Memory Interface 16

16.1 OVERVIEW

This chapter presents some examples that illustrate basic considerations for interfacing memory to ADSP-2100 Family processors. An example of a multiple paging scheme for data memory is included. Memory-mapped I/O is also demonstrated.

16.2 PROGRAM MEMORY

ADSP-2100 Family processors have a 14-bit program memory address (PMA) bus and a 24-bit program memory data (PMD) bus.

The ADSP-2100A has an additional address pin, the PMDA pin. (See the *ADSP-2100 User's Manual* for details on PMDA). The PMDA signal can be used as a fifteenth PMA bit. The lower half (16K) of program memory stores the instructions, which are 24 bits wide. The upper half (16K) of program memory stores program memory data that is 16 bits wide and left-justified (occupies data bits 23-8). Generally, the program memory data space is used to store constants, such as filter coefficients.

The following program memory example consists of 6 Cypress 8K x 8 static RAMs (CY7C185). They are configured into two banks of 8K x 24-bit wide memory. One bank is shown in Figure 16.1, on the next page. In each bank of RAM, PMD0-7 from the ADSP-2100 Family processor is connected to the D0-D7 pins of one RAM, PMD8-15 to the D0-D7 pins of the second RAM and PMD16-23 to the D0-D7 pins of the third RAM. PMA0-12 from the ADSP-2100 Family processor are connected to the A0-A12 pins of each of the RAMs.

The $\overline{\text{RD}}$ and $\overline{\text{WR}}$ lines are connected directly to the memories while the $\overline{\text{PMS}}$ line is used by the decoder.

16 Memory Interface

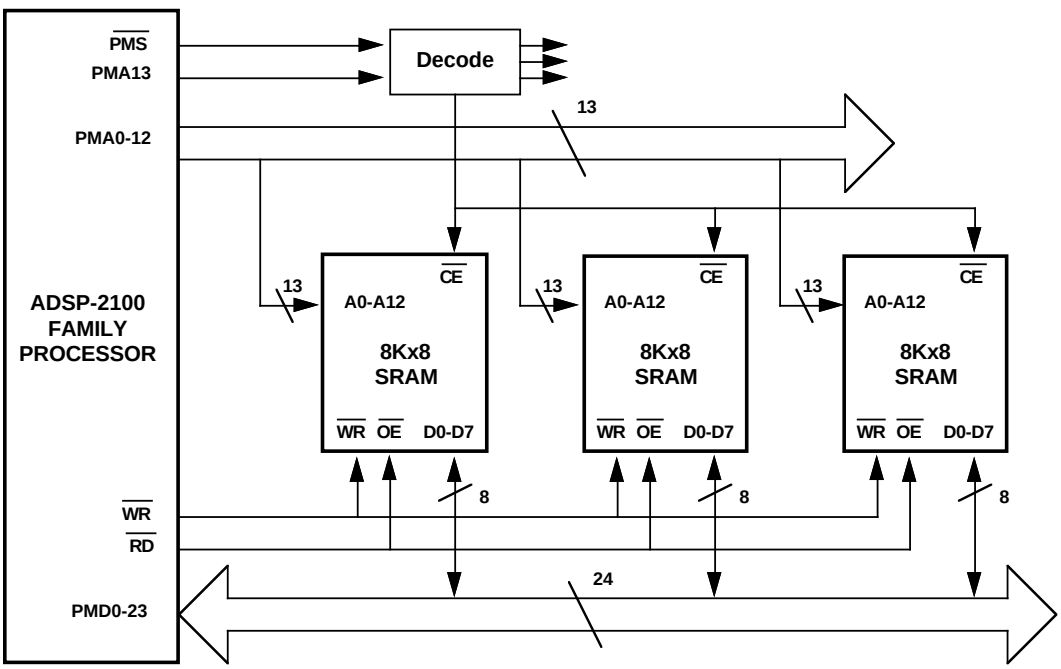

Figure 16.1 Program Memory Interface (One Bank)

16.2.1 Program Memory Bank Enables

Only one bank of program memory RAM is enabled at a time. Bank enables are generated by decoding PMA13 and PMS.

On the ADSP-2100A only, the PMDA signal acts as a fifteenth address bit. When PMDA is low, it selects the lower half of program memory (H#0000-3FFF, instruction space); when it is high, it selects the upper half of program memory (H#4000-7FFF, data space). Each half of program memory consists of two banks. PMA13 enables either the upper or lower half of memory.

16.3 DATA MEMORY

Data memory is generally used to store acquired data. The ADSP-2100 Family processors have a 16-bit wide data memory data (DMD) bus and 14-bit wide data memory address (DMA) bus. The 16-bit wide data bus allows direct interface to 16-bit A/D and D/A converters. The 14-bit wide address bus permits addressing of up to 16K of data memory.

Data memory paging can be used to allow ADSP-2100 Family processors to access several 16K x 16 data memory pages. The pages are in parallel, that is, they each use the same address space. One data memory page is enabled at a time via a data memory page select register, which is mapped into program memory data space.

Each page of data memory in the example uses four 8K x 8 Cypress static RAMs (CY7C185), configured in two 8K x 16 banks of RAM as shown in Figure 16.2. In each bank, DMA0-12 from the ADSP-2100 Family processor are connected to the A0-A12 pins on both chips. DMD0-7 are connected to the D0-D7 pins on one RAM chip, and DMD8-15 are connected to the D0-D7 pins of the second RAM chip.

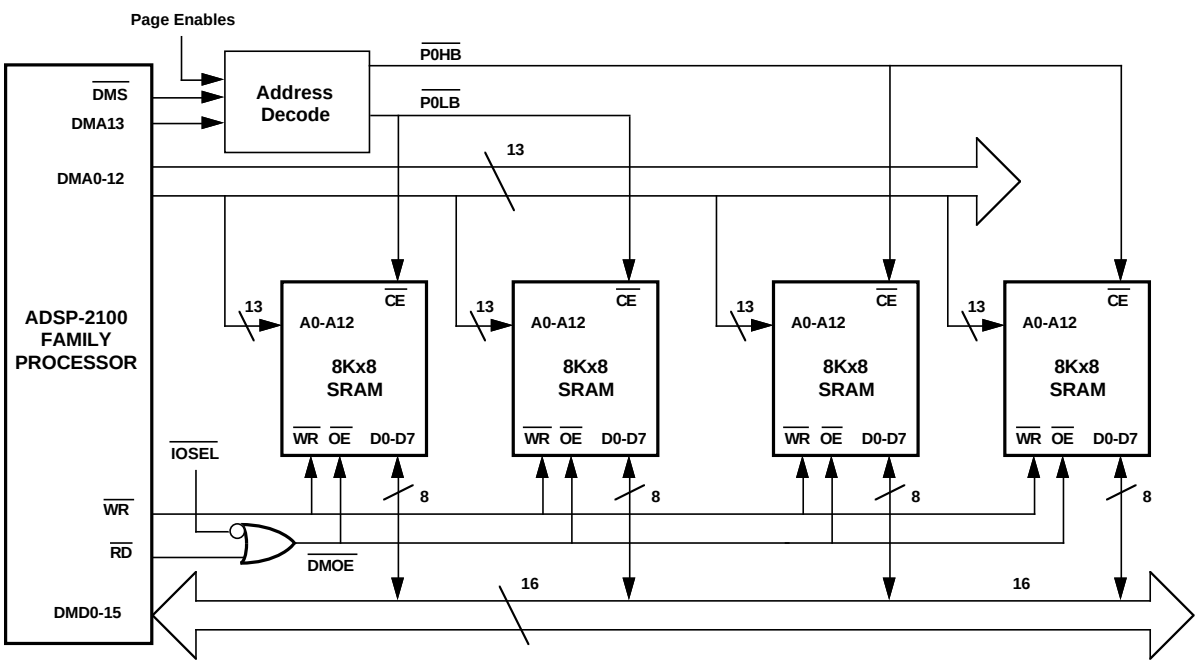

Figure 16.2 Data Memory Interface (One Page)

A bank enable signal, as described in "Data Memory Bank Enables," is connected to the chip enable (CE) pin of each RAM. The WR signal is connected to the WR pins of all RAM chips. The Data Memory Out Enable (DMOE) signal is connected to the OE pin on all RAMs. DMOE is generated by ORing RD and the inverse of the I/O Select signal (IOSEL), to prevent bus contention during I/O accesses. (See "I/O Memory Mapping" for more information on IOSEL.) When a bank enable signal goes low, both RAM chips in the corresponding bank are enabled.

16 Memory Interface

16.3.1 Data Memory Page Enables

The data memory page is selected through a reserved location in program memory. For example, the software can select a data memory page by writing the page number in binary to program memory location H#3FFE. This value appears on PMD8-23, and the LSBs (bits 8, 9 and 10) are decoded in hardware to generate the data memory page enable (see Table 16.2 for values).

	Data Memory Page Enable (Active Low)							
PMD10-8	DMPG0	DMPG1	DMPG2	DMPG3	DMPG4	DMPG5	DMPG6	DMPG7
000	0	1	1	1	1	1	1	1
001	1	0	1	1	1	1	1	1
010	1	1	0	1	1	1	1	1
011	1	1	1	0	1	1	1	1
100	1	1	1	1	0	1	1	1
101	1	1	1	1	1	0	1	1
110	1	1	1	1	1	1	0	1
111	1	1	1	1	1	1	1	0

Table 16.2 Data Memory Page Enables

PMD8-10 from the ADSP-2100 Family processor are input to a 3-to-8 decoder with latchable outputs, as shown in Figure 16.3. The 3-to-8 decoder is always enabled; its latched outputs are only updated on the rising edge of its Latch Enable (LE) input. The outputs of the 3-to-8 decoder are the data memory page enables (DMPGx, where x is the page number).

The LE input for the 3-to-8 decoder is generated as follows. Location H#3FFE is decoded using two 8-bit bus comparators (74AHCT521). The Q inputs to the bus comparators are hardwired to H#3FFE. The P inputs are connected to the program memory address bus. The outputs of the comparators, which go low when the Q and P inputs are the same, are ORed to create the data memory page select enable signal (DMPSE). DMPSE is NORed with the PMWR signal to generate the LE input to the 3-to-8 decoder.

When PMWR goes high, the decoded value of PMD8-10 is latched into the output of the 3-to-8 decoder, selecting the data memory page. A data memory page remains selected until another is selected or the system is reset. At power-up and after reset, the data memory page defaults to page zero.

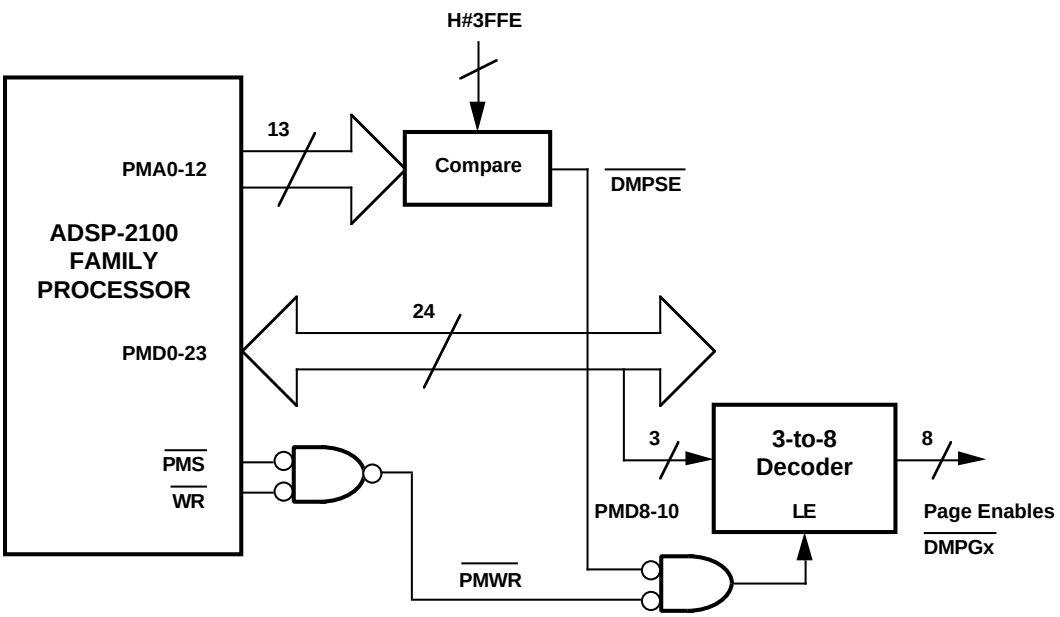

Figure 16.3 Data Memory Page Enables

16.3.2 Data Memory Bank Enables

The bank enables are generated from the most significant DMA bit, DMA13, and the page enable signals, DMPG0 and DMPG1, as shown in Table 16.3. DMA13 and an inverted version of DMA13 are each ORed with the data memory page enable signal to select one of the two banks in the enabled page. If a data memory page is not enabled, no RAM banks can be enabled.

			Data Memory Bank Enables			
DMPG0	DMPG1	DMA13	P0LB	P0HB	P1LB	P1HB
1	0	0	0	1	1	1
1	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Table 16.3 Data Memory Bank Enable Decoding

16 Memory Interface

16.4 I/O CONFIGURATION

This example maps I/O accesses into data memory; I/O devices can occupy certain memory space locations in place of memory. The I/O devices are written to and read from as if they were memory. The memory locations that map to I/O are called I/O locations.

I/O devices can be memory-mapped into program memory data space as well as data memory. A key difference to consider is:

- The program memory data bus is 24 bits wide, to accommodate instructions. In program memory data accesses, PMD7-0 are read from or written to the PX register.

See the *ADSP-2100 User's Manual* for further information.

16.4.1 I/O Memory Mapping

This example maps 16 I/O locations into the last 16 locations of each data memory page, H#3FE0-3FFF. Either I/O or memory can be used at these locations; a hardware switch selects I/O or memory. When I/O is enabled and a data memory access to an I/O location is executed, the IOSEL signal is generated. The IOSEL signal enables the I/O device to drive the data memory data bus and is also used to disable memory, preventing an I/O device and memory from driving the data bus at the same time. IOSEL should be qualified with a data memory read or data memory write to ensure a proper I/O access. When I/O is not enabled, data memory accesses to the I/O locations are executed as memory accesses and no IOSEL is generated (IOSEL remains high).

There are eight I/O switches, one for each of the eight pages of data memory that the example system can use. The I/O switch selects either memory or I/O for the last 16 locations of its data memory page. In this way, I/O can be enabled for any combination of data memory pages.

The IOSEL signal is generated when DMA5-13 are high (for addresses H#3FE0-3FFF). DMA5-13 are connected to a 12-input NAND gate (74S134) as shown in Figure 16.4. (Unused inputs are pulled high.) The output of the NAND gate is the IOSEL signal. Any address equal to or greater than H#3FE0 generates the IOSEL signal.

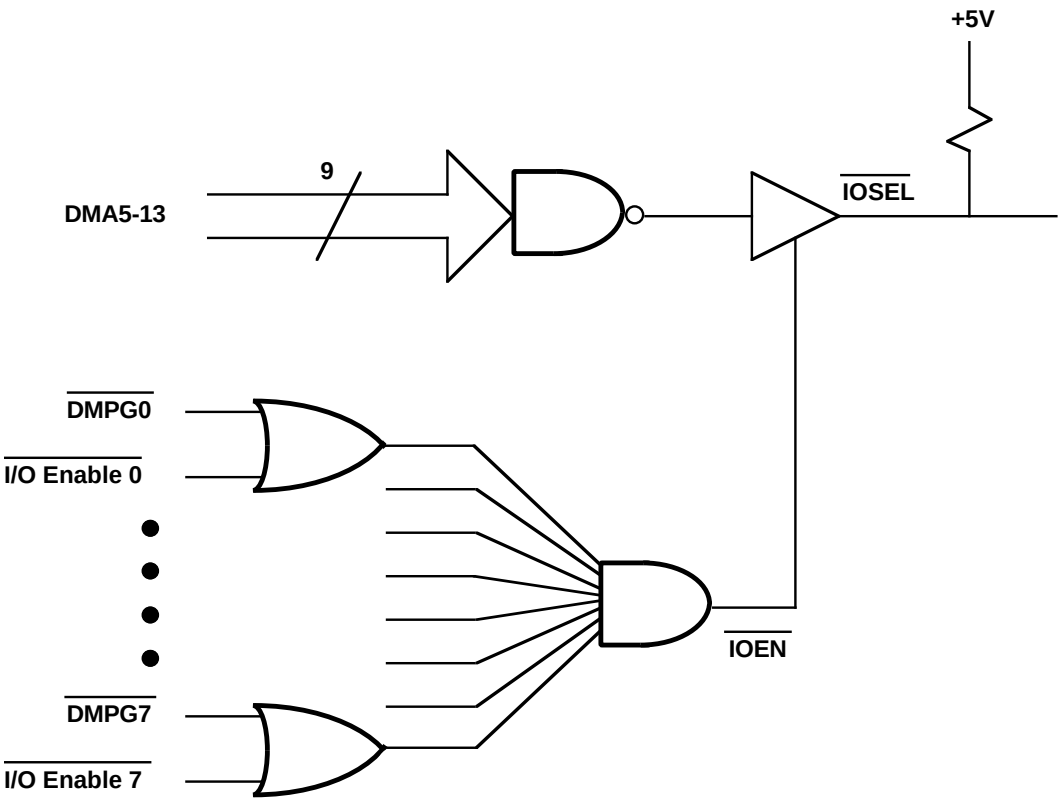

Figure 16.4 Generating I/O Select Signal

The 74S134 that generates IOSEL has a tristate output which is controlled by the IOEN signal. Each of the eight (active low) data memory page enable signals is ORed with its data memory page I/O enable switch (active low). The eight signals are connected to an 8-input NAND gate. The inverted output of the 8-input NAND gate (IOEN) is active only if a data memory page enable and the corresponding I/O enable switch are both active. Because only one data memory page is enabled at a time, only one input to the 8-input NAND gate can go low at a time. This allows you to set the I/O switches in any combination; IOSEL is not active until a page with I/O enabled is selected.

16.4.2 I/O Switches

I/O location H#3FFF is used by the example system to read the settings of four SPDT switches. The four general-purpose switches set the values of bits 0-3 at that location. An executing program can read the bits to determine the switch settings and then branch accordingly.

The values of the SPDT switches are output through a tristatable buffer (74F244A) to the 4 LSBs of the DMD bus. The buffer is enabled by decoding address H#3FFF as shown in Figure 16.5. The IOSEL signal is generated when DMA5-13 are high. Location H#3FFF (DMA0-13 high) can therefore be decoded from an inverted IOSEL signal and DMA0-4. These signals are input to an 8-input NAND gate (74ALS30). (The unused inputs are pulled high.) When the data memory address is to H#3FFF and I/O is enabled, the output of the NAND gate, EN, is asserted low.

The EN signal and the DMRD signal are ORed and input to the enable pin on the 74F244A. When a read to I/O location H#3FFF is executed, first EN goes low, then DMRD goes low, enabling the 74F244A. On the rising edge of DMRD, the output of the 74F244A is clocked into the ADSP-2100 Family processor via the DMD bus.

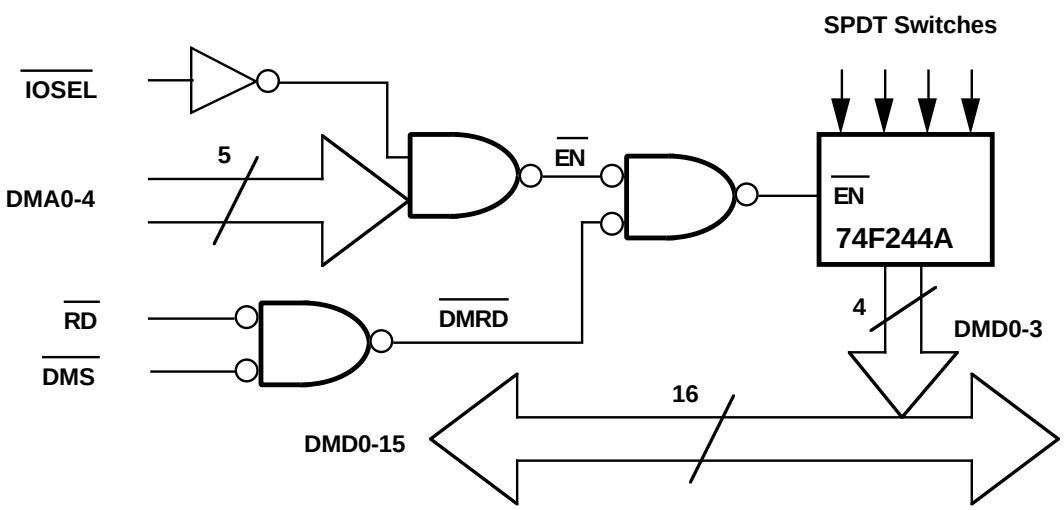

Figure 16.5 Decoding for I/O Switches