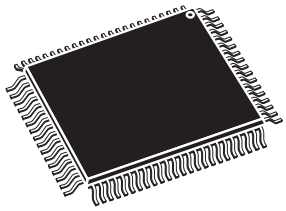
DSM2150F5V

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

FEATURES SUMMARY

- **Glueless Connection to DSP**
  - Easily add memory, logic, and I/O to the External Port of ADSP-218x, 219x, 2106x, 2116x, 2153x, and TS101 families of DSPs from Analog Devices, Inc.
- **Dual Flash Memories**
  - Two independent Flash memory arrays for storing DSP code and data
  - Capable of read-while-write concurrent Flash memory operation
  - Device can be configured as 8-bit or 16-bit
  - Built-in programmable address decoding logic allows mapping individual sectors of each Flash array to any address boundary
  - Each Flash sector can be write protected
- **512 KByte Main Flash memory**
  - Ample storage for boot loading DSP code/ data upon reset and subsequent code swaps
  - Large capacity for storing tables and constants or for data recording
- **32 KByte Secondary Flash memory**
  - Smaller sector size ideal for storing calibration and configuration constants. Eliminate external serial EEPROM.
  - Optionally bypass internal DSP boot ROM during start-up and execute code directly from Secondary Flash. Use for custom start-up code and In-Application Programming (IAP).
- **Up to 40 Multifunction I/O Pins**
  - Increase total DSP system I/O capability
  - I/O controlled by DSP software or PLD logic
- **General purpose PLD**
  - Use for peripheral glue logic to keypads, control panel, displays, LCDs, and other devices
  - Over 3,000 gates of PLD with 16 macro cells
  - Eliminate PLDs and external logic devices
  - Create state machines, chip selects, simple shifters and counters, clock dividers, delays
  - Simple PSDsoft Express™ development software, free from *www.st.com/psd*

Figure 1. Packages



TQFP80 (T)

- **In-System Programming (ISP) with JTAG**
  - Program entire chip in 15-35 seconds with no involvement of the DSP
  - Optionally links with DSP JTAG debug port
  - Eliminate need for sockets and pre-programming of memory and logic devices
  - ISP allows efficient manufacturing and product testing supporting Just-In-Time inventory
  - Use low-cost FlashLINK™ cable with any PC. Available from *www.st.com/psd*.
- **Content Security**
  - Programmable Security Bit blocks access of device programmers and readers
- **Operating Range**
  - Vcc: 3.3V ± 10%, Temp: –40°C to +85°C
- **Zero-Power Technology**
  - 50 µA standby current typical
- **Flash Memory Speed, Endurance, Retention**
  - 120 ns, 100K cycles, 15 year retention

TABLE OF CONTENTS

SUMMARY DESCRIPTION

The DSM2150F5V is an 8 or 16-bit system memory device for use with the Analog Devices DSPs. 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 flexible two-chip solution for DSP designs. On-chip integrated memory decode logic makes it easy to map dual banks of Flash memory to the DSPs in a variety of ways for bootloading or bypassing DSP boot ROM, 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 theneed for sockets and pre-programmed memory and logic devices. End products may be manufactured with a blank DSM device soldered down and programmed at the end of the assembly line in 15 to 35 seconds with no involvement of the DSP. Rapidly program test code, then application code as determined by Just-In Time inventory requirements. Additionally, JTAG ISP reduces development time by turning fast iterations of DSP code in the lab. The DSM JTAG interface may be optionally chained with the DSP JTAG debug interface (except ADSP-218x). Code updates in the field require no product disassembly. The FlashLINK™ JTAG programming cable costs $59 USD and plugs into any PC parallel port. Programming through conventional device insertion programmers is also available using PSDpro from STMicroelectronics and other 3rd party programmers. See *www.st.com/psd*.

DSM devices add programmable logic (PLD) and up to 32 configurable I/O pins to the DSP system. The state of I/O pins can be driven by DSP software or PLD logic. PLD and I/O configuration are programmable by JTAG ISP. 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, muxes, etc., eliminating the need for small external PLDs and logic devices. Configuration of PLD, I/O, and Flash memory mapping is easily entered in a point-and-click environment using the software development tool, PSDsoft Express™, available at no charge from www.st.com/psd. The two-chip DSP/DSM combination is ideal for systems having 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 data bus, even with security bit set.

Figure 2. System Block Diagram, Two Chip Solution

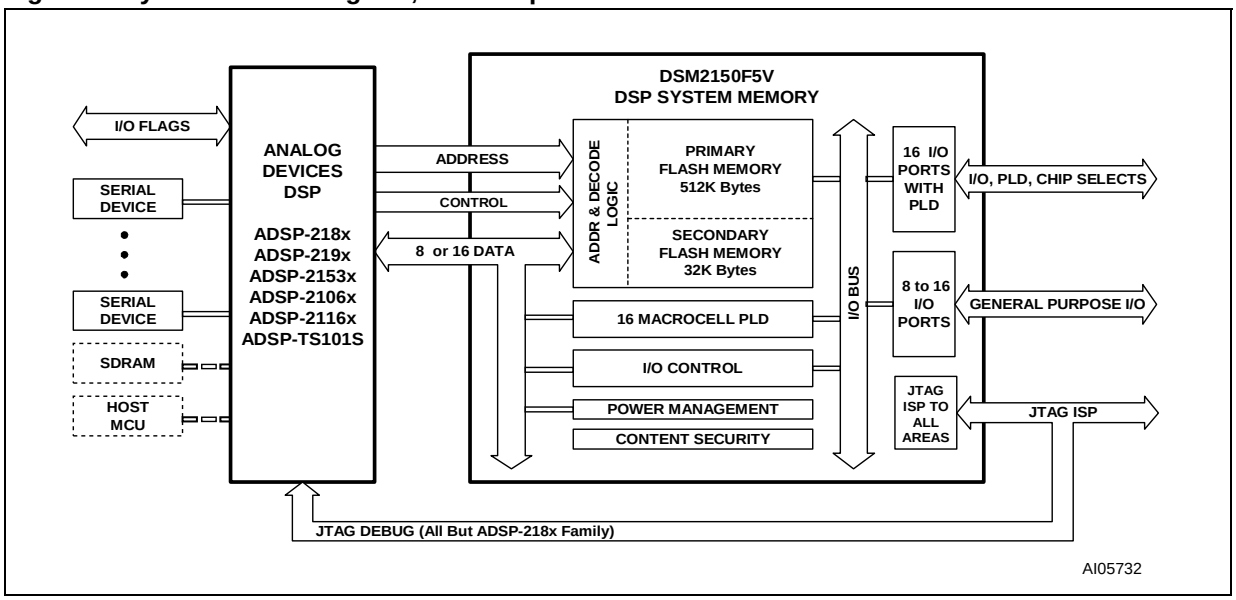


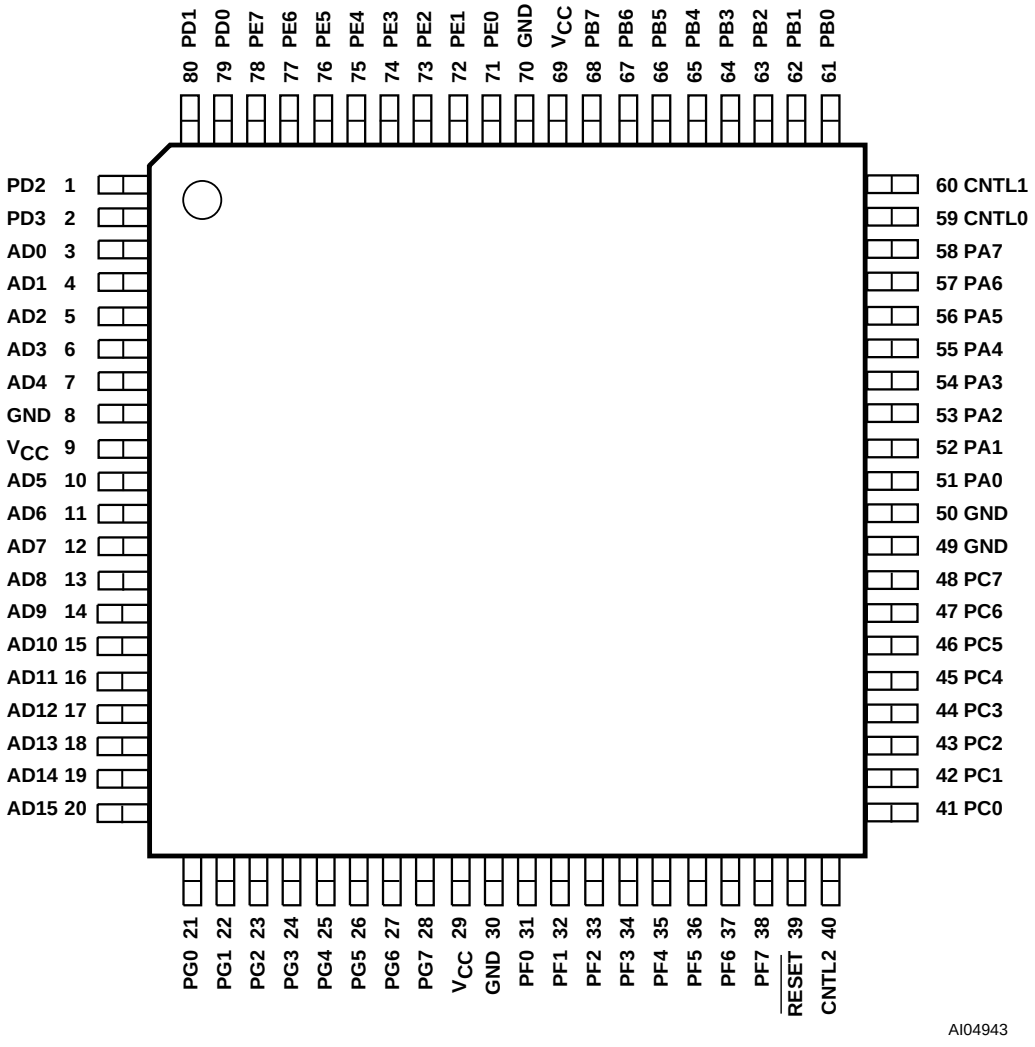
Table 1. DSM2150F5V DSP Memory System Devices

Part Number	Main Flash Memory	Secondary Flash Mem	PLD	I/O Ports	V_{CC} and I/O	Mem Speed	Package	Op Temp
DSM2150F5V-12T6	512 KBytes, eight 64KByte sectors	32 KBytes, four 8 KByte sectors	16 macro cells	Up to 40	3.3V ±10%	120 ns	80-pin TQFP	–40°C to +85°C

Table 2. Compatible Analog Devices DSPs

DSP Part Number	Core Operating Voltage	I/O Voltage
ADSP-2183, 2184L, 2185L, 2186L, 2187L	3.3V	3.3V
ADSP-2185M, 2186M, 2188M, 2189M	2.5V	3.3V
ADSP-2184N, 2185N, 2186N, 2187N, 2188N, 2189N	1.8V	3.3V
ADSP-2191M, 2195M, 2196M	2.5V	3.3V
Blackfin ADSP-21532S	3.3V	3.3V
Blackfin ADSP-21535P	1.5V	3.3V
Sharc ADSP-21060L, 21061L, 21062L, 21065L	3.3V	3.3V
Sharc ADSP-21160M	2.5V	3.3V
Sharc ADSP-21160N, 21161N	1.8V	3.3V
Tiger Sharc ADSP-TS101S	1.0V	3.3V

Figure 3. TQFP Connections



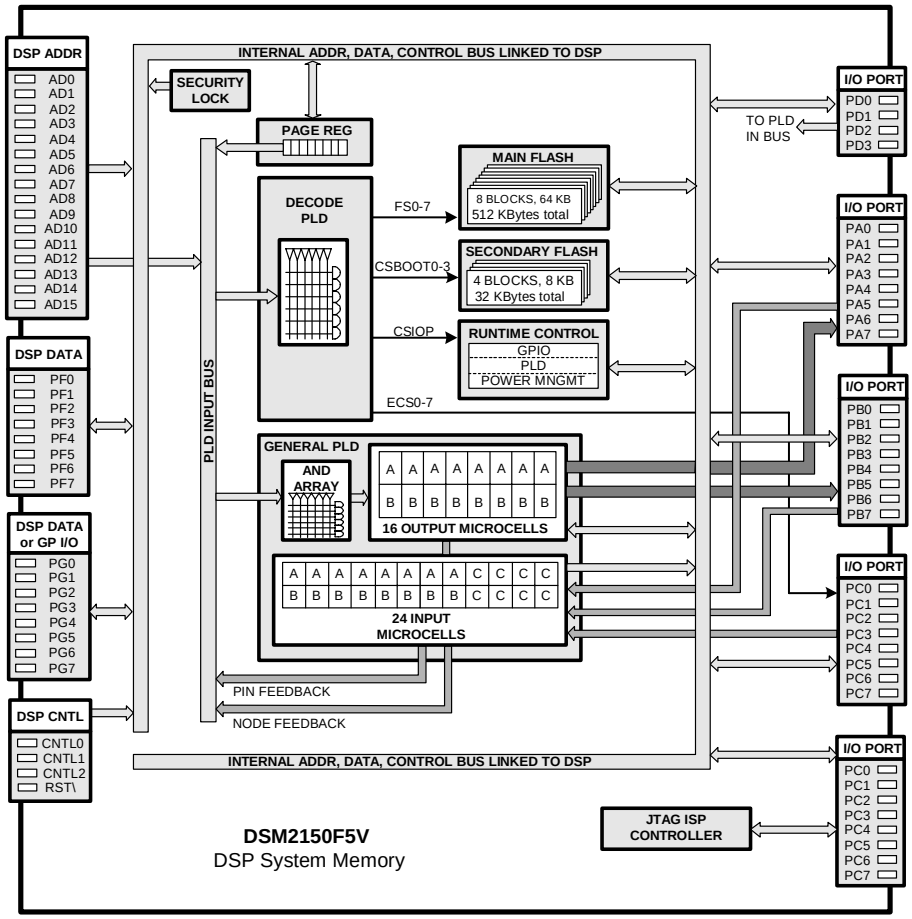
ARCHITECTURAL OVERVIEW

Major functional blocks are shown in Figure 4.

DSP Address/Data/Control Interface

These DSP signals attach directly to the DSM for a glueless connection. An 8-bit or 16-bit data connection is formed and 16 or more DSP address lines can be decoded as well as various DSP memory strobes; i.e. $\overline{BMS}$, $\overline{RD}$, $\overline{AWE}$, $\overline{IOMS}$, $\overline{MSx}$, etc. The data path width must be specified as 8-bits or 16-bits in PSDsoft Express. This configuration is a static, meaning the data path width cannot switch between 8-bits and 16-bits during runtime. Port F is used for 8-bit data path, Ports F and G are used for 16-bit data path. There are many different ways the DSM2150F5 can be configured and used depending on system requirements. See Appendices for example connections between the DSM2150 and different DSPs.

Figure 4. Block Diagram



Main Flash Memory

The 4M bit (512 KByte) Main Flash memory is divided into eight equally-sized 64 KByte 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 are easily placed in flash memory using PSDsoft Express, the software development tool.

Secondary Flash Memory

The 256 Kbit (32 KByte) Secondary Flash memory is divided into eight equally-sized 8 KByte 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 smaller sectors. 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 (bypass internal DSP boot ROM). Storing code here can keep the entire Main Flash free of initialization code for clean software partitioning. If only one or more 8 KByte sectors are needed for start-up code, the remaining sectors of Secondary Flash may be used for data storage.

In-Application-Programming (IAP) may be implemented using Secondary Flash. For example, code to implement IAP over a USB channel may be stored here. The DSP executes code from Secondary Flash array while erasing and writing new code to the Main Flash array as it is received over the USB channel. Any communication channel that the DSP supports can be used for IAP.

Secondary Flash may also be used as an extension to Main Flash memory producing a total of 544 KBytes

Miscellaneous: Main and Secondary Flash memories are totally independent, allowing concurrent operation. 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 array 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 drive external chip select signals on Port C 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 24 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.

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 73 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 A or Port B.

IMCs: Inputs from pins on Ports A, B or 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 for control and status registers. 50 registers are used from 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. Some DSPs can access *csiop* registers using I/O memory with the $\overline{\text{IOMS}}$ strobe (if equipped). *csiop* registers are bytes. When the DSM is configured for 16-bit operation, csiop registers are read in byte pairs at even addresses only. Care should be taken while writing *csiop* registers to ensure the proper byte is written within the byte pair. This is not a problem for DSPs that support the $\overline{\text{BHE}}$ (Byte High Enable) signal on the CNTL2 input pin, or

$\overline{WRL}$, $\overline{WRH}$ (write low byte, write high byte) on the CNTL0 and PD3 input pins of the DSM2150F5V.

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 both PLDs. The page register can be used for special memory mapping requirements and also for general logic.

I/O Ports

The DSM has 52 individually configurable I/O pins distributed over the seven ports (Ports A, B, C, D, E, F, and G). At least 32 I/O are available when DSM2150F5 is connected with 8-bit data path, and at least 24 I/O are available with 16-bit data path. 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.)

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 E. 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 15 to 35 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 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 E, TSTAT and $\overline{TERR}$ in addition to TMS, TCK, TDI and TDO.

The FlashLINK™ JTAG programming cable is available from STMicroelectronics for $USD59 and PSDsoft Express software is available at no charge from www.psdst.com. 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 33.

Power Management

The DSM has bits in *csiop* 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, other bits in two *csiop* registers can be set by the DSP to selectively block signals from entering the CPLD which reduces power consumption.

Both Flash memories automatically go to standy current between accesses. No user action required.

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 80-pin TQFP-package in Figure 3, and their description in Table 3.

Table 3. Pin Description

Pin Name	Type	Description
AD0-15	In	Sixteen address inputs from the DSP.
CNTL0	In	Active low write strobe input from the DSP, typically connected to DSP $\overline{WR}$ signal. Also functions as $\overline{WRL}$ for DSPs which use $\overline{WRL}$ strobe when writing low byte only in 16-bit word.
CNTL1	In	Active low read strobe input from the DSP.
CNTL2	In	Programmable control input. CNTL2 may be used for $\overline{BHE}$ (Byte High Enable) when DSM2150F5 is configured for 16-bit operation. BHE = 0 will allow a byte write from data lines D8-D15 ignoring data lines D0-D7. $\overline{BHE}$ = 1 will allow a byte write from D0-D7 ignoring datalines D8-D15. DSP read operatons are not affected by $\overline{BHE}$ (always read both bytes).
$\overline{Reset}$	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 configurable Port A 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 (McellA0-7) outputs. 3. Inputs to the PLDs (via Input Macrocells). Can be used to input address A16 and above. Note: PA0-PA7 may be configured at run-time as standard CMOS or Open Drain Outputs.
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 (McellB0-7 or McellC0-7) outputs. 3. Inputs to the PLDs (via Input Macrocells). Can be used to input address A16 and above. Note: PB0-PB7 may be configured at run-time as 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. DPLD chip-select outputs (ECS0-7, does not consume MicroCells). 3. Inputs to the PLDs (via Input Macrocells). Can be used to input address A16 and above. Note: PC0-PC7 may be configured at run-time as standard CMOS or Faster Slew Rate Output.
PD0-3	I/O	Four 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). Can be used to input address A16 and above. 3. PD1 can be configured as CLKIN, a common clock input to PLD. 4. PD2 can be configured as $\overline{CSI}$, active low Chip Select Input to select Flash memory. Flash memory is disabled to conserve more power when $\overline{CSI}$ is logic high. 5. PD3 can be used for $\overline{WRH}$ strobe from DSP to write high byte only for 16-bit configuration.
PE0-7	I/O	Eight configurable Port E signals with the following functions: 1. MCU I/O – DSP may write or read pins directly at runtime with csiop registers. 2. PE0, PE1, PE2, and PE3 can form the JTAG IEEE-1149.1 ISP serial interface as signals TMS, TCK, TDI, and TDO respectively. 3. PE4 and PE5 can form the enhanced JTAG signals TSTAT and $\overline{TERR}$ respectively. Reduces ISP programming time up to 30% when used in addition to the standard four JTAG signals: TDI, TDO, TMS, TCK. 4. PE4 can 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. Note 1: PE0-PE7 may be configured at run-time as either standard CMOS or Open Drain Outputs. Note 2: The JTAG ISP pins may be multiplexed with other I/O functions.
PF0-7	I/O	Port F connects to eight data bus signals, D0 - D7 from DSP.
PG0-7	I/O	Port G connects to eight data bus signals, D8 - D15 from DSP if 16-bit data path is used. Otherwise, PG0-PG7 can be used for general purpose MCU I/O pins. Note: PG0-PG7 may be configured at run-time as standard CMOS or Open Drain Outputs.
V_{CC}		Supply Voltage
GND		Ground pins

RUNTIME CONTROL REGISTER DEFINITION

A block of 256 addresses are decoded inside the DSM2150F5 for control and status. 50 locations contain registers that the DSP accesses at runtime. The base address of the registers is called *csiop* (Chip Select I/O Port). Table 4 lists the registers and their offsets (in hexadecimal) from the *csiop* base. See Appendix A for bit definitions.

Note: 1. Do not write to unused locations, they should remain logic zero.
2. See Table 13 for register state at reset and at power-on.

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

Register Name	Port A	Port B	Port C	Port D	Port E	Port G	Other	Description
Data In	00	01	10	11	30	41		MCUI/O input mode. Read to obtain current logic level of Port pins. No writes.
Data Out	04	05	14	15	34	45		MCU I/O output mode. Write to set logic level on Port pins. Read to check status.
Direction	06	07	16	17	36	47		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	08	09	18	19	38	49		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	0A	0B	1A					Read to obtain state of IMCs. No writes.
Enable Out	0C	0D	1C					Read to obtain the status of the output enable logic on each I/O Port driver. No writes.
Output Macrocells A							20	Read to get logic state of output of OMC bank A. Write to load registers of OMC bank A.
Output Macrocells B							21	Read to get logic state of output of OMC bank B. Write to load registers of OMC bank B.
Mask Macrocells A							22	Write to set mask for loading OMCs in bank A. Logic 1 in a bit position will block reads/writes of the corresponding OMC. Logic 0 will pass OMC value. Read to check status.
Mask Macrocells B							23	Write to set mask for loading OMCs in bank B. Logic 1 in a bit position will block reads/writes of the corresponding OMC. Logic 0 will pass OMC value. Read to check status.
Main Flash Sector Protect							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.
Memory_ID0							F0	Read to get size of Main Flash memory. No Writes.
Memory_ID1							F1	Read to get size of 2nd Flash memory. No Writes.

DETAILED OPERATION

Figure 4 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 64 KByte sectors. The Secondary Flash memory array is divided into four equal 8 KByte 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 7). 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/$\overline{\text{Busy}}$ (PE4). This signal can be used to output the Ready/$\overline{\text{Busy}}$ status of the device. 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 write operations to invoke an internal algorithm, then a single data byte (or word if DSM2150F5 is configured for 16-bit operation) is written to the Flash memory array, then programming status is checked by a read operation or by checking the Ready/$\overline{\text{Busy}}$ pin (PE4). This “unlocking” sequence optionally may be bypassed by using the Unlock Bypass command to reduce programming time .

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 when the DSM2150F5 is configured to operate as an 8-bit device. Table 6 lists instruction sequences when the DSM2150 is configured for 16-bit operation. These instruction sequences are different combinations of individual write and read operations.

IMPORTANT: The DSP cannot 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 from 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 ROM or SRAM device.

Table 5. Instruction Sequences for 8-bit Operaton[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 at addr XXX01h			
Read Memory Sector Protection Status[6,7,8]	Write AAh to XX555h	Write 55h to XXAAAh	Write 90h to XX555h	Read value at addr XXX02h			
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 addr in FS0-7 or CSBOOT0-3						
Resume Sector Erase[12]	Write 30h to addr in FS0-7 or CSBOOT0-3						
Reset Flash[6]	Write F0h to addr in FS0-7 or CSBOOT0-3						
Unlock Bypass	Write AAh to XX555h	Write 55h to XXAAAh	Write 20h to XX555h				
Unlock Bypass Program[13]	Write A0h to addr in FS0-7 or CSBOOT0-3	Write (program) data to addr					
Unlock Bypass Reset[14]	Write 90h to addr in FS0-7 or CSBOOT0-3	Write 00h to addr in FS0-7 or CSBOOT0-3					

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. 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. Likewise, directing command to any Secondary Flash sector (CSBOOT0-3) will invoke erase of all four sectors.
10. DSP writes command sequence to initial segment to be erased, then writes the byte 30h to additional sectors to be erased. 30h must be addressed to one of the other Flash memory segments (FS0-7 or CSBOOT0-3) for each additional segment (write 30h to any address within a desired sector). No more time than $t_{TIMEOUT}$ 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.
13. The Unlock Bypass instructionis required prior to the Unlock Bypass Program Instruction.
14. The Unlock Bypass Reset Flash instruction is required to return to reading memory data when the device is in Umlock Bypass mode.

Table 6. Instruction Sequences for 16-bit Operaton[1,2,3,4,15]

Instruction Sequence	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
Read Memory Contents[5]	Read word from even addr						
Read Flash Identifier (Main Flash only)[6,7]	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX90h to XXAAAh	Read identifier at addr XXX02h			
Read Sector ProtectStatus[6,7,8]	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX90h to XXAAAh	Read value at addr XXX04h			
Program a Flash word	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XXA0h to XXAAAh	Write word to even address			
Flash Bulk Erase[9]	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX80h to XXAAAh	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX10h to XXAAAh	
Flash Sector Erase[10]	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX80h to XXAAAh	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX30h to new Sectr	Write XX30h to new Sector
Suspend Sector Erase[11]	Write XXB0h to even addr in FS0-7 or CSBOOT0-3						
Resume Sector Erase[12]	Write XX30h to even addr in FS0-7 or CSBOOT0-3						
Reset Flash[6]	Write XXF0h to even addr in FS0-7 or CSBOOT0-3						
Unlock Bypass	Write XXAAh to XXAAAh	Write XX55h to XX554h	Write XX20h to XXAAAh				
Unlock Bypass Program[13]	Write XXA0h to even addr in FS0-7 or CSBOOT0-3	Write word to even addr					
Unlock Bypass Reset[14]	Write XX90h to even addr in FS0-7 or CSBOOT0-3	Write XX00h to even addr in FS0-7 or CSBOOT0-3					

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. 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 XX00h for an unprotected sector, and XX01h 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. Likewise, directing command to any Secondary Flash sector (CSBOOT0-3) will invoke erase of all four sectors.
10. DSP writes command sequence to initial segment to be erased, then writes the word XX30h to additional sectors to be erased. XX30h must be addressed to one of the other Flash memory segments (FS0-7 or CSBOOT0-3) for each additional segment (write XX30h to any address within a desired sector). No more time than $t_{TIMEOUT}$ 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.
13. The Unlock Bypass instructionis required prior to the Unlock Bypass Program Instruction.
14. The Unlock Bypass Reset Flash instruction is required to return to reading memory data when the device is in Umlock Bypass mode.
15. All bus cycles in an instruction sequence are writes or reads to an even address (XXAAAh or XX554h), and only the low byte, D0-D7, is significant (upper byte on D8-D15 is ignored). A Flash memory Program bus cycle writes a word to an even address.

Instruction Sequences

An instruction sequence consists of a sequence of specific write or read operations.

IMPORTANT:

When the DSM2150F5 is configured for 8-bit operations, all instruction sequences consist of byte write and read operations on an even or odd address boundary. Flash memory locations are programmed in bytes to even or odd addresses.

When the DSM2150F5 is configured for 16-bit operation, all instruction sequences consist of word write and read operations on even address boundaries only. The lower byte on D0-7 is significant and the upper byte on D8-15 is ignored during instructions and status. Flash memory locations are programmed in 16-bit words to even addresses only.

Each byte/word written to the device is received and sequentially decoded and not executed as a standard write operation to the memory array until the entire command string has been received. The instruction sequence is executed when the correct number of bytes/words are properly received and the time between two consecutive bytes/words is shorter than the time-out period, $t_{TIMEOUT}$. 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/words or time-out between two consecutive bytes/words 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 and Table 6:

Flash memory:

- Read memory contents
- Read Main Flash Identifier value
- Read Sector Protection Status
- Program a Byte/Word
- Erase memory by chip or sector
- Suspend or resume sector erase
- Reset to Read Array mode
- Unlock Bypass Instructions

For efficient decoding of the instruction sequences, the first two bytes/words of an instruction sequence are the coded cycles and are followed by an instruction byte/word or confirmation byte/word. The coded cycles consist of writing the data AAh to address XX555h (or XXAAh to address XXAAAh for 16-bit mode) during the first cycle and data 55h to address XXAAAh (or XX55h to address XX554 for 16-bit mode) during the second cycle. Address inpur signals A12 and above 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 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 or 6). 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. Bytes are read from even or odd addresses when the DSM2150F5 is configured for 8-bit operation. Only 16-bit words are read from even addresses when the DSM2150F5 is configured for 16-bit operations.

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 or 6). During the read operation the appropriate internal Sector Select (*FS0-FS7*) must be active. The identifier is E8h (or XXE8h for 16-bit mode). 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 or 6). The read operation will produce 01h (XX01h for 16-bit mode) if the Flash sector is protected or 00h (XX00h or 16-bit mode) if the sector is not protected. Internal Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) designates the Flash memory sector whose protection has to be verified.

Alternatively, 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 7. 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.
3. When the DSM2150F5 is configured for 16-bit operation, DQ8-DQ15 are not significant and can be ignored.

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 7. The status bits can be read as many times as needed. DQ8 - DQ15 are insignificant and can be ignored when the DSM2150F5 is configured to operate in 16-bit mode, however, the read operation must occur on an even address boundary.

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 18, 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.

- 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/word 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 t_{TIMOUT}, and then returns to the previous addressed byte. No erasure is performed.

Toggle Flag (DQ6). The device offers an alternative 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 DSM2150F5 is configured to operate in 16-bit mode, status reads must occur at even addresses, DQ8 - DQ15 are insignificant and can be ignored.

When the internal cycle is complete, the toggling stops and the data read on the Data Bus is the addressed memory byte/word. The device is now accessible for a new read or write operation. The cycle is finished when two successive reads yield the same output data.

- The Toggle Flag (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/word 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 t_{TIMOUT} 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/word Program operation, Sector Erase, or Bulk Erase operation.

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, logic 0, to the erased state, logic 1, which is not valid. The Error Flag (DQ5) bit may also indicate a Time-out condition while attempting to program a byte/word.

In case of an error in a Flash memory Sector Erase or byte/word Program cycle, the Flash memory sector in which the error occurred or to which the programmed byte/word 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/words. The Erase Time-out Flag (DQ3) bit is reset to 0 after a Sector Erase cycle for a time period t_{TIMOUT} 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 the DSM2150F5 is configured for 8-bit operation, Flash memory locations are programmed in 8-bit bytes to even or odd addresses.

When the DSM2150F5 is configured for 16-bit operation, Flash memory locations are programmed in 16-bit words to even addresses only. However, some DSPs support the BHE (byte high enable) signal on the DSM2150F5V CNTL2 input or the WRL, WRH (write low byte, write high byte) signals on the CNTL0 and PD3 inputs. In these cases, a DSP write operation can be directed to an individual byte (upper or lower) of a byte-pair. These signals do not effect read operations, only writes. Reads are always by 16-bits from an even address.

$\overline{\text{BHE}}$ signal on CNT2 input. See Table 8. Even-byte refers to locations with address A0 equal to 0, and odd byte as locations with A0 equal to 1.

Table 8. 16-Bit Data Bus with BHE

BHE	A0	D15-D8	D7-D0
0	0	Odd Byte	Even Byte
0	1	Odd Byte	—
1	0	—	Even Byte

$\overline{\text{WRL}}$ and WRH signals on CNT0 and PD3 inputs. See Table 9. Even-byte refers to locations with address A0 equal to 0, and odd byte as locations with A0 equal to 1.

Table 9. 16-Bit Data Bus with WRH and WRL

WRH	WRL	D15-D8	D7-D0
0	0	Odd Byte	Even Byte
0	1	Odd Byte	—
1	0	—	Even Byte

When a byte/word 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. The DSP may erase the entire Flash memory array all at once or individual sector-by-sector, but not byte-by-byte (or word-by-word for 16-bit mode). However, the DSP may program Flash memory byte-by-byte (or word-by-word for 16-bit mode).

The Flash memory requires the DSP to send an instruction sequence to program a byte or to erase sectors (see Table 5 or 6).

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/$\overline{\text{Busy}}$ (pin PE4).

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 5 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/word to be programmed in Flash memory to check status. For 16-bit operation, the status location read must be at an even address and D8-D15 can be ignored. The Data Polling Flag (DQ7) bit of this location becomes the compliment of bit 7 of the original data byte/word 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 5).

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

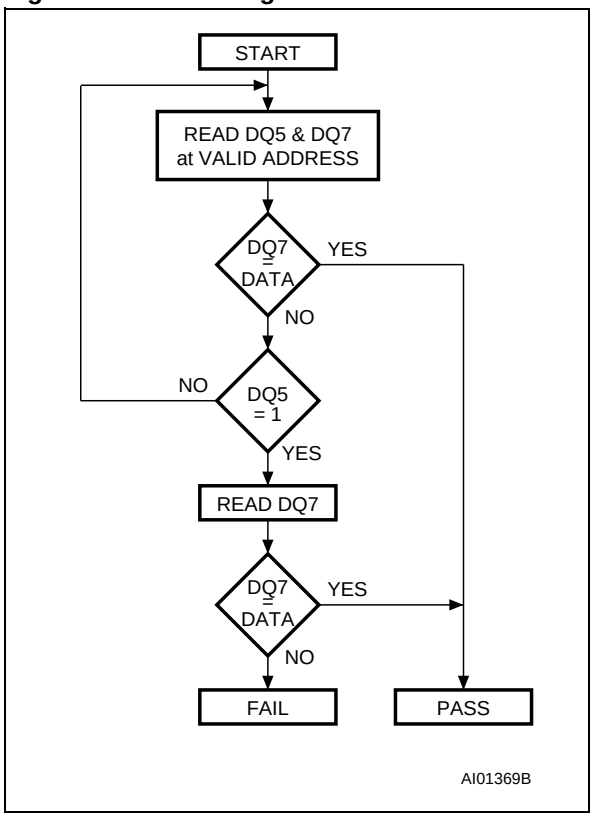
It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte/word hat was written to the Flash memory with the byte/word that was intended to be written.

When using the Data Polling method during an Erase cycle, Figure 5 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 (must be even address for 16-bit mode) 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 5. Data Polling Flowchart



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 6.

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 generate eight External Chip Select (ECS0-ECS7) signals on Port C.

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), 24 Input Macrocells (IMC), and the AND Array.

Table 10. DPLD and CPLD Inputs

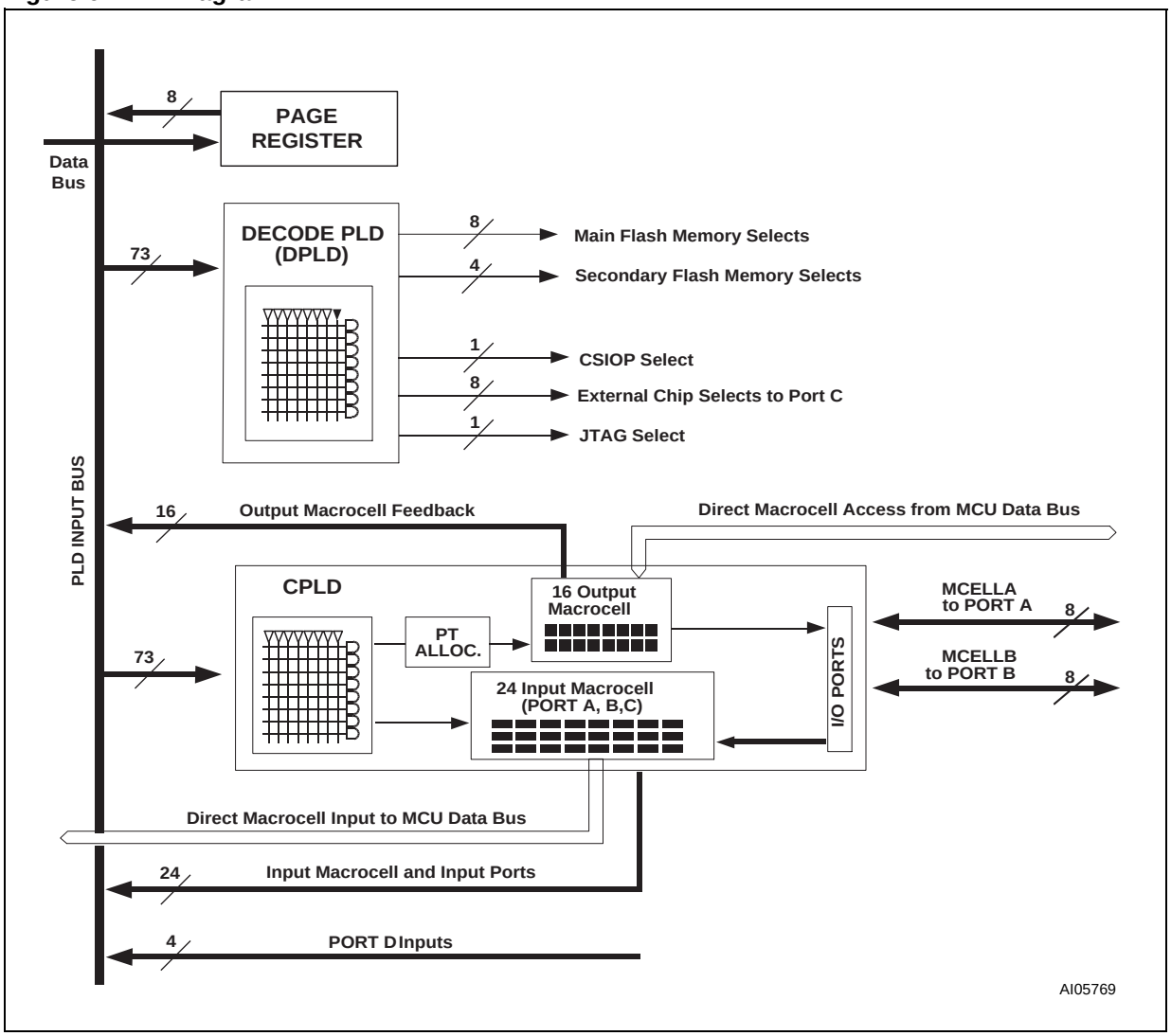
Input Source	Input Name	Number of Signals
DSP Address Bus[1]	A15-A0	16
DSP Control Signals[2]	CNTL2-CNTL0	3
Reset	$\overline{RST}$	1
PortA Input Macrocells	PA7-PA0	8
PortB Input Macrocells	PB7-PB0	8
PortC Input Macrocells	PC7-PC0	8
Port D Inputs	PD3-PD0	4
Page Register	PG7-PG0	8
Macrocell A Feedback	MCELLA FB7-0	8
Macrocell B Feedback	MCELLB FB7-0	8
Flash memory Program Status Bit	Ready/$\overline{Busy}$	1

Note: 1. DSP address lines above A15 may enter the DSM device on any pin on ports A, B, C or D. See Appendices for recommended connections.
2. Additional DSP control signals may enter the DMS device on any pin on Ports A, B, C, or D. See Appendices for recommended connections.

The AND Array is used to form product terms. These product terms are configured from the logic definition entered in PSDsoft Express. A PLD Input Bus consisting of 73 signals is connected to the PLDs. Input signals are shown in Table 10.

Turbo Bit. The PLDs in the device can minimize power consumption by switching to standby when inputs remain unchanged for an extended time t_{TURBO}. 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, seven 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 6. PLD Diagram

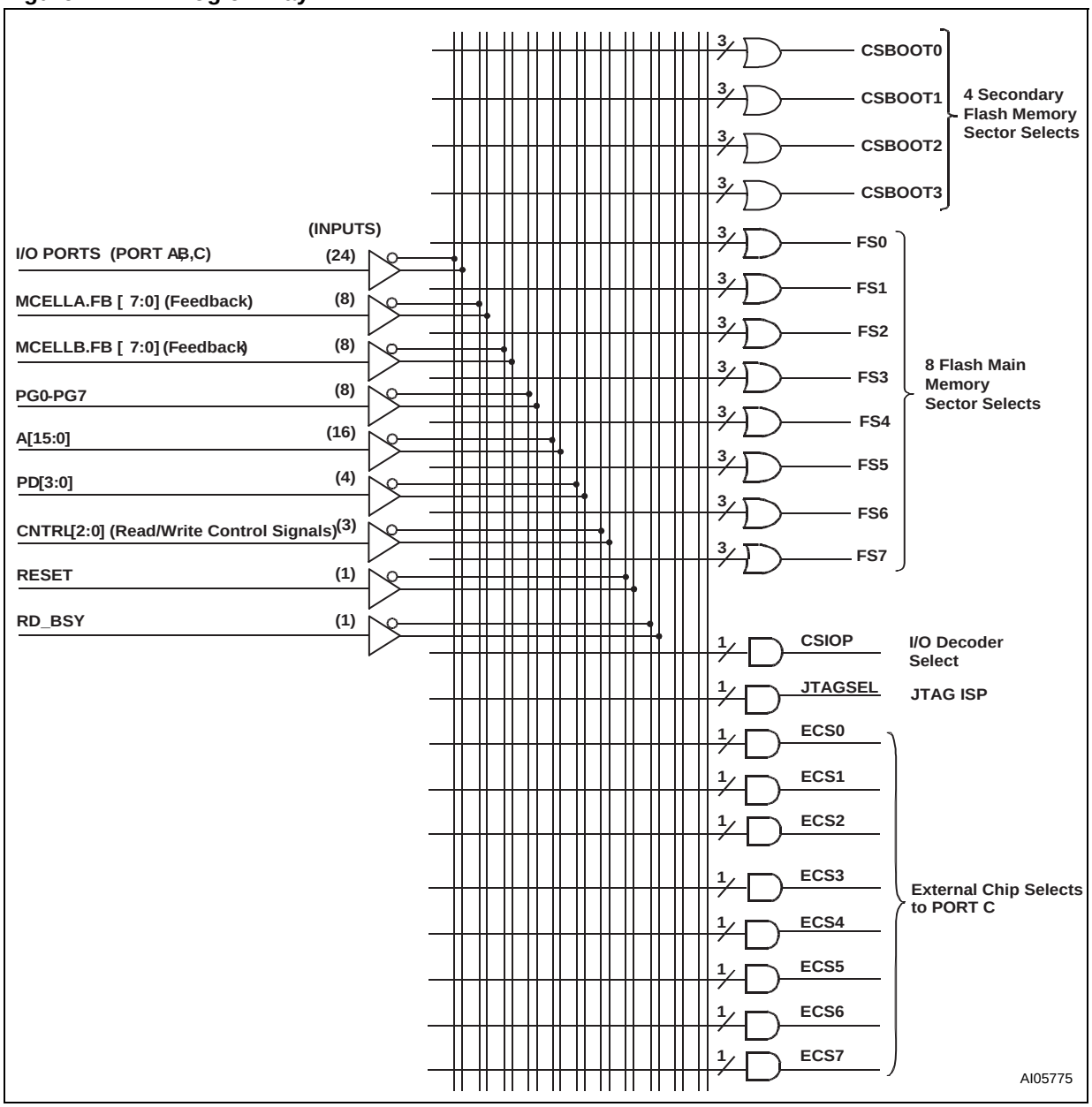


Decode PLD (DPLD)

The DPLD, shown in Figure 7, 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 E when multiplexing JTAG signals with general I/O signals)
- 8 external chip select output signals for Port C pins, each with one product term.

Figure 7. 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.

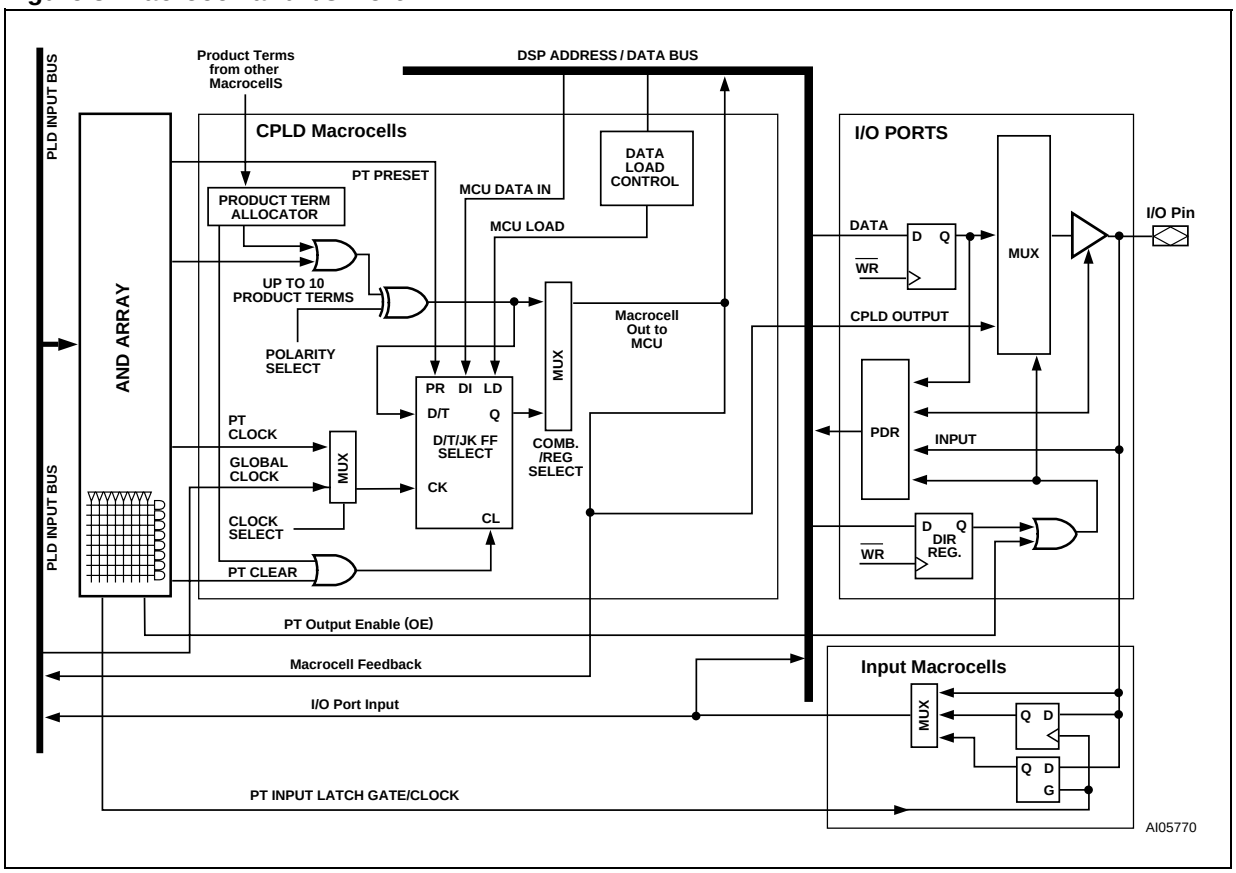
The CPLD has the following blocks:

- 24 Input Macrocells (IMC)
- 16 Output Macrocells (OMC)
- Product Term Allocator
- AND Array capable of generating up to 190 product terms
- Two I/O Ports.

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

The IMCs and OMCs 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 OMC or read data from both the IMCs and OMCs. 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 8. Macrocell and I/O Port



Output Macrocell (OMC). Eight of the OMCs, McellA0-McellA7, are connected to Port A pins. The other eight Macrocells, McellB0-McellB7, are connected to Ports B pins. OMCs may be used for internal feedback (buried registers), or their outputs may be routed to external Port pins.

The OMC architecture is shown in Figure 9. As shown in the figure, there are native product terms available from the AND Array, and borrowed product terms available (if unused) from other OMC. The polarity of the product term is controlled by the XOR gate. The 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 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.

Table 11. Output Macrocell Port and Data Bit Assignments

Output Macrocell	Port Assignment	Native Product Terms	Maximum Borrowed Product Terms	Data Bit for Loading or Reading in 16-bit Mode	Data Bit for Loading or Reading in 8-bit Mode
McellA0	Port A0	3	6	D0	D0
McellA1	Port A1	3	6	D1	D1
McellA2	Port A2	3	6	D2	D2
McellA3	Port A3	3	6	D3	D3
McellA4	Port A4	3	6	D4	D4
McellA5	Port A5	3	6	D5	D5
McellA6	Port A6	3	6	D6	D6
McellA7	Port A7	3	6	D7	D7
McellB0	Port B0	4	5	D8	D0
McellB1	Port B1	4	5	D9	D1
McellB2	Port B2	4	5	D10	D2
McellB3	Port B3	4	5	D11	D3
McellB4	Port B4	4	6	D12	D4
McellB5	Port B5	4	6	D13	D5
McellB6	Port B6	4	6	D14	D6
McellB7	Port B7	4	6	D15	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:

- McellA0-McellA7 all have three native product terms and may borrow up to six more
- McellB0-McellB3 all have four native product terms and may borrow up to five more
- McellB4-McellB7 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.

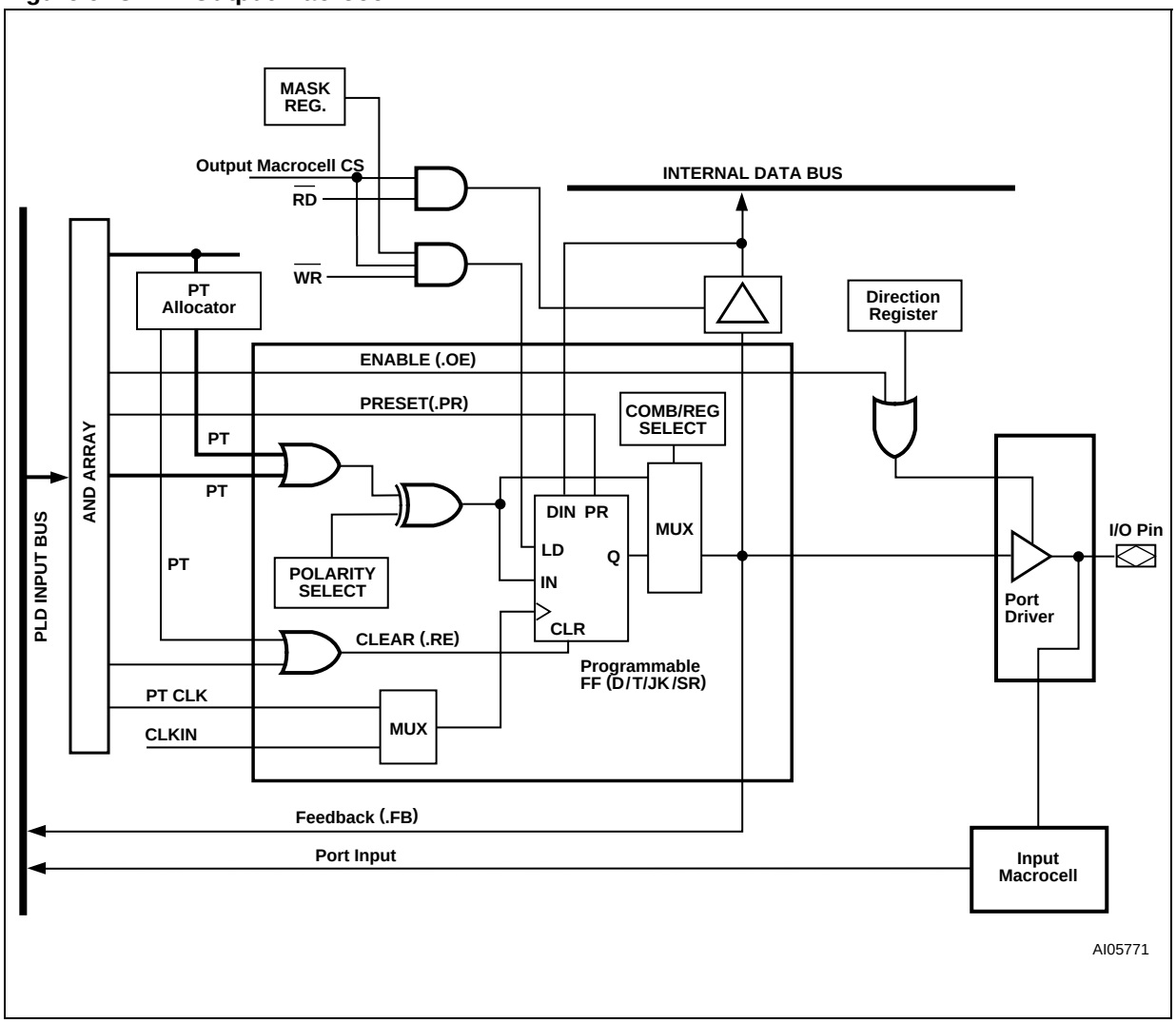
If an equation requires more product terms than are available to it through product term allocation, then “external” product terms are required, which consumes other OMC. This is called product term expansion and also happens automatically in PSDsoft Express™ as needed. Product tern expansion causes 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 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 MCELLA0-7 and MCELLB0-7 (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 OMC on the trailing edge of Write Strobe coming from CNTL0.

Figure 9. CPLD Output Macrocell

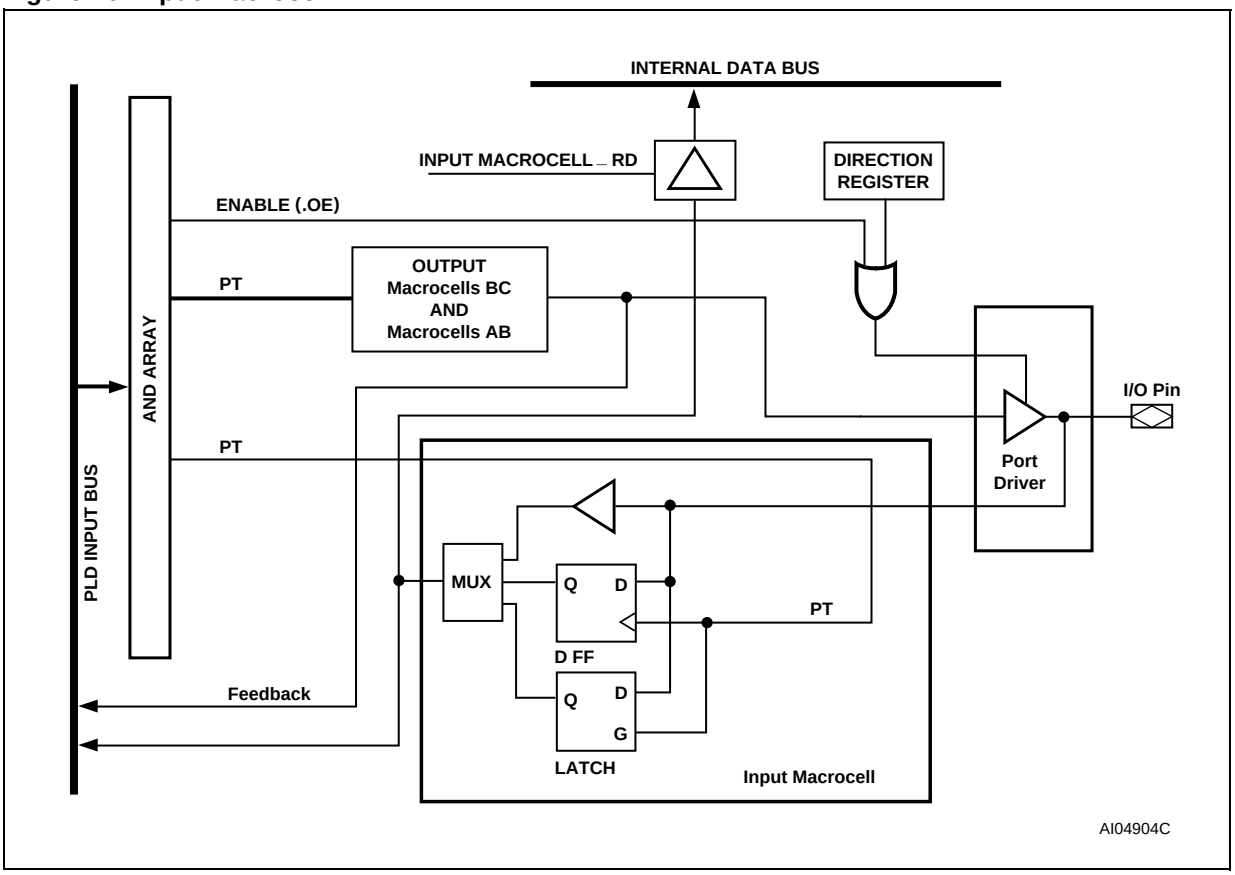


The OMC Mask Register. There is one Mask Register for each of the two groups of eight OMCs. The Mask Registers can be used to block the loading of data to individual OMCs. The default value for the Mask Registers is 00h, which allows loading of all the OMCs. When a given bit in a Mask Register is set to a 1, the DSP is blocked from writing to the associated OMC. For example, suppose McellA0-3 are being used for a state machine. You would not want a DSP write to McellA to overwrite the state machine registers. Therefore, you would want to load the Mask Register for McellA group with the value 0Fh.

The Output Enable of the OMC. The 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 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 10. Input Macrocell



Input Macrocells (IMC). The CPLD has 24 IMCs, one for each pin on Ports A, B and C. The architecture of the IMCs is shown in Figure 10. 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 Appendices 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™.

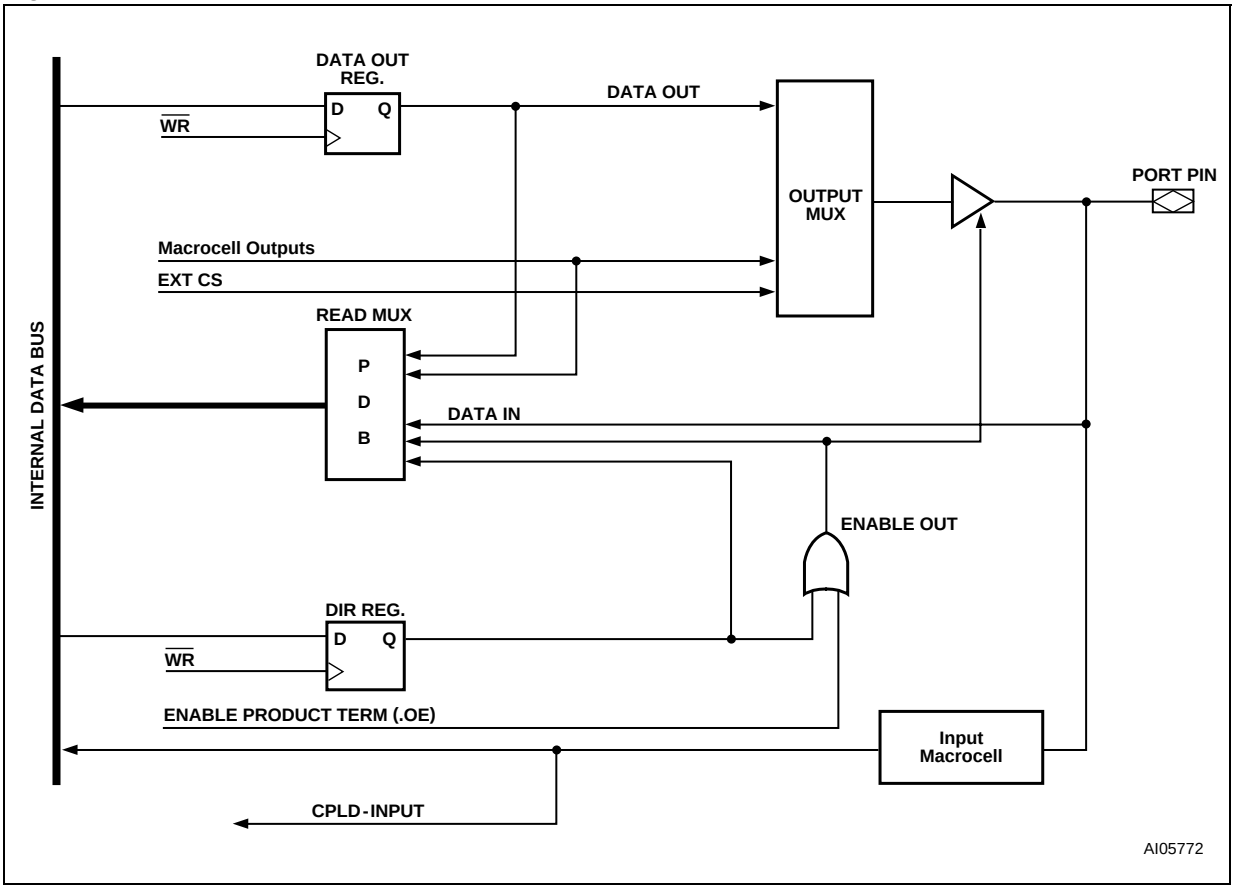
I/O Ports

There are seven programmable I/O ports: Ports A, B, C, D, E, F, and G. However, typically only four of these ports are available in 8-bit DSP data configuration, and 3 ports with 16-bit data. Each of the ports is eight bits except Port D, which is 4 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
- *csiop* Port registers
- Port Data Registers
- Individual Port functionality.

Figure 11. General Port Architecture



General Port Architecture. The general architecture of the I/O Port block is shown in Figure 12. Individual Port architectures are shown in Figure n to Figure 14. 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.

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 ESC0-7 from the DPLD to Port C 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 outputs, Direction and Drive 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 A, B, and C have 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 25.

Table 12. Port Operating Modes

Port Mode	Port A	Port B	Port C	Port D	Port E	Port F	Port G
MCU I/O	Yes	Yes	Yes	Yes	Yes	No	Yes[2]
DSP data bus for 8-bit config	No	No	No	No	No	Yes	No
DSP data bus for 16-bit config	No	No	No	No	No	Yes	Yes
PLD Input though IMC	Yes	Yes	Yes	No	No	No	No
PLD Input directly	No	No	No	Yes	No	No	No
McellA Outputs	Yes	No	No	No	No	No	No
McellB Outputs	No	Yes	No	No	No	No	No
Additional External CS Outputs	No	No	Yes	No	No	No	No
JTAG ISP	No	No	No	No	Yes[1]	No	No

Note: 1. Can be multiplexed with other I/O functions.
2. Only in 8-bit DSP data bus configuation.

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

MCU I/O Mode. In MCU I/O mode, DSP I/O Ports are expanded. 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* registers (Data In, Data Out, and Direction) that implement MCU I/O mode are defined Table 4 and Appendix A.

Data In Register for MCU I/O mode. 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 for MCU I/O Mode. 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 for MCU I/O mode. The Direction Register, in conjunction with the output enable, controls the direction of data flow in the I/O Ports. Any bit set to 1 in the Direction Register causes the corresponding pin to be an output, and any bit set to 0 causes it to be an input. The default mode for all port pins is input. Figure n shows the Port Architecture for Ports A, B and C. The direction of data flow for 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.

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. See Appendix A for Drive Register bit definitions.

DSP Data Bus. Port F is used for DSP data lines D0-D7 when DSM2150F5V is configured for 8-bit operation. Port G is additionally used for DSP data lines D8-D15 when configured for 16-bit operation.

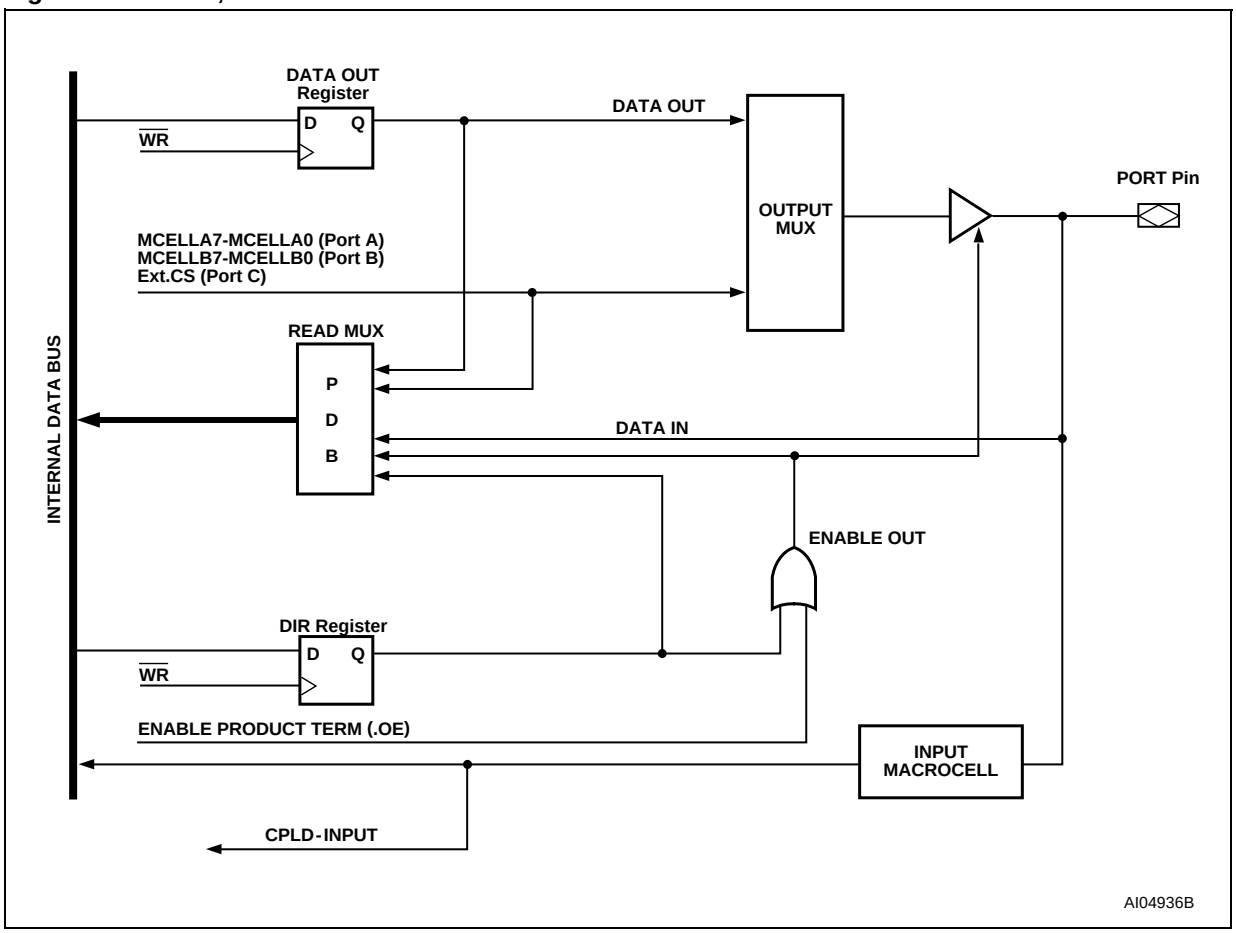
PLD Inputs. Inputs from Ports A, B, and C to the DPLD and CPLD come through IMCs. Inputs from Port D to PLDs are routed directly in and do not use IMCs.

PLD Outputs. Outputs from the CPLD to Port A come from the OMC group MCELLA0-7. Likewise, Port B is driven by MCELLB0-7. Outputs from the DPLD to Port C come from the external chip select logic block ECS0-7.

JTAG In-System Programming (ISP). Some of the pins on Port E implement IEEE 1194.1 JTAG bus for In-System Programming (ISP). You can multiplex the function of these Port E JTAG pins with other functions. See the section entitled “Programming In-Circuit Using JTAG ISP”, and Application Note *AN1153*.

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

Figure 12. Port A, B and C Structure

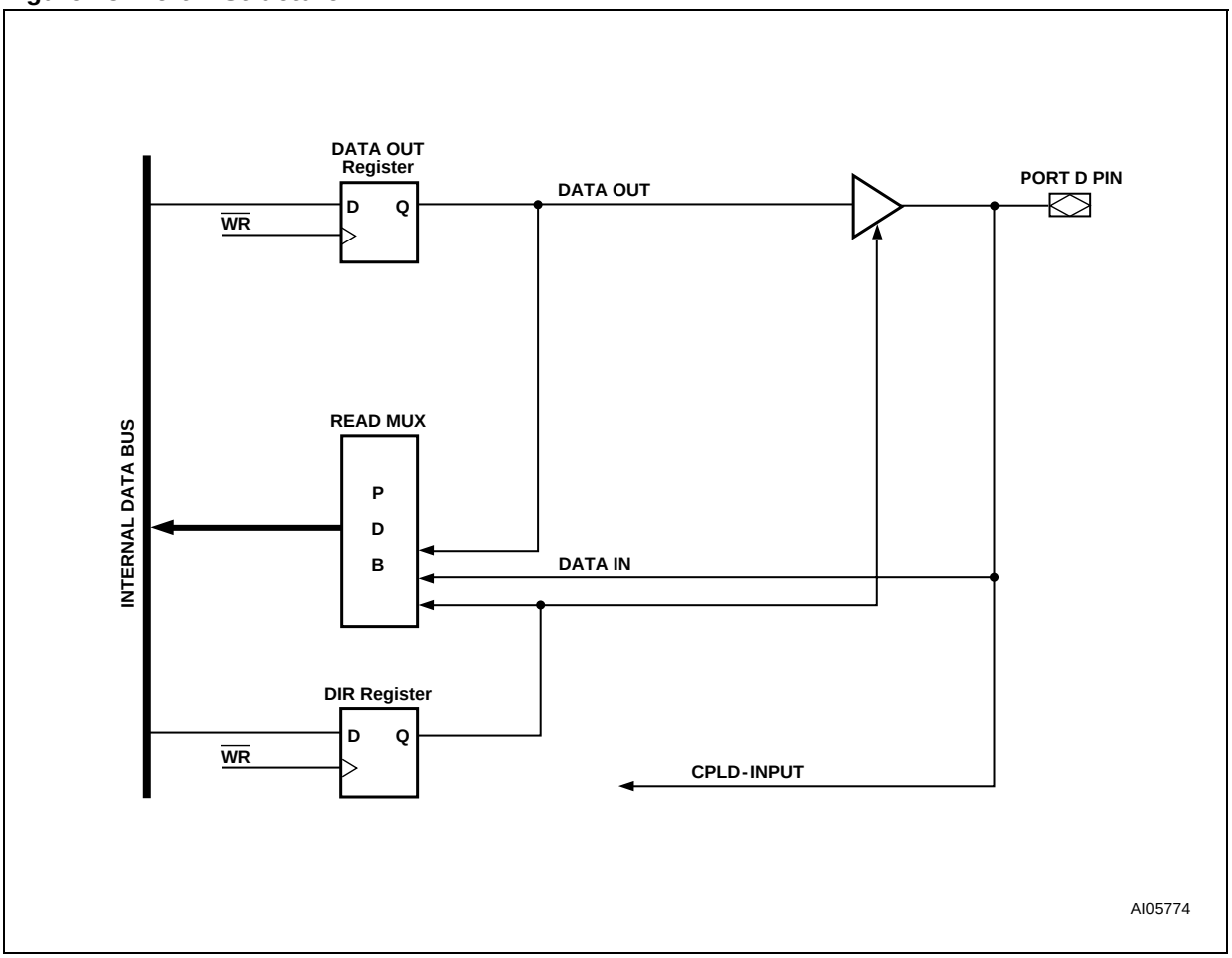


Ports A, B and C – Functionality and Structure

Ports A and B have similar functionality and structure, as shown in Figure 12. The two ports can be configured to perform one or more of the following functions:

- MCU I/O Mode
- CPLD Output – Macrocells McellA7-McellA0 can be connected to Port A. McellB7-McellB0 can be connected to Port B.
- DPLD Output - External Chip Select (ECS7-ECS0) can be connected to Port C.
- CPLD Input – Via the Input Macrocells (IMC).
- Open Drain/Slew Rate – pins PC7-PC0 can be configured to fast slew rate. Pins PA7-PA0, PB7-PB0, and PG7-PB0 and can be configured to Open Drain mode.

Figure 13. Port D Structure



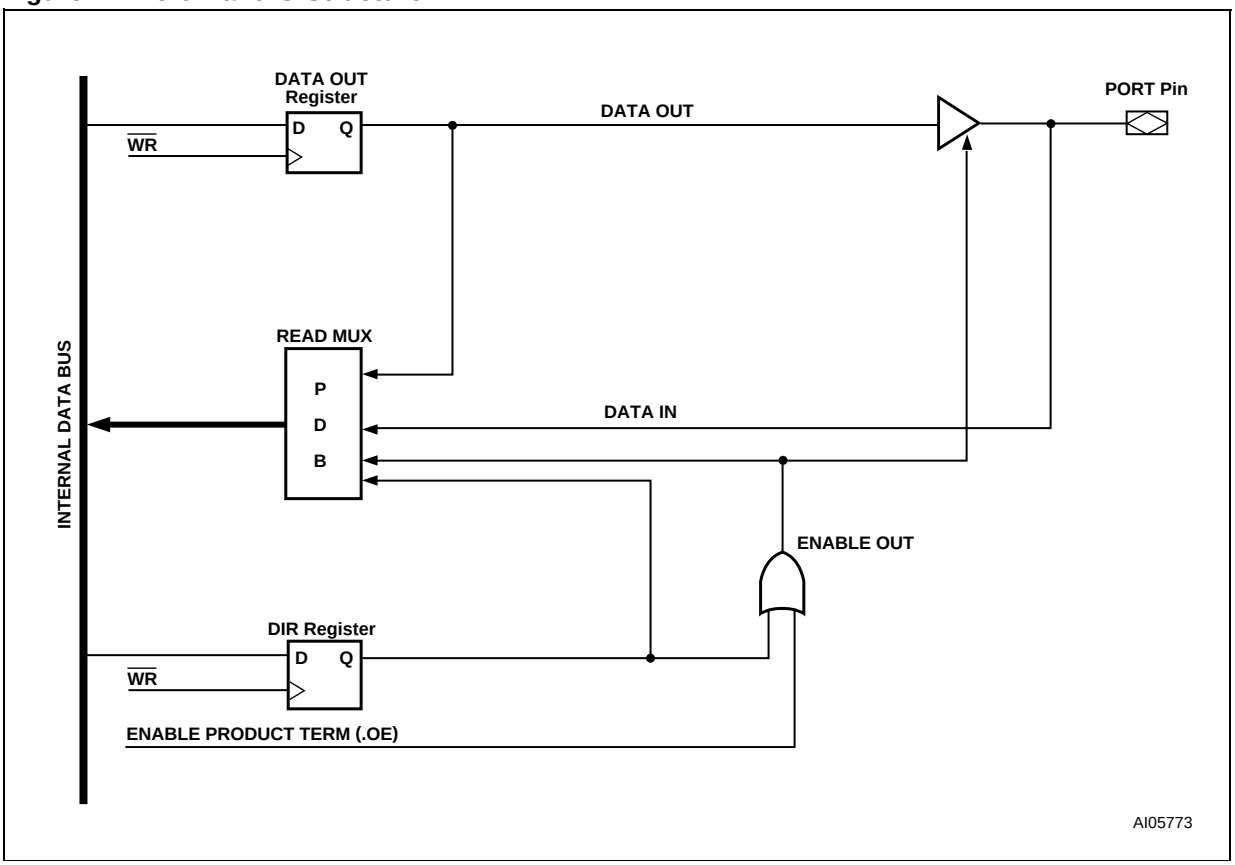
Port D – Functionality and Structure

Port D has four I/O pins. See Figure 13. Port D can be configured to perform one or more of the following functions:

- MCU I/O mode
- CPLD Input – direct input to the CPLD, no Input Macrocells (IMC)

  Port D pins can be configured in PSDsoft Express as input pins for other dedicated functions:
- CLKIN (PD1) as input to the Macrocells Flip-flops and APD counter
- PSD Chip Select Input ($\overline{CSI}$, PD2). Driving this signal High disables the Flash memory, SRAM and CSIOP.
- Write High-Byte input ($\overline{WRH}$, PD3) used for some 16-bit DSP connections.

Figure 14. Port E and G Structure



Port E – Functionality and Structure

Port E can be configured to perform one or more of the following functions (see Figure 14):

- MCU I/O Mode
- In-System Programming (ISP) – JTAG port can be enabled for programming/erase of the PSD device. (See the section entitled “Programming In-Circuit using JTAG ISP”, on page 33, for more information on JTAG programming.)
- Open Drain – pins can be configured in Open Drain Mode

Port F – Functionality and Structure

Port F will always be connected to DSP data bus D7-D0.

Port G – Functionality and Structure

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

- Connected to DSP data bus D15-D8 in 16-bit configuration.
- MCU I/O Mode in 8-bit configuration.
- Open Drain – pins can be configured in Open Drain Mode in 8-bit configuration.

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 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 address 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.
- 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 address 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.
- 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 16), so blocking unused PLD inputs can significantly lower PLD operating frequency and power consumption. The DSP also has the option of blocking certain PLD inputs when not needed, then letting them pass for when needed for specific logic operations. Table 4 and Appendix A define the PMMR registers.

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 Read Array 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.

Upon Power On reset, internal sector selects FS0-7 and CSBOOT0-7 must all be inactive and Write Strobe ($\overline{WR}$, CNTL0) inactive (logic 1) for maximum security of the data contents and to remove the possibility of a byte/word 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}.

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 15 shows the timing of the Power-up and warm reset.

I/O Pin, Register and PLD Status at Reset. Table 13 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.

Figure 15. Reset ($\overline{RESET}$) Timing

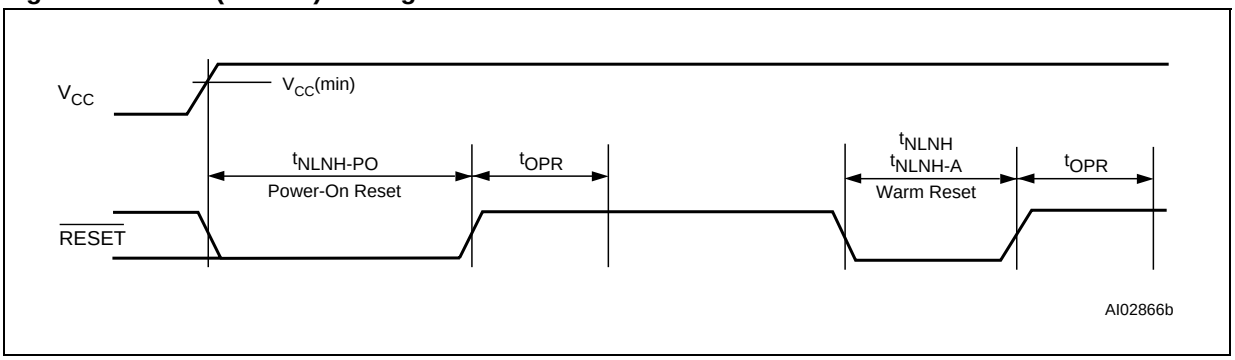


Table 13. Status During Power-On Reset, Warm Reset and Power-down Mode

Port Configuration	Power-On Reset	Warm Reset
MCU I/O	Input mode	Input mode
PLD Output	Valid after internal PSD configuration bits are loaded (almost immediately)	Valid

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

PROGRAMMING IN-CIRCUIT USING JTAG ISP

In-System Programming (ISP) can be performed through the JTAG signals on Port E. 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 15to 35 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 E, TSTAT and $\overline{\text{TERR}}$ in addition to TMS, TCK, TDI and TDO. See Table 14. The FlashLINK™ JTAG programming cable available from STMicroelectronics for $USD59 and PSDsoft Express software that is available at no charge from www.st.com/psd 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 E 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 A, B, C, or D must be dedicated to switch the Port E 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 E 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 14. JTAG Port Signals

Port E Pin	JTAG Signals	Description
PE0	TMS	Mode Select
PE1	TCK	Clock
PE2	TDI	Serial Data In
PE3	TDO	Serial Data Out
PE4	TSTAT	Status
PE5	$\overline{\text{TERR}}$	Error Flag

JTAG Extensions. 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.

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

TSTAT and $\overline{\text{TERR}}$ can be configured as open-drain type signals with PSDsoft Express. This facilitates a wired-OR connection of TSTAT signals from multiple DSM2150F5V devices and a wired-OR connection of $\overline{\text{TERR}}$ signals from those same

devices. This is useful when several devices are “chained” together in a JTAG environment. PSDsoft Express puts TSTAT and $\overline{\text{TERR}}$ 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

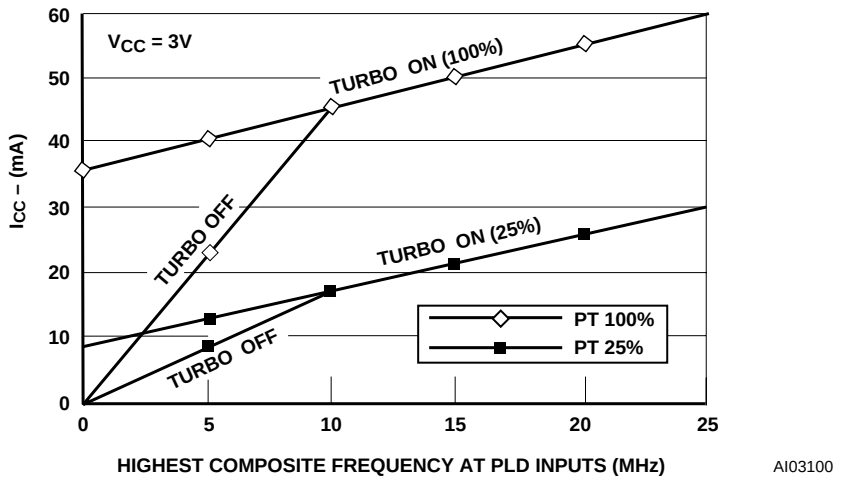
❏ AC Timing Specification

- PLD Timing
  - Combinatorial Timing
  - Synchronous Clock Mode
  - Asynchronous Clock Mode
  - Input MicroCell 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 16 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 16.
- In the PLD timing parameters, add the required delay when Turbo bit is 0.

Figure 16. 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 15. 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	4.0	V
V_{CC}	Supply Voltage	–0.6	4.0	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 16. 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 17. AC Measurement Conditions

Symbol	Parameter	Min.	Max.	Unit
C_L	Load Capacitance	30		pF

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

Figure 17. AC Measurement I/O Waveform

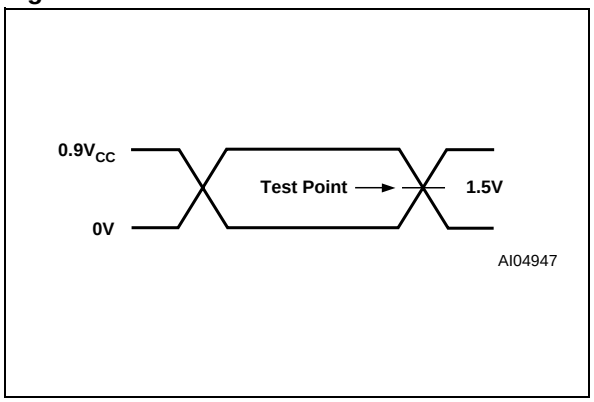


Figure 18. AC Measurement Load Circuit

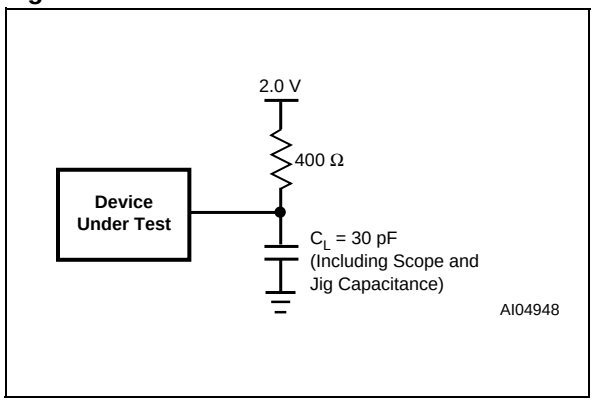


Table 18. Capacitance

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 19. 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 19. Switching Waveforms – Key

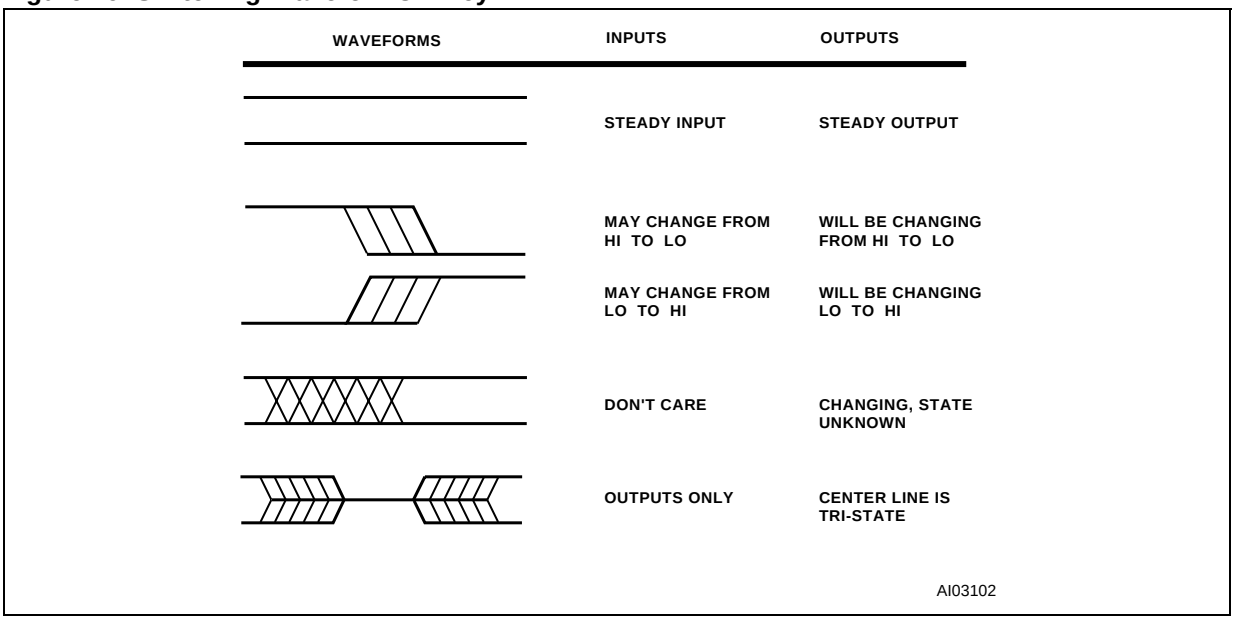


Table 20. DC Characteristics

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		I_{OH} = –20 µA, V_{CC} = 3.0 V	2.9	2.99		V
			I_{OH} = –1 mA, V_{CC} = 3.0 V	2.7	2.8		V
I_{STBY}	Stand-by Supply Current		$\overline{CSI}$ >V_{CC} –0.3 V (Note 2,3,4)		50	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 6)	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)	PLD AC Adder			(see note 5)			
	Flash memory AC Adder				1.5	2.0	mA/ MHz

Note: 1. $\overline{Reset}$ 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 ($\overline{CSI}$ >V_{CC} –0.3 V) or the DSP is not changing state of any address signal.
3. PLD is in non-Turbo mode, and none of the PLD inputs are switching.
4. No inputs floating, must be solid logic 1 or 0 (pullup to Vcc or gnd, or actively driven)
5. See Figure 16 for the PLD current calculation.
6. I_{OUT} = 0 mA, meaning outputs are driving no loads.

Table 21. CPLD Combinatorial Timing

Symbol	Parameter	Conditions	-12		PT Aloc	Turbo Off	Slew Rate[1]	Unit
			Min	Max				
t_{PD}	CPLD Input Pin/Feedback to CPLD Combinatorial Output			43	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 MicroCell		27	Add 4			ns
t_{TURBO}	Minimum time between switching of any PLD inputs which prevents PLD from entering standby mode.	PLD inputs that switch less frequently than this parameter allow standby PLD mode.	100					ns

Note: 1. Fast Slew Rate output available on Port C and Port F.

Table 22. CPLD MicroCell Synchronous Clock Mode Timing

Symbol	Parameter	Conditions	-12		PT Aloc	Turbo Off	Slew Rate[1]	Unit
			Min	Max				
f_{MAX}	Maximum Frequency External Feedback	$1/(t_S+t_{CO})$		20.4				MHz
	Maximum Frequency Internal Feedback (f_{CNT})	$1/(t_S+t_{CO}–10)$		25.6				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CH}+t_{CL})$		35.7				MHz
t_S	Input Setup Time		23		Add 4	Add 20		ns
t_H	Input Hold Time		0					ns
t_{CH}	Clock High Time	Clock Input	14					ns
t_{CL}	Clock Low Time	Clock Input	14					ns
t_{CO}	Clock to Output Delay	Clock Input		26			Sub 6	ns
t_{ARD}	CPLD Array Delay	Any MicroCell		27	Add 4			ns
t_{MIN}	Minimum Clock Period[2]	$t_{CH}+t_{CL}$	28					ns

Note: 1. Fast slew rate output available on Port C and Port F.
2. CLKIN (PD1) $t_{CLCL} = t_{CH} + t_{CL}$.

Table 23. CPLD MicroCell Asynchronous Clock Mode Timing

Symbol	Parameter	Conditions	-12		PT Aloc	Turbo Off	Slew Rate	Unit
			Min	Max				
f_{MAXA}	Maximum Frequency External Feedback	$1/(t_{SA}+t_{COA})$		20.8				MHz
	Maximum Frequency Internal Feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		26.3				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CHA}+t_{CLA})$		30.3				MHz
t_{SA}	Input Setup Time		10		Add 4	Add 20		ns
t_{HA}	Input Hold Time		12					ns
t_{CHA}	Clock High Time		18			Add 20		ns
t_{CLA}	Clock Low Time		15			Add 20		ns
t_{COA}	Clock to Output Delay			38		Add 20	Sub 6	ns
t_{ARD}	CPLD Array Delay	Any MicroCell		27	Add 4			ns
t_{MINA}	Minimum Clock Period	$1/f_{CNTA}$	38					ns

Figure 20. Input to Output Disable / Enable

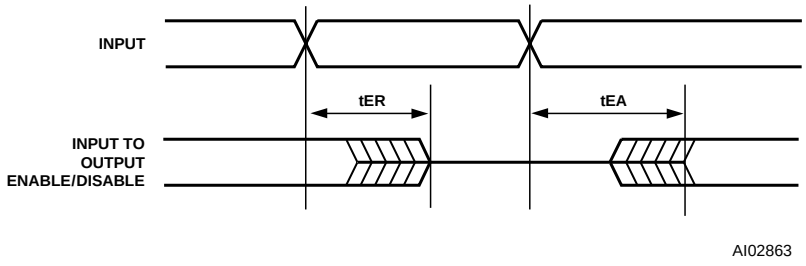


Figure 21. Asynchronous Reset / Preset

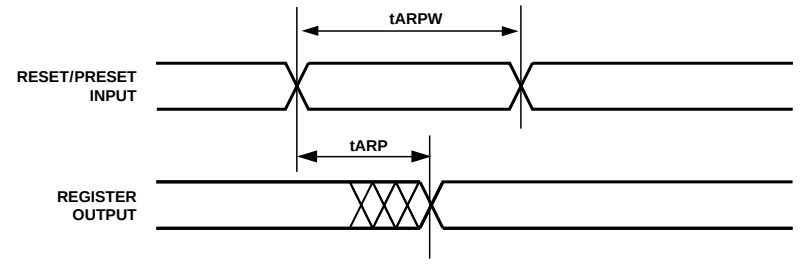


Figure 22. Synchronous Clock Mode Timing – PLD

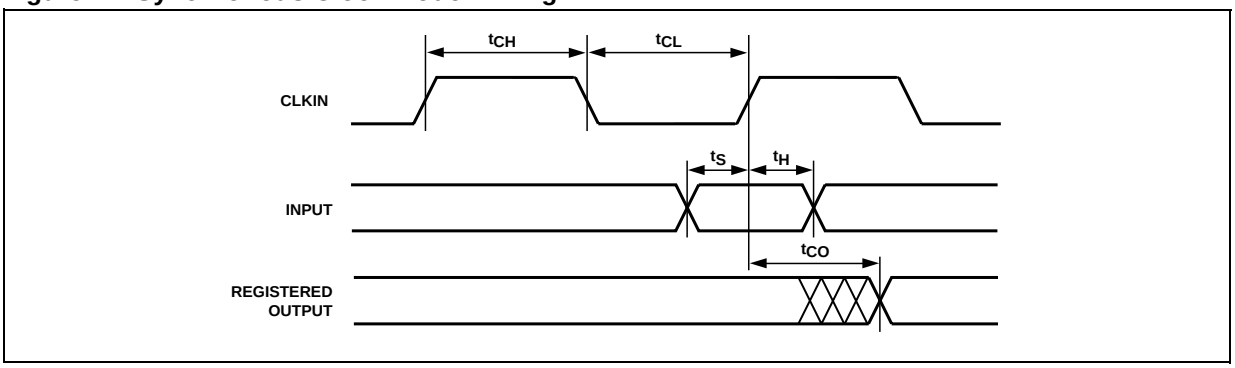


Figure 23. Asynchronous Clock Mode Timing (product term clock)

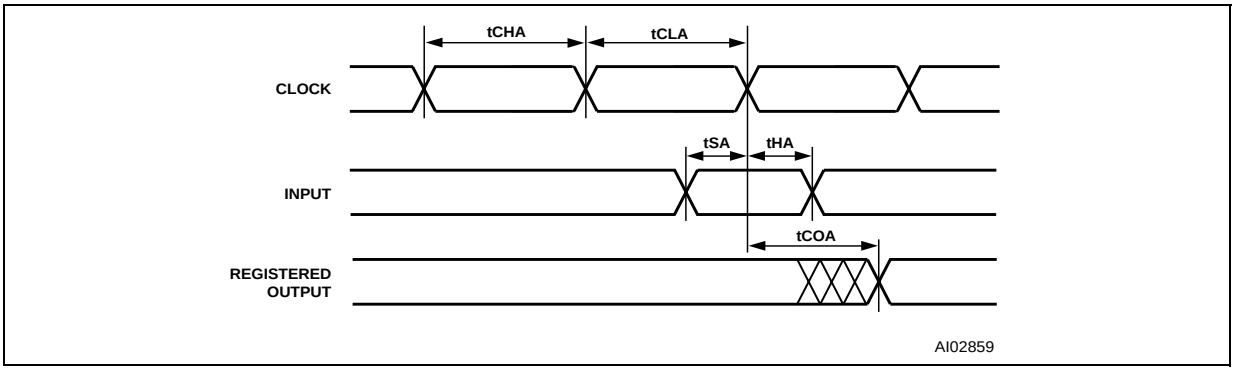


Table 24. Input MicroCell Timing

Symbol	Parameter	Conditions	-12		PT Aloc	Turbo Off	Unit
			Min	Max			
t_{IS}	Input Setup Time	(Note 1)	0				ns
t_{IH}	Input Hold Time	(Note 1)	23			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 A, B, and C relative to register/latch clock from the PLD.

Figure 24. Input MicroCell Timing (product term clock)

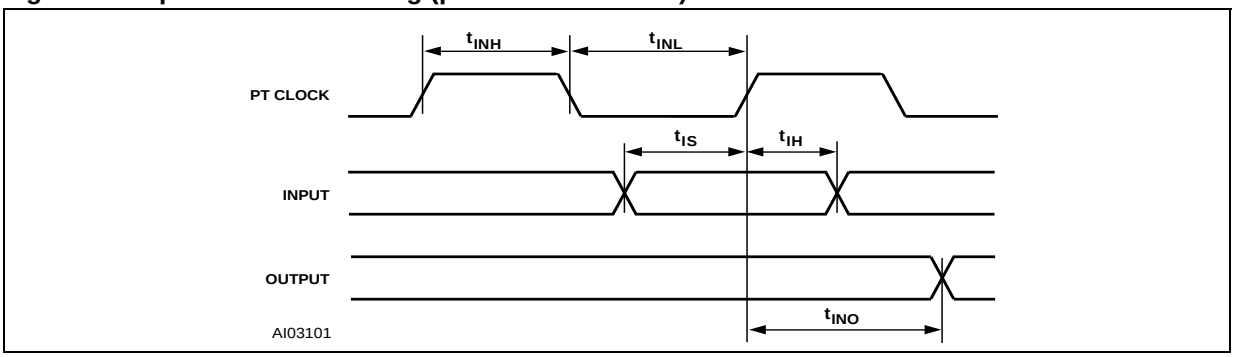


Table 25. Read Timing

Symbol	Parameter	Conditions	-12		Turbo Off	Unit
			Min	Max		
t_{AVQV}	Address Valid to Data Valid	(Note [1])		120	Add 20	ns
t_{SLQV}	CS Valid to Data Valid			120		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 25. Read Timing

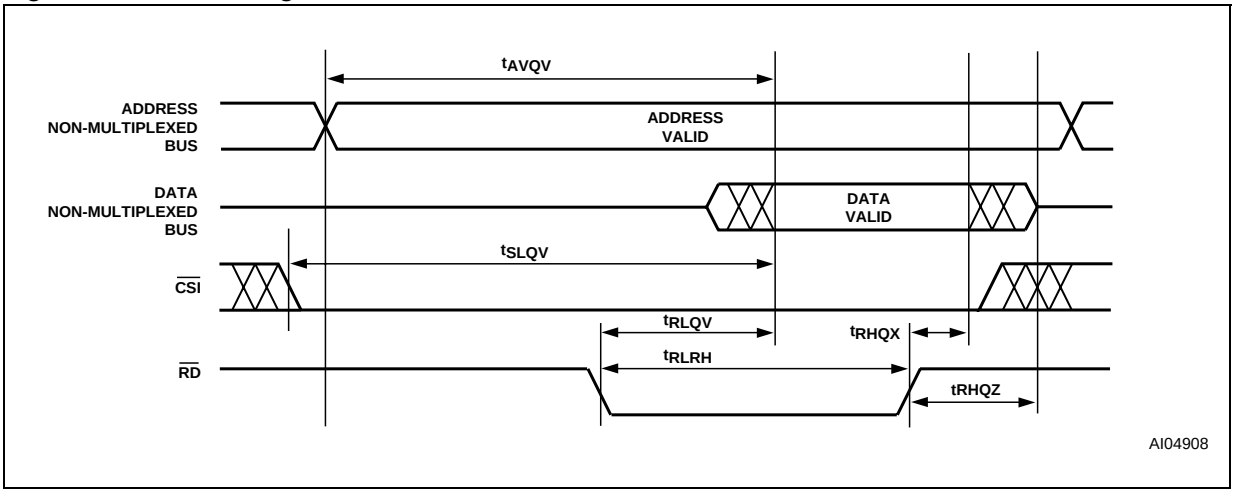


Table 26. Write Timing

Symbol	Parameter	Conditions	-12		Unit
			Min	Max	
t_{AVWL}	Address Valid to Leading Edge of $\overline{WR}$	(Note 1)	8		ns
t_{SLWL}	$\overline{CS}$ Valid to Leading Edge of $\overline{WR}$		8		ns
t_{DVWH}	$\overline{WR}$ Data Setup Time		45		ns
t_{WHDX_8}	$\overline{WR}$ Data Hold Time for 8-bit mode		5		ns
t_{WHDX_16}	$\overline{WR}$ Data Hold Time for 16-bit mode	(Note 5)	8		ns
t_{WLWH}	$\overline{WR}$ Pulse Width		45		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	(Note4)	0		ns
t_{WHPV}	Trailing Edge of $\overline{WR}$ to Port Output Valid Using I/O Port Data Register			33	ns
t_{DVMV}	Data Valid to Port Output Valid Using MicroCell Register Preset/Clear	(Note 3)		68	ns
t_{WLMV}	$\overline{WR}$ Valid to Port Output Valid Using MicroCell 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.
5. tWHAX_16 is 11 ns when writing to the Output Microcells

Figure 26. Write Timing

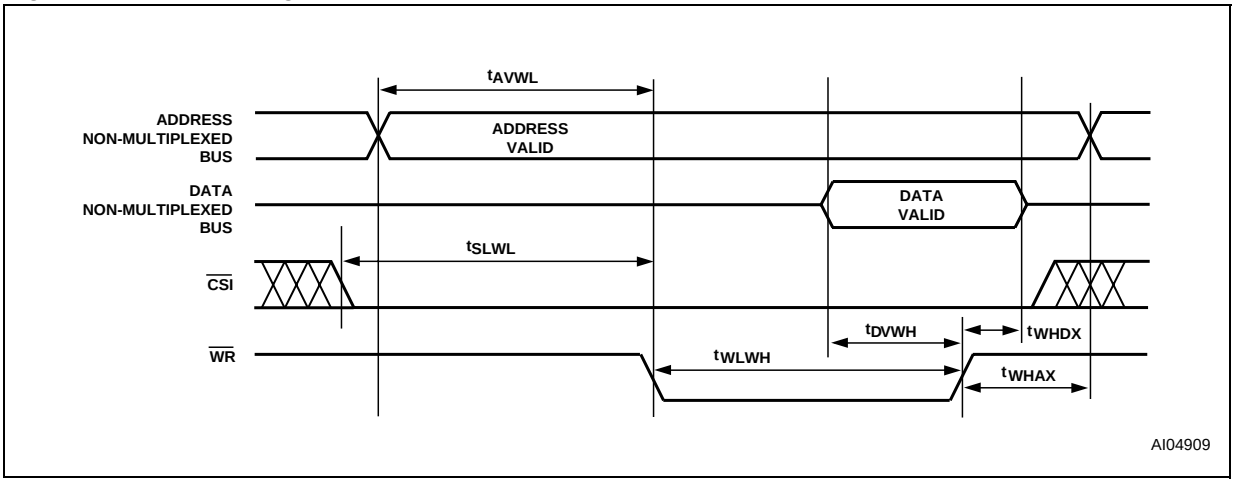


Table 27. Flash Memory Program, Write and Erase Times

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Flash Bulk Erase[1] (pre-programmed)		3	30	s
	Flash Bulk Erase (not pre-programmed)		10		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
$t_{TIMEOUT}$	Sector Erase Time-Out			80	µs
t_{Q7VQV}	DQ7 Valid to Output (DQ7-DQ0) Valid (Data Polling)[2]			30	ns
t_{TIMOUT}	Toggle Flag toggles after Suspend Sector Erase Instruction	0.1		15	µs

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.

Figure 27. Reset ($\overline{RESET}$) Timing

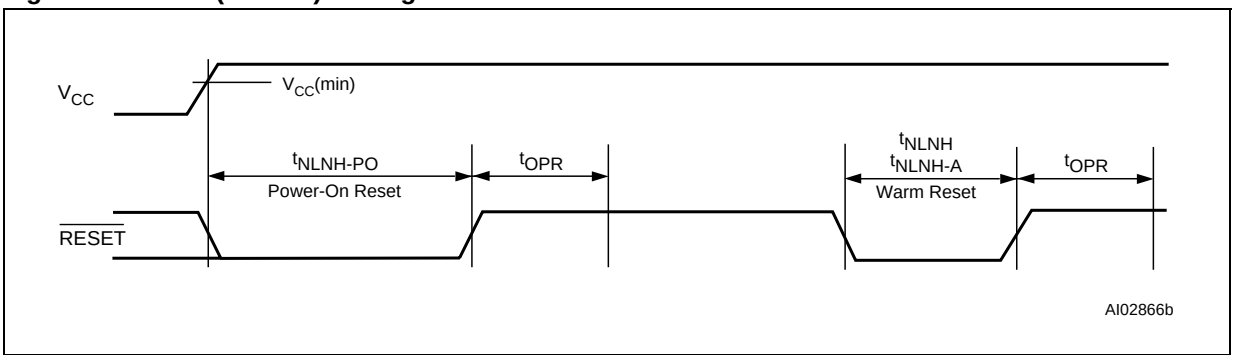


Table 28. Reset ($\overline{Reset}$) Timing

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_{NLNH-A}	Warm Reset Active Low time [2]		25		us
t_{OPR}	RESET High to Operational Device			300	ns
t_{READ_ARRAY}	Flash memory returns to read mode after Flash Reset Intruction	If Flash Program, Erase, or Error condition was in progress		25	us

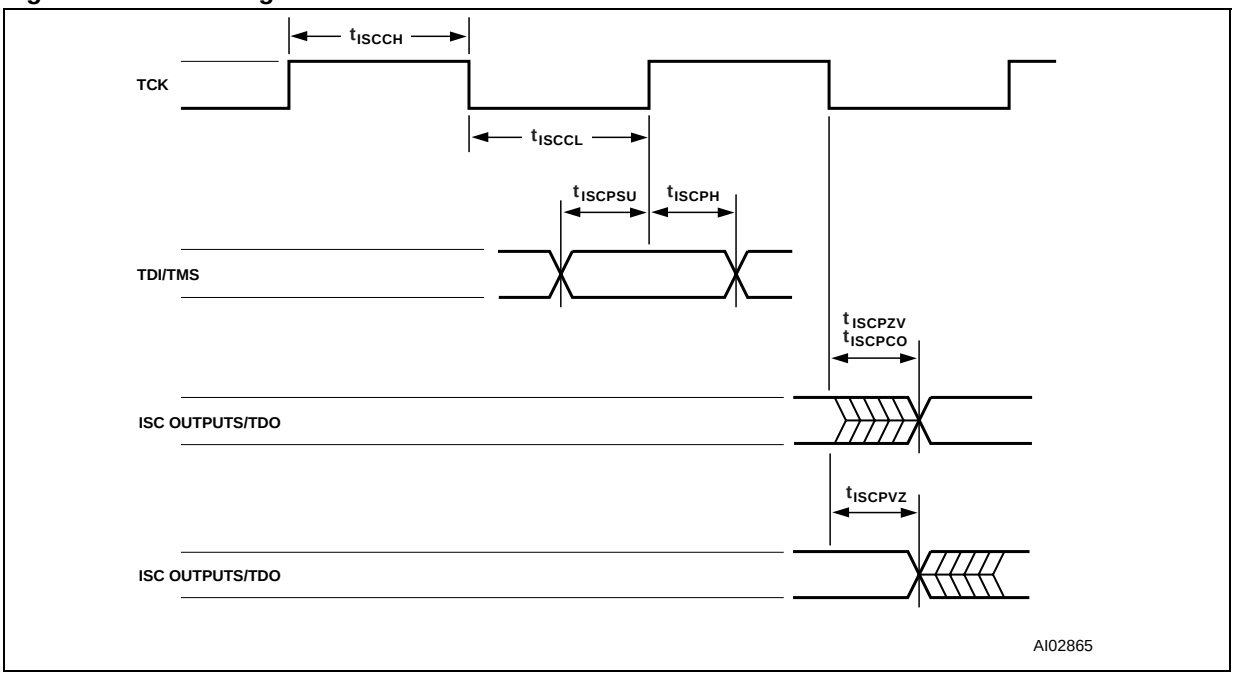
Note: 1. Reset ($\overline{RESET}$) does not reset Flash memory Program or Erase cycles.
2. Warm reset aborts Flash memory Program or Erase cycles, and puts the device in Read mode.

Table 29. ISC Timing

Symbol	Parameter	Conditions	-12		Unit
			Min	Max	
t_{ISCCF}	Clock (TCK, PC1) Frequency (except for PLD)	(Note 1)		12	MHz
t_{ISCCH}	Clock (TCK, PC1) High Time (except for PLD)	(Note 1)	40		ns
t_{ISCCL}	Clock (TCK, PC1) Low Time (except for PLD)	(Note 1)	40		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		12		ns
t_{ISCPH}	ISC Port Hold Up Time		5		ns
t_{ISCPCO}	ISC Port Clock to Output			32	ns
t_{ISCPZV}	ISC Port High-Impedance to Valid Output			32	ns
t_{ISCPVZ}	ISC Port Valid Output to High-Impedance			32	ns

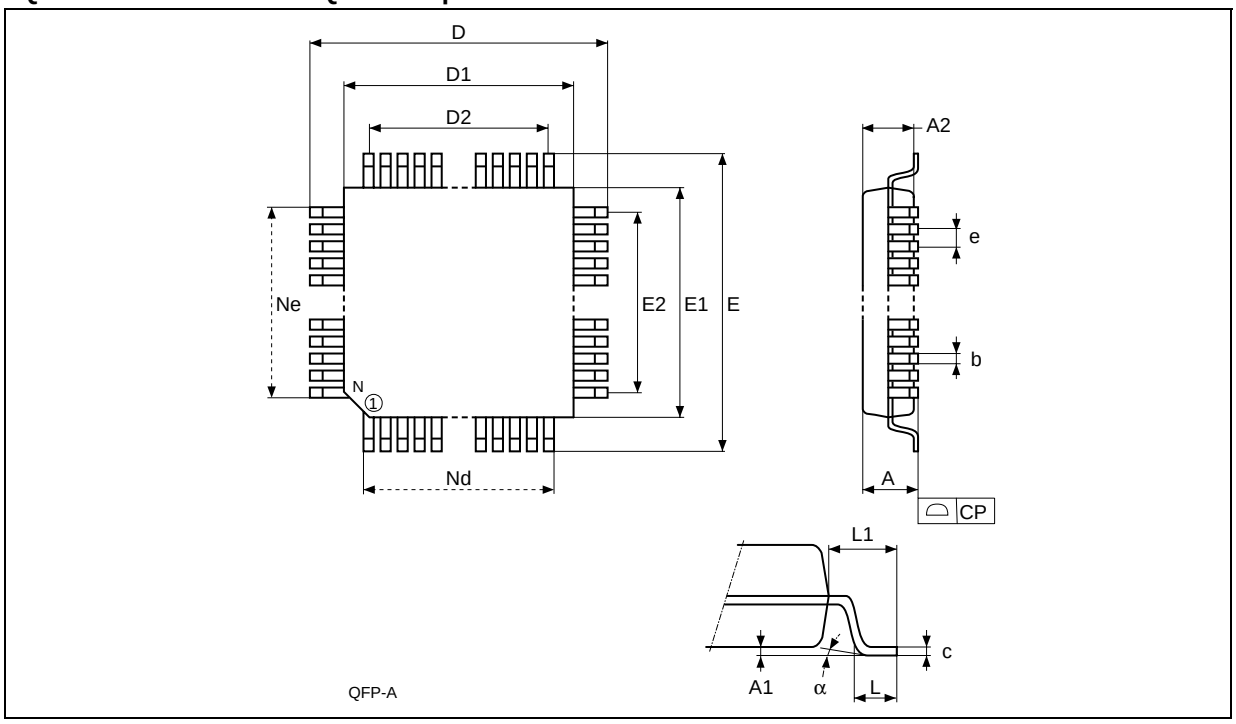
Note: 1. For non-PLD Programming, Erase or in by-pass mode.
2. For Program or Erase PLD only.

Figure 28. ISC Timing



PACKAGE MECHANICAL

TQFP80 - 80 lead Plastic Quad Flatpack



Note: Drawing is not to scale.

TQFP80 - 80 lead Plastic Quad Flatpack

Symb.	mm			inches		
	Typ.	Min.	Max.	Typ.	Min.	Max.
A			1.200			0.0472
A1		0.050	0.150		0.0020	0.0059
A2		0.950	1.050		0.0374	0.0413
α	3.5°	0.0°	7.0°	3.5°	0.0°	7.0°
b	0.220	0.170	0.270	0.0087	0.0067	0.0106
c		0.090	0.200		0.0035	0.0079
D	14.000			0.5512		
D1	12.000			0.4724		
D2	9.500	—	—	0.3740	—	—
E	14.000			0.5512		
E1	12.000			0.4724		
E2	9.500	—	—	0.3740	—	—
e	0.500	—	—	0.0197	—	—
L	0.600	0.450	0.750	0.0236	0.0177	0.0295
L1	1.000			0.0394		
CP	0.080			0.0031		
N	80			80		
Nd	20			20		
Ne	20			20		

Table 30. Pin Assignments – TQFP80

Pin No.	Pin Assignments	Pin No.	Pin Assignments	Pin No.	Pin Assignments	Pin No.	Pin Assignments
1	PD2	21	PG0	41	PC0	61	PB0
2	PD3	22	PG1	42	PC1	62	PB1
3	AD0	23	PG2	43	PC2	63	PB2
4	AD1	24	PG3	44	PC3	64	PB3
5	AD2	25	PG4	45	PC4	65	PB4
6	AD3	26	PG5	46	PC5	66	PB5
7	AD4	27	PG6	47	PC6	67	PB6
8	GND	28	PG7	48	PC7	68	PB7
9	V_{CC}	29	V_{CC}	49	GND	69	V_{CC}
10	AD5	30	GND	50	GND	70	GND
11	AD6	31	PF0	51	PA0	71	PE0
12	AD7	32	PF1	52	PA1	72	PE1
13	AD8	33	PF2	53	PA2	73	PE2
14	AD9	34	PF3	54	PA3	74	PE3
15	AD10	35	PF4	55	PA4	75	PE4
16	AD11	36	PF5	56	PA5	76	PE5
17	AD12	37	PF6	57	PA6	77	PE6
18	AD13	38	PF7	58	PA7	78	PE7
19	AD14	39	$\overline{\text{RESET}}$	59	CNTL0	79	PD0
20	AD15	40	CNTL2	60	CNTL1	80	PD1

PART NUMBERING

Table 31. Ordering Information Scheme

Example: DSM21 50 F5 V - 12 T 6

Device Type
DSM21=DSP System Memory for ADSP-21XXX Famly

DSM Series
50 = 4.25M bit, Dual Array Flash, 40 I/O

Main Flash Memory Density
F5 = 4Mbit

Operating Voltage (Vcc)
V = 3.3V ± 10%

Access Time
12 = 120 nsec

Package
T = 80-pin TQFP

Temperature Range
6 = –40 to 85°C (Industrial)

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.

APPENDIX A. *CSIOP* REGISTER BIT DEFINITIONS

Table 32. Data-In Registers – Ports A, B, C, D, E, G

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

Note: Bit Definitions (Read-only registers):
Read Port pin status when Port is in MCU I/O input mode.

Table 33. Data-Out Registers – Ports A, B, C, D, E, G

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

Note: Bit Definitions:
Latched data for output to Port pin when pin is configured in MCU I/O output mode.

Table 34. Direction Registers – Ports A, B, C, D, E, G

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

Note: Bit Definitions:
Port pin <**i**> 0 = Port pin <i> is configured in Input mode (default).
Port pin <**i**> 1 = Port pin <i> is configured in Output mode.

Table 35. Drive Registers – Ports A, B, E, G

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

Note: Bit Definitions:
Port pin <**i**> 0 = Port pin <i> is configured for CMOS Output driver (default).
Port pin <**i**> 1 = Port pin <i> is configured for Open Drain output driver.

Table 36. Drive Registers – Port C

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

Note: Bit Definitions:
Port pin <**i**> 0 = Port pin <i> is configured for CMOS Output driver (default).
Port pin <**i**> 1 = Port pin <i> is configured in Slew Rate mode.

Table 37. Enable-Out Registers – Ports A, B, C

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

Note: Bit Definitions (Read-only registers):
Port pin <**i**> 0 = Port pin <i> is in tri-state driver (default).
Port pin <**i**> 1 = Port pin <i> is enabled.

Table 38. Input Macrocells – Ports A, B, C

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

Note: Bit Definitions (Read-only registers):
Read Input Macrocell (IMC7-IMC0) status on Ports A, B and C.

Table 39. Output Macrocells A Register

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

Note: Bit Definitions:
Write Register: Load MCellA7-MCellA0 with 0 or 1.
Read Register: Read MCellA7-MCellA0 output status.

Table 40. Output Macrocells B Register

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

Note: Bit Definitions:
Write Register: Load MCellB7-MCellB0 with 0 or 1.
Read Register: Read MCellB7-MCellB0 output status.

Table 41. Mask Macrocells A Register

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

Note: Bit Definitions:
McellA<i>_Prot 0 = Allow MCellA<i> flip-flop to be loaded by MCU (default).
McellA<i>_Prot 1 = Prevent MCellA<i> flip-flop from being loaded by DSP.

Table 42. Mask Macrocells B Register

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

Note: Bit Definitions:
McellB<i>_Prot 0 = Allow MCellB<i> flip-flop to be loaded by MCU (default).
McellB<i>_Prot 1 = Prevent MCellB<i> flip-flop from being loaded by DSP.

Table 43. Flash Memory Protection Register

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

Note: Bit Definitions (Read-only register):
Sec<i>_Prot 1 = Primary Flash memory Sector <i> is write protected.
Sec<i>_Prot 0 = Primary Flash memory Sector <i> is not write protected.

Table 44. Flash Boot Protection Register

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

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

Table 45. JTAG Enable Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used	not used	not used	not used	not used	not used	not used	JTAGEnable

Note: Bit Definitions:
JTAGEnable 1 = JTAG Port is enabled.
JTAGEnable 0 = JTAG Port is disabled.

Table 46. Page Register

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

Note: Bit Definitions:
Configure Page input to PLD. Default is PGR7-PGR0=0.

Table 47. PMMR0 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used (set to 0)	not used (set to 0)	PLD MCells CLK	PLD Array CLK	PLD Turbo	not used (set to 0)	not used (set to 0)	not used (set to 0)

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

Note: Bit Definitions:

PLD Turbo 0 = PLD Turbo is on.
1 = PLD Turbo is off, saving power.

PLD Array CLK 0 = CLKIN to the PLD AND array is connected. Every CLKIN change powers up the PLD when Turbo bit is off.
1 = CLKIN to the PLD AND array is disconnected, saving power.

PLD MCells CLK 0 = CLKIN to the PLD Macrocells is connected.
1 = CLKIN to the PLD Macrocells is disconnected, saving power.

Table 48. PMMR2 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used (set to 0)	PLD Array WRH	not used (set to 0)	PLD Array CNTL2	PLD Array CNTL1	PLD Array CNTL0	not used (set to 0)	PLD Array Addr

Note: For Bit 4, Bit 3, Bit 2: See Table 47 for the signals that are blocked on pins CNTL0-CNTL2.

Note: Bit Definitions:

PLD Array Addr 0 = Address A7-A0 are connected to the PLD array.
1 = Address A7-A0 are blocked from the PLD array, saving power.

PLD Array CNTL2 0 = CNTL2 input to the PLD AND array is connected. Every CLKIN change powers up the PLD when Turbo bit is off.
1 = CNTL2 input to the PLD AND array is disconnected, saving power.

PLD Array CNTL1 0 = CNTL1 input to the PLD AND array is connected. Every CLKIN change powers up the PLD when Turbo bit is off.
1 = CNTL1 input to the PLD AND array is disconnected, saving power.

PLD Array CNTL0 0 = CNTL0 input to the PLD AND array is connected. Every CLKIN change powers up the PLD when Turbo bit is off.
1 = CNTL0 input to the PLD AND array is disconnected, saving power.

PLD Array WRH 0 = WRH input to the PLD AND array is connected.
1 = WRH input to the PLD AND array is disconnected, saving power.

Table 49. Memory_ID0 Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used (set to 0)	not used (set to 0)	not used (set to 0)	not used (set to 0)	F_size 3	F_size 2	F_size 1	F_size 0

Note: Bit Definitions:

F_size[3:0] 5h = Primary Flash memory size is 4 Mbit
6h = Primary Flash memory size is 8 Mbit

Table 50. Memory_ID1 Register

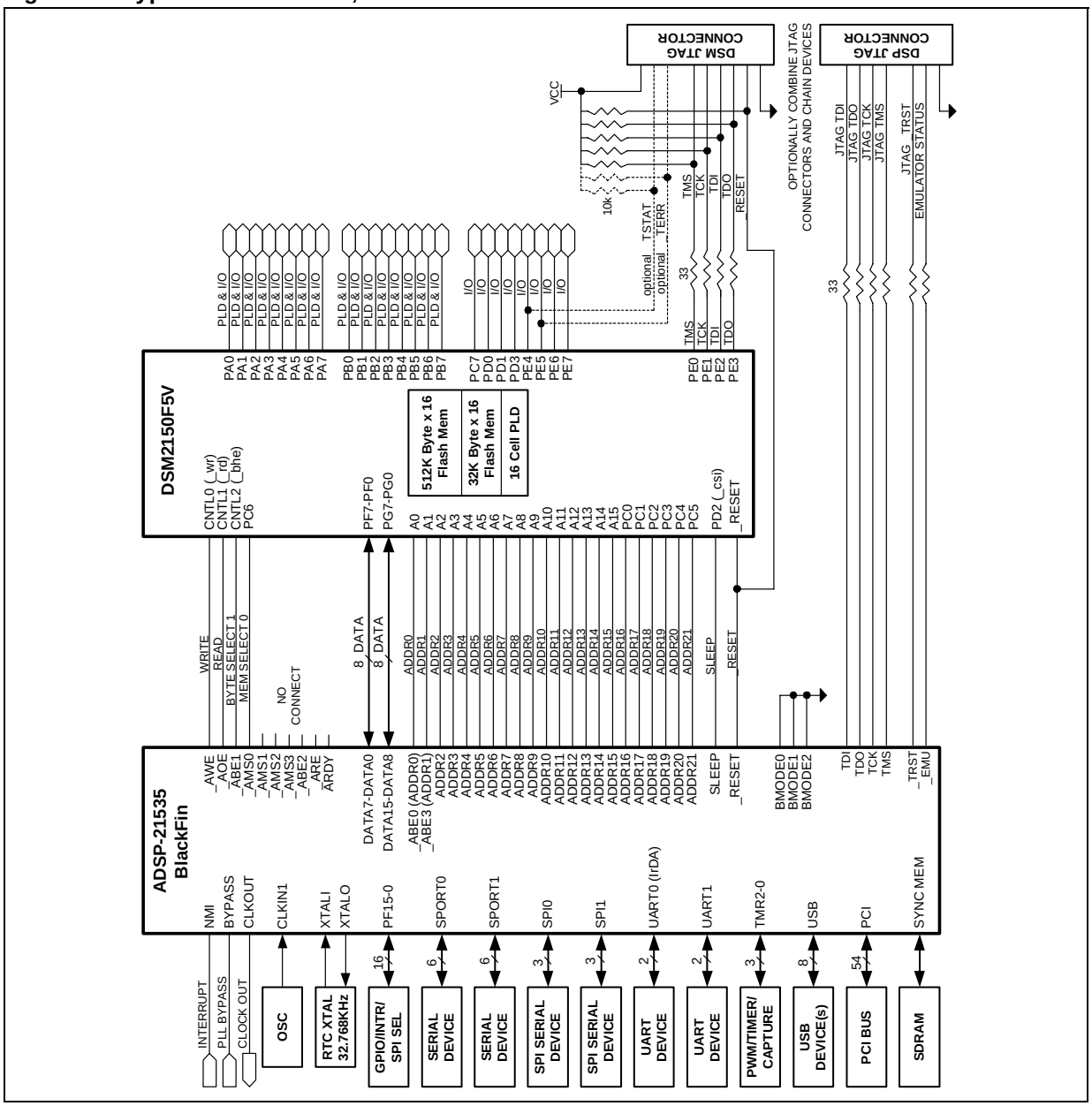
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
not used (set to 0)	not used (set to 0)	not used (set to 0)	not used (set to 0)	B_size 3	B_size 2	B_size 1	B_size 0

Note: Bit Definitions:

B_size[3:0] 2h = Secondary NVM size is 256 Kbit
3h = Secondary NVM size is 512 Kbit

APPENDIX B. TYPICAL CONNECTIONS, DSM2150F5V AND ADSP-21535 BLACKFIN DSP

Figure 29. Typical Connections, DSM2150F5V and ADSP-21535 Blackfin DSP



Typical Memory Map, DSM2150F5V and ADSP21535 BLACKFIN DSP

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 30 shows one possible system memory map.

In this example, the DSP will bypass it's internal boot ROM at power-on and begin executing code directly from the DSM2150F5V secondary Flash memory. While executing this code, the DSP will load the contents of the DSM2150F5V main Flash memory into the ADSP-21535 internal SRAM, then execute code from that high performance SRAM.

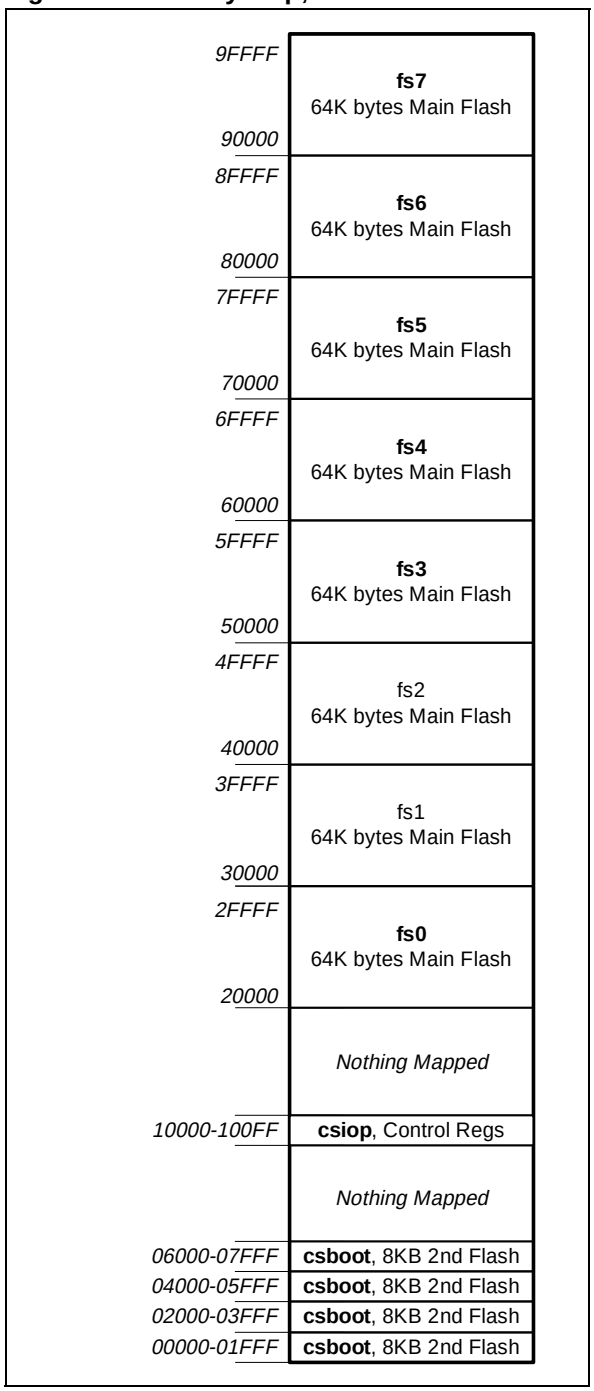
The advantage of this is speed, flexibility, IAP, clean software partitioning, and parameter storage.

- Loading external Flash memory to internal SRAM by 16-bits is faster than booting by 8-bits. Also, subsequent loading of new memory overlays during runtime is also faster by 16-bits.
- Bypassing internal DSP boot ROM and executing from DSM secondary memory provides total flexibility to meet system requirements. Like having custom boot ROM programmable by JTAG.
- In-Application Programming (IAP) can be implemented by placing custom loader code in DSM secondary flash which, when executed, allows the DSP to receive data over any communication channel (i.e. USB) and write new code/data the DSM main flash memory. Since the DSM Flash arrays are independent, it is possible to read from the secondary flash while writing to the main Flash.
- Since the DSM secondary Flash has smaller sector sizes, small data sets and calibration constants may be stored there. EEPROM emulation techniques can be used.
- Placing start-up and IAP code in DSM secondary Flash keeps it totally separate DSM main flash memory, affording clean software partitioning. This also ensures robust system operaton since start-up code will always be there and removed from accitental writes or erasures of DSM main flash.

The nomenclature *fs0..fs7* in Figure 30 are designators for the individual sectors of Main Flash memory, 64 KBytes each. *csboot0..csboot3* are designators for the individual Secondary Flash memory segments, 8 KBytes 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 30. Memory Map, ADSP-21535



Specifying the Memory Map with PSDsoft Express™

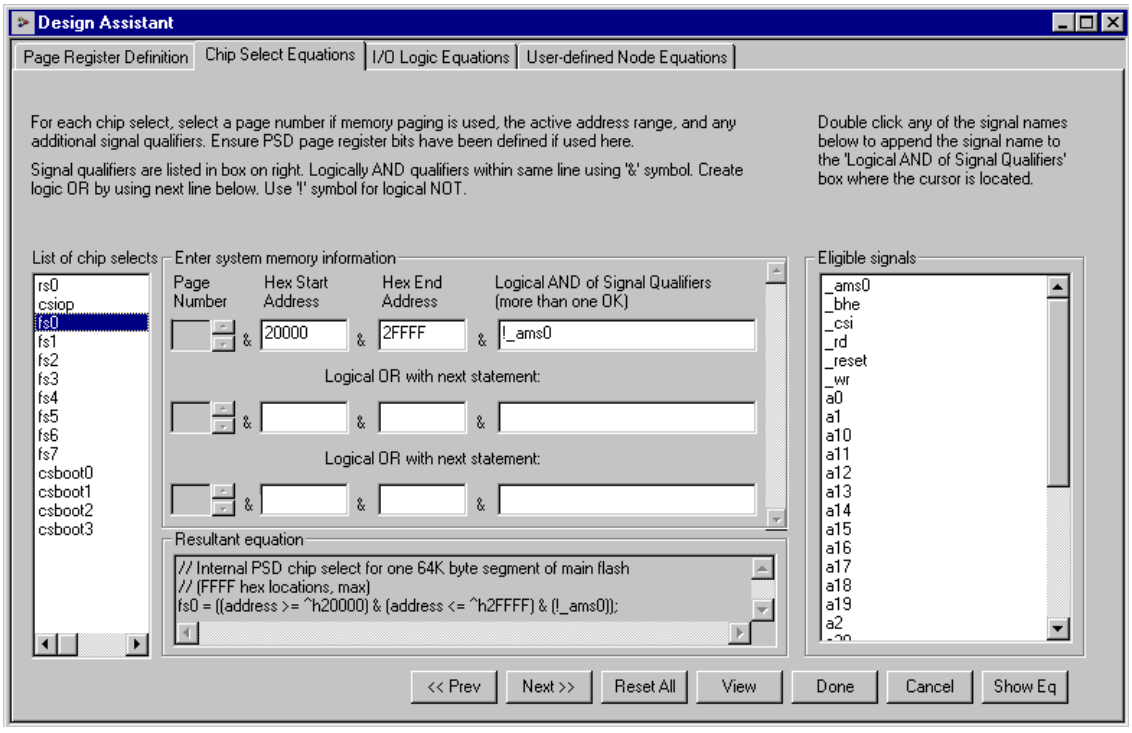
The memory map shown in Figure 30 can be easily implemented using PSDsoft Express™ in a point-and-click environment. PSDsoft Express™ will generate Hardware Definition Language (HDL) statements of the ABEL language. Figure 31 shows the resulting equations generated by PSDsoft Express™.

Figure 31. HDL Statements Generated from PSDsoft Express to Implement Memory Map

```
csiop = ((address >= ^h10000) & (address <= ^h100FF) & (!_ams0));
fs0 = ((address >= ^h20000) & (address <= ^h2FFFF) & (!_ams0));
fs1 = ((address >= ^h30000) & (address <= ^h3FFFF) & (!_ams0));
fs2 = ((address >= ^h40000) & (address <= ^h4FFFF) & (!_ams0));
fs3 = ((address >= ^h50000) & (address <= ^h5FFFF) & (!_ams0));
fs4 = ((address >= ^h60000) & (address <= ^h6FFFF) & (!_ams0));
fs5 = ((address >= ^h70000) & (address <= ^h7FFFF) & (!_ams0));
fs6 = ((address >= ^h80000) & (address <= ^h8FFFF) & (!_ams0));
fs7 = ((address >= ^h90000) & (address <= ^h9FFFF) & (!_ams0));
csboot0 = ((address >= ^h0000) & (address <= ^h1FFF) & (!_ams0));
csboot1 = ((address >= ^h2000) & (address <= ^h3FFF) & (!_ams0));
csboot2 = ((address >= ^h4000) & (address <= ^h5FFF) & (!_ams0));
csboot3 = ((address >= ^h6000) & (address <= ^h7FFF) & (!_ams0));
```

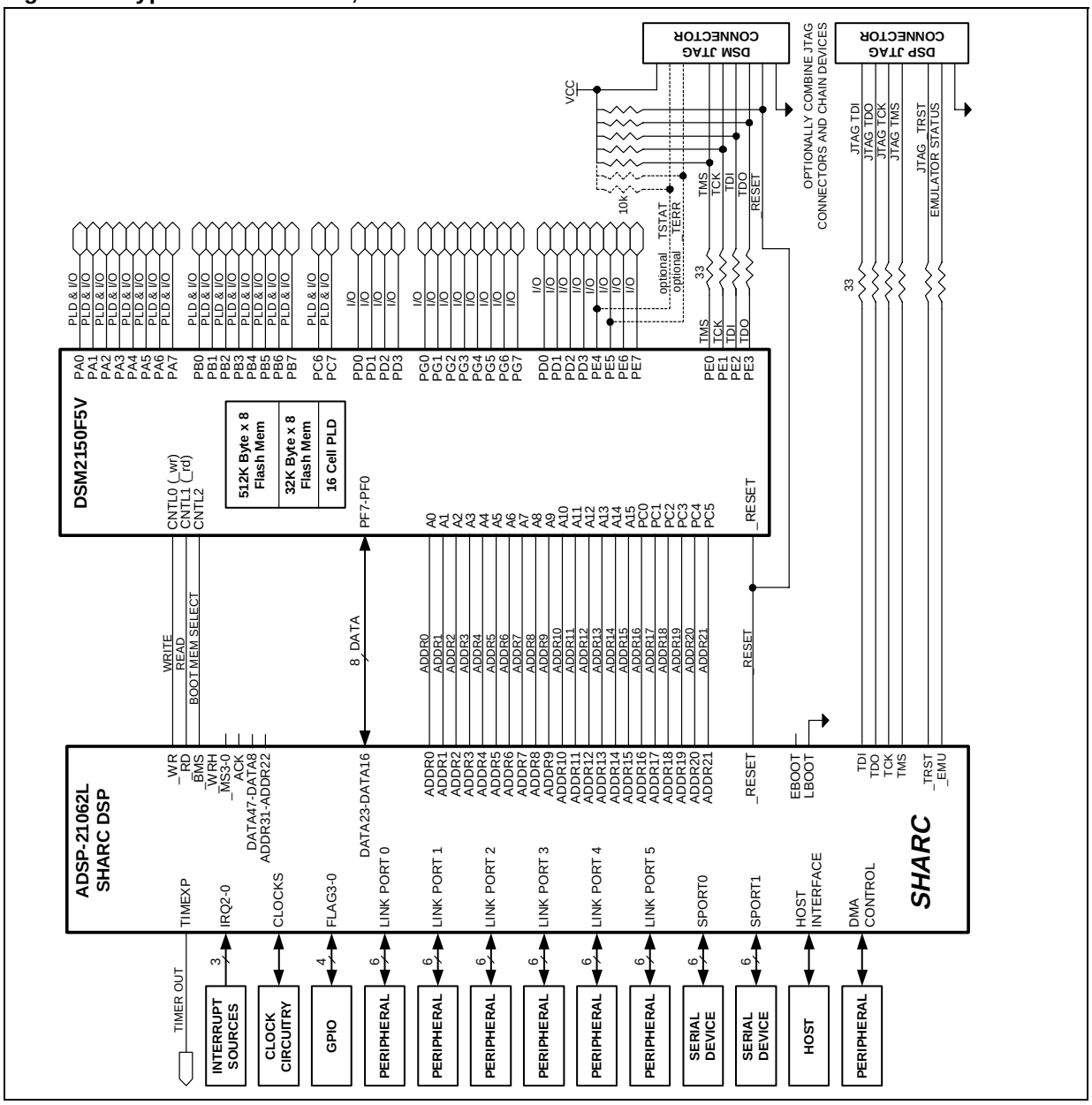
Specifying these equations using PSDsoft Express™ is very simple. Figure 32 shows how to specify the equation for the 64 KByte Flash memory segment, *fs0*. Notice *fs0* is qualified with the signal $\overline{\text{AMS0}}$. This specification process is repeated for all other Flash memory segments, the *csiop* register block, and any external chip select signals that may be needed.

Figure 32. PSDsoft Express™ Memory Mapping



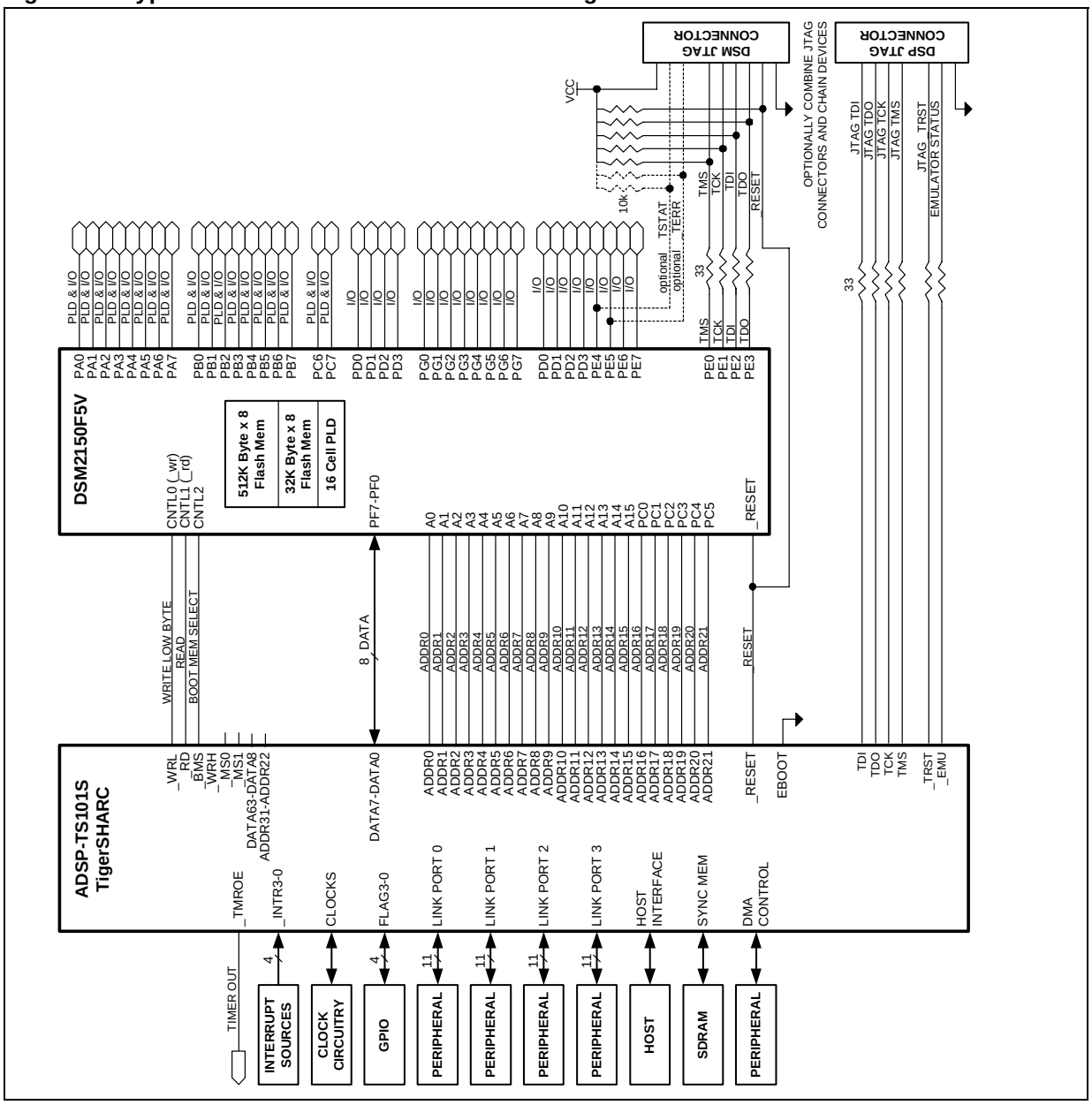
APPENDIX C. TYPICAL CONNECTIONS, DSM2150F5V AND ADSP-21062 SHARC DSP

Figure 33. Typical Connections, DSM2150F5V and ADSP-21062 Sharc DSP



APPENDIX D. TYPICAL CONNECTIONS, DSM2150F5V AND ADSP-TS101S TIGERSHARC DSP

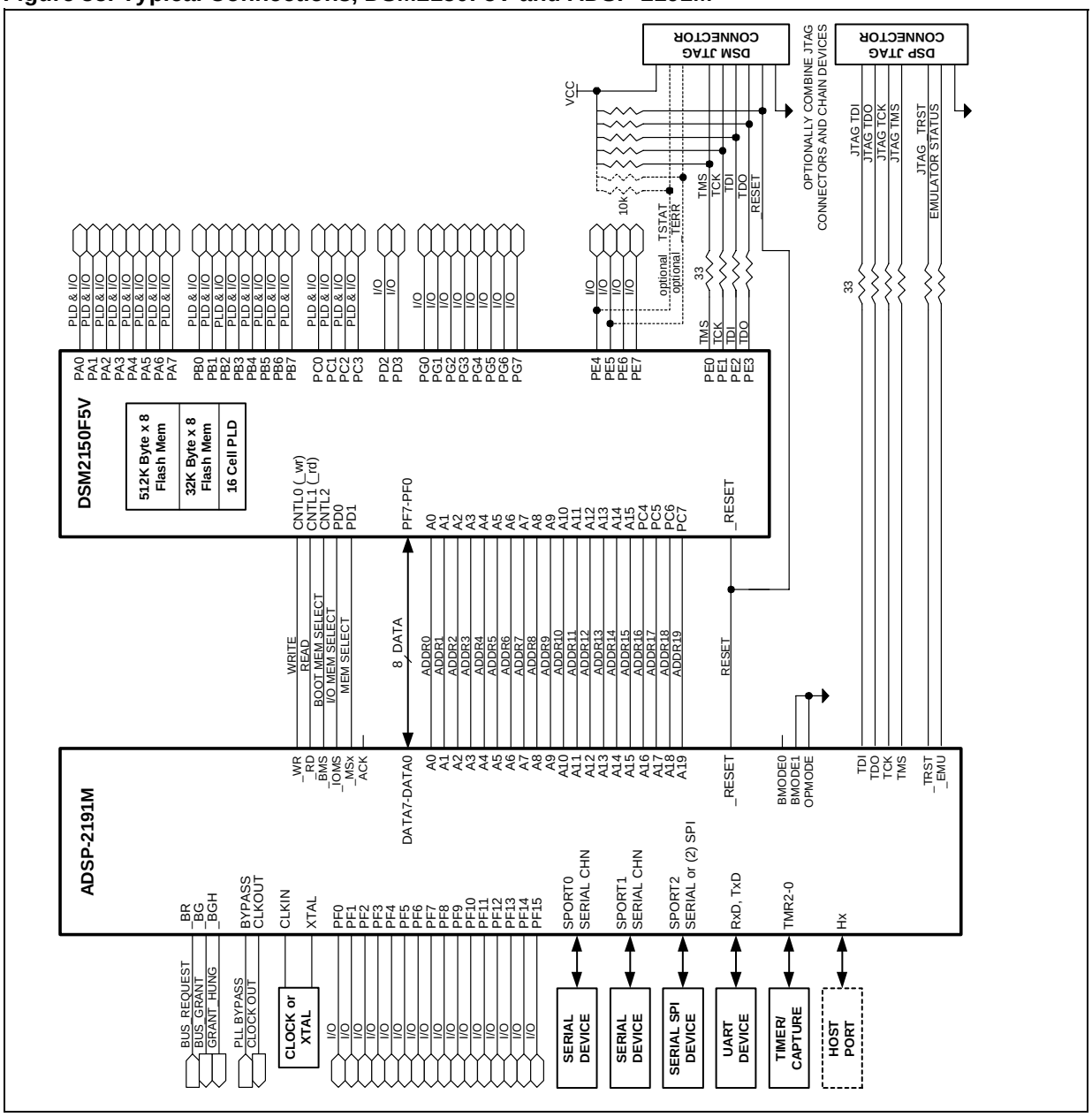
Figure 34. Typical Connections for ADSP-TS101S TigerSHARC DSP



APPENDIX E. TYPICAL CONNECTIONS, DSM2150F5V AND ADSP-2191

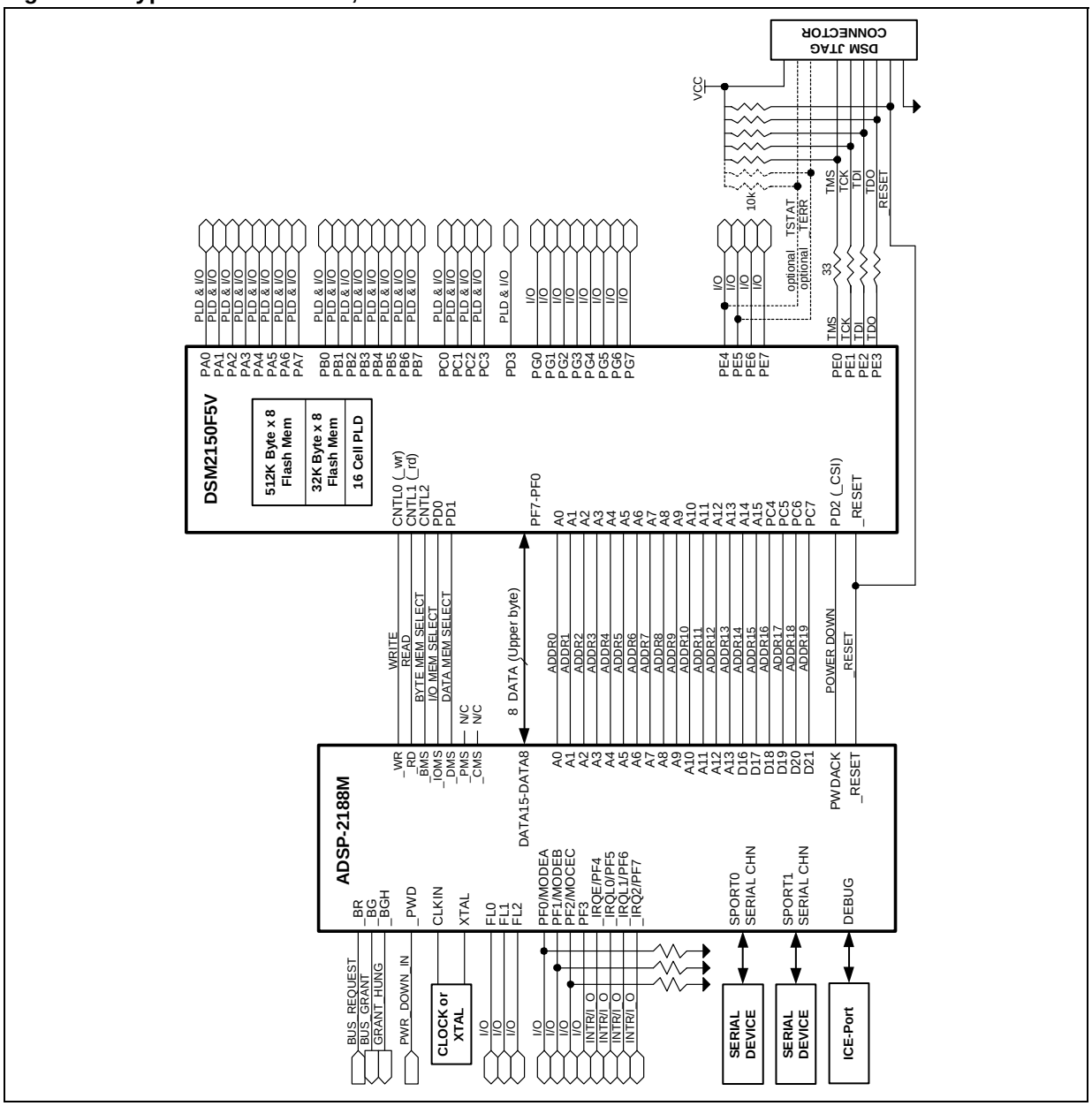
Note: DSP HCLK is limited to 45 MHz to satisfy DSM2150F5V parameter, t_{AVWL}. If HCLK is greater than 45 MHz, a non-inverting buffer should be placed between $\overline{WR}$ signal output from ADSP-219x and CNTL0 input to DSM2150. This delays the falling edge of $\overline{WR}$ to satisfy t_{AVWL}, However, the Write Hold Enable (E_WHE) memory space setting of the ADSP-2191 must be set to hold the DSP address after the delayed rising edge of $\overline{WR}$.

Figure 35. Typical Connections, DSM2150F5V and ADSP-2191M



APPENDIX F. TYPICAL CONNECTIONS, DSM2150F5V AND ADSP-2188M

Figure 36. Typical Connections, DSM2150F5V and ADSP-2188M



REVISION HISTORY

Table 51. Document Revision History

Date	Rev.	Description of Revision
14-Feb-2002	1.0	Document written