AN1429
APPLICATION NOTE

Interfacing the 3V PSD813FxV Family with the TI TMS320VC5402 DSP

CONTENTS

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 imbedded 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.

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 memory into the fast DSP SRAM for full speed operation.

This Application Note addresses the ease of interfacing the PSD813F5V with the TMS32VC5402 DSP. Familiarity with the PSD813F is assumed. Please reference “PSD813F Data Sheet” for a detailed description of the device. The VC5402 DSP is designed for use in wireless communications and telephony systems where low power, low voltage and size are critical. The 3V PSD813FxV family of Zero Power parts meets

these criteria and enables the core DSP design to be done with two chips.

PSD813F1 ARCHITECTURE

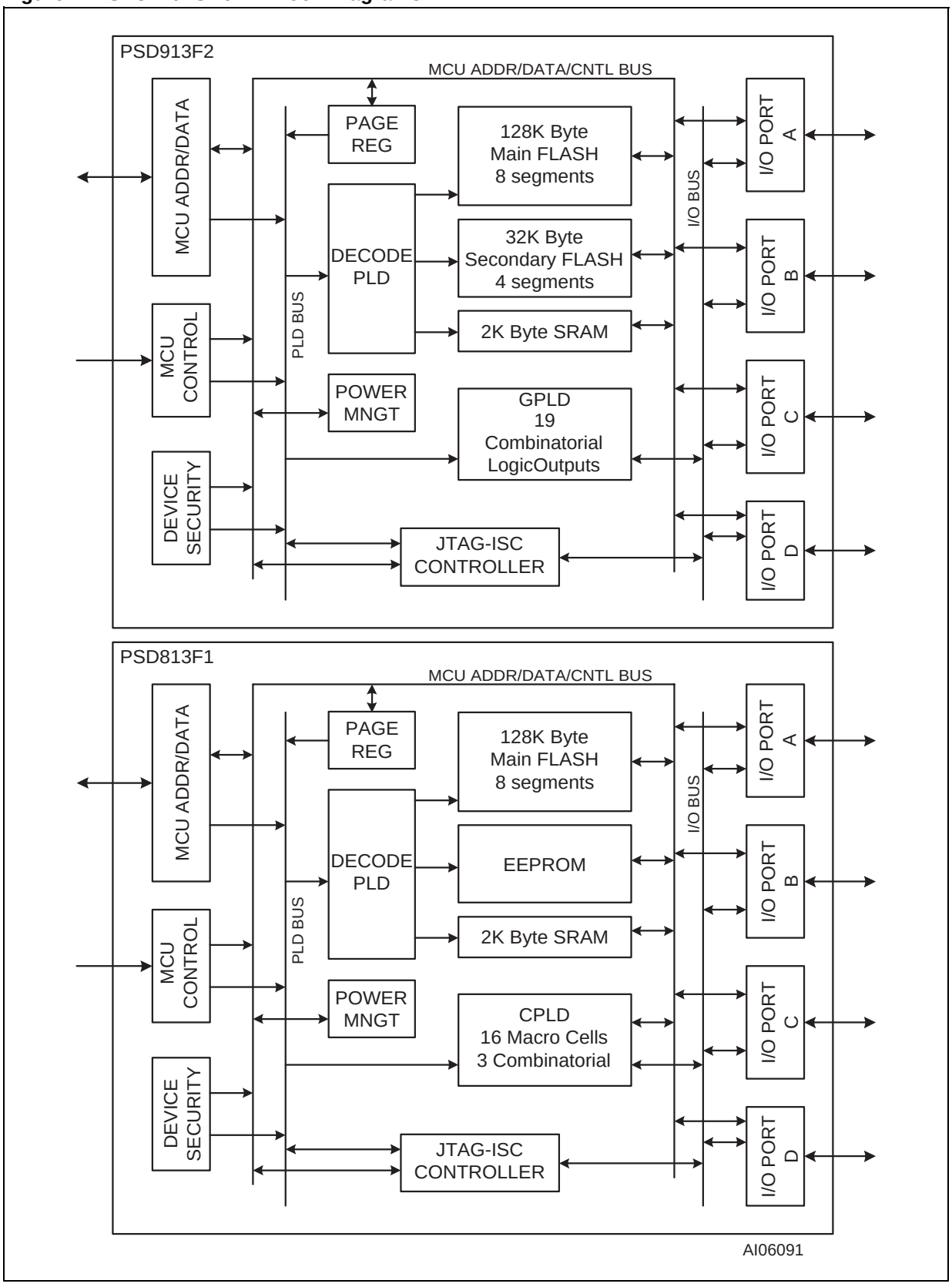
The PSD8XX family is complemented by a lower-cost PSD9XX family. Figure 1 is a block diagram of the PSD8XX and PSD9XXF. Table 1 shows a comparison of the functional differences in the memory and CPLD options. On-chip features supply the key elements to implement a two-chip DSP System. Some devices have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

- Programmable bus interface to DSPs that are capable of accessing external 8-bit boot code and/or program code.
- Programmable bus interface to DSPs with external 8-bit boot code and/or program code.
- 128-256 Kbytes of main Flash memory, divided into eight equal individually protected sectors.
- Separate 32 Kbytes EEPROM or Flash Boot memory divided into four equal blocks.
- Concurrent programming of the Flash or EEPROM/Boot Flash memories allows execution from one memory while reprogramming the other.
- 2Kbytes or 8 Kbytes scratch-pad SRAM.
- Two Flash-based PLDs with 16 Output Micro⇔Cells and 24 Input Micro⇔Cells.
- 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-bit Page Register to expand the address space by a factor of 256.
- JTAG compliant 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 Kbit (8 Sectors)	Additional Memory for Boot and/or Data (4 Sectors)	SRAM Kbit	PLD
PSD813F1	1024	256 Kbit EEPROM	16	Sequential
PSD813F2	1024	256 Kbit Flash	16	Sequential
PSD813F3	1024	None	16	Sequential
PSD813F4	1024	256 Kbit Flash	None	Sequential
PSD813F5	1024	None	None	Sequential
PSD833F2	1024	256 Kbit Flash	64	Sequential
PSD834F2	2048	256 Kbit Flash	64	Sequential
PSD913F2	1024	256 Kbit Flash	16	Combinatorial
PSD934F2	2048	256 Kbit Flash	64	Combinatorial

Figure 1. PSD8XX/PSD9XX Block Diagrams



The VC5402 has a basic on-chip Boot Loader and 16K words of on-chip DARAM. Program code is downloaded from external Flash memory to fast internal DARAM for execution after system reset. The low-cost PSD813F5V (no secondary boot memory) is selected for this design to take advantage of the TMS320VC5402 resident boot loader. The following design parameters are assumed for using the PSD813F5V without the Flash Boot memory:

1. The initial firmware is programmed into the PSD Flash Memory through the JTAG interface on Port C of the PSD during manufacturing.
2. The firmware, containing the VC5402 Serial Port control code to download future code updates into the PSD813F5V Flash memory, is downloaded to the DSP DARAM during the Boot operation after Power On Reset is over.
3. The PSD813F5V Page Register is used to expand external Local memory beyond 64K words.

DEVELOPMENT SYSTEMS

The PSD family is supported by PSDsoft Express, a software development tool that runs on Windows 95 and 98 and Windows 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 Express 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 and supports device chaining of multiple PSDs and devices from other manufacturers. FlashLink is available on *www.st.com/psm* for $69USD.

PROGRAMMING THE PSD813F5V IN-CIRCUIT USING THE JTAG 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 through this interface.

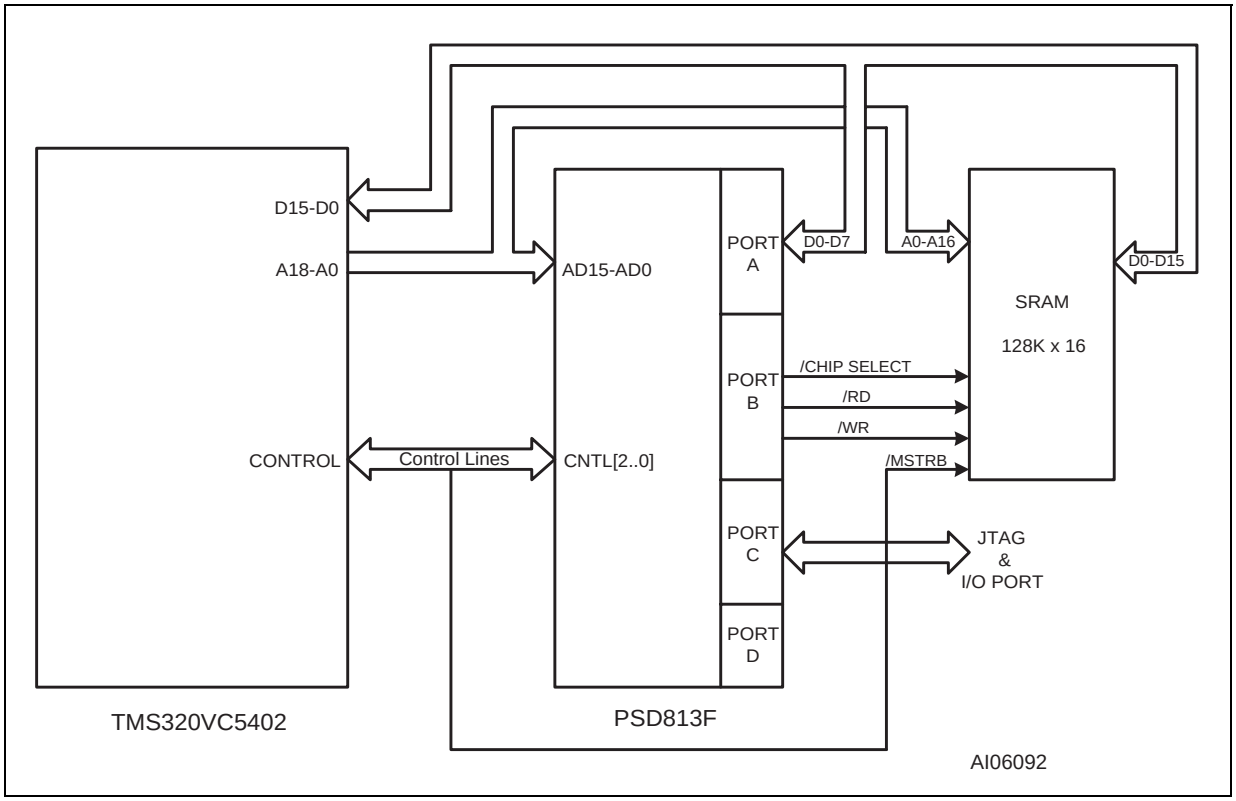
Port C also gives the option to multiplex its JTAG pins with the PSD813F5V general I/O lines. This option, if used, frees up the JTAG pins for I/O functions after JTAG programming is completed. This option is enabled by the following three lines of code in PSDabel, and its hardware implementation is 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
```

INTERFACING THE PSD813F5V WITH THE TMS320VC5402

Figure 2 is a Block diagram that shows the implementation of a two-chip System (three, if external SRAM is required) using the PSD813F5V and the TMS320VC5402. All glue logic, Flash memory, bus interface logic, I/O, chip selects and PLDs are contained in one chip.

Figure 2. Block Diagram – Minimized DSP System



PSD813F5V Bus Interface

The PSD813F5V 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 TMS320VC5402 used to access the Flash memory, PLD logic and I/O inside the PSD813F5V.

NOTE: These bus signals are also used to access the EEPROM/Flash Boot and SRAM inside the PSD813FxV family, if these options are required.

Table 2. Bus Interface Pin Functions

TMS320VC5402 Pin Functions	PSD813F5V Pin Functions	Pin Description
A15 – A0	AD15 – AD0	External Address bus addresses all external memory – Program, data and I/O.
A19 – A16	N.C.	External Address lines which address only external program space.
D15 – D8	N.C.	8 high byte bi-directional external Data bus lines.
D7 – D0	Port A PA7 – PA0	8 low byte bi-directional external Data bus lines. Port A is used as the 8-bit Data bus into the ZPSD813FV.

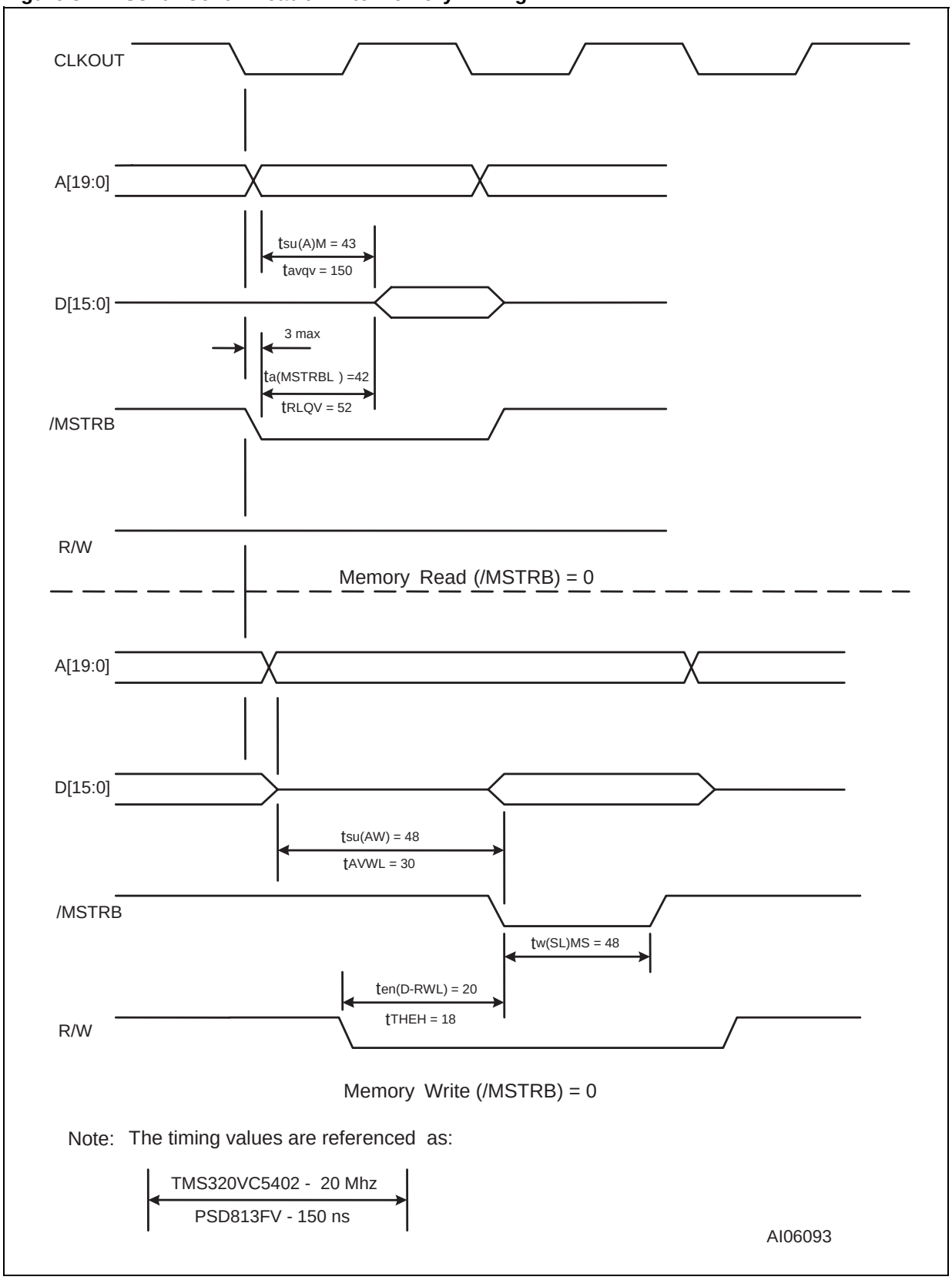
R/W	CNTL0	Read/write signal is used to access external memory and devices.
/DS	CNTL1	Data, program and I/O select signal is driven low to access external memory space.
/MSTRB	Port B PB7	Memory strobe signal is driven low when accessing external data or program memory.
/IOSTRB	Port D PD0	I/O strobe signal is driven low when accessing external I/O.

TMS320VC5402 Bus Interface Timing Calculation

The TMS320VC5402 has an internal software-programmable wait-state generator (SWWSR) which can extend external data, program and I/O bus cycles up to fourteen machine cycles. At reset, the SWWSR is initialized for seven wait states on all external memory accesses. All calculations are based on the R/W, /MSTRB and /IOSTRB. /DS, /PS and /IS are active low for the duration of a valid address and do not figure in the calculations. Figure 3 compares the critical read/write timing differences between the TMS320VC5402-20 MHz (-50ns) and PSD813F5V-150ns. The timing diagram is based on the following parameters:

1. The TMS320VC5402 Rev.B must be operated with the on-chip oscillator. It does not support an external clock source.
2. The maximum crystal frequency is 20Mhz, but the internal PLL circuitry can multiply the crystal frequency by one of 31 possible ratios. For this application, the DSP Clock Mode register (CLKMD is set by the external mode pins – CLKMD1, CLKMD2 and CLKMD3 to generate a system clock of PLLx1 (system clock frequency = crystal frequency).

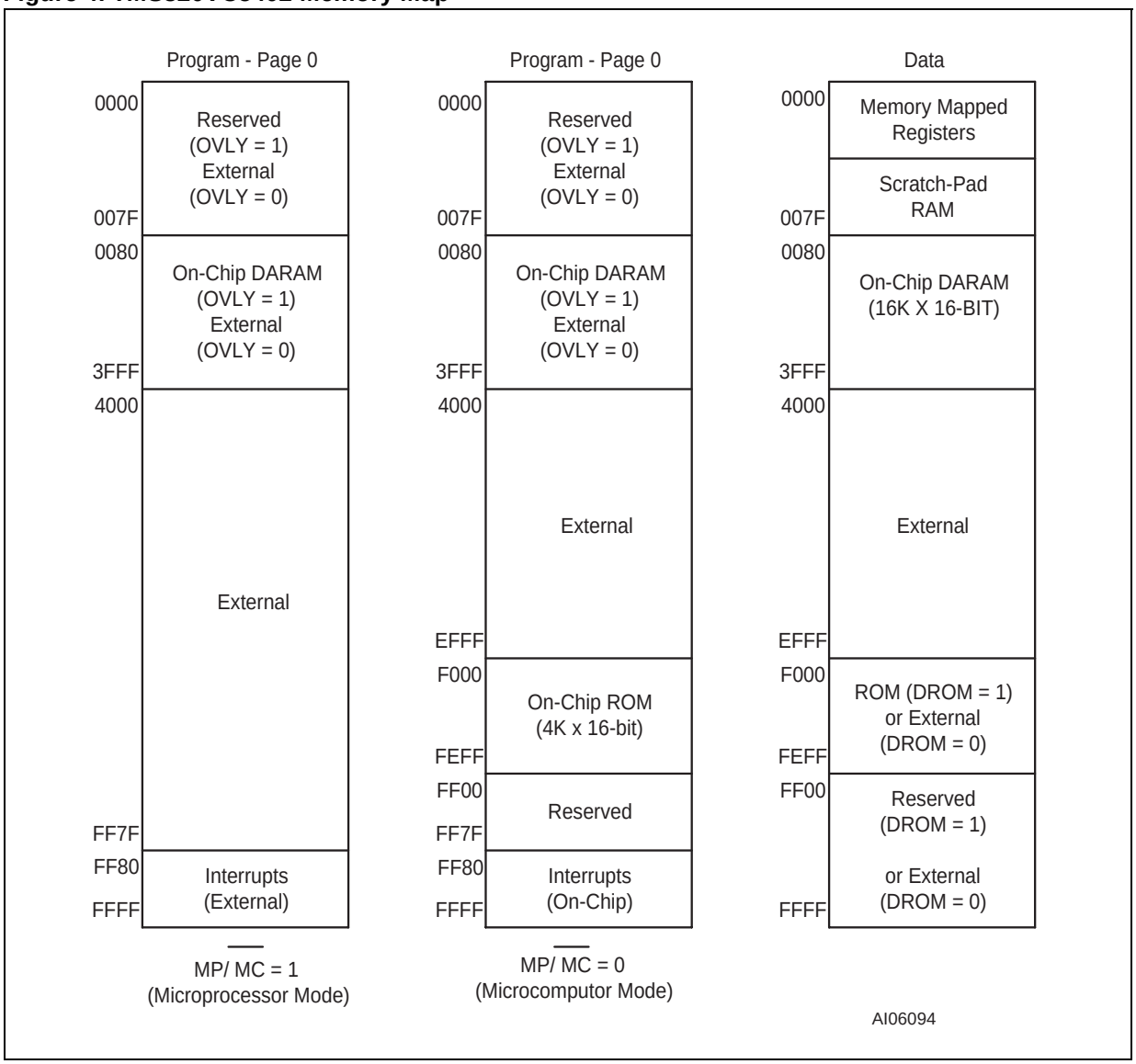
Figure 3. TMS320VC5402 Read / Write Memory Timing



TMS320VC5402 Memory Map

The TMS320VC5402 memory map is illustrated in Figure 4. The TMS320VC5402 has 20 Address lines (A0-A19) which can access 64K words of external data and up to 1M words of program memory. A16-A19 are only used to access program memory via the internal programmable bank-switching logic. The internal 8-bit Page register of the PSD813F5V is used to allow the DSP to access external data in excess of 64K words.

Figure 4. TMS320VC5402 Memory Map



The Processor Mode Status Register (PMST) shown in Table 3 shows the various Memory Mapping options available to configure the TMS320VC5402 Memory Map for optimum system performance:

Table 3. Processor Mode Status Register

15-7	6	5	4	3	2	1	0
IPTR	MP/$\overline{MC}$	OVLY	AVIS	DROM	CLKOFF	SMUL	SST

Table 4 describes only the control bits and their settings that are used to set up the System Memory Map for this Application Note:

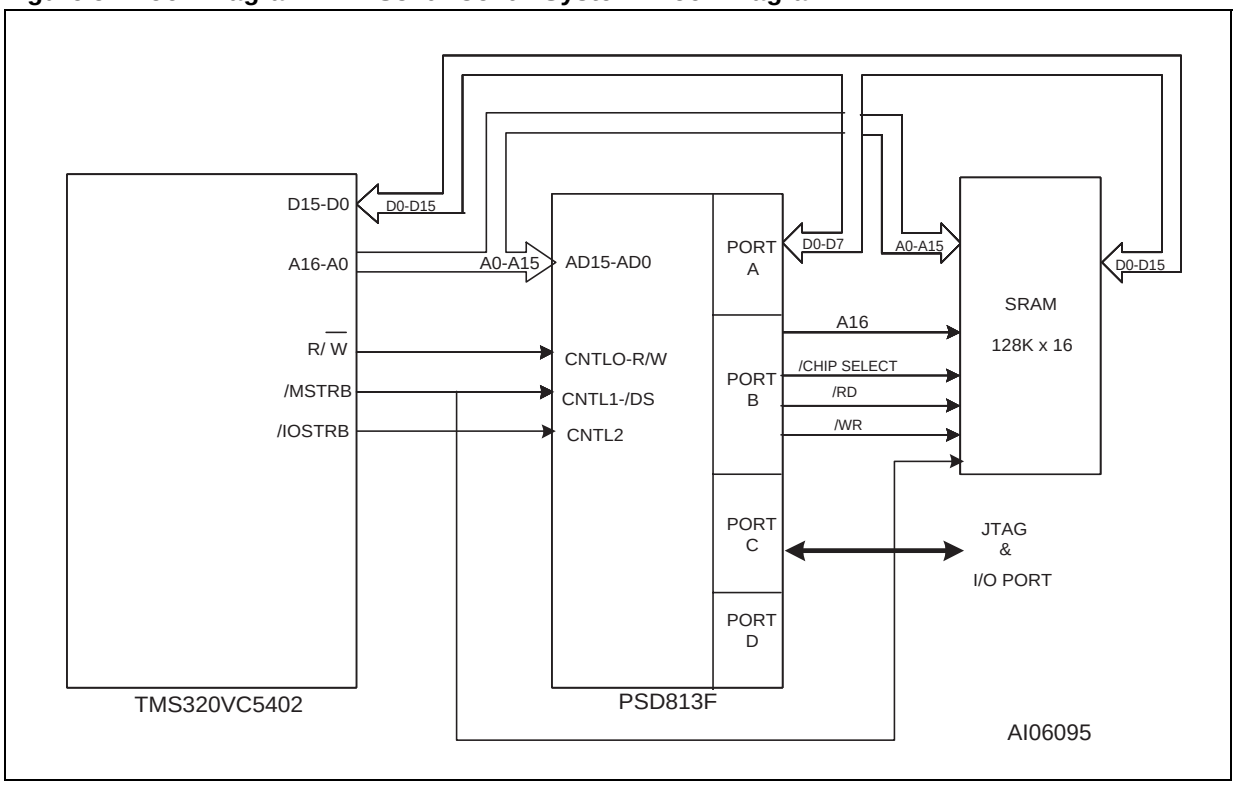
Table 4. Processor Mode Status Register Bit Summary

Bit	Name	Reset Value	Function
6	MP/ $\overline{MC}$	MP/ $\overline{MC}$ pin	Microprocessor / Microcontroller mode. MP/ $\overline{MC}$ enables / disables the on-chip ROM that is addressable in program memory space. MP/ $\overline{MC}$ = 0 The on-chip ROM is enabled. MP/ $\overline{MC}$ is set to the external logic level on the MP/ $\overline{MC}$ pin that is sampled at reset. This pin is not sampled again until the next reset. This bit is also set or cleared by software
5	OVLY	0	RAM overlay. OVLY enables on-chip dual-access data RAM blocks to be mapped into program space. OVLY = 1. The on-chip RAM is mapped into program and data space. Data page 0 (0h –7Fh) is not mapped into program space.
3	DROM	0	Data ROM. DROM enables on-chip ROM to be mapped into data space. DROM = 0. The on-chip ROM is not mapped into data Space.

Interfacing to the TMS320VC5402 External Memory Bus

The Block Diagram of Figure 5 shows the bus interface between the TMS320VC5402 and the PSD813F5V. The TMS320VC5402 has 20 address lines (the four high-order address lines are used for paging external program memory. Because the Flash memory in the PSD813F5V is placed in data space, A19-A16 are not used. The PSD813F5V internal page register is used to configure the Flash memory into multiple pages. A16 for the external SRAM is generated by one of the page register bits. For this application, The SRAM is used for data storage, but the bus interface can be reconfigured to split the external SRAM between data and program memory.

Figure 5. Block Diagram – TMS320VC5402 System Block Diagram



Define the 320VC5402 Interface in PSDsoft Express Define PSD and MCU Utility

Figure 6 is the MCU and PSD Selection screen from the PSDsoft Express development software. For more information on the PSDsoft Express, see the on-line User Manual on the ST website listed on the back page. The bus configuration between the TMS320VC5402 and PSD813F5V is quickly configured by selecting the appropriate signals in this screen, as shown below:

- Type: Other
- Data Bus Width: 8-bit
- Address / Data Mode: Non-Mux
- Control Setting: R/W, /DS

Figure 7 is the schematic diagram of the TMS320VC5402 / PSD813F5V bus interface. The 128K words are used for data storage, but the /PS signal has been wired into the PSD813FV so that the external SRAM can be divided between program and data. The SRAM_cs SRAM chip select signal generated by the PSD and the system memory map would have to be modified by software to allow this feature.

Figure 6. PSDsoft Bus Configuration

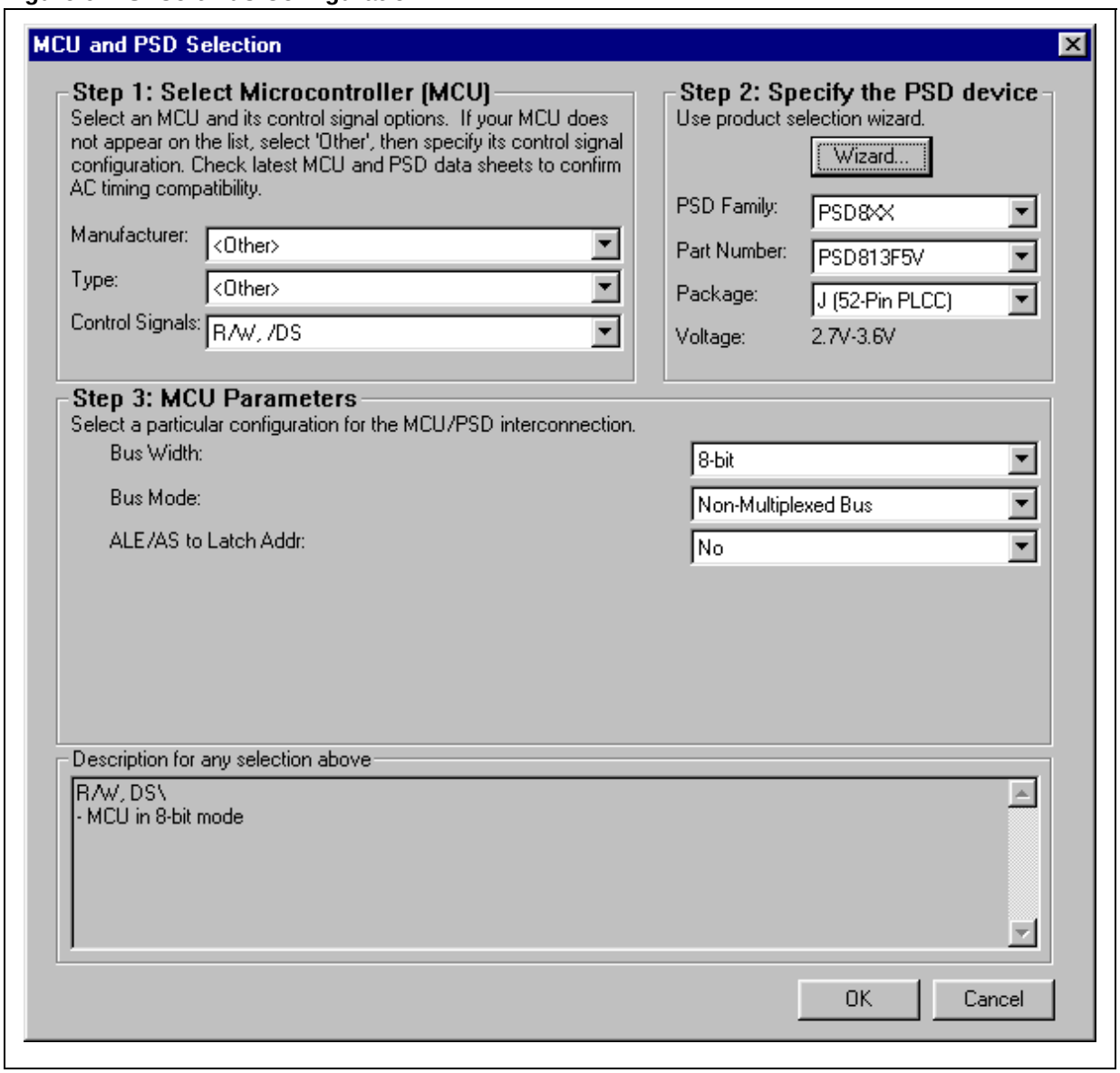
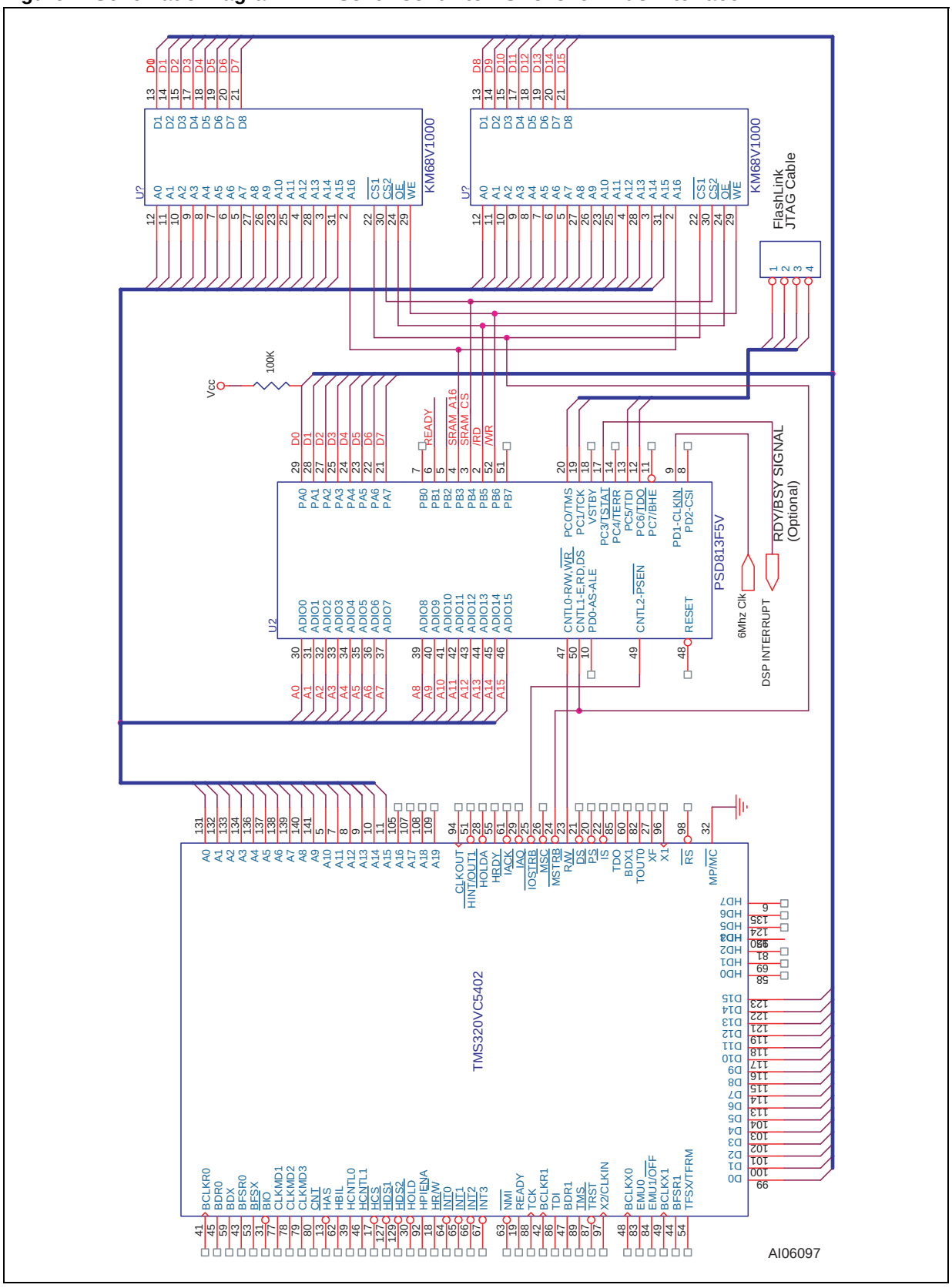


Figure 7. Schematic Diagram – TMS320VC5402 to PSD813F5V Bus Interface

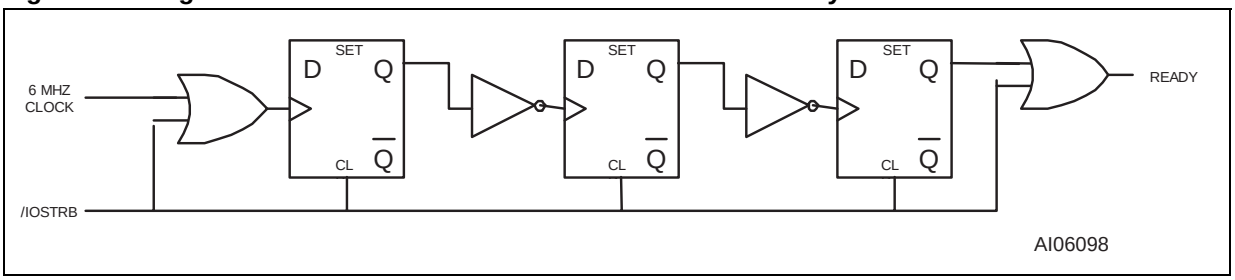


Accessing Slower External I/O and Peripherals

It sometimes becomes necessary to access slower external I/O and peripherals. This can be accomplished either by increasing the number of wait states or by stretching the external Ready pulse. Figure 8 illustrates one method of utilizing the available CPLD Micro⇔Cells to accomplish this task in hardware.

While IOSTRB is inactive (high), the 3-bit ripple counter is held in Reset, the Ready signal is high, and the external Clock is disabled. When IOSTRB goes active (low), the Ready pulse is forced low and the 3-bit counter begins to count up until the Q output of bit-3 goes high and forces the Ready pulse high. The Ready pulse in turn forces the IOSTRB signal high, disabling the counter.

Figure 8. Using the PSD813FV CPLD to Stretch the External Ready Pulse

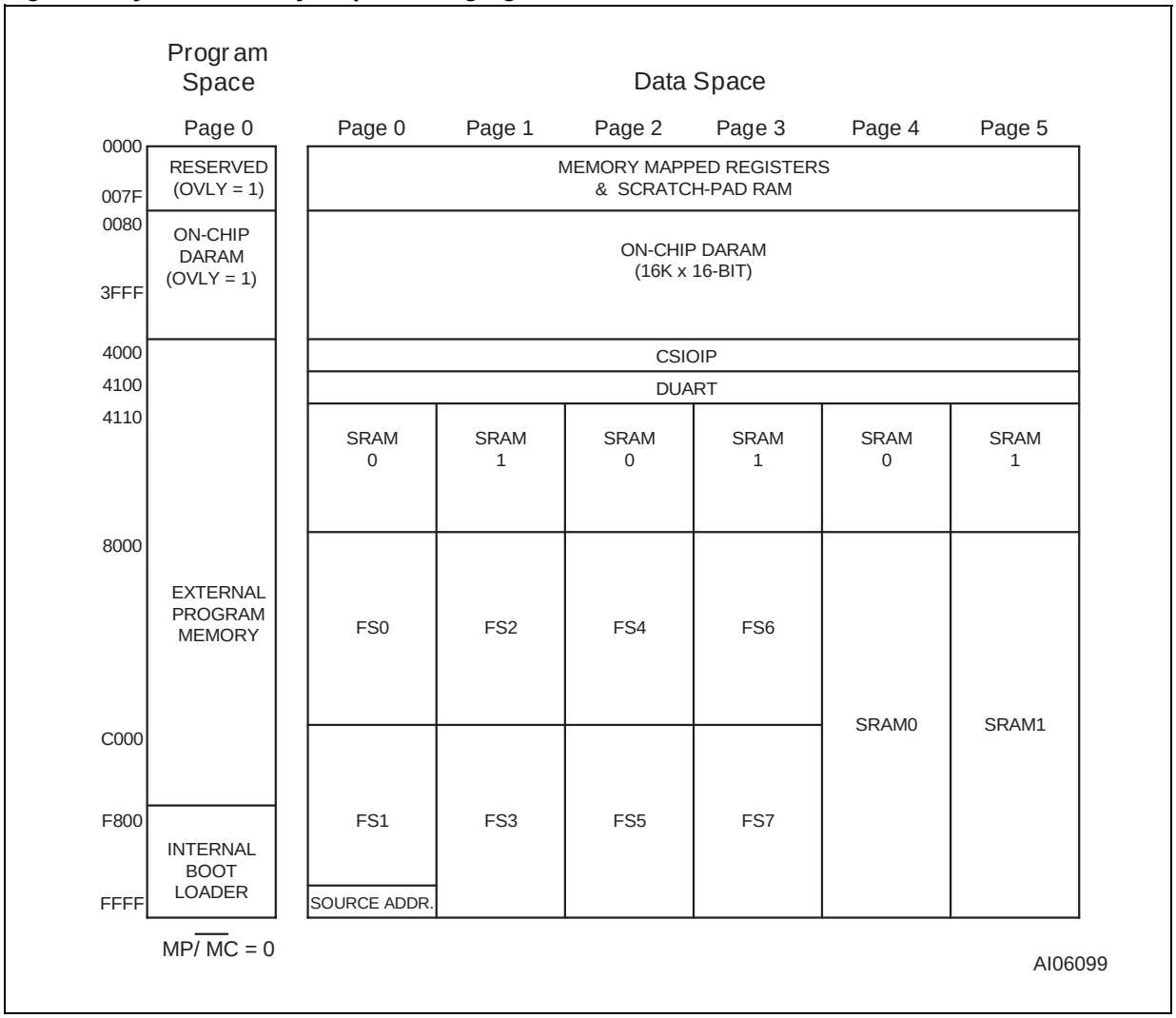


Define the PSD813F5V DPLD Functions in PSDsoft Express Edit/Add Logic Statements

Figure 9 is the system memory map created for this A/N. The data, I/O and program addresses are defined in the PSDsoft Express Edit/Add Logic Statements screen and implemented in the internal PSD813F5V Decoding PLD (DPLD). Three bits of the PSD813F5V Page Register are used to extend the external data address range beyond the 64K limitation of the TMS320VC5402. Since paging is used, an area in data memory containing routines common to all data memory pages – memory-mapped registers, scratch-pad RAM, I/O and external peripheral – must be accessible independent of which page the DSP is addressing. There are also several DSP system requirements that dictate the basic format of the system data memory map:

1. The DSP OVLY bit controls address 0 to 0x3FFF as being internal /external program space. When OVLY = 1, this range is always internal data space when accessed by the DSP as data.
2. If an SRAM block is not included in page 0, program code can only be downloaded to SRAM under DSP supervision. The boot loader by itself can not perform this procedure.
3. The first 32K words of SRAM are lost because the region 0 to 0x3FFF can not be accessed as data; program code can not be written into the SRAM.

Figure 9. System Memory Map with Paging



Accessing the PSD813F5V Internal Registers

The bank of internal control registers in the PSD813F5V (csiop + hXX) are 8-bits wide. The DSP data bus is 16-bits wide and accesses external memory locations on a word basis. When an internal register in the PSD813F5 is accessed, the DSP reads or writes a 16-bit word; only the lower 8 lines of the data bus are used to transfer data between the DSP and PSD. For this application, The internal PSD813F5V Control Registers are mapped into Data space and enabled during /MSTRB active.

When data is read from the internal registers, 16-bit data is read into the DSP accumulator. The high byte in the DSP accumulator is either ignored or masked out. When data is written to the internal registers, valid data must be located in the lower byte of the accumulator. The high byte is ignored by the PSD813F5V.

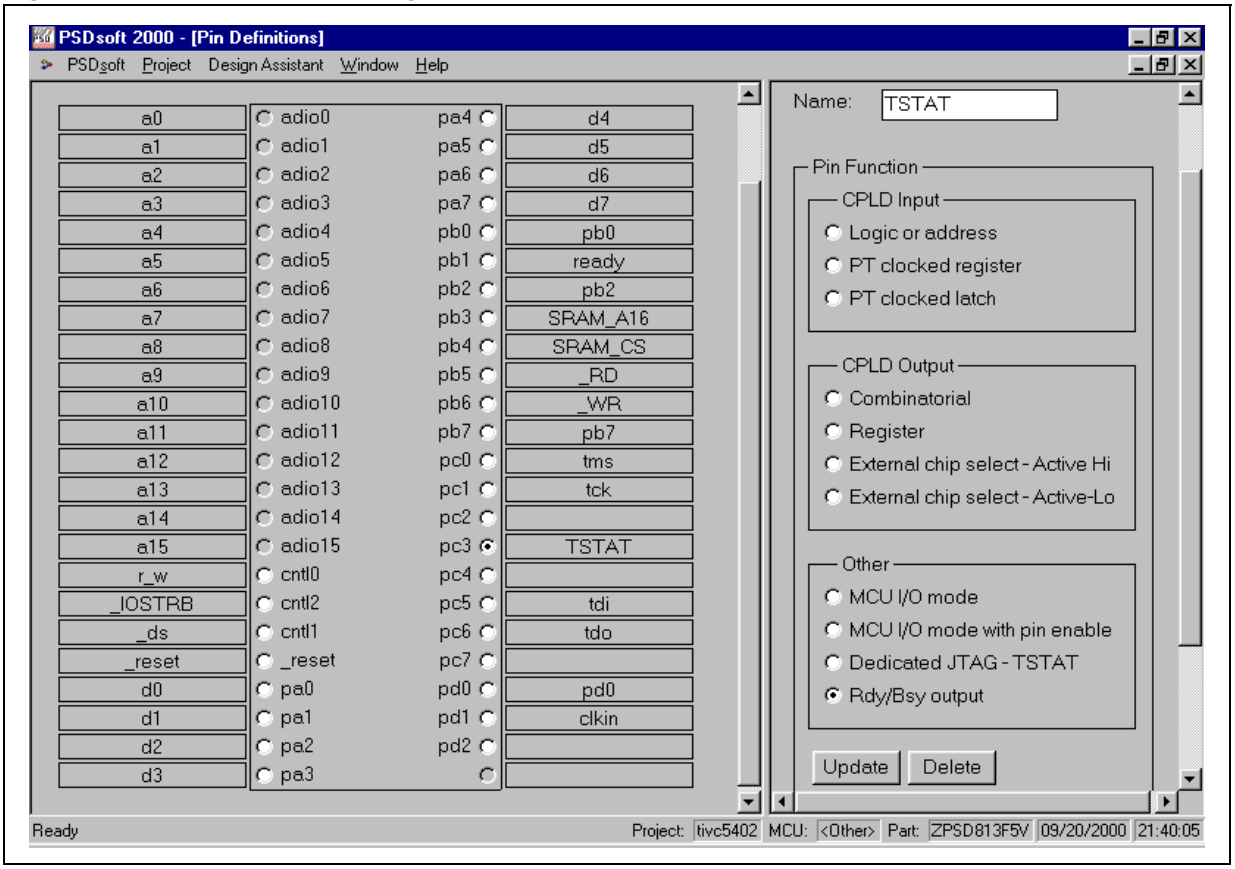
IN-APPLICATION RE-PROGRAMMING (IAP) USING THE PSD813F5V

The PSD813F5V (without internal Boot Flash) was selected to reduce the system cost and take advantage of the external SRAM or internal DARAM that can contain the program code to ISP the Flash memory through the DSP Serial Port Interface (SPI).

The TMS320VC5402 transmit and receive data buffers of the SPI are 16-bit buffers. The SPI control registers (SPC and SPCE) control the SPI communication format. When the FO bit in the SPC register = 1, and the FE bit in the SPCE register = 0, the SPI is formatted for 8-bit interface. An 8-bit word is stored in the lower eight bits of the 16-bit 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 written to the accumulator to store either in an assigned 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 perform 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 11 with the configuration sequence as follows:

Figure 10. Additional PSD Configuration Screen



1. Select pc3 from the PSDsoft Express Additional PSD configuration screen.
2. Type TSTAT in “Name” block.
3. Select “Rdy/Busy output” under “Other” block.

TMS320VC5402 BOOTLOADER

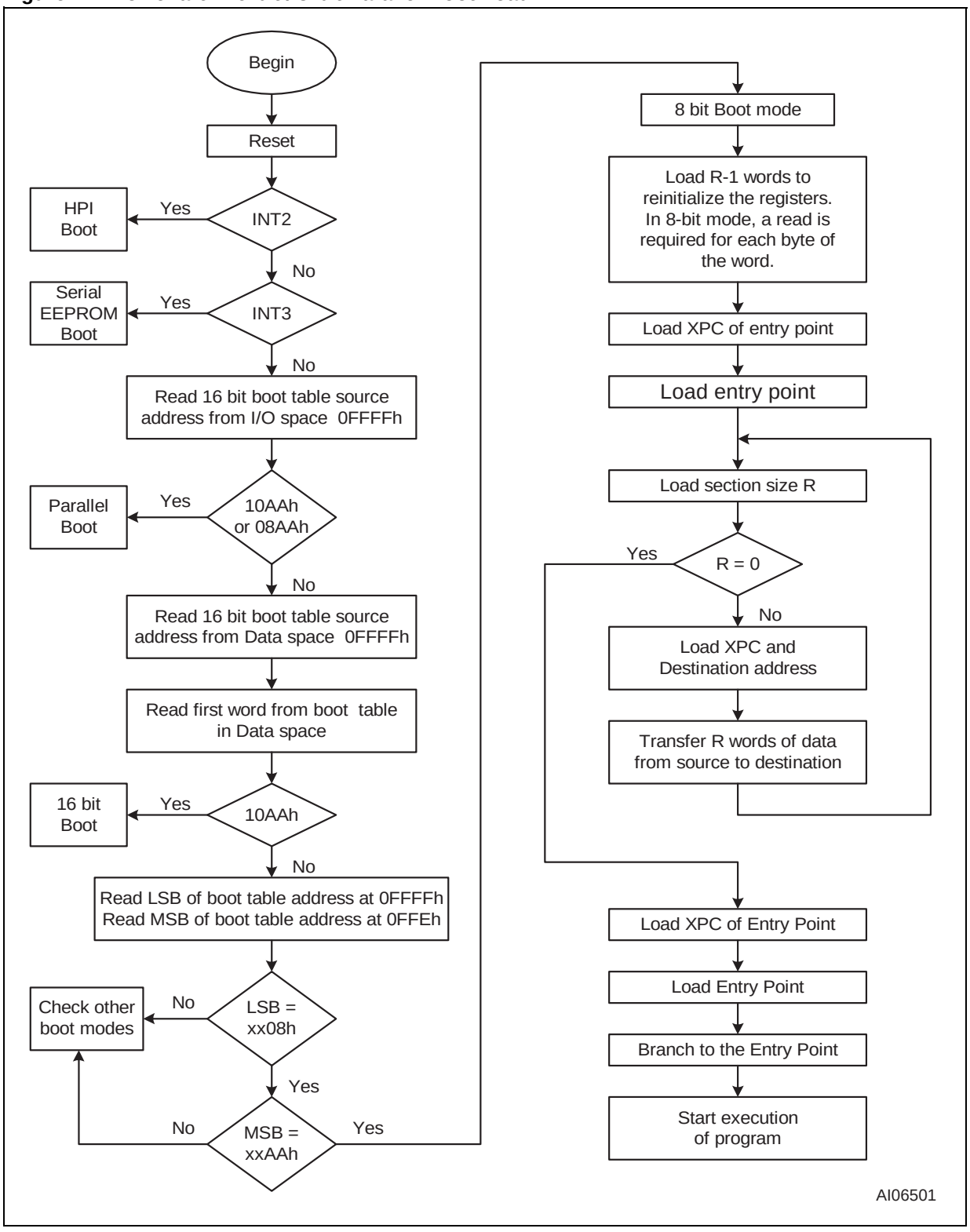
The internal bootloader is used to transfer program code from an external source to the selected program memory at power-up. Five different routines are available to download code depending on the system requirements. Since the code to be downloaded is stored in 8-bit Flash memory in global data space, the parallel boot option is selected.

If the MP/MC pin of the TMS320VC5402 is sampled low during a hard reset, execution begins at location FF80h of the on-chip ROM, which contains a branch instruction to the start of the bootloader program. The on-chip ROM is factory programmed with the boot load program.

The boot load program sets up the DSP status registers before initializing the boot load routine. Interrupts are globally disabled (INTM = 1), internal DARAM and external SRAM is mapped in program/data space (OVLY = 1), and seven wait states are programmed for the entire program and data spaces. The boot loader checks the Host Port Interface (HPI) first by sampling interrupt 2 (INT2). If INT2 is not latched, the boot routine skips HPI mode and samples interrupt 3 (INT3). If INT3 is not latched, the boot loader will try to read the source address from external I/O location 0FFFFh, which is accessed by IOSTRB. This entails having a non-volatile value (either 10AA or 08AA) at this address. If this is not possible, one way to bypass booting from I/O address is to pull up D0 of the DSP data bus with a weak resistor (100K) when accessing a non-existent address at 0FFFF in I/O space to guarantee that a logic “1” is present during this read. This will ensure that a valid keyword will not be present on the data bus, which will be tri-state during the read access.

The boot loader next reads the source address of the boot code from external Data space located in Flash data memory sector fs1 to determine if the interface is 8-bit or 16-bit. Once the boot loader determines that the VC5402 DARAM will be uploaded from external 8-bit memory (W = 08AA), the bootloader commences with the 8-bit upload subroutine. Figure 11 is the flowchart for the 16-bit/8-bit parallel boot load routine.

Figure 11. Flowchart - 16-bit / 8-bit Parallel Boot Load



SUMMARY

As DSPs continue to rapidly proliferate into markets such as communications, industrial, medical, signal conditioning, and hand held test equipment, the PSD813Fx 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 PSD813Fx 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 PSD813Fx now become both cost and performance effective.

Several features internal to the PSD813F5V were used to expand the limitations of the TMS320VC5402, and DSPs in general:

1. The JTAG-ISP channel is used to eliminate manufacturing complications associated with Flash memory and programmable logic.
2. The Page Register was used to expand the address range of the external data memory accessible to the DSP.
3. Flash memory allows the program code to be field updated through the serial port of the DSP while the DSP is running program code in the internal DARAM.
4. Expanded I/O was added to the system.
5. The internal PLD allows functional modifications to the I/O, these changes being made in software.

These changes have added to both the versatility and performance of the TMS320VC5402; future changes most likely will not require a hardware change to the 2-chip core.

APPENDIX

The Appendix contains the PSDsoft Express Design Assistant Summary listing all logic equations and showing how the PSD813F5V is configured to implement the example in this Application Note. Application Note AN1356 presents a step-by-step illustration of how to configure the Flash PSD Family. Although AN1356 uses the 16-bit Flash PSD4235G2 in the example, the software and procedure is the same for the 8-bit PSD813F5V.

```
******************************************************************************
                          PSDsoft Express Version 6.02
                       Summary of Design Assistant
******************************************************************************
PROJECT    : tivc5402                       DATE : 09/20/2000
DEVICE     : ZPSD813F5V                     TIME : 21:42:37
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        r_w          MCU bus control signal
cntl2        _IOSTRB      Logic or address
cntl1        _ds          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          pb0          MCU I/O mode
pb1          ready        Combinatorial
pb2          pb2          Logic or address
pb3          SRAM_A16     Combinatorial
pb4          SRAM_CS      External chip select - Active-Lo
pb5          _RD          Combinatorial
pb6          _WR          Combinatorial
pb7          pb7          MCU I/O mode
pc0          tms          Dedicated JTAG - TMS
pc1          tck          Dedicated JTAG - TCK
pc3          TSTAT        Rdy/Bsy output
pc5          tdi          Dedicated JTAG - TDI
pc6          tdo          Dedicated JTAG - TDO
pd0          pd0          MCU I/O mode
pd1          clkin        Common clock input, CLKIN

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

Node         Node
Name         Type
------------ ------------
bit1         Register
bit2         Register
bit3         Register

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

pgr0 is used for paging
pgr1 is used for paging
pgr2 is used for paging
pgr3 is not used
pgr4 is not used
pgr5 is not used
pgr6 is not used
pgr7 is used for logic, signal name: la16

Equations:
==========
```

```
csiop = ((address >= ^h4000) & (address <= ^h40FF));
fs0 = ((page == 0) & (address >= ^h8000) & (address <= ^hBFFF));
fs1 = ((page == 0) & (address >= ^hC000) & (address <= ^hFFFF));
fs2 = ((page == 1) & (address >= ^h8000) & (address <= ^hBFFF));
fs3 = ((page == 1) & (address >= ^hC000) & (address <= ^hFFFF));
fs4 = ((page == 2) & (address >= ^h8000) & (address <= ^hBFFF));
fs5 = ((page == 2) & (address >= ^hC000) & (address <= ^hFFFF));
fs6 = ((page == 3) & (address >= ^h8000) & (address <= ^hBFFF));
fs7 = ((page == 3) & (address >= ^hC000) & (address <= ^hFFFF));
! SRAM_CS = ((address >= ^h4100) & (address <= ^h7FFF) & (!SRAM_A16))
     # ((address >= ^h4100) & (address <= ^h7FFF) & (SRAM_A16))
     # ((page == 4) & (address >= ^h8000) & (address <= ^hFFFF) & (!SRAM_A16))
     # ((page == 5) & (address >= ^h8000) & (address <= ^hFFFF) & (SRAM_A16));
ready = bit3 # _IOSTRB;
bit1.ck = clkin # !_IOSTRB;
bit1.re = _IOSTRB;
bit2.ck = !bit1;
bit2.re = _IOSTRB;
bit3.ck = !bit2;
bit3.re = _IOSTRB;
```

Table 5. Document Revision History

Date	Rev.	Description of Revision
Oct-2000	1.0	Document written in the WSI format (AN073)
01-Mar-2002	2.0	Document converted to the ST format (AN1429)

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.