AN1174
APPLICATION NOTE

Use the PSD813F to Minimize Signal Conditioning System Hardware

CONTENTS

Use the PSD813F to Minimize Signal Conditioning System Hardware

By: Don Buccini

INTRODUCTION

The proliferation of inexpensive microcontrollers, DSPs and memory has fueled an explosive growth in hardware and systems that monitor and interface with the physical world. This physical world, which is perceived through the human senses, is analog in nature and must be converted into a format that is compatible with today's data acquisition technology. Specific sensors are designed to monitor this physical stimulus. The accelerating growth in markets such as environmental, process controls, automotive, HVAC, appliances, etc., is forecasted to grow the sensor market from $10.4 billion in 1995 to $20.1 billion in 2000. The data received from these sensors is conditioned and processed by microcontrollers and DSPs for subsequent data sampling, processing, and storage.

PURPOSE

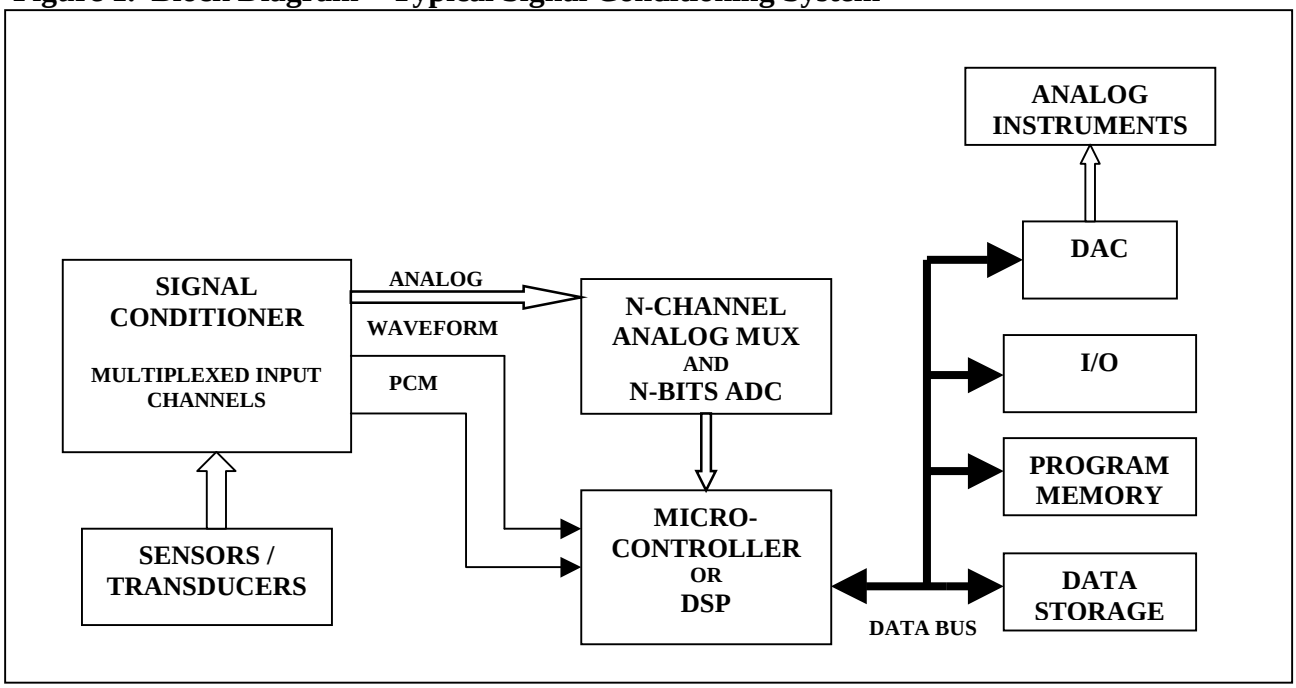
This Application Note focuses on the Signal Conditioning Marketplace and how the Flash-based PSD813F family greatly simplifies the hardware design effort while adding enhanced features such as Concurrent Memory and In-System-Programming for easy remote field upgrades. Familiarity with the PSD813F is assumed. Please reference "PSD813F Data Sheet" for a detailed description of the device. The design example will use the Analog Devices ADuC812 Multi-Channel 12-bit ADC with embedded 8051 MCU Core. This is a highly integrated device with 8 analog inputs and 12-bit ADC accuracy, allowing a 3-chip design solution, including the PSD813F, and external SRAM for data storage.

SIGNAL CONDITIONING OVERVIEW

Figure 1 is a Block Diagram showing the hardware components of a typical Signal Conditioning system. The individual components perform the following functions:

Sensor/Transducer…A Sensor is a device that monitors physical stimulus such as temperature, pressure, acceleration, etc. and converts it into a measurable output signal. A transducer is the circuitry associated with the specific sensor that creates the voltage or current excitation for further processing by the Signal Conditioner. This signal can be a variable voltage level, a pulse or periodic waveform, or other measurable formats.

Figure 1. Block Diagram – Typical Signal Conditioning System



Signal Conditioner…The Signal Conditioner optimizes the output voltage from the transducer to match the input range of the Analog-to-Digital converter (ADC). The most frequent form of signal conditioning is amplification of low level output signals. The Signal Conditioner also receives digital data from the Microcontroller/PC and converts it to an analog signal to drive analog meters, chart recorders and other analog instrumentation.

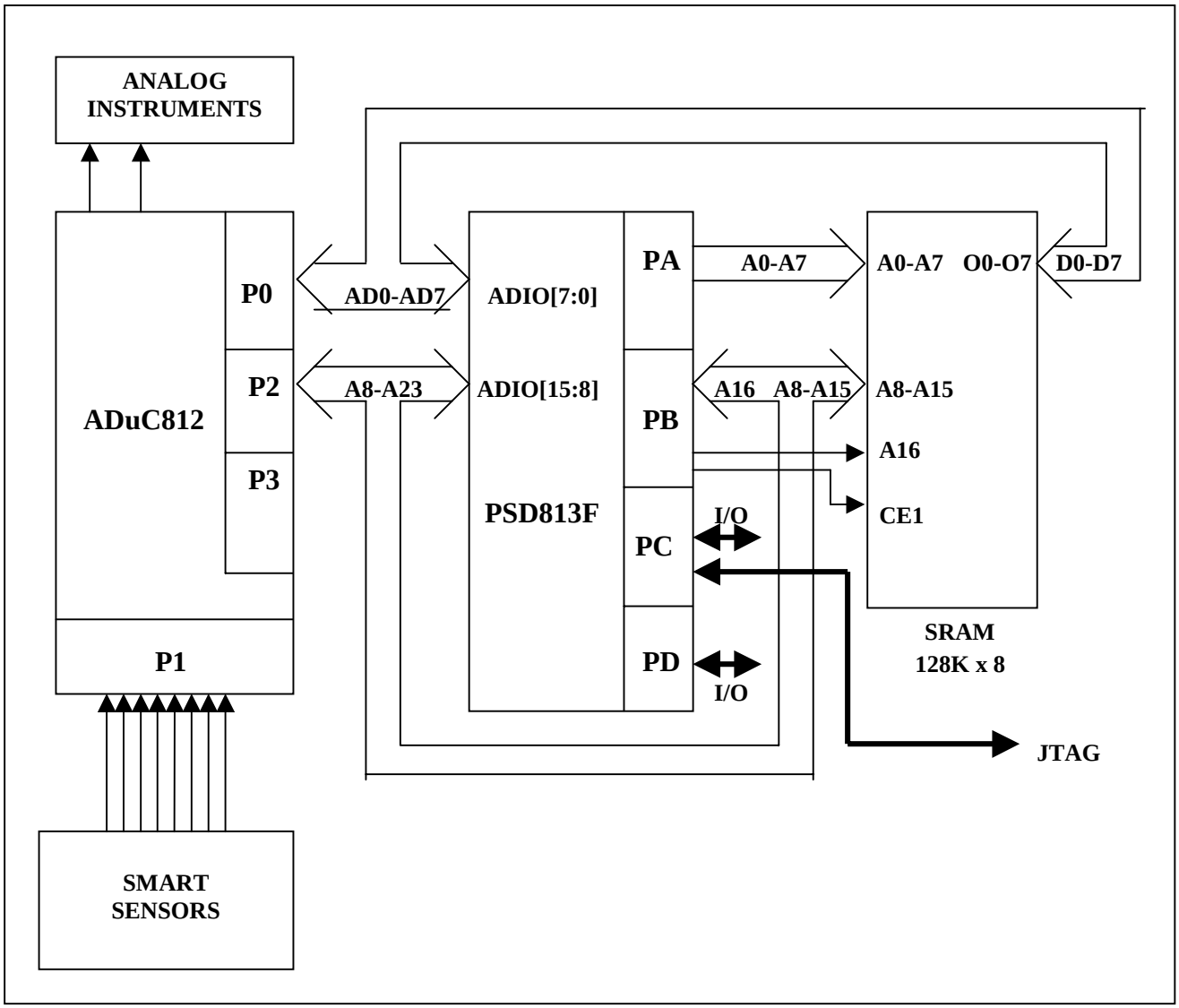
Analog Multiplexer and ADC…This Analog-to Digital Converter converts multiple analog sensor inputs, which are switched in sequence to the ADC, to an n-bit digital value. The sampling rate (frequency response) is dependent on the maximum clock rate of the ADC. This digital value is transferred to the microcontroller or PC for processing, displaying and storage.

Microcontroller…In addition to processing, displaying and storing the sampled data, the microcontroller performs a variety of other functions - I/O, controlling the Signal Conditioning component, chip selects for external peripherals, keyboard scan, etc.

Digital-to Analog Converter (DAC)…The data which is processed in digital format can be reformatted through the Digital-to Analog converter to generate a signal which can be used to drive analog instrumentation such as panel meters and chart recorders and graphs.

Figure 2 is a Block diagram illustrating how the PSD813F and ADuC812 reduce all of the hardware components of the Signal Conditioning system in Figure 1 (except the Sensors and Analog Instruments) to three ICs.

Figure 2. Block Diagram – Minimized Signal Conditioning System Hardware



PSD813F1 ARCHITECTURE

Figure 3 is a Block Diagram of the PSD813F1 that contains all of the options available in the PSD813F family. Table 1. Shows the function matrix of the family. The differences are in the memory options available on each part. The on-chip features supply many of the key elements to implement an effective two-chip Signal Conditioning System, depending on system requirements. The on-chip 32 Kbytes of byte-erasable Flash EEPROM may possibly be used in place of external SRAM. These features include:

* An easily programmable bus interface to the microcontroller with external 8-bit boot code and/or program code capability.

* 128 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.

* 2 Kbytes scratchpad SRAM (8 Kbytes in future pin-compatible upgrade).

* 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 MCU I/Os, PLD I/Os, latched MCU 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 or reprogramming of devices in the factory or field.

Table 1. PSD813F Product Matrix

Device	Flash Main Memory Kbit (8 Sectors)	Additional Memory for Boot and/or Data (4 Sectors)	SRAM Kbit
PSD813F1	1024	256 Kbit EEPROM	16
PSD813F2	1024	256 Kbit Flash	16
PSD813F3	1024	None	16
PSD813F4	1024	256 Kbit Flash	None
PSD813F5	1024	None	None

For this application, the PSD813F5 is used. The 8 Kbytes of Flash Memory in the ADuC812 will be used for both boot and program code, and the 1024 Kbytes Flash Memory in the PSD813F5 will be used for extended program code.

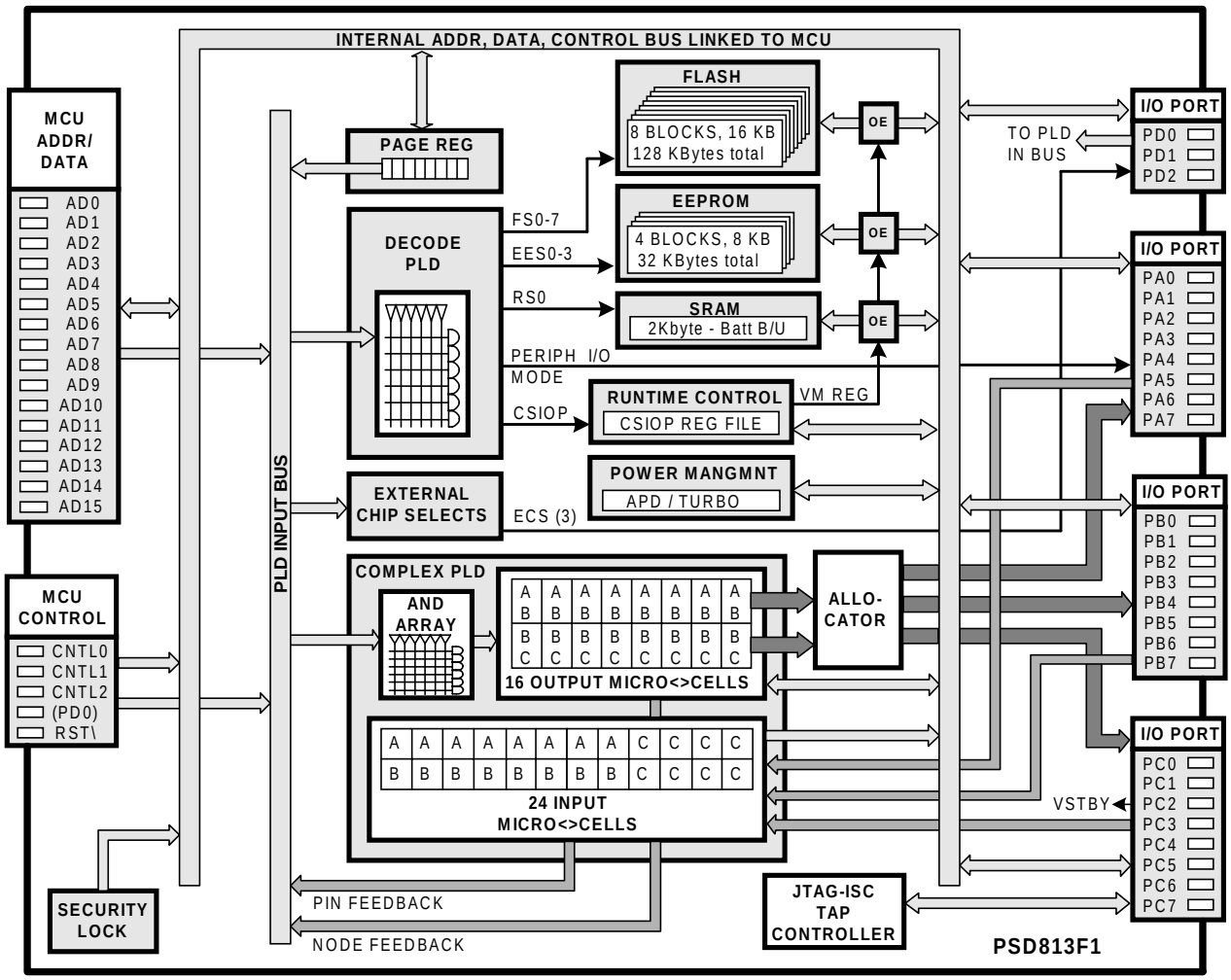
DEVELOPMENT SYSTEMS

The PSD family is supported by the ST Windows 95 & 98 and NT-based software development system, PSDsoft. This software contains several elements to simplify the design task.

The PLD design entry uses an HDL-PSDabel, which creates a minimized logic implementation, and provides logic simulation of the PLDs. The 8051 bus interface is defined in PSD Bus Configuration.

The PSDcompiler, comprised of a fitter and address translator, generates an object file from the PSDabel, PSDconfiguration and MCU firmware files. The object file is then downloaded to a programmer (ST PSDpro, ST FlashLink, Data I/O or other third party programmers for device programming) or to PSDsimulator for device-level simulation.

Figure 3. PSD813F1 block Diagram



ST offers two low-cost device programmers:

PSDpro....plugs into a PC/laptop parallel port and is a replacement for the ST MagicPro III.

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

PROGRAMMING THE PSD813F IN-CIRCUIT USING THE JTAG INTERFACE

Port C shares its I/O pins with the JTAG interface. This allows maximum utilization of Port C I/O lines if the mutiplexed option is selected, since Port C is freed up for I/O functions once the JTAG operation is complete. The PSD813F configuration, PLD logic, Flash memory and EEPROM can be programmed through this interface. The standard JTAG signals are TMS, TCK, TDI, and TDO. TSTAT and /TERR are JTAG extensions that can be used to speed up programming the PSD813F.

For a detailed description of the ST JTAG interface specification, please reference Application Note 054 "JTAG Information – PSD813F".

INTERFACING THE PSD813F TO THE ADuC812

PSD813F BUS INTERFACE

The PSD813F has a user-friendly programmable bus interface that is quickly configured to interface directly to most microcontrollers and General Purpose DSPs with no "glue logic". Table 2 lists the bus interface signals from the ADuC812 used to access the Flash memory, EEPROM, SRAM and PLD logic inside the PSD813F.

Table 2. Bus Interface Pin Functions

AduC812 Pin Functions	PSD813F Pin Functions	Pin Description
P0.0 – P0.7 (A/D0 – A/D7)	ADIO0 – ADIO7	Multiplexed low-order Address/Data Bus accesses external program or data memory
P2.0 – P2.7 (A8 – A15)	ADIO8 – ADIO15	High-order Address byte when accessing external 64 Kbytes program memory space
P2.0 – P2.1 (A16 – A17)	PB0 – PB1	High-order Address byte when accessing external 16 Mbytes data memory space PB0 & PB1 latch A16 & A17 during data access
P3.6 - /WR	CNTL0	Active low Write control pulse latches the data byte from Port 0 into the external data memory
P3.7 - /RD	CNTL1	Active low Read control pulse enables the external data memory to Port 0
/PSEN	CNTL2 - /PSEN	Active low Program Store Enable Pulse enables the external program memory during instruction fetch
ALE	PD0 - ALE	Active high Address Latch Enable pulse latches A0-A7 during external program memory fetch, and A0-A7 & A16-A23 during external data memory access

BUS INTERFACE TIMING CALCULATION

The ADuC812 has a unique way of accessing external memory. PSEN is used to access 64 Kbytes of external Program memory, identical to the way the 8051 core microcontroller fetches external program code. The difference is in how the ADuC812 accesses 16 Mbytes of external data memory. During an external data access, Port P2.0-P2.7 multiplexes A8-A15 with A16-A23. AD0-AD7 from port P0.0-P0.7 and A16-A23 are latched by ALE. A8-A15 are unlatched outputs from P2.0-P2.7 which are stable when data is present on P0.0-P0.6, similar to the 8051 external memory fetch operation. Figure 4. Is the timing diagram for external data read/write access.

ADuC812 MEMORY MAP

The ADuC812 has two separate 8-bit external address spaces – 64K Program and 16M Data. When EA=1, the first 8 Kbytes of program code are executed from the internal 8 Kbytes Flash EEPROM ; above 8 Kbytes, program code is executed from external program memory. When EA=0, all 64Kbytes of code are executed from external memory. The memory map is illustrated in Figure 5, along with their respective select signals. Address lines A23-A16 are multiplexed with A15-A8 to achieve the 16 Mbytes data address range without using bank switching. EA is tied high to Vcc for the system example in this application note.

Figure 4. AduC812 External Data Memory Read/Write Timing

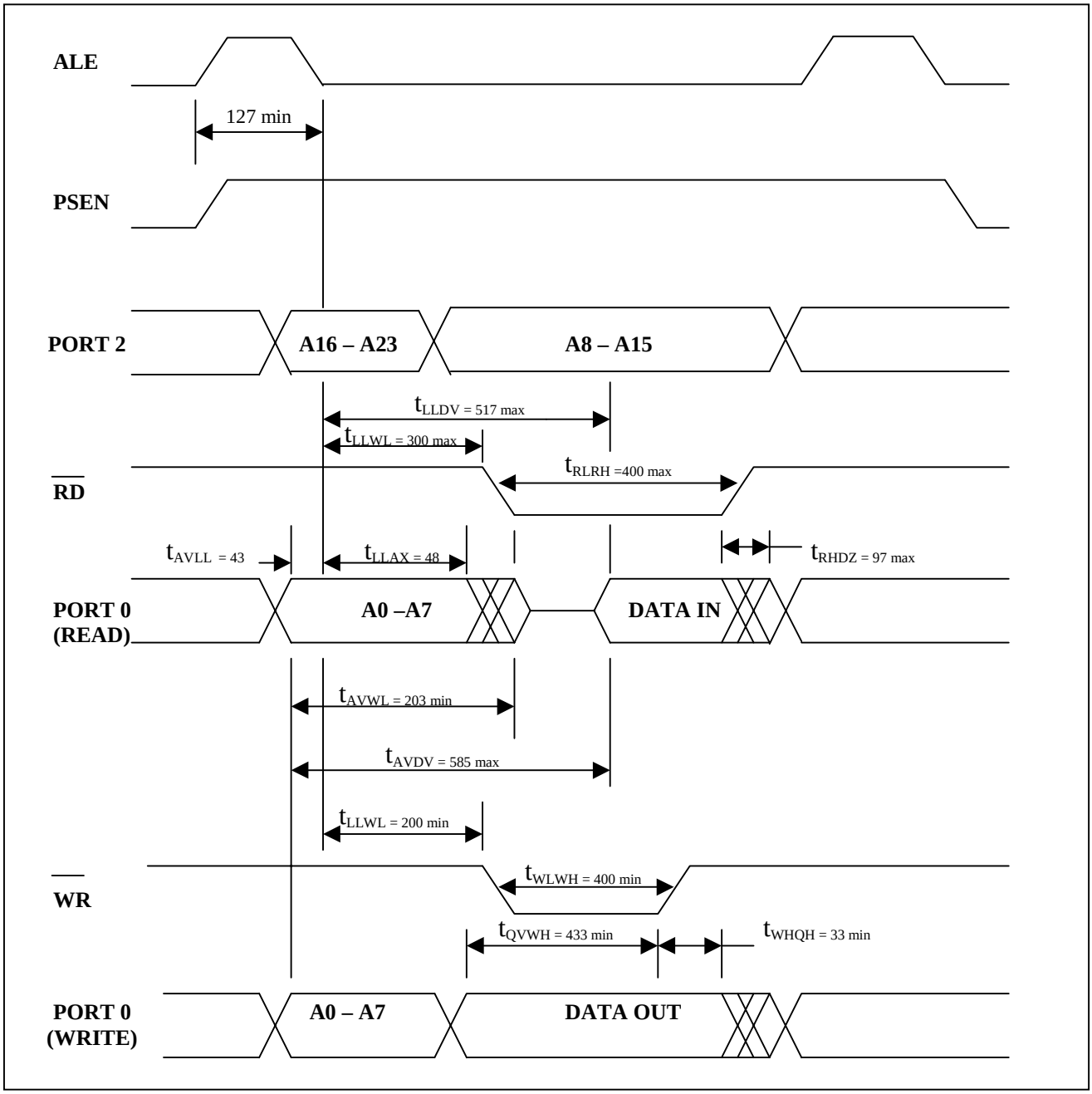
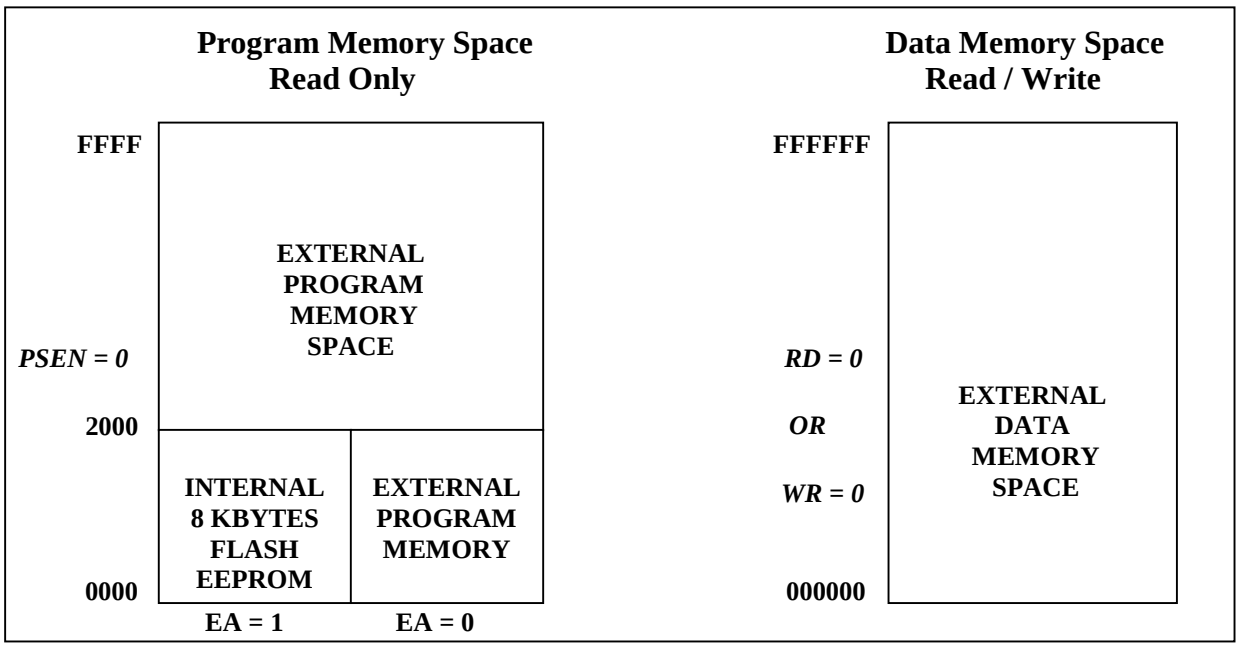


Figure 5. ADuC812 Program and Data Memory Map



INTERFACING TO THE ADuC812 EXTERNAL MEMORY BUS

The Block Diagram of Figure 6 illustrates the bus interface between ADuC812 and PSD813F5. The ADuC812 has 24 address lines and an 8-bit data bus to access external memory. When accessing external data memory, the ADuC812 outputs A0-A7 and A16-A23 simultaneously to be latched by ALE, before data is output on Port 0. The PSD813F5 latches A0-A7 on the dedicated AD0-AD7 pins. For this example, the PSD813F5 uses two PSD Port B I/O pins and two internal Input Micro⇔Cells to latch A16 and A17 with ALE; these latched address lines are brought out to two additional pins of Port B. This allows up to 256 Kbytes of external data SRAM, I/O and peripheral address locations to be accessed. The remaining external memory select signals are defined in PSDabel Design Entry.

Figure 6. Block Diagram – ADuC812 / PSD813F5 Bus Interface

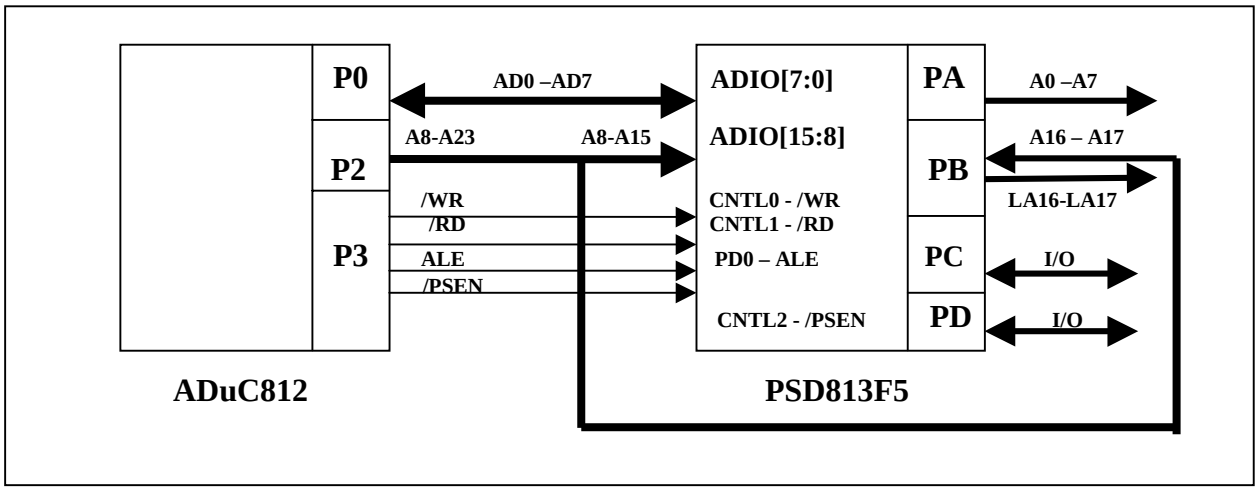


Figure 6 shows how latching A16 and A17 is implemented in the PSD813F5 PLD.

Figure 6. Latching A16 and A17 in the PSD813F5 CPLD

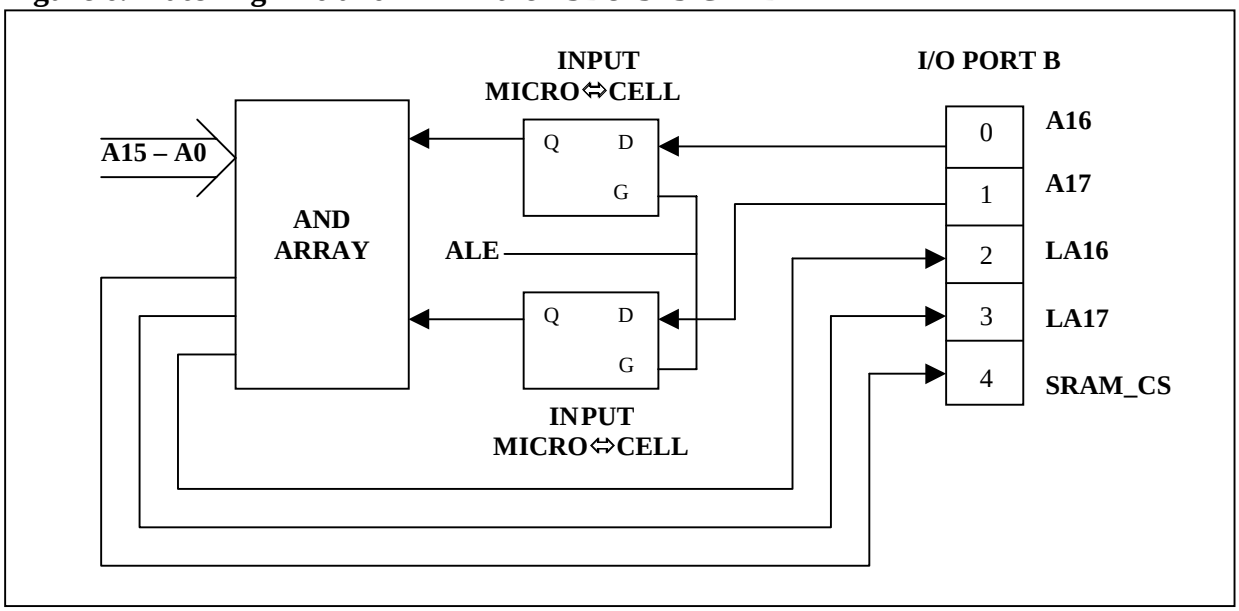


Figure 7 is the schematic diagram of the ADuC812 / PSD813F interface. The 128 Kbytes SRAM is used for data storage, but external data can be expanded to the full 16 Mbytes by using an external 8-bit latch to latch A16-A23, freeing up the I/O Pins of the PSD813F for other functions. An alternative would be to use the 32 Kbytes EEPROM in the PSD813F1, if this storage capacity and write speed is sufficient, and eliminate the external SRAM.

DEFINE THE ADuC812 BUS INTERFACE IN PSDsoft DEVICE CONFIGURATION

Figure 8 is the MCU Bus configuration screen imported from PSDsoft Device Configuration utility. The bus interface between the ADuC812 and PSD813F is quickly configured by selecting the appropriate signals in this screen. The following bus signals are selected in the MCU Bus Configuration screen in PSDsoft. A16 and A17 are connected to the CPLD I/O pins and are included in the internal equations generated by PSDabel Design Entry.

* Data Bus Width:	8-bit
* Address / Data Mode:	Mux
* Control Setting:	/WR, /RD, /PSEN
* Address Latch / Strobe Setup:	ALE = Active High
* VM Register Configuration:	Flash = Program Space

Figure 7. Schematic Diagram – ADuC812 to PSD813F5 Interface

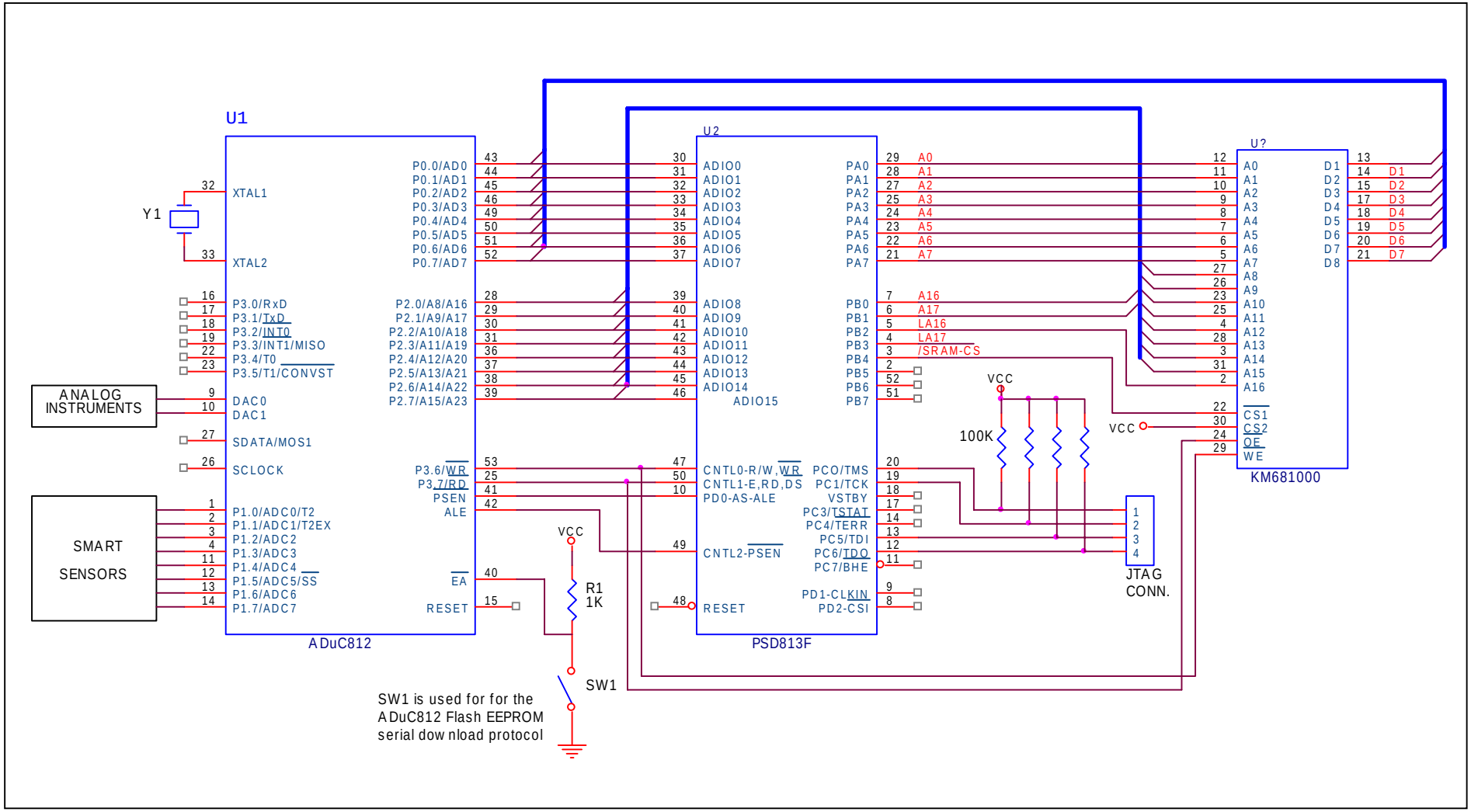
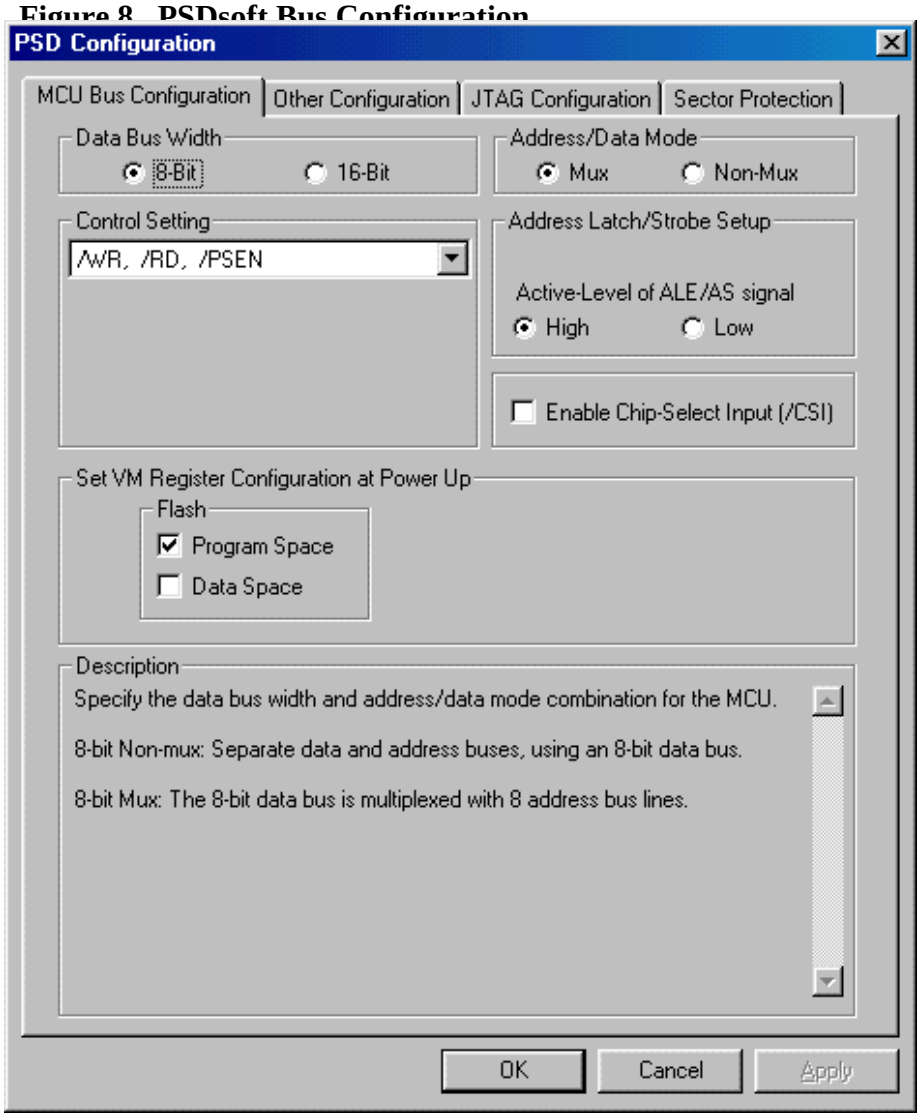


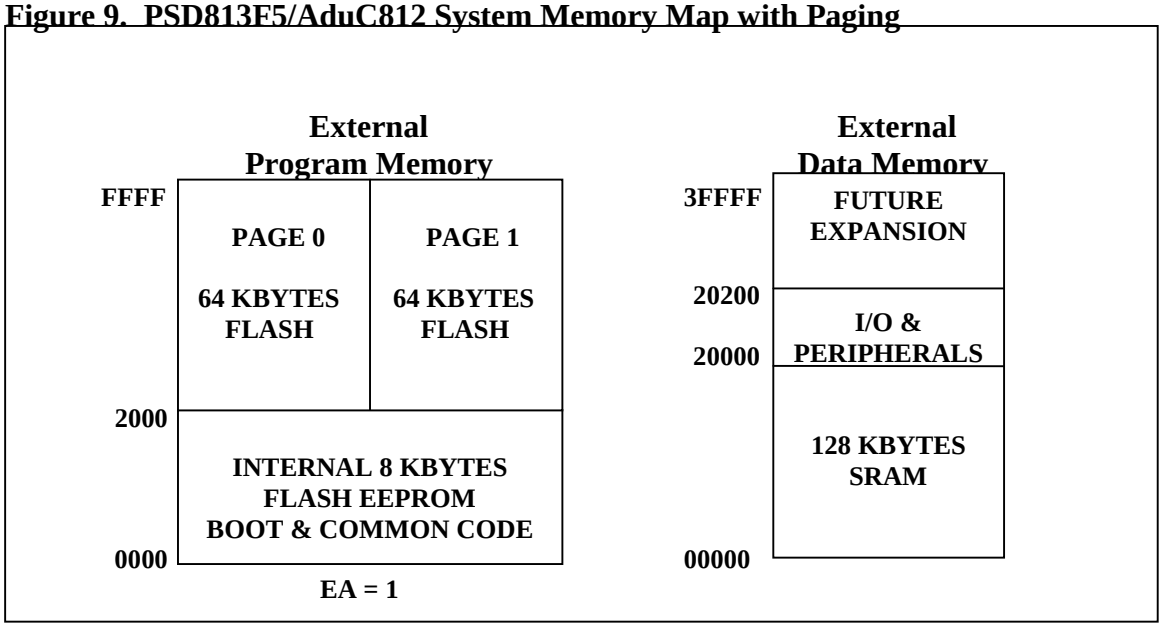
Figure 8. PSDsoft Bus Configuration



DEFINE THE PSD813F5 DECODING FUNCTIONS IN PSDabel DESIGN ENTRY

Figure 9 is the example system memory whose program, data and I/O addresses are defined in the PSDabel Design Entry Screen. The PSD813F5 Page Register is used to extend the external program address range of the 8031 beyond its 64K limitation. Since paging is used, an area in memory containing routines common to all program memory pages – initialization, I/O and SRAM access, page switching, device drivers, etc. - must be accessible independent of which page the MCU is executing code from. To simplify setting up the system memory map within the 64K program address range of the AduC812 – that is, external Flash memory starting at address location 200h, the upper 8 Kbytes of Sectors 3 and 7 in the PSD813F5 Flash memory are not used. The boot program and common program code is stored in the internal 8 Kbytes Flash EEPROM in the AduC812 and is directly accessible from both Page 0 and 1. When the Program Counter in the AduC812 is greater than 1FFFh, code is fetched from the external PSD Flash memory in Page 0 or 1.

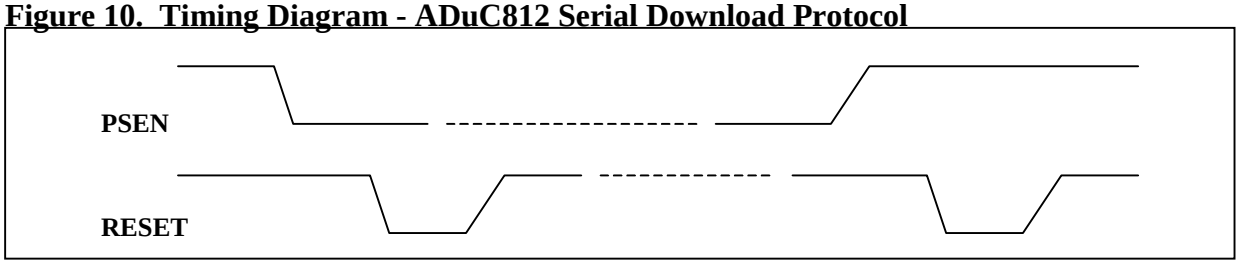
Figure 9. PSD813F5/AduC812 System Memory Map with Paging



ADuC812 FLASH EEPROM SERIAL DOWNLOAD PROTOCOL

The ADuC812 accomplishes In-System-Programming of code through the UART serial port (alternative to JTAG). This download is entered automatically by resetting the chip while PSEN is pulled low through a 1K resistor. Once the handshaking is completed and the entire hex file has been transmitted, control is reverted back to the internal EEPROM Flash by releasing the ground on PSEN and resetting the ADuC812. This is illustrated in Figure 10.

Figure 10. Timing Diagram - ADuC812 Serial Download Protocol

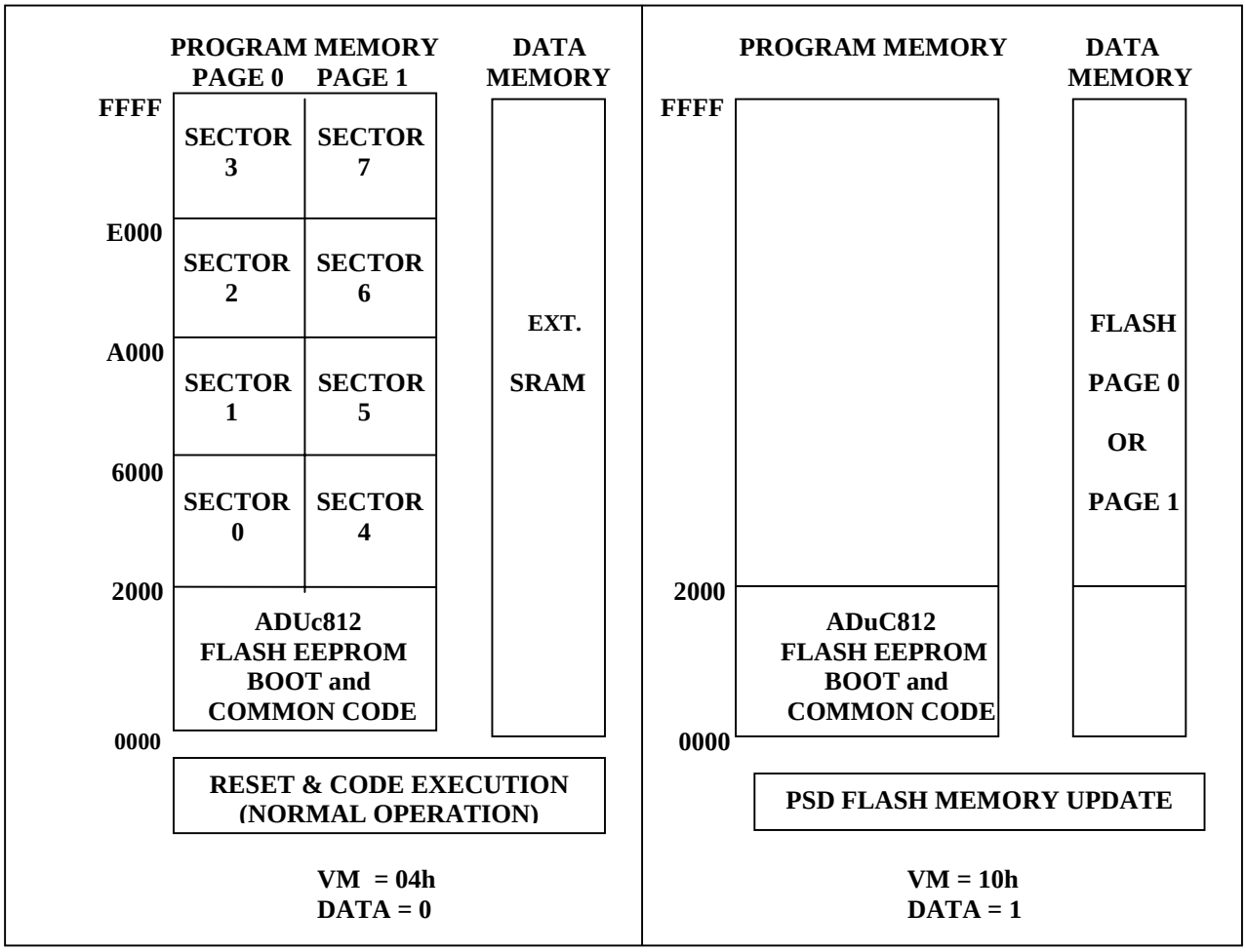


USING THE PSD813F CONCURRENT MEMORY FEATURE

(Please reference Figure 11). Typically, it is very difficult to implement In-System-Programming with a design using the 8031 architecture because of its separate program and data address spaces. The Flash program memory must be temporarily placed in data space while the boot memory is placed in program space to execute minimum code to upgrade the Flash memory. For this example, the boot resides in fixed common EEPROM program memory space at address 0000h-1FFF; The Flash memory is placed in data space and temporarily overlaps the external SRAM data space at address 2000h-FFFFh by setting the Data bit (Page Register Bit 7) to 1, disabling the external chip select signal (SRAM_CS) to the SRAM. The Data bit is

configured in the PSDabel Design Entry screen. Figure 11 shows how program and data memory spaces are temporarily swapped during In-System-Programming of Flash memory.

Figure 11. Memory Swapping - Program and Data Space



The PSD813F5 easily accomplishes memory swapping using its Concurrent Memory features:

1. **Volatile Memory (VM) Register**…is used exclusively for 8031-core designs and controls whether program or data memory space is accessed by PSEN and RD. Figure 12 shows the contents of the VM Register at Reset and when program and data spaces are swapped during the In-System Programming (ISP) operation.

   Reset…(Reference Figure 8). The PSDsoft Bus Configuration screen is used to initially place Flash memory in Program Space after Reset. All memory accesses above 1FFFh will fetch code from the PSD813F5 Flash memory. The Bus Configuration screen also configures the VM Register to be set to 04h at Reset, enabling PSEN to read Flash memory while disabling Read.
   In-System-Programming…Setting the VM Register to10h enables Read and Write to access Flash memory and disables PSEN.

2. **Page Register**…Any of the bits in the 8-bit Page Register which are not used to set up Flash memory pages can be used as control bits for such functions as swapping program and data memory spaces and disabling chip selects to overlapping memory addresses during In-System Programming.

The procedure to update code in the PSD813F5 Flash memory by ISP is straightforward:

1. The Data bit (Page Register Bit 7) is set to disable the SRAM_CS chip enable signal to the external SRAM; this allows Flash memory to temporarily overlap the SRAM in data space.
2. Set the VM Register to 10h; this places Flash program memory in data space by enabling the RD and WR signals from the 8031-core to access the Flash memory. Program code can now be downloaded via the UART Serial Port.
3. When ISP code download is complete, the Data bit is reset and the VM Register is reset back to 04h, restoring code and data back to their respective memory spaces.

Figure 12. VM Register Codes for Memory Swapping Example

Function/ HexCode		Bit 7 PIO_EN	Bit 6 Not Used	Bit 5 Not Used	Bit 4 Fl_Data	Bit 3 EE_Data	Bit 2 FL_Code	Bit 1 EE_Code	Bit 0 SRAM _Code
System Reset (04h)		0	0	0	0 = RD can't access Flash	0	1 = PSEN can access Flash	0	0
Flash Update (10h)		0	0	0	1 = RD can access Flash	0	0 = PSEN can't access Flash	0	0

SUMMARY

The PSD813F has universal applications across all markets. Of no less importance is the rapidly growing Signal Conditioning market. The highly integrated PSD813F, with the inclusion of previously difficult functions to implement – Flash PLDs, Concurrent Memory and ISP – allows the design engineer to add enhanced features to existing and new products with little or no additional cost.

APPENDIX

The Appendix contains the PSDabel listing showing how the PSD813F5 is configured to implement the example in this Application Note.

module ADuC812

```
title 'Minimal Signal Conditioning System using the PSD813F5
and the Analog Devices ADuC812';

"**** By: Don Buccini
"**** Date 3/19/99

"PIN DECLARATIONS

"*********** MCU Bus Interface signal declarations *****************

"**** The following are 8031 bus input signals to the PSD PLDs.

wr       pin;     "CNTL0 Input:(pin 47)- write strobe
rd       pin;     "CNTL1 Input:(pin 50)- read strobe
psen     pin;     "CNTL2 Input:(pin 49)- program store enable
ale      pin;     "PD0 Input:(pin 10)- address latch enable
reset    pin;     "Input:(pin 48)- system reset
a15..a0  pin;     "Input:(pins 46..39,37..30)- demuxed address

"**** In addition to making these declarations, use the PSD
"**** Configuration utility and make these selections:
"****    * 8-bit muxed data bus
"****    * /WR, /RD, /PSEN for control setting
"****    * Active high level for ALE/AS
"****    * Enable CSi if used in application
"****    * Set the VM register.

"******** Port A, B, C, D pin declaration ****************************

"**** Port A pin assignments

"**** Port A is used to latch the low-order address bits.

la0   pin; "I/O (pin 29)- Port A pin pa0
la1   pin; "I/O (pin 28)- Port A pin pa1
la2   pin; "I/O (pin 27)- Port A pin pa2
la3   pin; "I/O (pin 25)- Port A pin pa3
la4   pin; "I/O (pin 24)- Port A pin pa4
la5   pin; "I/O (pin 23)- Port A pin pa5
la6   pin; "I/O (pin 22)- Port A pin pa6
la7   pin; "I/O (pin 21)- Port A pin pa7

"**** Port B pin assignments

a16      pin 7 istype 'reg'; "Port B pin pb0
a17      pin 6 istype 'reg'; "Port B pin pb1
la16     pin 5;              "Port B pin pb2
la17     pin 4;              "Port B pin pb3
SRAM_CS  pin 3;              "Port B pin pb4
pb5      pin;                "I/O (pin 2)- Port B pin pb5
pb6      pin;                "I/O (pin 52)- Port B pin pb6
pb7      pin;                "I/O (pin 51)- Port B pin pb7
```

```
"**** Port C pin assignments

"**** Port C pins can be used for I/O and/ors pecial functions
"**** such as the IEEE 1149.1 JTAG interface.

pc0   pin; "I/O (pin 20)- Port C pin pc0, or JTAG TMS
pc1   pin; "I/O (pin 19)- Port C pin pc1, or JTAG TCK
pc2   pin; "I/O (pin 18)- Port C pin pc2, or VSTBY
pc3   pin; "I/O (pin 17)- Port C pin pc3, or JTAG TSTAT, or Stdby On
pc4   pin; "I/O (pin 14)- Port C pin pc4, or JTAG TERR\, or Rdy/Busy
pc5   pin; "I/O (pin 13)- Port C pin pc5, or JTAG TDI
pc6   pin; "I/O (pin 12)- Port C pin pc6, or JTAG TDO
pc7   pin; "I/O (pin 11)- Port C pin pc7

"**** Port D pin assignments

"pd0 (pin 10) is assigned above as the ALE signal from the
"microcontroller and is not available for use as general I/O.

//clkin  pin;    "Port D pin pd1 (pin 9) can be used as a common
               "clock (clkin) to the PLDs and the power down
               "circuitry. If not used as a common clock, this
               "pin may be used as general I/O.

//pd2    pin;  "Port D pin pd2 (pin 8) can be used as general I/O
               "or the global PSD chip select (CSi). If CSi is
               "desired, do not declare the pin, go to the PSD
               "Configuration utility and enable CSi.

"******** DPLD Outputs and other internal node declaration ********

"**** Main Flash memory segments ************************
fs7..fs0 node;  "This declaration is supported by all PSD8XX devices

"**** PSD Control Registers *****************************
csiop node;   "This declaration is needed for all PSD8XX devices.

"**** Internal PSD Page Register bits
pgr0 node;  "This one-bit page register is used to divide the PSD813F5
            "Flash memory into two pages.

"**** Important. If page register bits are not used as address
"****            extension bits (such as pgr0, pgr1, etc), but are
"****            used to manipulate access of memory, then they should be
"****            declared as individual page bit node numbers to be
"****            compatible with the future mapping feature of PSDsoft.  Here
"****            are the node numbers associated with page register bits.
"****
"****              Page register bit        Internal node number
"****                   pgr7                     node 117
"****                   pgr6                     node 116
"****                   pgr5                     node 115
"****                   pgr4                     node 114
"****                   pgr3                     node 113
"****                   pgr2                     node 112
"****                   pgr1                     node 111
```

```
"****                    pgr0                    node 110

data node 117;    " This page register bit (pgr7) will be used to
                  " disable the external SRAM CS while the PSD813F5
                  " Flash memory is switched to the 80C31 data space
                  "(rd and /wr are used to update Flash memory through
                  "the 80C31 UART.

"**** JTAG port select ********************************

jtagsel node;   "Used to select the JTAG port active with a product term

"*****************************************************************

"DEFINITIONS

X  = .x.;        "Don't care symbol

page = [pgr0]; "You can use up to eight bits for memory paging.
               "Here, only one bit is used to define two memory pages.

address = [a17,a16,a15..a0];  "De-muxed microcontroller address signals

laddr = [a17,a16];

EQUATIONS

"******** DPLD equations ***************************************

"**** Generate active high chip selects for the main Flash segments.
"**** Each segment is 16K bytes for the PSD813FX devices.
"**** All PSD8XX devices support fs7..fs0.

fs0 = (address >= ^h2000) & (address <= ^h5FFF) & (page == 0);
fs1 = (address >= ^h6000) & (address <= ^h9FFF) & (page == 0);
fs2 = (address >= ^hA000) & (address <= ^hDFFF) & (page == 0);
fs3 = (address >= ^hE000) & (address <= ^hFFFF) & (page == 0);
fs4 = (address >= ^h2000) & (address <= ^h5FFF) & (page == 1);
fs5 = (address >= ^h6000) & (address <= ^h9FFF) & (page == 1);
fs6 = (address >= ^hA000) & (address <= ^hDFFF) & (page == 1);
fs7 = (address >= ^hE000) & (address <= ^hFFFF) & (page == 1);

"**** Generate active high chip select for the PSD control registers.
"**** 256 contiguous bytes must be decoded for all PSD8XX devices.

csiop = (address >= ^h20000) & (address <= ^h200FF) & (page == X);

"******** GPLD/ECSPLD Equations ********************************

"**** Active low chip select for an external 128K byte SRAM:
!SRAM_CS = (address >= ^h00000) & (address <= ^h1FFFF) & !data;
laddr.ld = ale;
la16 = a16;
la17 = a17;
end
```

Table 1. Document Revision History

Date	Rev.	Description of Revision
Sep-1999	1.0	Document written (AN060) in the WSI format
03-Jan-2002	1.1	Front page, and back two pages, in ST format, added to the PDF file References to Waferscale, WSI, EasyFLASH and PSDsoft 2000 updated to ST, ST, Flash+PSD and PSDsoft Express

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.