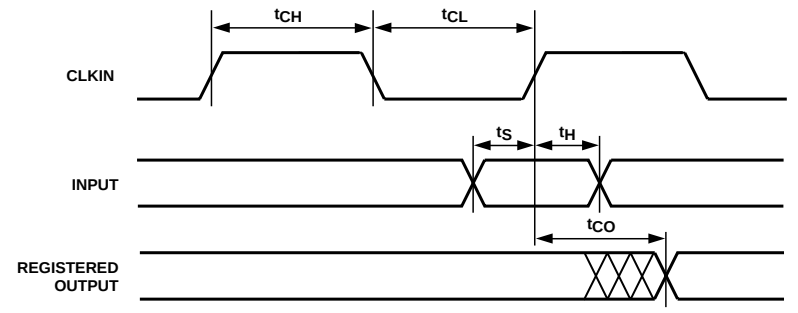


Figure 32. Asynchronous Clock Mode Timing (product term clock)

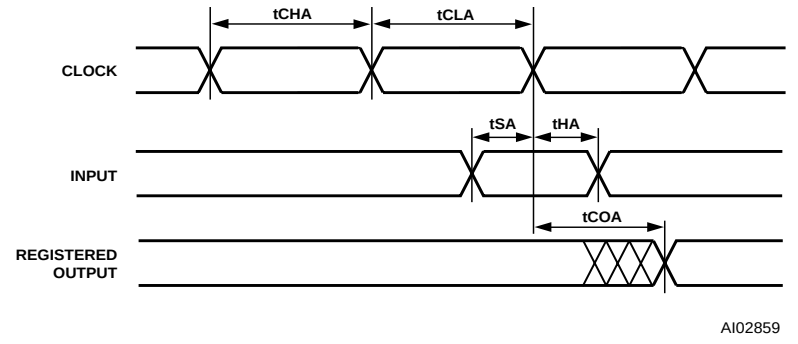


Table 30. Input Macrocell Timing

Symbol	Parameter	Conditions	-15		PT Aloc	Turbo Off	Unit
			Min	Max			
t_{IS}	Input Setup Time	(Note 1)	0				ns
t_{IH}	Input Hold Time	(Note 1)	25			Add 20	ns
t_{INH}	NIB Input High Time	(Note 1)	13				ns
t_{INL}	NIB Input Low Time	(Note 1)	13				ns
t_{INO}	NIB Input to Combinatorial Delay	(Note 1)		62	Add 4	Add 20	ns

Note: 1. Inputs from Port B, and C relative to register/latch clock from the PLD.

Figure 33. Input Macrocell Timing (product term clock)

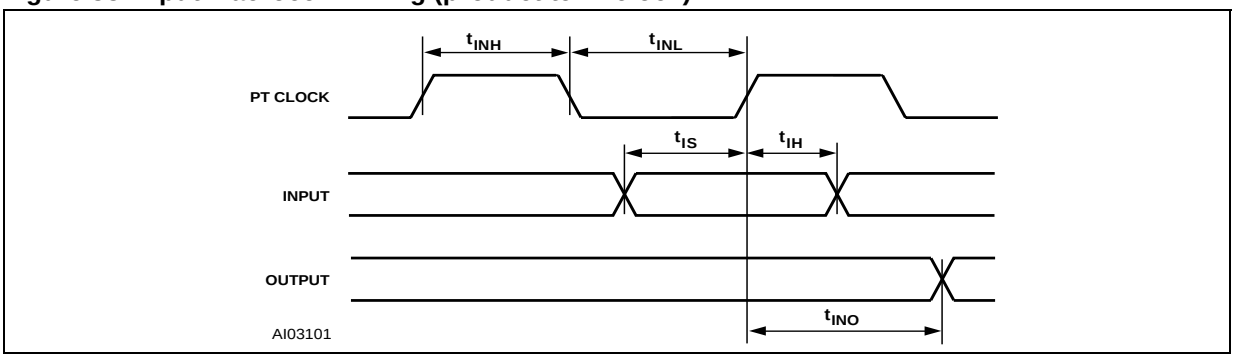


Table 31. Read Timing

Symbol	Parameter	Conditions	-15		Turbo Off	Unit
			Min	Max		
t_{AVQV}	Address Valid to Data Valid	(Note 1)		150	Add 20	ns
t_{SLQV}	CS Valid to Data Valid			150		ns
t_{RLQV}	$\overline{RD}$ to Data Valid 8-Bit Bus			35		ns
t_{RHQX}	$\overline{RD}$ Data Hold Time		1			ns
t_{RLRH}	$\overline{RD}$ Pulse Width		40			ns
t_{RHQZ}	$\overline{RD}$ to Data High-Z			20		ns

Note: 1. Any input used to select an internal DSM function.

Figure 34. Read Timing

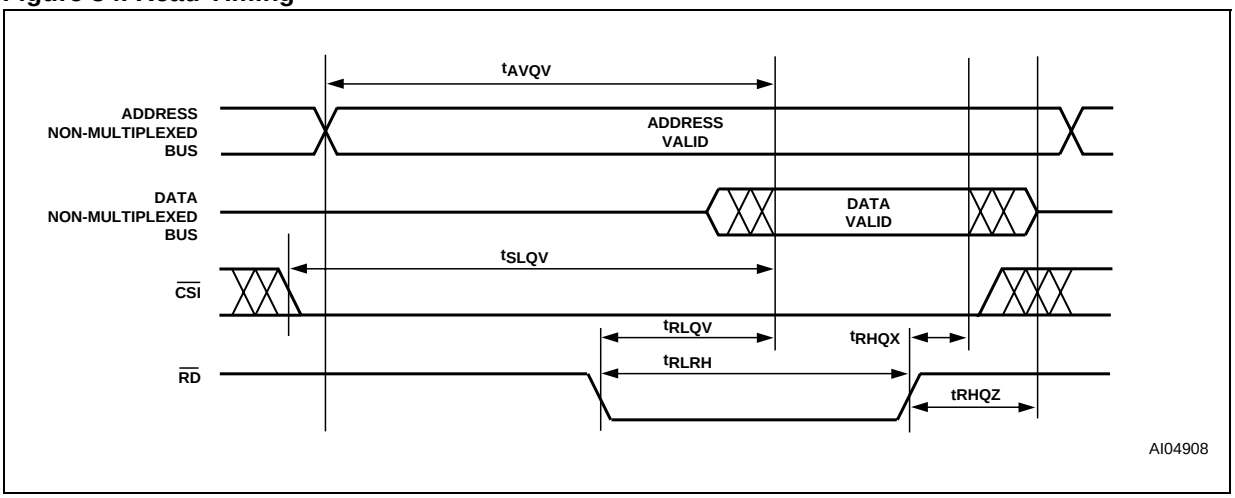


Table 32. Write Timing

Symbol	Parameter	Conditions	-15		Unit
			Min	Max	
t_{AVWL}	Address Valid to Leading Edge of $\overline{WR}$	(Notes [1])	0		ns
t_{SLWL}	$\overline{CS}$ Valid to Leading Edge of $\overline{WR}$		0		ns
t_{DVWH}	$\overline{WR}$ Data Setup Time		45		ns
t_{WHDX}	$\overline{WR}$ Data Hold Time		2		ns
t_{WLWH}	$\overline{WR}$ Pulse Width		48		ns
t_{WHAX1}	Trailing Edge of $\overline{WR}$ to Address Invalid		1.75		ns
t_{WHAX2}	Trailing Edge of $\overline{WR}$ to DPLD Address Invalid	(Note[4])	0		ns
t_{WHPV}	Trailing Edge of $\overline{WR}$ to Port Output Valid Using I/O Port Data Register			35	ns
t_{DVMV}	Data Valid to Port Output Valid Using Macrocell Register Preset/Clear	(Note [3])		70	ns
t_{WLMV}	$\overline{WR}$ Valid to Port Output Valid Using Macrocell Register Preset/Clear	(Note [2])		70	ns

Note: 1. Any input used to select an internal PSM function.
2. Assuming data is stable before active write signal.
3. Assuming write is active before data becomes valid.
4. TWHAX2 is the address hold time for DPLD inputs that are used to generate Sector Select signals for internal DSM memory.

Figure 35. Write Timing

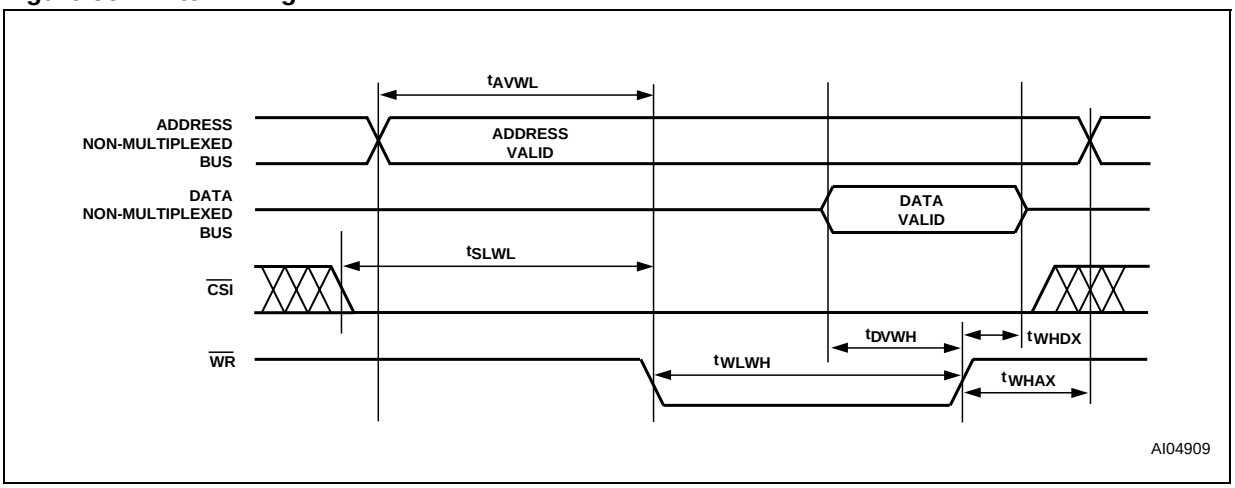


Table 33. Flash Memory Program, Write and Erase Times

Symbol	Parameter	Min.	Typ.	Max.	Unit
	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	1200	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 tQ7VQV time units before the data byte, DQ0-DQ7, is valid for reading.

Table 34. Reset ($\overline{\text{Reset}}$) Timing

Symbol	Parameter	Conditions	Min	Max	Unit
t_{NLNH}	RESET Active Low Time [1]		300		ns
$t_{NLNH-PO}$	Power On Reset Active Low Time		1		ms
t_{OPR}	RESET High to Operational Device			300	ns

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

Figure 36. Reset ($\overline{\text{RESET}}$) Timing

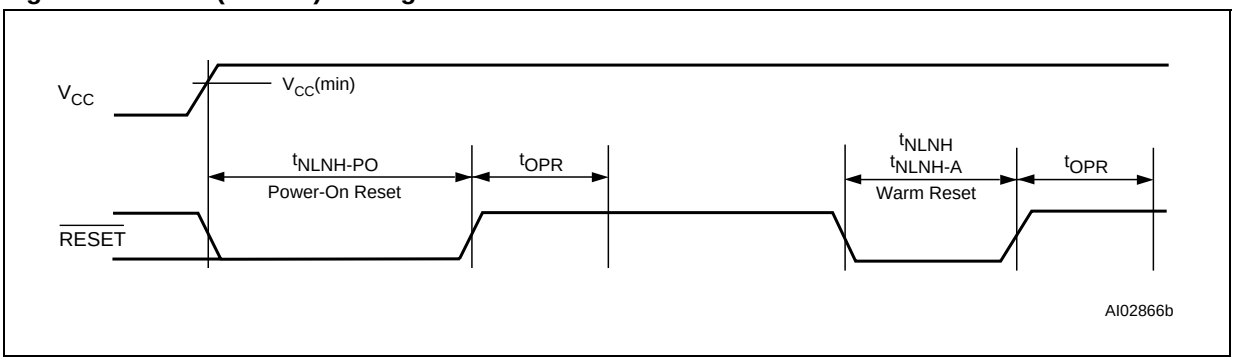
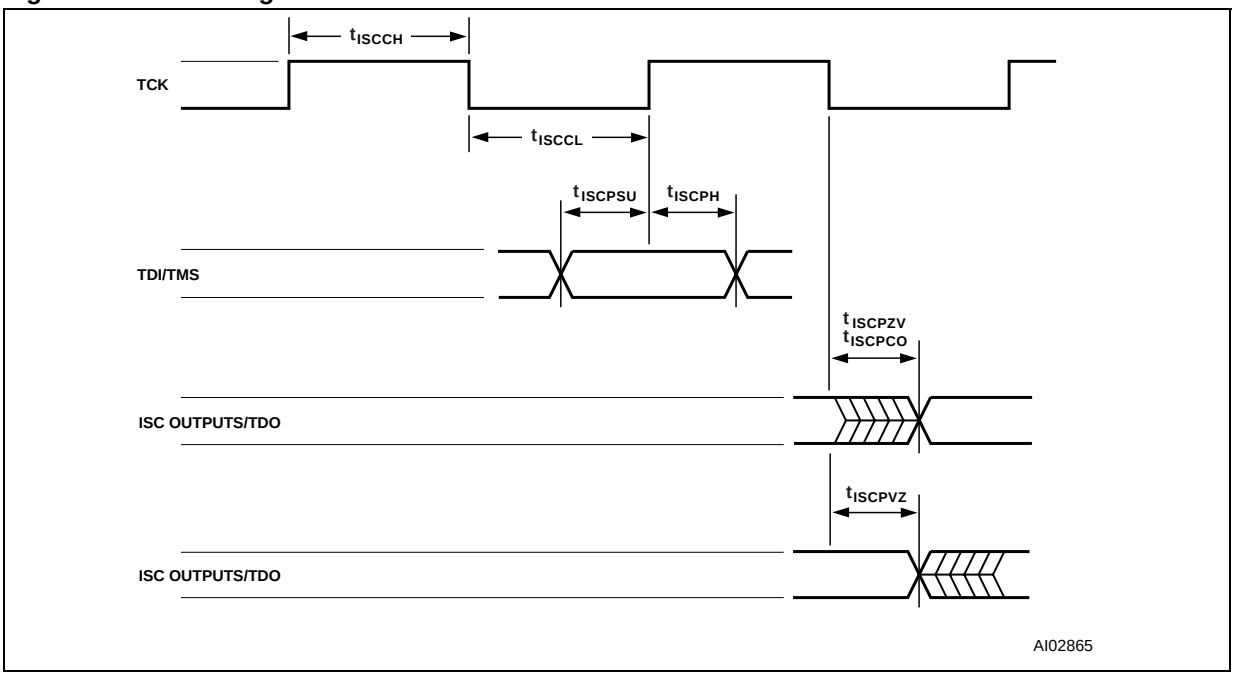


Table 35. ISC Timing

Symbol	Parameter	Conditions	-15		Unit
			Min	Max	
t_{ISCCF}	Clock (TCK, PC1) Frequency (except for PLD)	(Note 1)		10	MHz
t_{ISCCH}	Clock (TCK, PC1) High Time (except for PLD)	(Note 1)	45		ns
t_{ISCCL}	Clock (TCK, PC1) Low Time (except for PLD)	(Note 1)	45		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		13		ns
t_{ISCPH}	ISC Port Hold Up Time		5		ns
t_{ISCPCO}	ISC Port Clock to Output			36	ns
t_{ISCPZV}	ISC Port High-Impedance to Valid Output			36	ns
t_{ISCPVZ}	ISC Port Valid Output to High-Impedance			36	ns

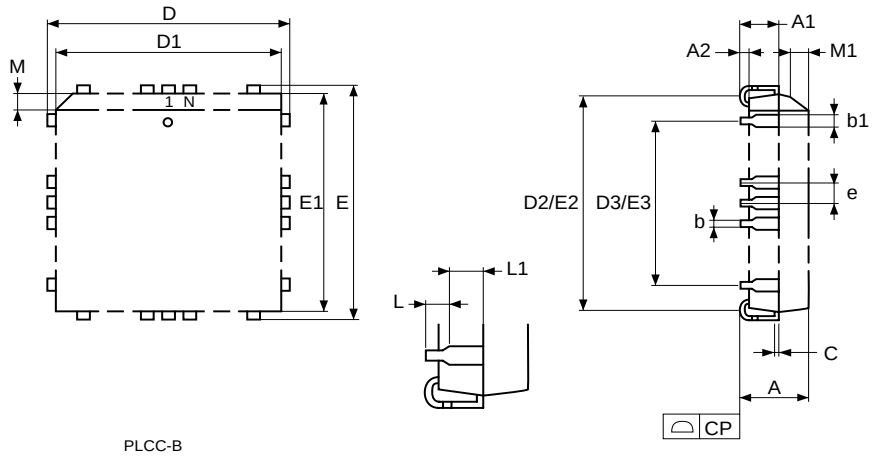
Note: 1. For non-PLD Programming, Erase or in ISC by-pass mode.
2. For Program or Erase PLD only.

Figure 37. ISC Timing



PACKAGE MECHANICAL

PLCC52 – 52 lead Plastic Leaded Chip Carrier, rectangular, Package Outline



Note: Drawing is not to scale.

PLCC52 – 52 lead Plastic Leaded Chip Carrier, rectangular, Package Mechanical Data

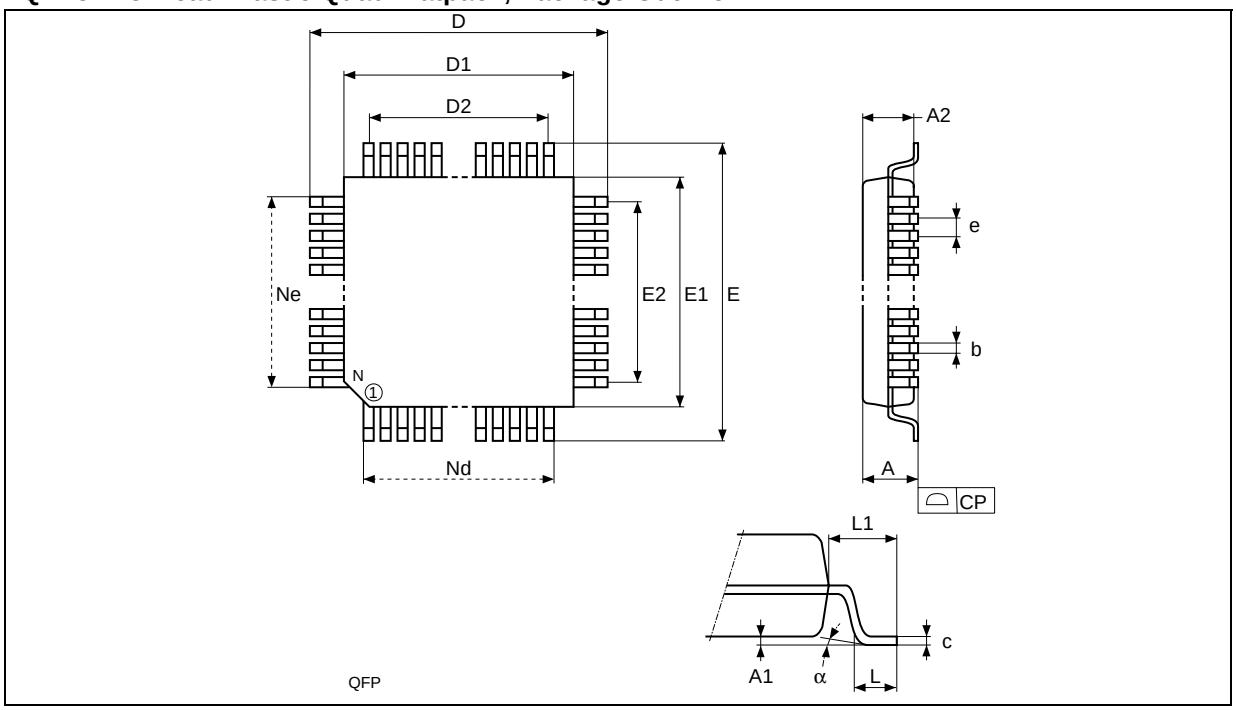
Symbol	mm			inches		
	Typ.	Min.	Max.	Typ.	Min.	Max.
A		4.19	4.57		0.165	0.180
A1		2.54	2.79		0.100	0.110
A2		–	0.91		–	0.036
B		0.33	0.53		0.013	0.021
B1		0.66	0.81		0.026	0.032
C		0.246	0.261		0.0097	0.0103
D		19.94	20.19		0.785	0.795
D1		19.05	19.15		0.750	0.754
D2		17.53	18.54		0.690	0.730
E		19.94	20.19		0.785	0.795
E1		19.05	19.15		0.750	0.754
E2		17.53	18.54		0.690	0.730
e	1.27	–	–	0.050	–	–
R	0.89	–	–	0.035	–	–
N	52			52		
Nd	13			13		
Ne	13			13		

Table 36. Assignments – PLCC52

Pin No.	Pin Assignments
1	GND
2	PB5
3	PB4
4	PB3
5	PB2
6	PB1
7	PB0
8	PD2
9	PD1
10	PD0
11	PC7
12	PC6
13	PC5
14	PC4
15	V_{CC}
16	GND
17	PC3
18	PC2 (VSTBY)
19	PC1
20	PC0
21	PA7
22	PA6
23	PA5
24	PA4
25	PA3
26	GND

Pin No.	Pin Assignments
27	PA2
28	PA1
29	PA0
30	AD0
31	AD1
32	AD2
33	AD3
34	AD4
35	AD5
36	AD6
37	AD7
38	V_{CC}
39	AD8
40	AD9
41	AD10
42	AD11
43	AD12
44	AD13
45	AD14
46	AD15
47	CNTL0
48	$\overline{\text{RESET}}$
49	CNTL2
50	CNTL1
51	PB7
52	PB6

PQFP52 - 52 lead Plastic Quad Flatpack, Package Outline



Note: Drawing is not to scale.

PQFP52 - 52 lead Plastic Quad Flatpack, Package Mechanical Data

Symb.	mm			inches		
	Typ.	Min.	Max.	Typ.	Min.	Max.
A			2.35			0.093
A1			0.25			0.010
A2	2.00	1.80	2.10	0.079	0.077	0.083
b		0.22	0.38		0.009	0.015
c		0.11	0.23		0.004	0.009
D	13.20	12.95	13.45	0.520	0.510	0.530
D1	10.00	9.90	10.10	0.394	0.390	0.398
D2	7.80	–	–	0.307	–	–
E	13.20	12.95	13.45	0.520	0.510	0.530
E1	10.00	9.90	10.10	0.394	0.390	0.398
E2	7.80	–	–	0.307	–	–
e	0.65	–	–	0.026		
L	0.88	0.73	1.03	0.035	0.029	0.041
L1	1.60	–	–	0.063		
α		0°	7°		0°	7°
N	52			52		
Nd	13			13		
Ne	13			13		
CP			0.10			0.004

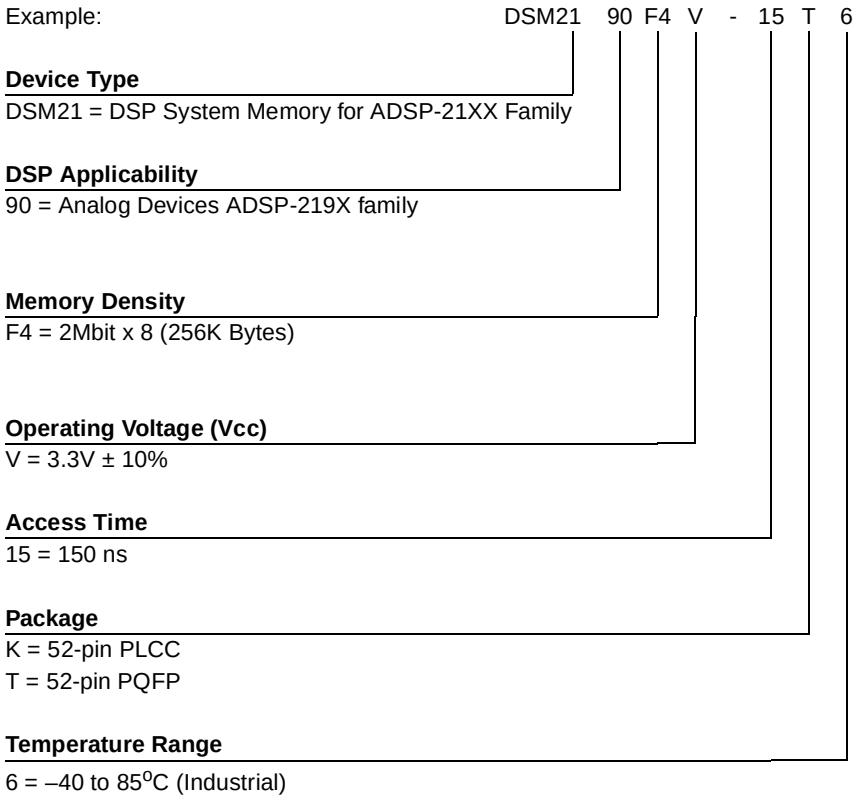
Table 37. Pin Assignments – PQFP52

Pin No.	Pin Assignments
1	PD2
2	PD1
3	PD0
4	PC7
5	PC6
6	PC5
7	PC4
8	V_{CC}
9	GND
10	PC3
11	PC2
12	PC1
13	PC0
14	PA7
15	PA6
16	PA5
17	PA4
18	PA3
19	GND
20	PA2
21	PA1
22	PA0
23	AD0
24	AD1
25	AD2
26	AD3

Pin No.	Pin Assignments
27	AD4
28	AD5
29	AD6
30	AD7
31	V_{CC}
32	AD8
33	AD9
34	AD10
35	AD11
36	AD12
37	AD13
38	AD14
39	AD15
40	CNTL0
41	$\overline{\text{RESET}}$
42	CNTL2
43	CNTL1
44	PB7
45	PB6
46	GND
47	PB5
48	PB4
49	PB3
50	PB2
51	PB1
52	PB0

PART NUMBERING

Table 38. Ordering Information Scheme



For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest ST Sales Office.

REVISION HISTORY

Table 39. Document Revision History

Date	Rev.	Description of Revision
27-Aug-2001	1.0	Document written
06-Nov-2001	1.1	Document released
17-Dec-2001	1.2	PQFP52 package mechanical data updated
18-Sep-2002	1.3	JTAG Debug bus separated from JTAG ISP bus