AN1427
APPLICATION NOTE

Interfacing the PSD813F5
with the ADSP-21061 SHARC DSP

CONTENTS

■ (See next page)

Contents

1.0 Introduction

The Digital Signal Processing Marketplace is typically divided into two specific areas:

Function and Algorithm Specific ICs….are non-programmable DSPs integrated with other peripherals. They consist of modem chips, DVDs, MPEG and Video Decoders, etc.

General Purpose Programmable DSPs….are flexible DSPs that are used in a broad spectrum of products. They typically use a microcontroller for control, as well as additional I/O and programmable logic.

Most general purpose DSPs have internal 8-bit Boot Load routines embedded in ROM which take advantage of slower, less expensive external Flash and EPROMS to store non-volatile program code to upload into fast internal SRAM at reset.

2.0 Purpose

Although the Flash PSD8XX family has become an ideal peripheral for 8-bit microcontrollers, many companies using the PSD in DSP-based products have shown that it makes an excellent peripheral for DSPs. The PSD8XX provides programmable logic and the required bus interfacing to implement a clean two-chip solution.

The PSD JTAG port allows In-System Programming (ISP) of a completely blank PSD8XX device soldered to the board with no involvement of the DSP, which is ideal for first time programming during manufacturing. The PSD8XX also offers In Application re-Programming (IAP), in which the DSP participates by executing UART download code from the small flash memory in the PSD while writing new code into the large flash memory in the PSD. This unique concurrent operation of PSD memories offers many IAP options. After IAP is complete, the DSP can copy the contents of the PSD main flash into the fast DSP SRAM for full speed operation.

This Application Note addresses the ease of interfacing the PSD8XXF with the ADSP-21061 DSP. Familiarity with the PSD8XXF is assumed. Please reference "PSD813F Data Sheet" for a detailed description of the device. The 21061 DSP is optimized for high performance signal computing for Speech, Sound, Graphics and Imaging applications. The PSD8XXF family of Zero Power parts meets these criteria and enables the core DSP design to be done with two chips.

3.0 PSD813F1 Architecture

The PSD8XX family is complemented by a lower-cost PSD9XX family. Figure 1 is a block diagram of the PSD8XX and PSD9XXF. Table 1 shows a comparison of the functional differences in the memory and CPLD options. On-chip features supply the key elements to implement a two-chip DSP System. Some devices have 32K bytes of byte-erasable EEPROM that may be used in place of external SRAM in some designs. Flash PSD features include:

1. Programmable bus interface to DSPs that are capable of external 8-bit boot code and/or program code.

2. 128-256 Kbytes of main Flash memory divided into eight individually selectable segments, each with optional sector protection.

3. Separate 32 Kbytes EEPROM or Secondary Flash memory divided into four individually selectable segments, each with optional sector protection.

4. Concurrent memory operation of main Flash and secondary memory (EEPROM or Flash) allows execution from one memory while reprogramming the other.

5. 2 Kbytes or 8 Kbytes of SRAM.

6. Two Flash-based PLDs with 16 Output Micro⇔Cells and 24 Input Micro⇔Cells.

7. 27 individually configurable I/O Port pins. Each may be defined as DSP I/Os, PLD I/Os, latched DSP address outputs or special function I/Os.

8. 8-bit Page Register to expand the address space by a factor of 256.

9. JTAG-ISP serial port for true In-System Programming (ISP) of blank devices and reprogramming of devices in the factory or field.

Table 1. PSD8XXF and PSD9XX Product Matrix

Device	Flash Main Memory 8 Sectors (Kbit)	Secondary Memory for Boot and/or Data 4 Sectors (Kbit)	SRAM (Kbit)	PLD
PSD813F1	1024	256 EEPROM	16	Sequential
PSD813F2	1024	256 Flash	16	Sequential
PSD813F3	1024	None	16	Sequential
PSD813F4	1024	256 Flash	None	Sequential
PSD813F5	1024	None	None	Sequential
PSD833F2	1024	256 Flash	64	Sequential
PSD834F2	2048	256 Flash	64	Sequential
PSD913F2	1024	256 Flash	16	Combinatorial Only
PSD934F2	2048	256 Flash	64	Combinatorial Only

An ADSP-21061 DSP is used in this application that has an on-chip Boot Loader feature and does not require an external Boot block. We'll take advantage of this and select the low-cost PSD813F5 that has main flash only (no secondary boot memory).

4.0 Development Systems

The PSD family is supported by PSDsoft Express, a software development tool that runs on Windows 95 & 98 and NT. This tool has point and click features for DSP bus interface configuration, and uses an HDL (PSDabel) to define general programmable logic within the PLD. DSP firmware is imported and merged to create a single object file to program into the PSD. PSDsoft supports two device programmers directly (ST PSDpro, ST FlashLink). The generated object file is also compatible with third-party programmers. See web site for list (*www.st.com/psm).*

ST offers two low-cost device programmers:

PSDpro….plugs into a PC/laptop parallel port and replaces the ST MagicPro III.

FlashLink….is a low cost cable that plugs into a PC/laptop parallel port to support JTAG-ISP programming. FlashLink is controlled by PSDsoft Express and supports device chaining of multiple PSDs and devices from other manufacturers.

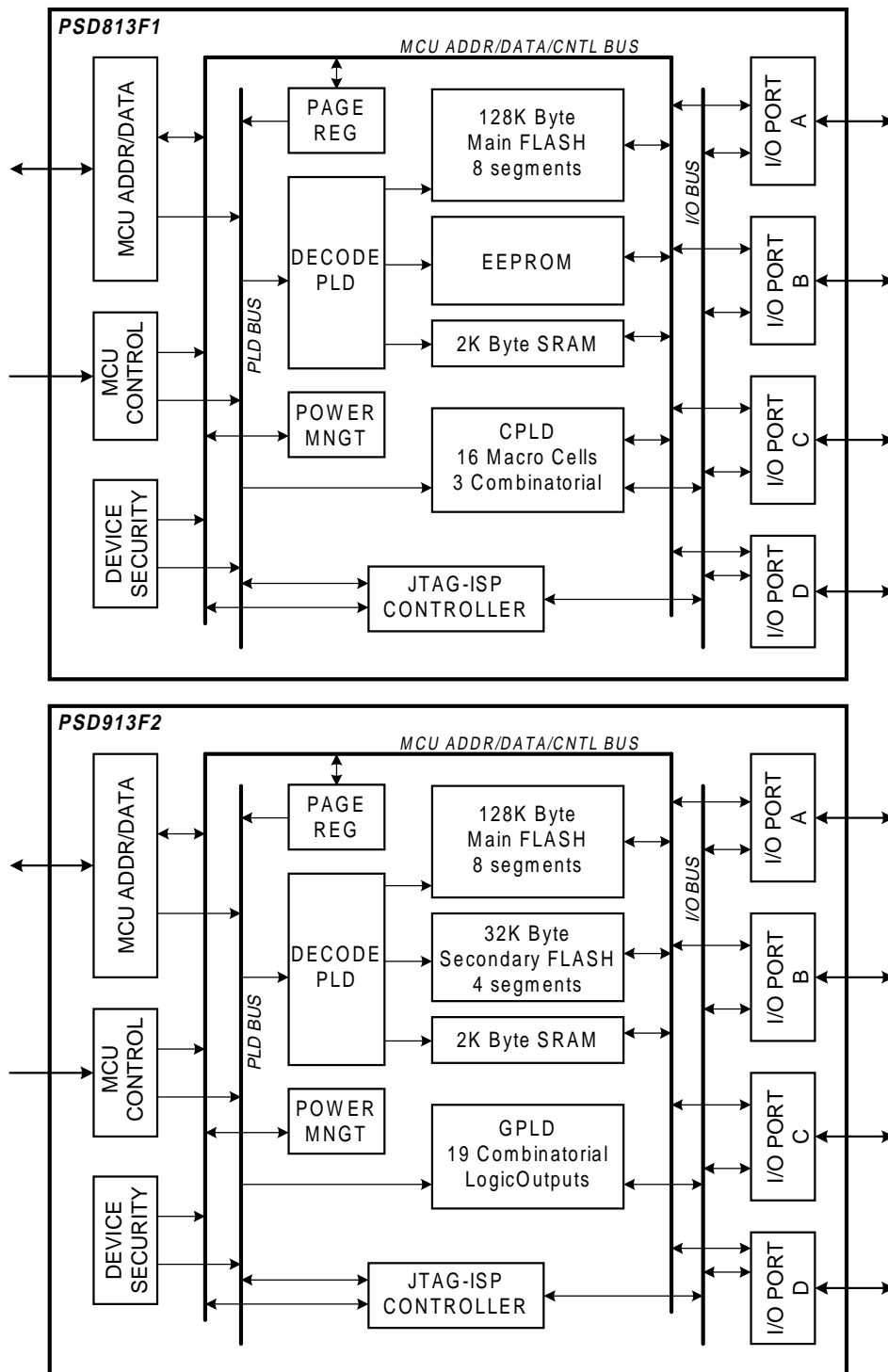**5.0 Programming the PSD813F In-Circuit using the JTAG ISP Interface**

The ability to initially program a new system board with a blank Flash memory soldered directly to it has solved many manufacturing logistics problems – no sockets or individual labels are required; inventory of non-volatile program memory chips is reduced to one package; the PLD is programmed at the same time as the memory chip. One system board can be built and inventoried. Any options can be programmed into the Flash memory at board level testing.

Port C I/O lines are used to interface to the standard JTAG signals – TMS, TCK, TDI and TDO. TSTAT and TERR are optional JTAG-ISP extensions that can be monitored to decrease the programming time of the PSD813F. The PSD configuration, PLD logic, Flash memory and optional Flash Boot/EEPROM can be programmed simultaneously through this interface.

Port C also gives the option to multiplex its JTAG pins with the PSD813F general I/O lines. This option, if used, frees up the JTAG pins for I/O functions after JTAG programming is completed. This option is enabled by the following three lines of code in PSDabel, and its hardware implementation is illustrated in Application Note 054 "JTAG Information – PSD813F":

```
jen      pin  11;        "Port C pin pc7 is used as external JTAG multiplex enable
jtagsel  node;           "Selects JTAG port active using internal product term
jtagsel = !jen;          "Switches Port C between JTAG and I/O
```

Figure 1. – PSD8XX/PSD9XX Block Diagrams

PSD813F1

MCU ADDR/DATA/CNTL BUS

MCU ADDR/DATA

MCU CONTROL

DEVICE SECURITY

PLD BUS

PAGE REG

DECODE PLD

POWER MNGT

128K Byte Main FLASH 8 segments

EEPROM

2K Byte SRAM

CPLD 16 Macro Cells 3 Combinatorial

JTAG-ISP CONTROLLER

I/O BUS

I/O PORT A

I/O PORT B

I/O PORT C

I/O PORT D

PSD913F2

MCU ADDR/DATA/CNTL BUS

MCU ADDR/DATA

MCU CONTROL

DEVICE SECURITY

PLD BUS

PAGE REG

DECODE PLD

POWER MNGT

128K Byte Main FLASH 8 segments

32K Byte Secondary FLASH 4 segments

2K Byte SRAM

GPLD 19 Combinatorial LogicOutputs

JTAG-ISP CONTROLLER

I/O BUS

I/O PORT A

I/O PORT B

I/O PORT C

I/O PORT D

4

6.0 Interfacing the PSD813F5 with the ADSP-21061

Figure 2 is a Block diagram that shows the bus interface implementation of a two-chip system using the PSD813F5 and the ADSP-21061. All glue logic, Flash memory, bus interface logic, I/O, chip selects and PLDs are contained in one chip.

Figure 2. – Block Diagram – Bus Interface for Minimized DSP System
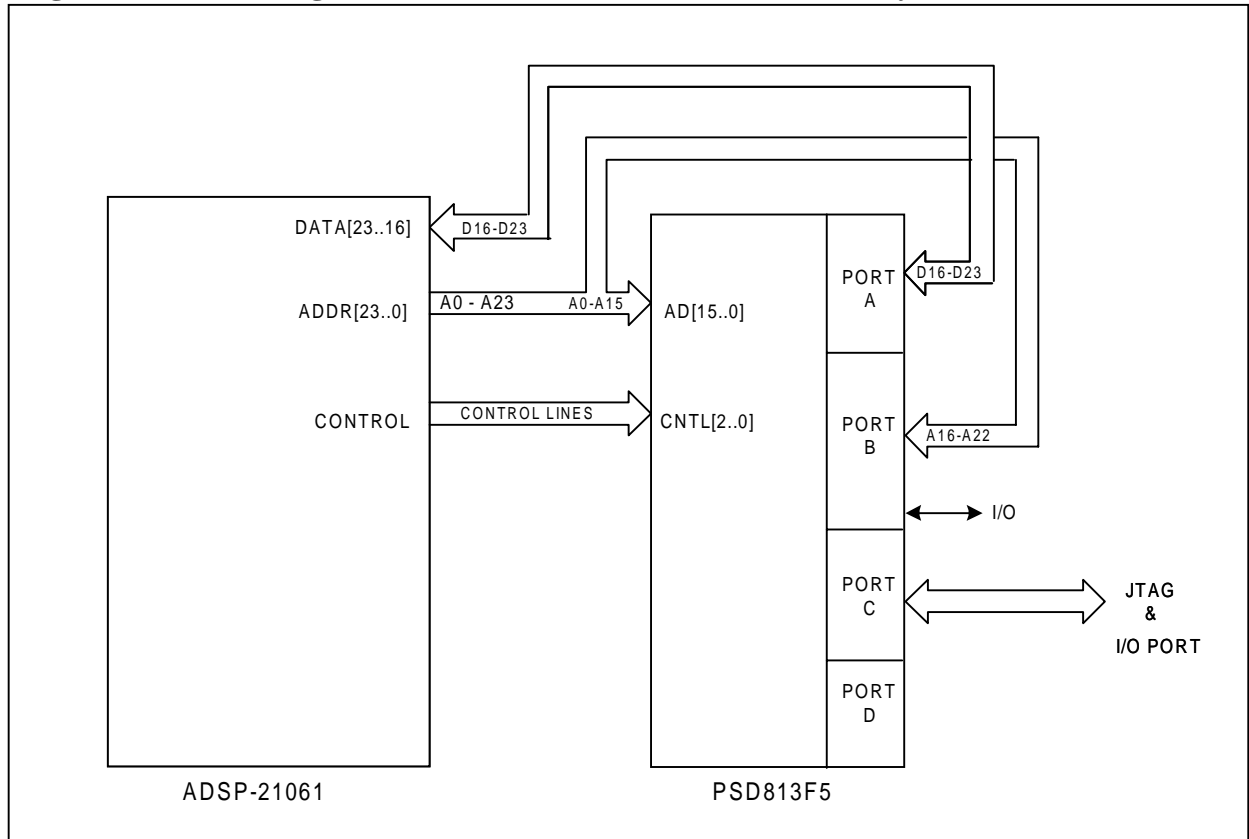
6.1. PSD813F5 Bus Interface

The PSD813F5 has a user-friendly programmable bus interface that is quickly configured to interface directly to most General Purpose DSPs with no "glue logic". Table 2 lists the bus interface signals from the ADSP-21061 used to access the Flash memory, PLD logic and I/O inside the PSD813F5.
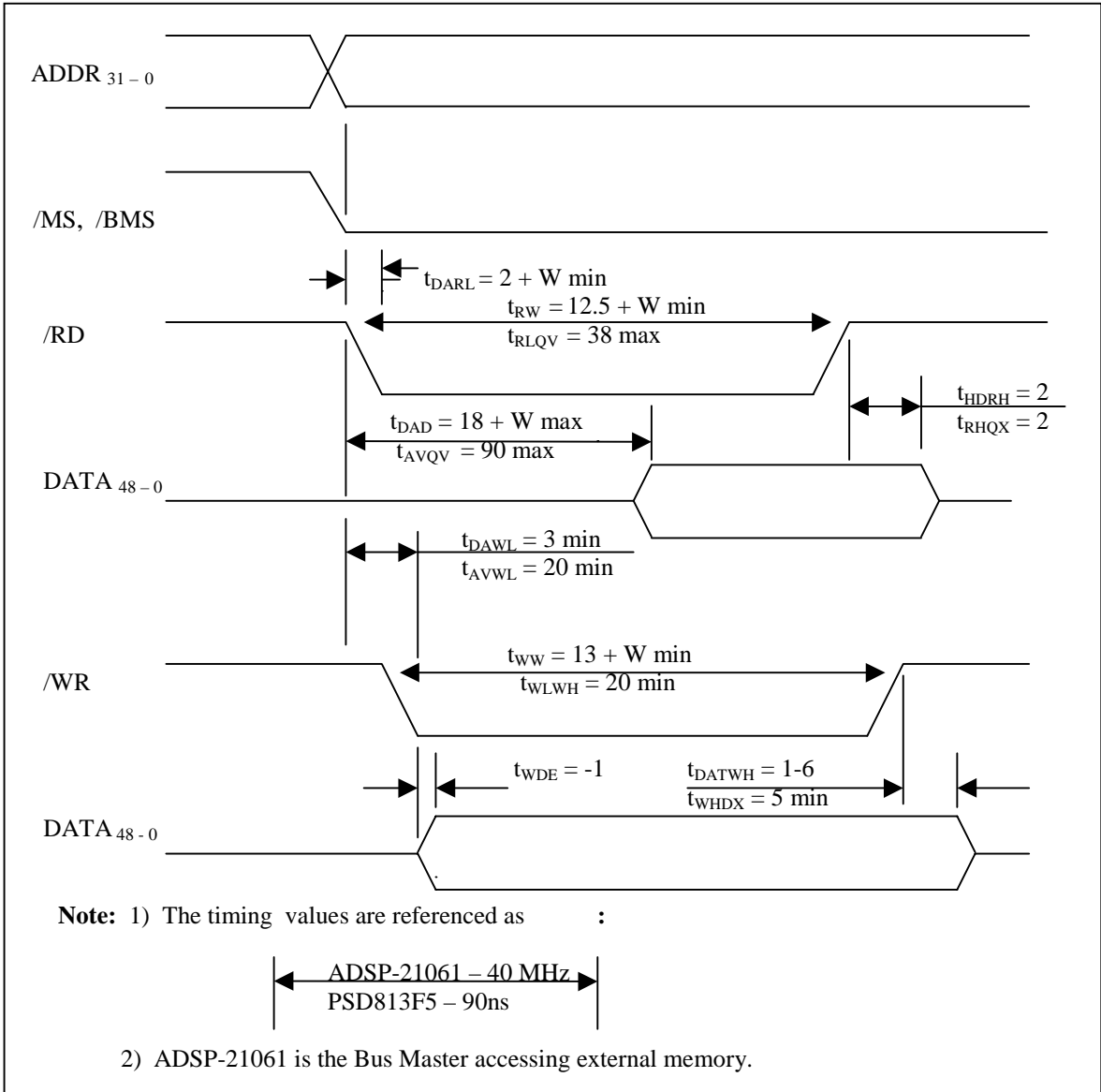
Table 2. – ADSP-21061 Bus Interface Pin Functions

ADSP-21061 Pin Functions	PSD813F5 Pin Functions	Pin Description
ADDR31 – ADDR24 ADDR23 – ADDR16 ADDR15 – ADDR0	NC PortB PB7 – PB0 AD15 – AD0	External Address Bus addresses external Memory and I/O in Bus Master mode.
DATA47 – DATA24 DATA22 – DATA16 DATA15 – DATA0	NC PortA PA6 – PA0 NC	48 bi-directional external data bus lines. DATA22 – DATA16 are used to boot load from an external 8-bit Flash memory.
/BMS	CNTL2	Boot Memory Select. /BMS is output in the Bus Master mode (EBOOT = 1, LBOOT = 0) to select the external Flash Boot memory.
/RD	CNTL1	Memory Read Strobe is output in Bus Master mode to read from external memory and I/O.
/WR	CNTL0	Memory Write Strobe is output in Bus Master mode to write to external memory and I/O.
/MS3 - /MS1 MS0	NC PortD PD0	Memory Select Lines are active low chip selects for the four banks of external memory.

6.2 ADSP-21061 Bus Interface Timing Calculation

The ADSP-21061 has an internal programmable WAIT register that generates 0 to 7 separate wait states for each of the four external memory banks. All Flash Program memory and I/O are located in Bank 0. If the EEPROM/Boot Flash and SRAM options are used, they also are located in Bank 0. After the DSP is reset, all wait state registers are default to 6 wait states to allow access to slow memory. Figure 3 compares the read/write timing differences between the ADSP-21061-40 MHz and PSD813F5-90ns. Three wait states are required to access the Program Flash memory and I/O in the PSD813F5 during normal program execution. All calculations are based on the /RD, /WR, /BMS and /MS0 bus interface signals.

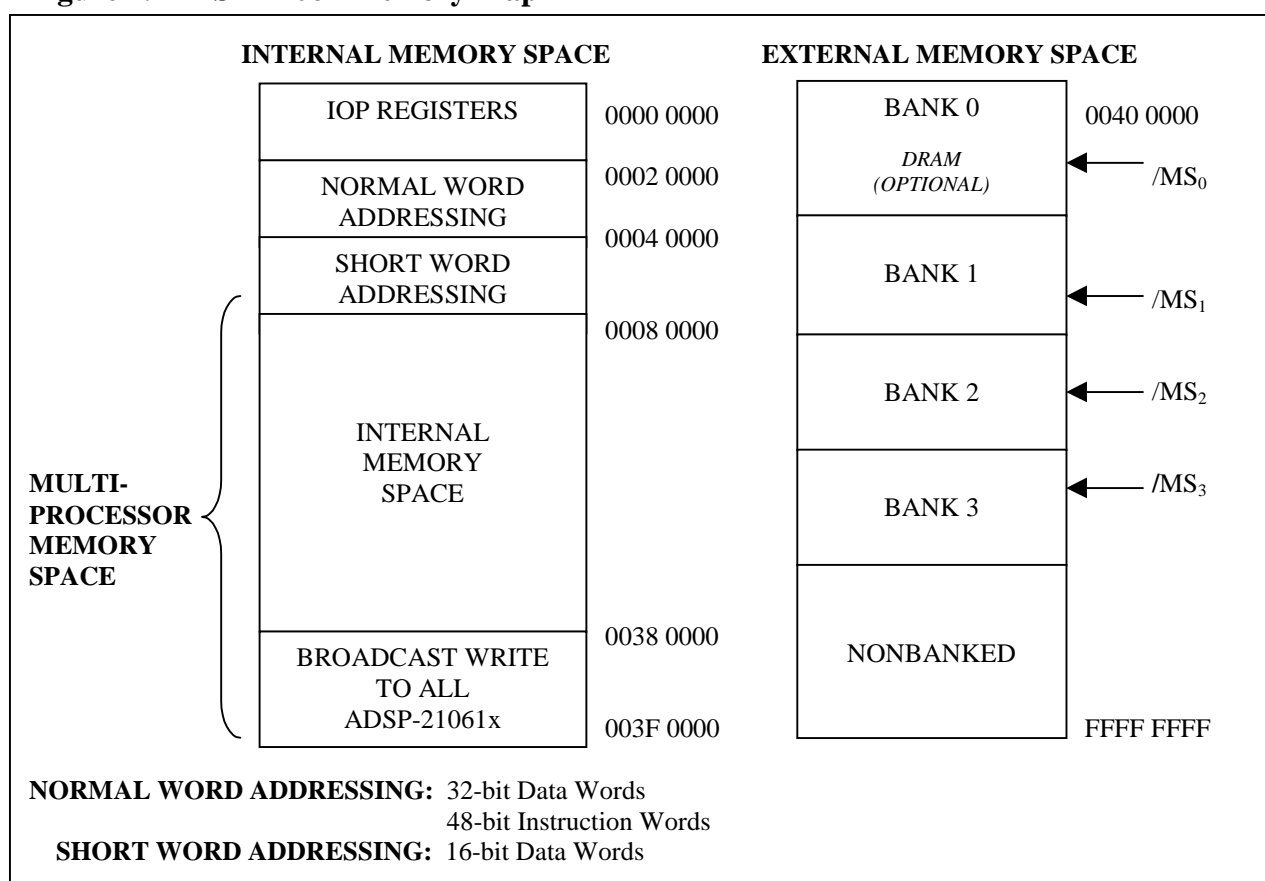**Figure 3. ADSP-21061 Read / Write Memory Timing**

ADDR $_{31-0}$

/MS, /BMS

$t_{DARL} = 2 + W$ min

/RD

$t_{RW} = 12.5 + W$ min
$t_{RLQV} = 38$ max

$t_{HDRH} = 2$
$t_{RHQX} = 2$

$t_{DAD} = 18 + W$ max
$t_{AVQV} = 90$ max

DATA $_{48-0}$

$t_{DAWL} = 3$ min
$t_{AVWL} = 20$ min

$t_{WW} = 13 + W$ min
$t_{WLWH} = 20$ min

/WR

$t_{WDE} = -1$

$t_{DATWH} = 1-6$
$t_{WHDX} = 5$ min

DATA $_{48-0}$

Note: 1) The timing values are referenced as **:**

ADSP-21061 – 40 MHz
PSD813F5 – 90ns

2) ADSP-21061 is the Bus Master accessing external memory.

6.3 ADSP-21061 Memory Map

The ADSP-21061 contains 1 Mbit configurable on-chip Dual-Port SRAM, organized as two banks of 0.5 Mbit each. One bank is used primarily to store instructions and data, and the other bank is used to store data; other combinations are allowed. The memory is configured as 32K Words Data Memory (32-bit), 16K Words Program Memory (48-bit), or combinations of both. The memory map is shown in Figure 4.

Four Gigawords of external memory (32-bit address) are accessed via the External Port, starting at address 0040 0000. The optional Flash Boot memory also starts at this address. Four Memory Select lines (MS0 - MS3) are used as chip selects to access their corresponding external memory banks.

Figure 4. ADSP-21061 Memory Map

NORMAL WORD ADDRESSING: 32-bit Data Words
                                                48-bit Instruction Words
  SHORT WORD ADDRESSING: 16-bit Data Words

6.4 Interfacing to the ADSP-21061 External Memory Bus

The Block Diagram of Figure 5 shows the bus interface between the ADSP-21061 and the PSD813F5. Since the Boot Code in external Flash memory begins at address 0040 0000, 23 of the 32 address lines are used by the PSD813F5. Also, only data lines DATA16-DATA22 are utilized to transfer 8-bit program/data to the DSP. Paging is not required, due to the 32 available external address lines.

Figure 5. Block Diagram – ADSP-21061 System Block Diagram
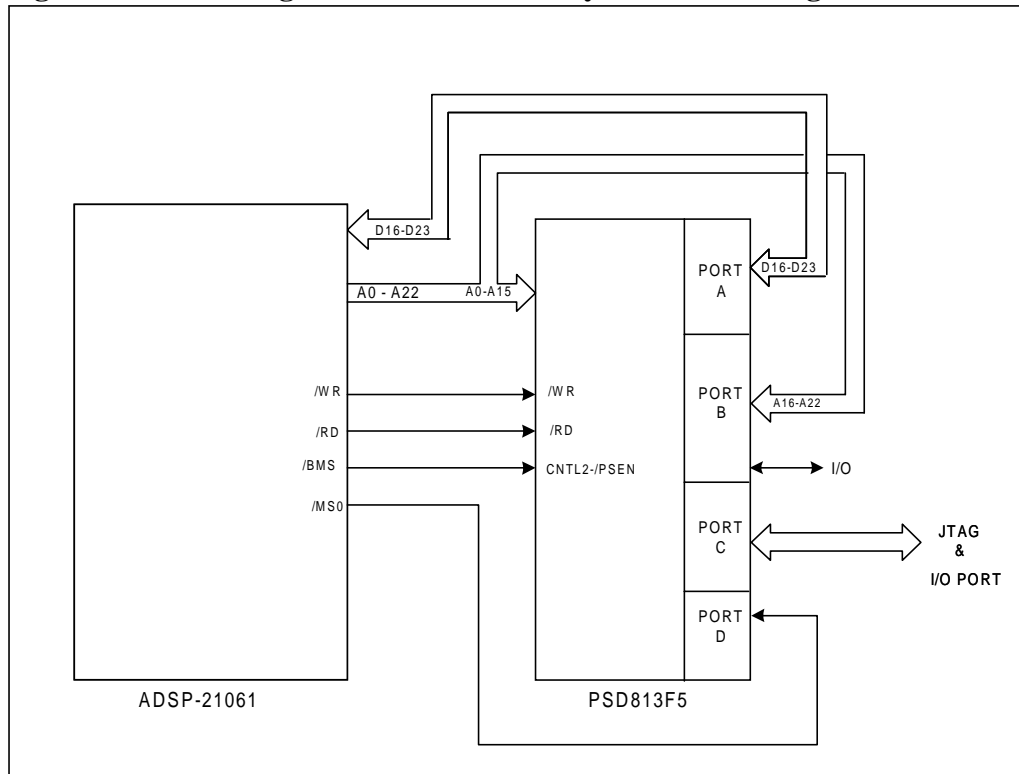
6.5. Define the ADSP-21061 Interface in PSDsoft Express Define PSD and MCU Utility

Figure 6 is the MCU and PSD Selection screen imported from PSDsoft Express Define PSD and MCU utility. Selecting the following appropriate signals in this screen quickly configures the bus configuration between the ADSP-21061 and PSD813F5. /MSO is connected to the PSD813F5 DPLD through Port D (pin PD0) and included in the internal chip select equations generated under the Chip Select Equations tab in the Design Assistant screen.

* Type: Other
* Data Bus Width: 8-bit
* Address / Data Mode: Non-Mux
* Control Setting: /WR, /RD
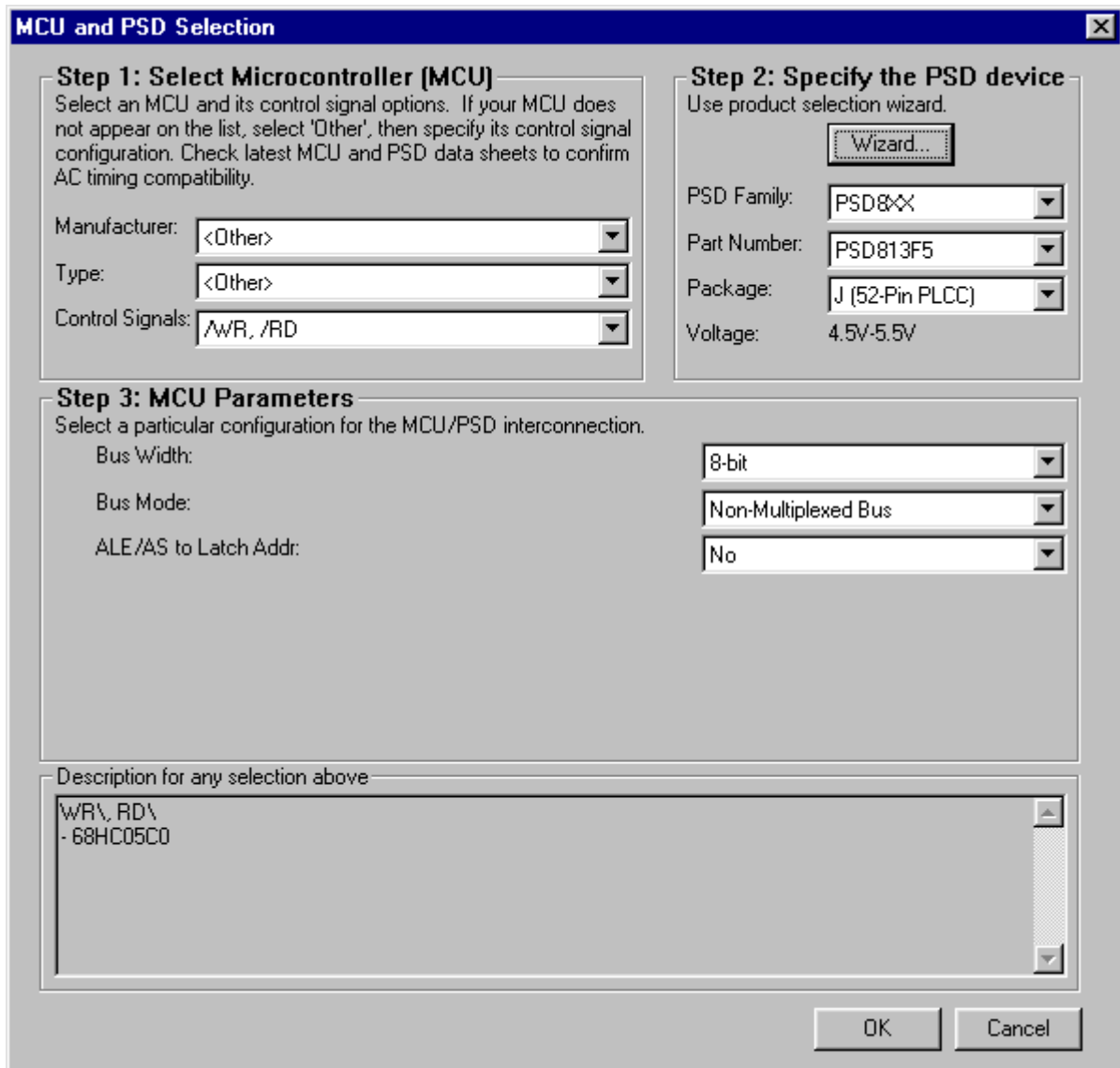
Figure 6. Define PSD and MCU

Figure 7 is the schematic diagram of the ADSP-21061 / PSD813F5 bus interface. The 128K bytes of PSD813F5 Flash Memory reside in data space, because program code will not be executed from external Flash memory; it will be downloaded to the internal DSP DARAM. This IAP feature allows updated program code to be downloaded to the Flash memory through the DSP Serial Port for future download to the DARAM for program execution.

Figure 7. Schematic Diagram – ADSP-21061 to PSD813F5 Bus Interface
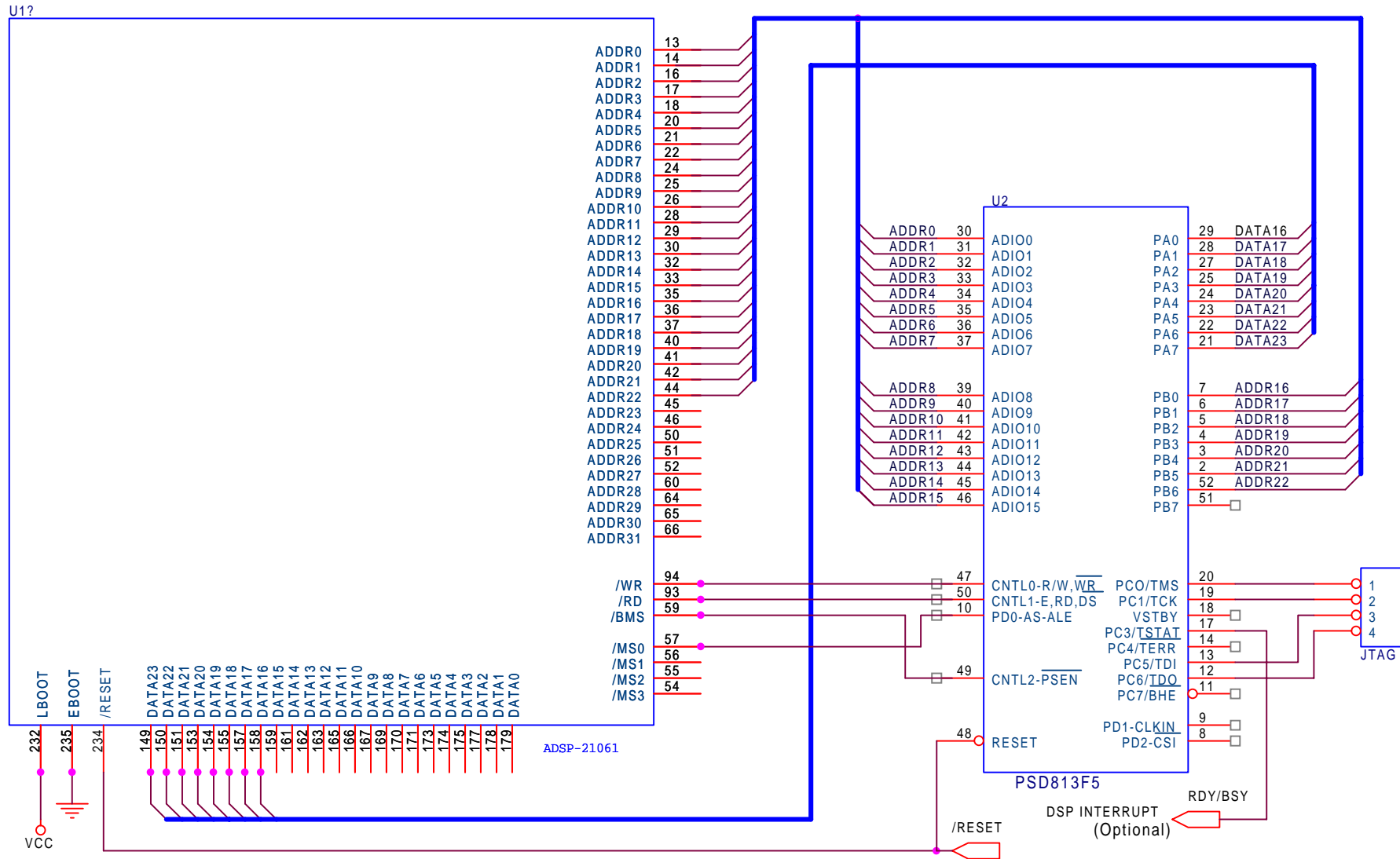
6.6. Define the PSD813F5 DPLD Functions in PSDsoft Express Edit/Add Logic Statements

Figure 8 is the system memory map created for this application note defining the internal decoding functions for the ADSP-21061 based system. No paging is required, due to its 4-gigaword external memory address range. All functions of the PSD813F5 are mapped in Bank 0 of the external memory space. For this example, the external memory bank size is set to one megaword by setting the MSIZE bit field of the SYSCON Register. The I/O and chip select addresses are defined in the PSDsoft Express Design Assistant screen under "Chip Select Equations" and implemented in the internal PSD813F5 Decoding PLD (DPLD). See Appendix for PSDsoft Express Design Assistant Summary.

Figure 8. ADSP-21061 System Memory Map
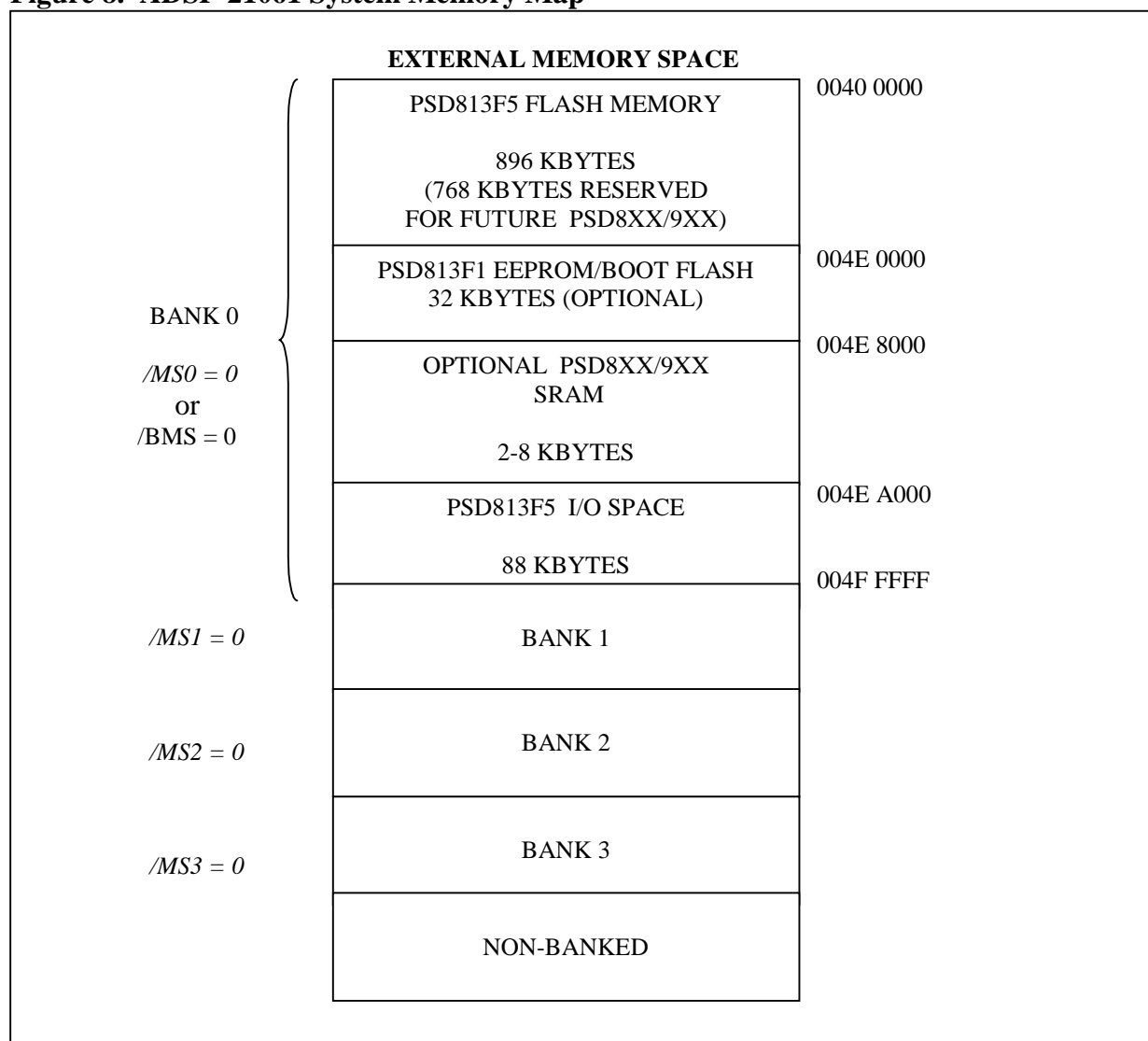
6.7Accessing the PSD813F5 Internal Registers

The bank of internal control registers in the PSD813F5 (csiop + hXX) are 8-bits wide and are not accessible on a 16-bit word boundary; they must be addressed on 8-bit byte boundaries. The Boot Select Override (BSO) bit in the ADSP-21061 SYSCON Register provides this capability. Also, the internal registers must be mapped in the memory area that is accessible by /BMS

6.7.1 Reading the Contents of the Internal Registers

The BSO bit overrides the external memory selects and causes the /BMS pin to go active low for an external port DMA transfer. The bootstrap program should first set the BSO bit in the SYSCON register and then set up one of the four external DMA channels to access the external non-boot memory. While one of the external port DMA channels is being used with the BSO bit, the other three channels cannot be used.

When BSO = 1, /BMS is not asserted by a DSP access to the boot code, but by a DMA transfer. This allows the bootstrap program in the DSP core to access the PSD813F5 internal registers instead of boot memory.

6.7.2 Writing to the Internal Registers

The BSO bit in the SYSCON register allows the /BMS signal to write to external 8-bit memory under software control. To write to the internal registers with /BMS active, use DMA channels 7, 8, or 9, but not channel 6, which is hardwired for a special 8-bit boot read mode. Since BMS memory space is 8-bits wide and no 8-bit packing mode is available for these write accesses, the DSP shifter must be used to place data on data lines DATA16-DATA23 for each write.

7.0. In Application re-Programming (IAP) using the ADSP-21061 Serial Port

The PSD813F5 (without secondary memory) was selected to reduce the system cost and take advantage of the DSP DARAM that can contain the program code for IAP of PSD Flash memory through the DSP Serial Port.

The synchronous serial port of the ADSP-21061 is used for field updates to the program code that resides in one or more sectors of the PSD813F5 Flash memory; these sectors are located in Bank 0 of external memory space. Each chip select for each Flash sector – fs0..fs7 – has three product terms that enable control of individual Flash sector(s) to easily switch between /BMS for downloading Boot code during power-up and reset, and /MS0 to field upgrade program code as if it were data memory. The PSDabel file in the Appendix shows how the abel equations are written to swap all eight Flash sectors between program and data control. Page Register pgr7 is used a SWAP bit to switch the Flash memory chip select(s) control between /BMS and /MS0.

The ADSP-21061 has 32-bit transmit and receive data buffers with selectable word lengths from 3 to 32 bits. An 8-bit programmable word length is right-justified in the receive and transmit buffers. When a boot program update is received by the serial port, the 8-bit data in the receive data buffer is shifted left to align with the external data bus bits DATA23-DATA16 and written

to the accumulator to store either in an asigned section of the DSP DARAM allocated as a buffer for the uploaded program code, or write to the Flash memory on a byte-by-byte basis.

The byte-by byte write sequence to the Flash memory can be speeded up dramatically by configuring the RDY/Busy polling bit to Port C (pin PC3) and using it as an interrupt input to the DSP. Once a byte write command is issued, the time required to program the byte can now be executed in background mode. The DSP can be performing other tasks until the RDY/Busy pin signals that the byte has been successfully programmed and generates an interrupt.

The RDY/Busy bit is hardware configured as an output interrupt pin as shown in Figure 10 with the configuration sequence as follows:

1. Select "pc3" from the PSDsoft Express Pin Definition screen.
2. Select "Rdy/Bsy output" in "Other" block.

Figure 9. PSDsoft Express – Pin Definition Screen
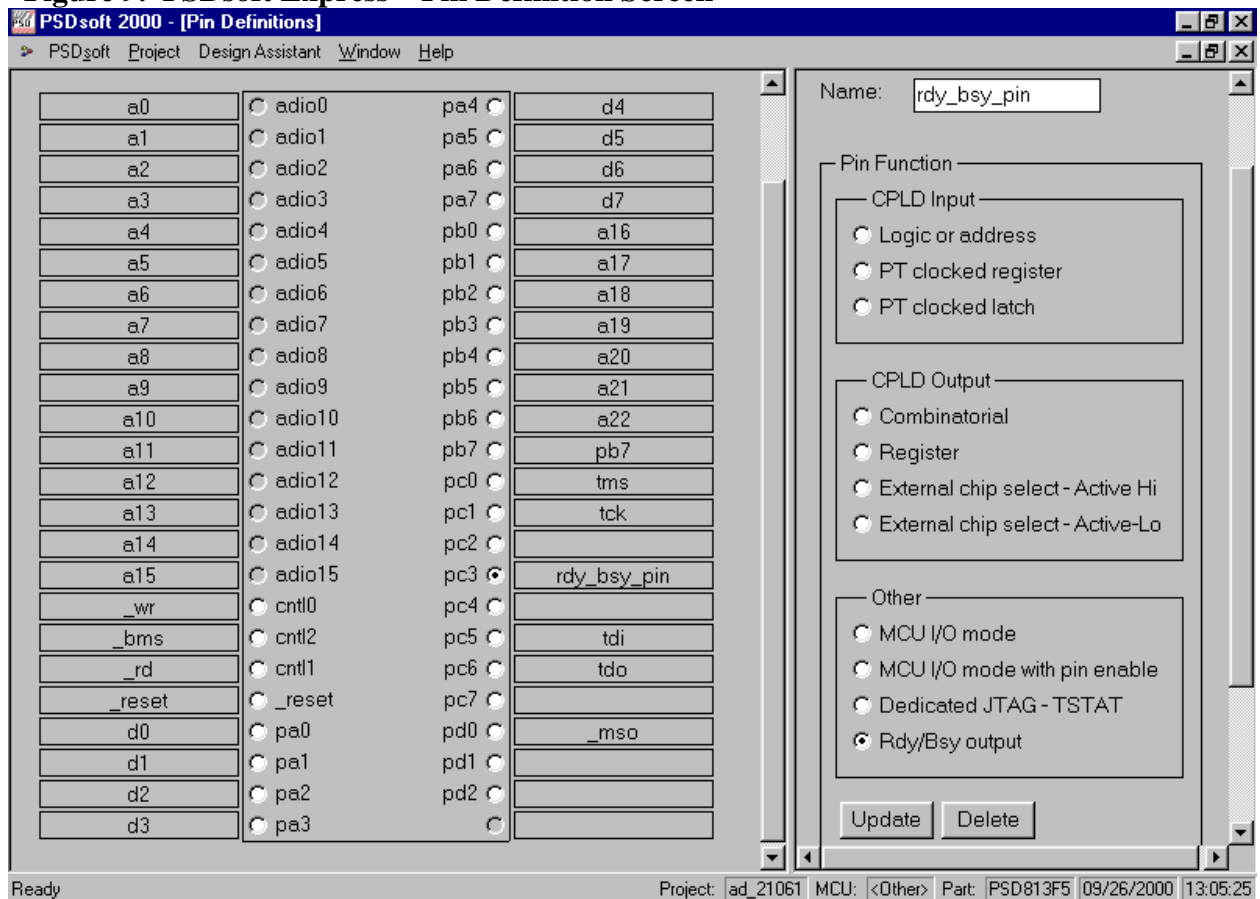
ADSP-21061 Boot Loader

The ADSP-21061 supports three modes of booting: EPROM, host and link port. Each bootmode packs boot code into 48-bit instructions and uses DMA Channel 6 to transfer the instructions to internal DARAM. Table 4 lists the available Boot Mode options. EBOOT = 1 and LBOOT = 0 is selected for this application.

Table 4. ADSP-21061 Boot Mode Options

EBOOT	LBOOT	BMS	Boot Mode
1	0	output	Boot memory is loaded from an 8-bit External Flash/EPROM
0	0	1 (input)	Boot memory is loaded from a 16-bit host processor
0	1	1 (input)	Link Port accesses 4-bit boot memory through DMA Channel and ext.clock
0	0	0 (input)	No booting. ADSP-21061 executes code from external memory

EPROM/Flash booting is selected when EBOOT input is strapped high, configuring /BMS as an output to be used as the boot Flash chip select. EPROM/Flash mode only loads 256 instructions during boot loading. The ADSP-21061 must have access to the boot/Flash memory after completion of the bootloading to download the entire application into DARAM. The primary configuration of the DSP DMA Channel 6 is used for Flash and host booting. Table 5 shows how the DMA Channel 6 parameter registers are initialized at reset for Flash booting.

Table 5. DMA Channel 6 Registers Initialization Values

Parameter Register	Initialization Value	Description
II6	0x0002 0000	Starting address of internal memory space
C6	0100	Contents of Count Register to transfer 256 words to internal memory
EI6	0x0040 0000	Starting address of external memory space
EC6	0600	External Count Register – transfers 256 words (6 bytes/word)

8.1. Boot Loader Sequence

After system reset (/RESET = 1), the following sequence of events occur:

1. The ADSP-21061 enters an idle state. The Program Counter (PC) is set to address 0x0002 0004.
2. The DMA Parameter Registers for DMA Channel 6 are initialized as shown in Table 5.
3. /BMS becomes the Boot Flash memory chip select.
4. 8-bit Master Mode DMA transfers from Flash to internal SRAM memory begin, using external port data bus lines DATA23 - DATA16.
5. The external address lines ADDR31 – ADDR0 start at address 0x0040 0000 and increment after each access.
6. The /RD strobe asserts as in a normal memory access, with six wait states.

When the External Count Register (EC6) reaches zero, the following wake-up sequence occurs:

1. The DMA transfer stops.
2. The External Port DMA Channel 6 interrupt (EPOI) is activated.
3. /BMS is deactivated and normal external memory selects are enabled.
4. The ADSP-21061 vectors to the EPOI interrupt vector at 0x0002 0040.
5. The ADSP-21061 booting mode is complete and instructions are executing normally.

8.0. Summary

As DSPs continue to rapidly proliferate into markets such as communications, industrial, medical, signal conditioning, and hand held test equipment, the PSD813F and DSP form an ideal 2-chip core with on-chip PLD and 27 I/O lines that can be individually configured to perform any function required by the system design. Using the PSD813F as an 8-bit boot loader in both high speed and low speed systems is an ideal and rapid design alternative to a discrete solution. Inexpensive slower memory and PLDs integrated in the PSD813F now become both cost and performance effective.

Several features internal to the PSD813F5 were used to expand the limitations of the ADSP-21061, and DSPs in general:

1. Flash memory allows IAP update of the program code in the field through the serial port of the DSP while the DSP is running program code in the internal DARAM.
2. JTAG-ISP simplifies manufacturing.
3. Expanded I/O was added to the system.
4. The internal Flash PLD allows design changes, in logic, I/O and memory mapping, to be made by software modifications instead of board level hardware changes.

These changes have added to both the versatility and performance of the ADSP-21061; future changes most likely will not require a hardware change to the 2-chip core.

9.0. Appendix

The Appendix contains the PSDsoft Express Design Assistant Summary showing how the PSD813F5 is configured to implement the example of this application note.

```
    ********************************************************************
                       PSDsoft Express Version 6.02
                       Summary of Design Assistant
    ********************************************************************
    PROJECT    : ad_21061                       DATE : 09/26/2000
    DEVICE     : PSD813F5                        TIME : 17:27:05
    MCU        : <Other>
    ********************************************************************


    Pin Definitions:
    ================


    Pin            Signal         Pin
    Name           Name           Type
    ------------   ------------   ------------
    adio0          a0             Address line
    adio1          a1             Address line
    adio2          a2             Address line
    adio3          a3             Address line
    adio4          a4             Address line
    adio5          a5             Address line
    adio6          a6             Address line
    adio7          a7             Address line
    adio8          a8             Address line
    adio9          a9             Address line
    adio10         a10            Address line
    adio11         a11            Address line
    adio12         a12            Address line
    adio13         a13            Address line
    adio14         a14            Address line
    adio15         a15            Address line
    cntl0          _wr            MCU bus control signal
    cntl2          _bms           Logic or address
    cntl1          _rd            MCU bus control signal
    reset          _reset         RESET input
    pa0            d0             Data line
    pa1            d1             Data line
    pa2            d2             Data line
    pa3            d3             Data line
    pa4            d4             Data line
    pa5            d5             Data line
    pa6            d6             Data line
    pa7            d7             Data line
    pb0            a16            Logic or address
    pb1            a17            Logic or address
    pb2            a18            Logic or address
    pb3            a19            Logic or address
    pb4            a20            Logic or address
    pb5            a21            Logic or address
    pb6            a22            Logic or address
    pb7            pb7            Logic or address
    pc0            tms            Dedicated JTAG - TMS
    pc1            tck            Dedicated JTAG - TCK
    pc3            rdy_bsy_pin    Rdy/Bsy output
    pc5            tdi            Dedicated JTAG - TDI
    pc6            tdo            Dedicated JTAG - TDO
    pd0            _mso           Logic or address
```

```
User defined nodes:
===================

None defined

Page Register settings:
=======================

pgr0 is not used
pgr1 is not used
pgr2 is not used
pgr3 is not used
pgr4 is not used
pgr5 is not used
pgr6 is not used
pgr7 is used for logic, signal name: swap

Equations:
==========

csiop = ((address >= ^h4EA000) & (address <= ^h4EA0FF) & (!_bms));
fs0 = ((address >= ^h400000) & (address <= ^h403FFF) & (!_bms & !swap))
    # ((address >= ^h400000) & (address <= ^h403FFF) & (!_mso & swap));
fs1 = ((address >= ^h404000) & (address <= ^h407FFF) & (!_bms & !swap))
    # ((address >= ^h404000) & (address <= ^h407FFF) & (!_mso & swap));
fs2 = ((address >= ^h408000) & (address <= ^h40BFFF) & (!_bms & !swap))
    # ((address >= ^h408000) & (address <= ^h40BFFF) & (!_mso & swap));
fs3 = ((address >= ^h40C000) & (address <= ^h40FFFF) & (!_bms & !swap))
    # ((address >= ^h40C000) & (address <= ^h40FFFF) & (!_mso & swap));
fs4 = ((address >= ^h410000) & (address <= ^h413FFF) & (!_bms & !swap))
    # ((address >= ^h410000) & (address <= ^h413FFF) & (!_mso & swap));
fs5 = ((address >= ^h414000) & (address <= ^h417FFF) & (!_bms & !swap))
    # ((address >= ^h414000) & (address <= ^h417FFF) & (!_mso & swap));
fs6 = ((address >= ^h418000) & (address <= ^h41BFFF) & (!_bms & !swap))
    # ((address >= ^h418000) & (address <= ^h41BFFF) & (!_mso & swap));
fs7 = ((address >= ^h41C000) & (address <= ^h41FFFF) & (!_bms & !swap))
    # ((address >= ^h41C000) & (address <= ^h41FFFF) & (!_mso & swap));
```

Table 1. Document Revision History

Date	Rev.	Description of Revision
Sep-2000	2.1	Document written (AN071) in the WSI format
03-Jan-2002	2.2	Front page, and back two pages, in ST format, added to the PDF file

For current information on PSD products, please consult our pages on the world wide web:

www.st.com/psm

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.psd@st.com   (for application support)

ask.memory@st.com   (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.