Memory Interface 10

10.1 OVERVIEW

The ADSP-2100 family has a modified Harvard architecture in which data memory stores data and program memory stores both instructions and data. Each processor contains on-chip RAM and/or ROM, so that a portion of the program memory space and a portion of the data memory space reside on-chip. Each processor (except the ADSP-2181) also has a boot memory space in addition to the data and program spaces. The ADSP-2181 has a byte memory space instead of the boot memory space. The boot memory space and byte memory space can be used to load on-chip program memory with code from an external EPROM at reset.

In each ADSP-2100 family device, memory is connected with the internal functional units by four on-chip buses: the data memory address bus (DMA), data memory data bus (DMD), program memory address bus (PMA), and program memory data bus (PMD). The internal PMA bus and DMA bus are multiplexed into a single address bus which is extended off-chip. Likewise, the internal PMD bus and DMD bus are multiplexed into a single external data bus. The sixteen MSBs of the external data bus are used as the DMD bus: external bus lines D_{23-8} are used for DMD_{15-0}.

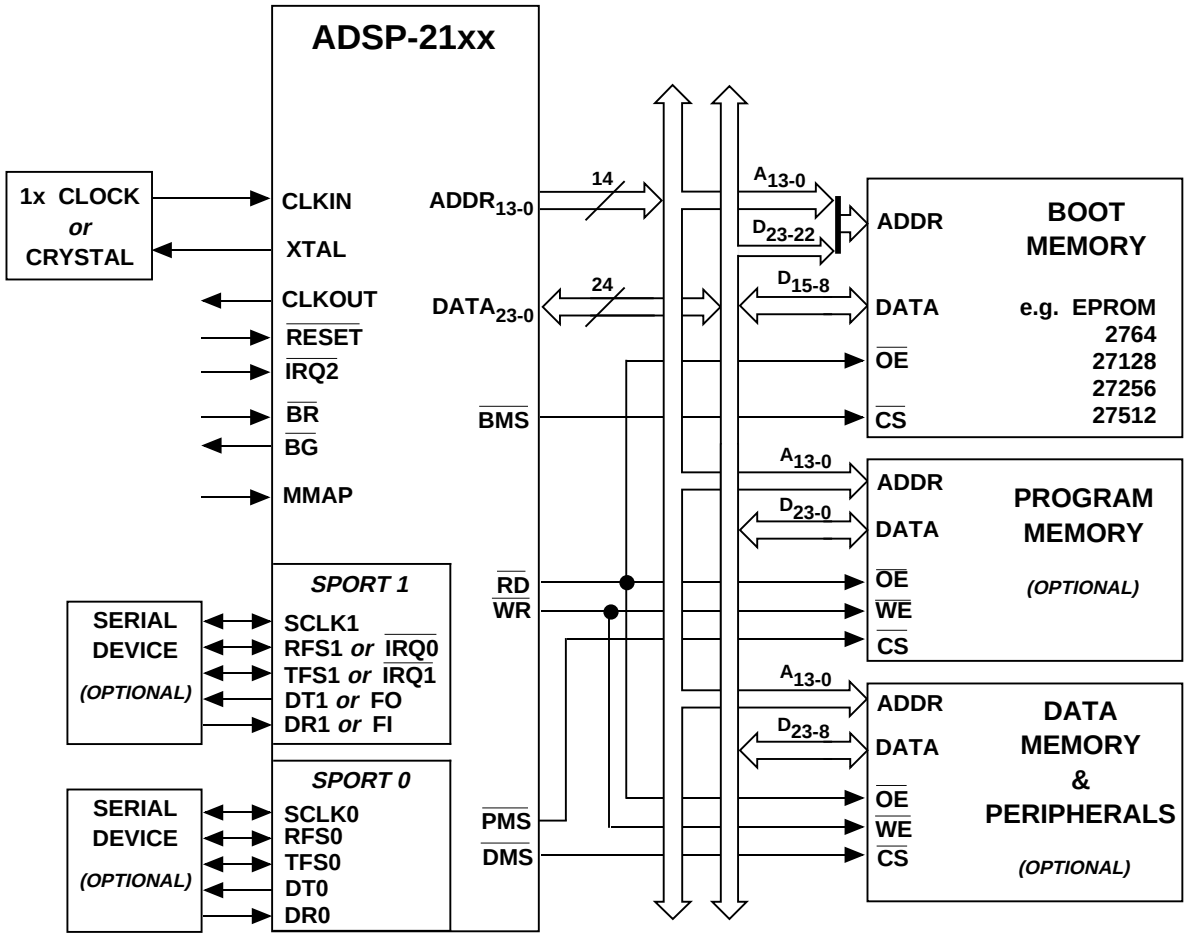
There are three separate memory spaces: data memory, program memory and boot (or byte) memory. The $\overline{PMS}$, $\overline{DMS}$, and $\overline{BMS}$ signals indicate which memory space is being accessed. Because the program memory and data memory buses are multiplexed off-chip, if more than one external transfer must be made in the same instruction there will be an overhead cycle required. There is no overhead if just one off-chip access (with no wait states) occurs in any instruction. Figure 10.1 shows the external memory buses and control signals (for all ADSP-21xx processors except the ADSP-2181).

All external memories may have automatic wait state generation associated with them. The number of wait states—each equal to one instruction cycle—is programmable.

10 Memory Interface

This chapter includes example timing diagrams for the memory interfaces of the ADSP-21xx processors. For each bus transaction, only the sequence of events is described; you must consult the processor data sheets for actual timing parameters. All timing diagrams use CLKOUT as a reference, which indicates the instruction execution rate.

The memory interfaces of the ADSP-2181 are described separately in the second half this chapter.



NOTES
1. Applies to all ADSP-21xx processors except ADSP-2181.
2. ADSP-2171 and ADSP-21msp58/59 use a 1/2x CLKIN signal.
3. Unused data bus lines may be left floating.
4. The two MSBs of the data bus (D23-22) are used to supply the two MSBs of the boot memory EPROM address. This is only required for the 27256 and 27512.

Figure 10.1 ADSP-21xx System With External Memory

10.2 PROGRAM MEMORY INTERFACE

This section describes the program memory interface of all ADSP-21xx processors except the ADSP-2181.

The processors address 16K of 24-bit wide program memory, up to 2K on-chip and the remainder external, using the control lines shown in Figure 10.1. The processors supply a 14-bit address on the program memory address bus (PMA) which is driven off-chip on the address bus in the case of external program memory accesses. Instructions or data are transferred across the 24-bit program memory data (PMD) bus which is also multiplexed off-chip. For a dual off-chip data fetch, the data from program memory is read first, then the data memory data. A program memory select pin, $\overline{\text{PMS}}$, indicates that the address bus is being driven with a program memory address and memory can be selected.

Two control lines indicate the direction of the transfer. Memory read ($\overline{\text{RD}}$) is active low signaling a read and memory write ($\overline{\text{WR}}$) is active low for a write operation. Typically, you would connect $\overline{\text{PMS}}$ to $\overline{\text{CE}}$ (Chip Enable), $\overline{\text{RD}}$ to $\overline{\text{OE}}$ (Output Enable) and $\overline{\text{WR}}$ to $\overline{\text{WE}}$ (Write Enable) of your memory.

10.2.1 External Program Memory Read / Write

On-chip memory accesses do not drive any external signals. $\overline{\text{PMS}}$, $\overline{\text{DMS}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ remain high (deasserted); the address and data buses are tristated. Off-chip program memory access happens in this sequence:

1. The processor places the address on the PMA bus, which is multiplexed off-chip, and $\overline{\text{PMS}}$ is asserted.

2. $\overline{\text{RD}}$ or $\overline{\text{WR}}$ is asserted.

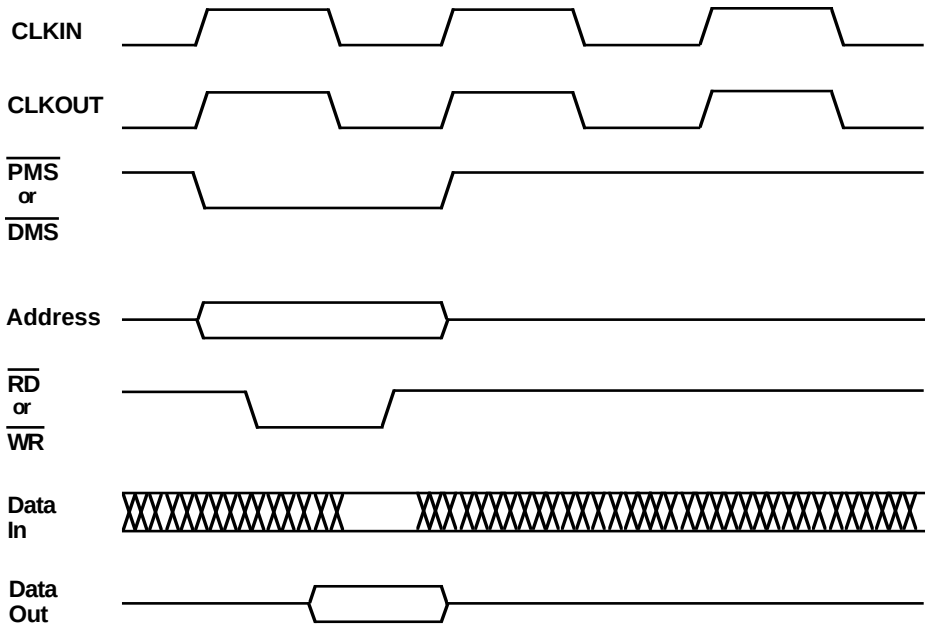
3. Within a specified time, data is placed on the data bus, multiplexed to the internal PMD bus.

4. The data is read or written and $\overline{\text{RD}}$ (or $\overline{\text{WR}}$) is deasserted.

5. $\overline{\text{PMS}}$ is deasserted.

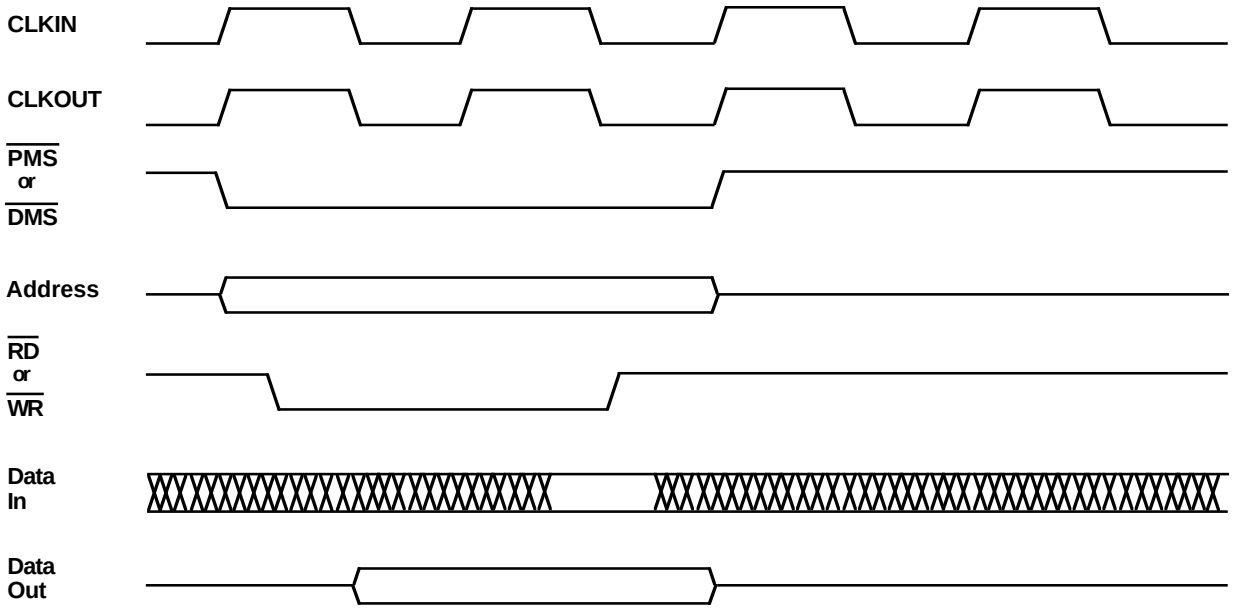
The basic read and write cycles are illustrated in Figure 10.2 on the next page. Figure 10.2A shows zero wait states and 10.2B shows the effect of one wait state.

10 Memory Interface



External Program/Data Memory Read/Write
PWAIT=0, DWAIT=0 (no wait states added)

Figure 10.2A Memory Read And Write, No Wait States



External Program/Data Memory Read/Write
PWAIT=1, DWAIT=1 (one wait state added)

Figure 10.2B Memory Read And Write, One Wait State

The program memory interface can generate 0 to 7 wait states for external memory devices. The program memory wait state field (PWAIT) in the system control register is shown in Figure 10.3. PWAIT defaults (after RESET) to seven wait states for program memory accesses.

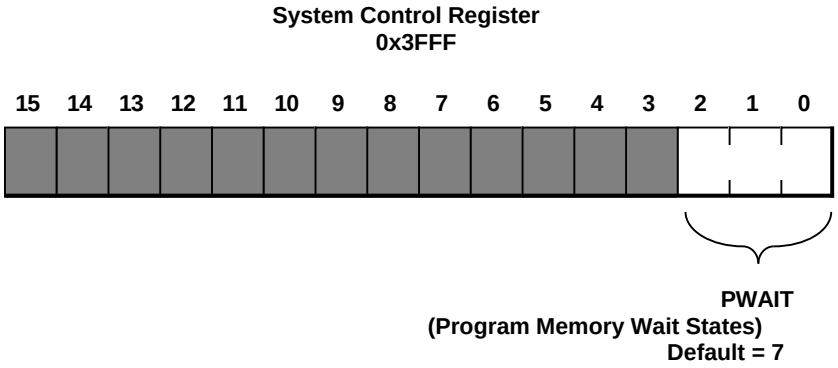


Figure 10.3 Program Memory Wait State Field In System Control Register

10.2.2 Program Memory Maps

For all RAM-based processors except the ADSP-2181, the program memory space is mapped in one of two configurations depending on the state of the MMAP pin. Figure 10.4 shows these configurations for the processors with 2K internal program memory (ADSP-2101, ADSP-2111, ADSP-2171, ADSP-21msp58), and Figure 10.5 shows the same information for the processors with 1K internal program memory (ADSP-2105, ADSP-2115).

When MMAP=0, internal RAM occupies 2K words beginning at address 0x0000. In this configuration, the boot loading sequence is automatically initiated when RESET is released (as described in "Boot Memory Interface").

When MMAP=1, words of external program memory begin at address 0x0000 and internal RAM is located in the upper 2K words, beginning at address 0x3800. In this configuration, program memory is not loaded although it can be written to and read from under program control.

The program memory space can hold instructions and data intermixed in any combination. The ADSP-21xx linker determines where to place relocatable code and data segments. You may specify absolute address placement for any module or data structure, including the code for the restart and interrupt vector locations. The restart vector is at program memory address 0x0000. The interrupt vector locations are given in Chapter 3 and in Appendix D.

10 Memory Interface

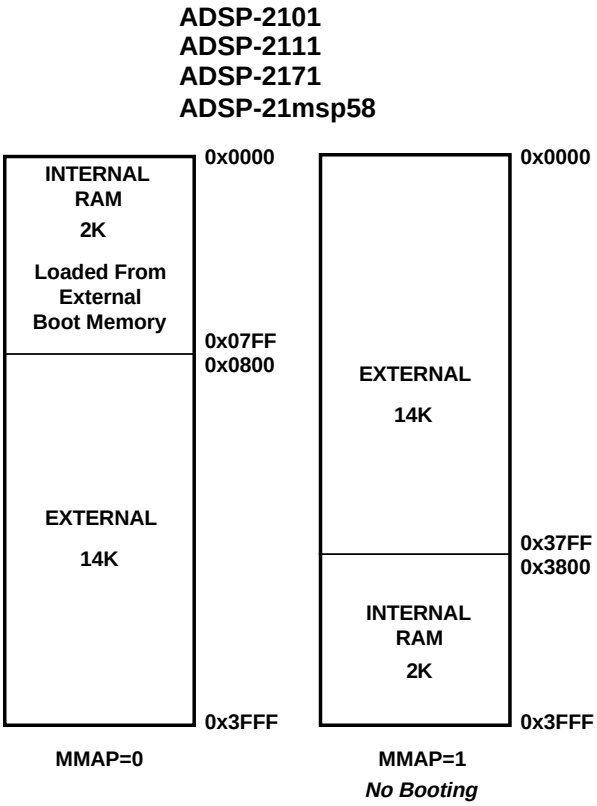


Figure 10.4 Program Memory Maps (2K internal RAM)

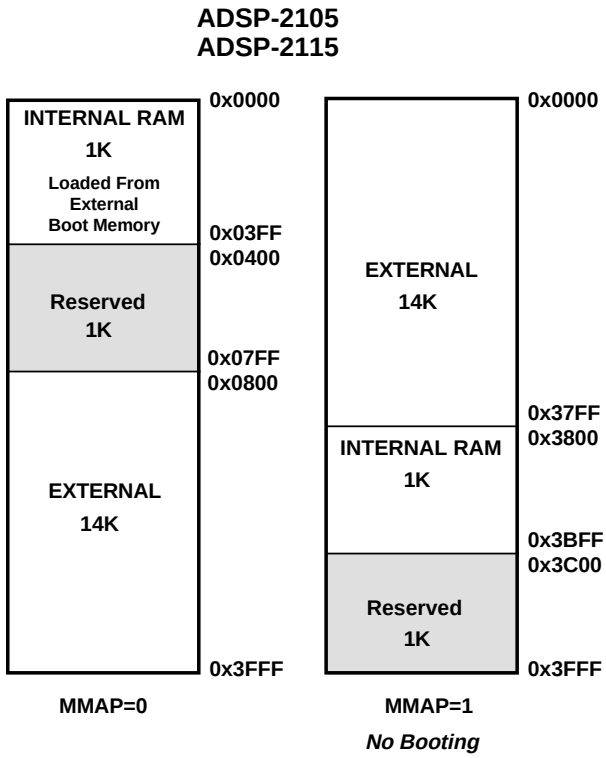


Figure 10.5 Program Memory Maps (1K internal RAM)

Internal program memory RAM is fast enough to supply an instruction and data in the same cycle, eliminating the need for cache memory. Consequently, if the processor is operating entirely from on-chip memory, it can fetch two operands and the next instruction on every cycle. It can also fetch any one of these three from external memory with no performance penalty.

10.2.3 ROM Program Memory Maps

The ADSP-2172 and ADSP-21msp59 processors contain mask-programmable ROM on-chip. The program memory maps for these processors are shown in Figures 10.6 and 10.7. The ADSP-2172 contains 8K of ROM and the ADSP-21msp59 contains 4K.

On the ADSP-2172 and ADSP-21msp59, the ROM is enabled by setting the ROMENABLE bit in the Data Memory Wait State control register (at address DM[0x3FFE]). When the ROMENABLE bit is set to 1, addressing program memory in the ROM range will access the on-chip ROM. When ROMENABLE is set to 0, addressing program memory in this range will access external program memory. The ROMENABLE bit is initialized to 0 after reset unless MMAP *and* BMODE=1.

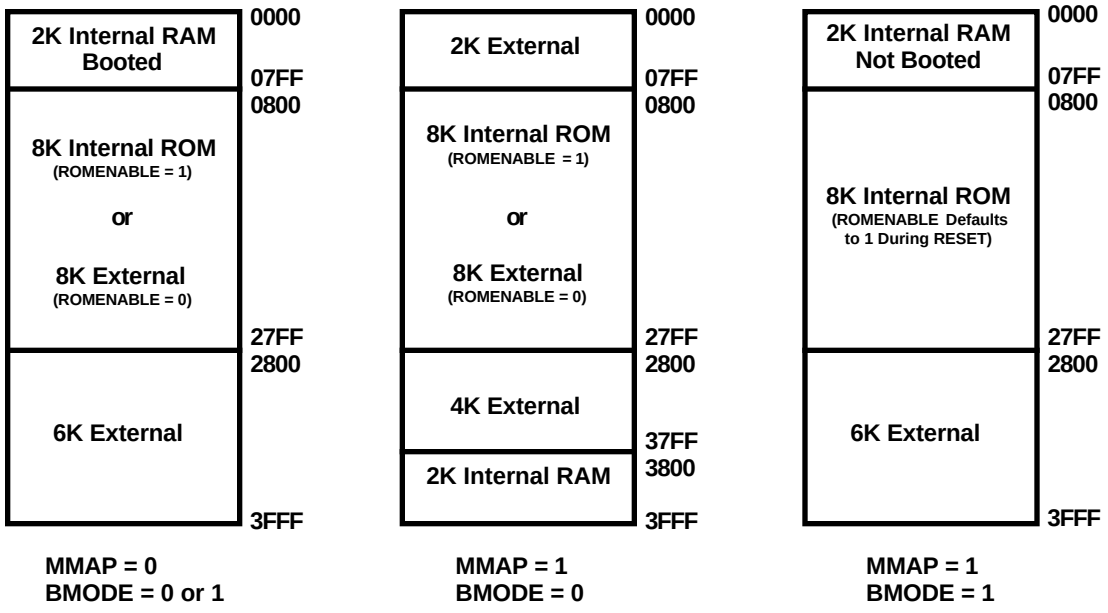


Figure 10.6 ADSP-2172 Program Memory Map

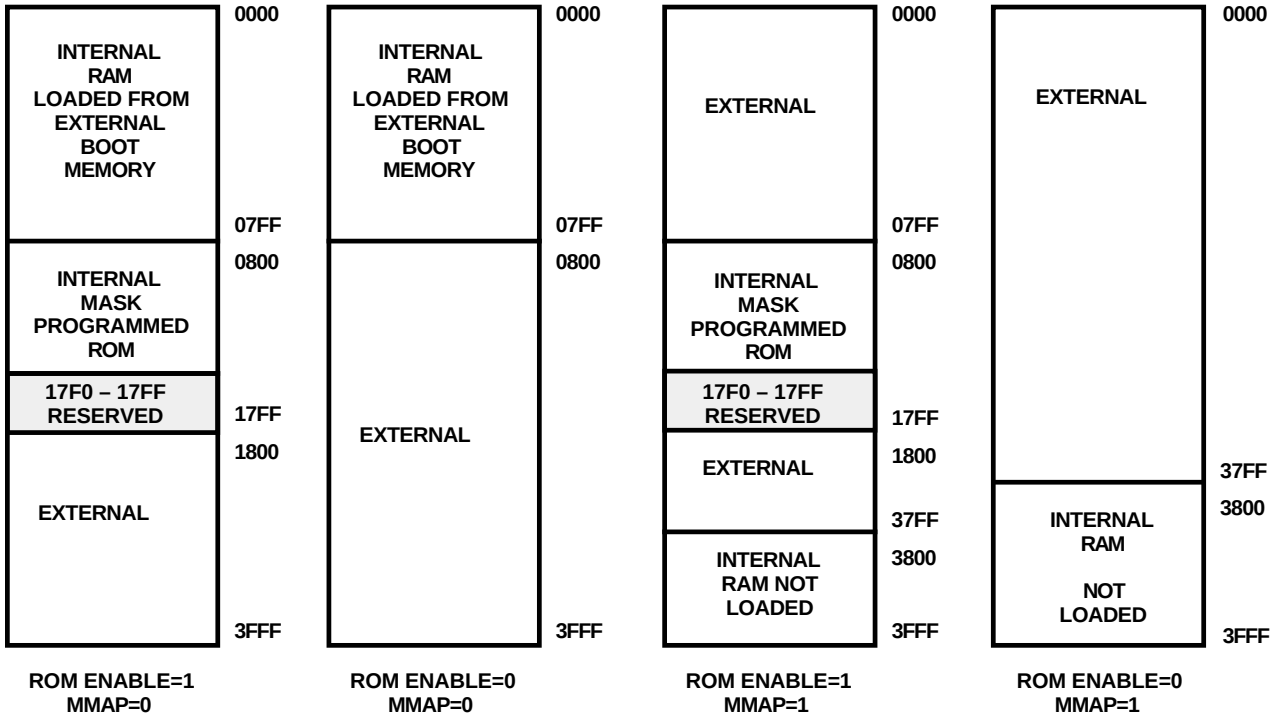


Figure 10.7 ADSP-21msp59 Program Memory Map

10 Memory Interface

When the MMAP and BMODE pins both are set to 1, the ADSP-2172 (or ADSP-21msp59) will operate in *standalone ROM execution* mode. When MMAP=1 and BMODE=1, the ROM is automatically enabled and execution begins from program memory location 0x0800 at the start of ROM. This lets an embedded design operate without external memory components. To operate in this mode, the ROM-coded program must copy an interrupt vector table to the appropriate locations in program memory RAM. In this mode, the ROMENABLE bit defaults to 1 during reset. Table 10.1 summarizes the booting and startup execution modes for the ADSP-2172 and ADSP-21msp59.

	BMODE = 0	*BMODE = 1*
MMAP = 0	Boot from EPROM, then execution starts at internal RAM location 0x0000	Boot from HIP, then execution starts at internal RAM location 0x0000
MMAP = 1	No booting, execution starts at external memory location 0x0000	Standalone mode, execution starts at internal ROM location 0x0800

Table 10.1 Booting Mode for ADSP-2172, ADSP-21msp59

The ADSP-216x processors are memory-variant versions of the ADSP-2101 and ADSP-2103 that contain factory-programmed on-chip ROM program memory. The ADSP-2161, ADSP-2163, and ADSP-2165 are 5.0V supply processors based on the ADSP-2101. The ADSP-2162, ADSP-2164, and ADSP-2166 are 3.3V supply processors based on the ADSP-2103. These devices offer different amounts of on-chip memory for program and data storage, as shown in Table 10.2.

Feature	*2161*	*2162*	*2163*	*2164*	*2165*	*2166*
Data Memory (RAM)	½K	½K	½K	½K	4K	4K
Program Memory (ROM)	8K	8K	4K	4K	12K	12K
Program Memory (RAM)	–	–	–	–	1K	1K

Table 10.2 ADSP-216x ROM-Programmed Processors

Figures 10.8, 10.9, and 10.10 show the program memory maps for the ADSP-2161/62, ADSP-2163/64, and ADSP-2165/66, respectively.

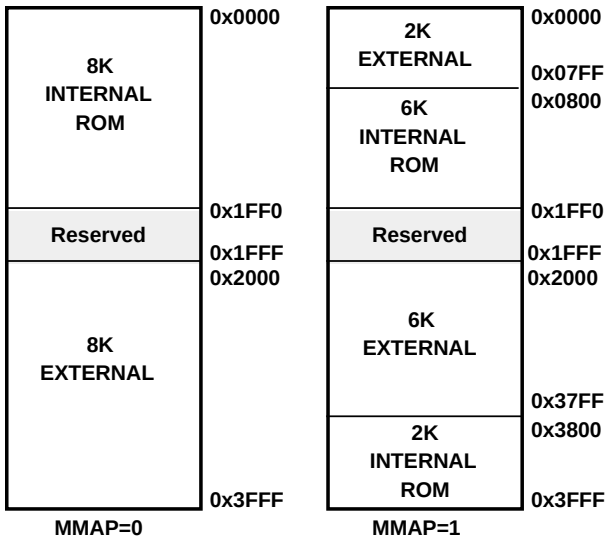


Figure 10.8 ADSP-2161/62 Program Memory Maps

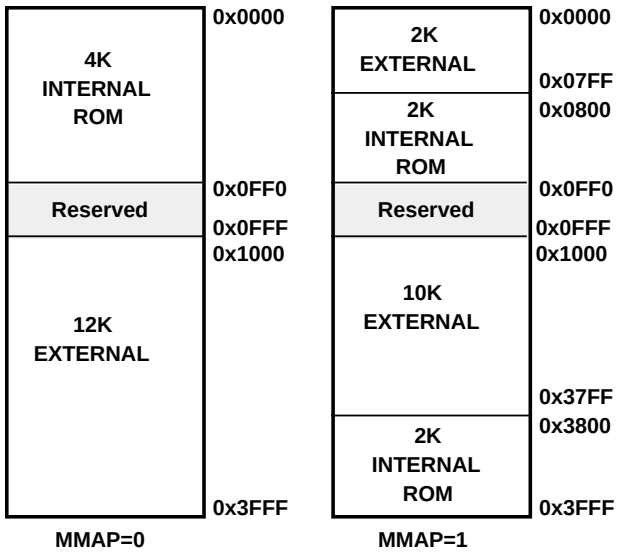


Figure 10.9 ADSP-2163/64 Program Memory Maps

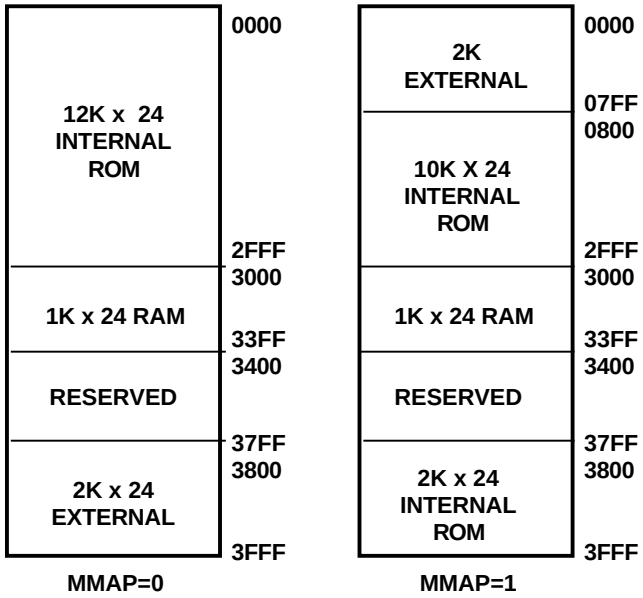


Figure 10.10 ADSP-2165/66 Program Memory Maps

10 Memory Interface

10.3 DATA MEMORY INTERFACE

This section describes the data memory interface of all ADSP-21xx processors except the ADSP-2181.

The processors supply a 14-bit address on the data memory address bus (DMA) which is multiplexed off-chip. Data is transferred across the upper 16 bits of the 24-bit memory data bus, which is also multiplexed off-chip. A data memory select pin, $\overline{\text{DMS}}$, indicates that the address bus is being driven with a data memory address and memory can be selected.

Two control lines indicate the direction of the transfer. Memory read ($\overline{\text{RD}}$) is active low signaling a read and memory write ($\overline{\text{WR}}$) is active low for a write operation. Typically, you would connect $\overline{\text{DMS}}$ to $\overline{\text{CE}}$ (Chip Enable), $\overline{\text{RD}}$ to $\overline{\text{OE}}$ (Output Enable) and $\overline{\text{WR}}$ to $\overline{\text{WE}}$ (Write Enable) of your memory.

10.3.1 External Data Memory Read/Write

Internal data memory accesses are transparent to the external memory interface. Only off-chip accesses drive the memory interface. Off-chip data memory accesses follow the same sequence as off-chip program memory accesses, namely:

1. The processor places the address on the DMA bus, which is multiplexed off-chip, and $\overline{\text{DMS}}$ is asserted.

2. $\overline{\text{RD}}$ or $\overline{\text{WR}}$ is asserted.

3. Within a specified time, data is placed on the data bus, multiplexed to the internal DMD bus.

4. The data is read or written and $\overline{\text{RD}}$ (or $\overline{\text{WR}}$) is deasserted.

5. $\overline{\text{DMS}}$ is deasserted.

The basic read and write cycles are illustrated in Figure 10.2.

For a dual off-chip data fetch, the data from program memory is read first, then the data memory data.

10.3.2 Data Memory Maps

The processors can address a total of 16K words of 16-bit data memory. On-chip data memory is 1K in size and starts at address 0x3800 on the ADSP-2101 and ADSP-2111. On-chip data memory is 512 locations in size on the ADSP-2105 and ADSP-2115, again starting at address 0x3800. On-chip data memory is 2K in size on the ADSP-2171 and ADSP-21msp58/59, beginning at address 0x3000.

The processors' control and status registers are mapped into the top 1K of data memory, addresses 0x3C00-0x3FFF. The rest of the top 1K is reserved. External data memory is available for additional data storage. Figures 10.11, 10.12, and 10.13 show the data memory maps for each ADSP-21xx processor.

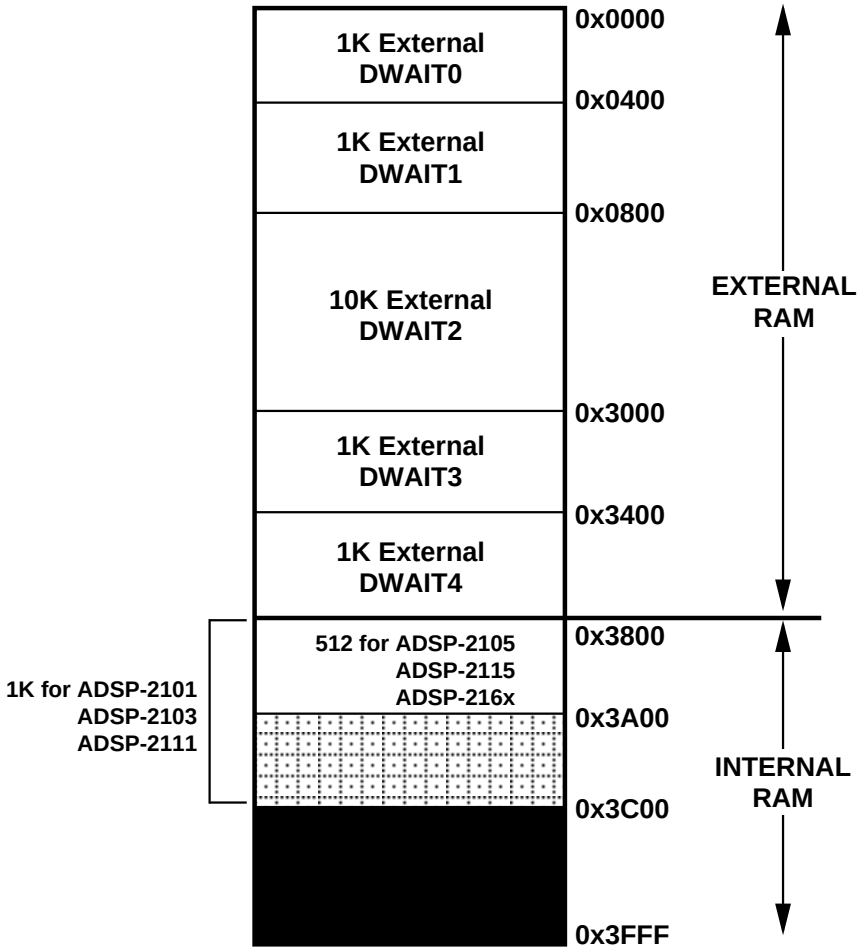


Figure 10.11 Data Memory Map (ADSP-2101, ADSP-2111, ADSP-2105, ADSP-2115, ADSP-2161/62/63/64)

10 Memory Interface

As shown in Figure 10.11, the ADSP-2101, ADSP-2111, ADSP-2105, ADSP-2115, and ADSP-2161/62/63/64 processors have five external wait state zones (DWAIT0–DWAIT4). Each of the five zones of external data memory has its own programmable number of wait states. Wait states are extra cycles that the processor either waits before latching data (on a read) or drives the data (on a write). This means that one zone of memory could be used for working with memory-mapped peripherals of one speed while another zone was used with faster or slower peripherals. Similarly, slower and faster memories can be used for different purposes, as long as they are located in different zones of the data memory map.

As shown in Figures 10.12 and 10.13, the ADSP-2171, ADSP-21msp58/59, and ADSP-2165/66 processors each have three wait state zones for external data memory.

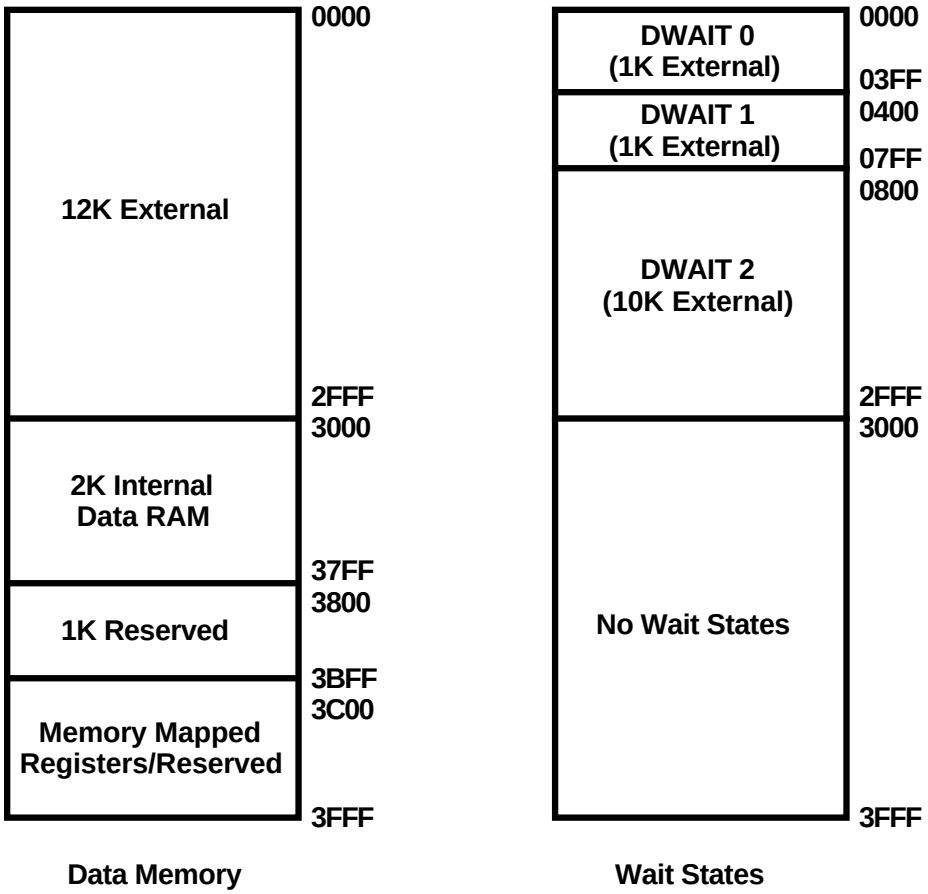


Figure 10.12 Data Memory Map (ADSP-2171, ADSP-21msp58/59)

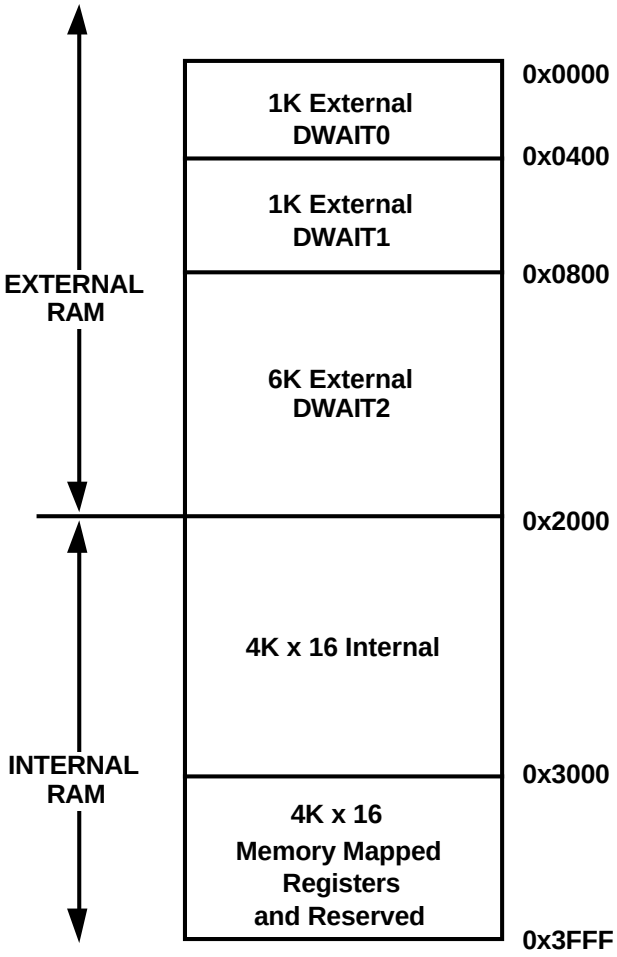


Figure 10.13 Data Memory Map (ADSP-2165/66)

The Data Memory Waitstate control register has a separate field for each zone of external memory. Each 3-bit field specifies the number (0-7) of wait states for the corresponding zone of memory; all zones default to 7 wait states after $\overline{\text{RESET}}$. Figure 10.14 shows this control register for the ADSP-2101, ADSP-2111, ADSP-2105, ADSP-2115, and ADSP-2161/62/63/64 processors. Figure 10.15 shows the register for the ADSP-2171/72 and ADSP-21msp58/59 processors; on the ADSP-2172 and ADSP-21msp59, one bit in this register is used to enable or disable the on-chip ROM.

10 Memory Interface

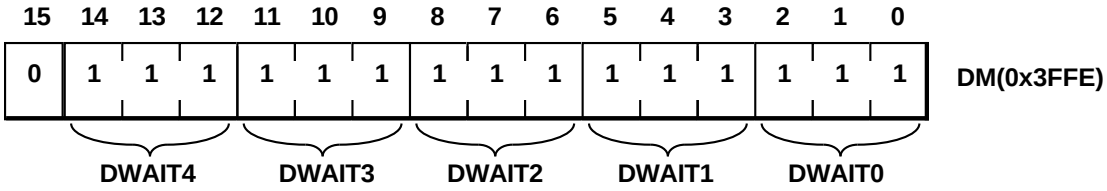


Figure 10.14 Data Memory Waitstate Control Register (ADSP-2101, ADSP-2111, ADSP-2105, ADSP-2115, ADSP-2161/62/63/64)

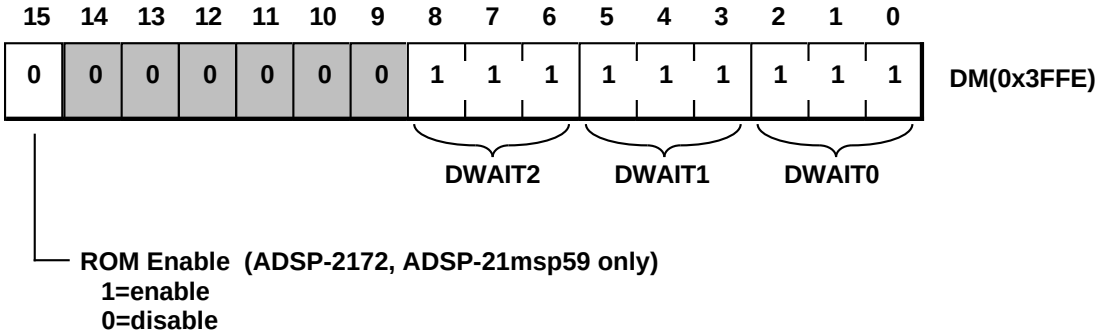


Figure 10.15 Data Memory Waitstate Control Register (ADSP-2171/72, ADSP-21msp58/59)

10.3.3 Memory-Mapped Peripherals

Peripherals requiring parallel communications and other types of devices can be mapped into external data memory. Communication takes the form of reading and writing the memory locations associated with the device. Some A/D and D/A converters require this type of interface. The .PORT directives in the System Builder and Assembler modules of the ADSP-2100 Family Development Software support this mapping.

Communication with a memory-mapped device consists simply of reading and writing the appropriate locations. By matching the access times of the external devices to the wait states specified for their zone of data memory, you can easily interface a variety of devices.

The 16 MSBs of the external data bus (D_{23-8}) are connected to the 16 LSBs of the internal DMD bus, so D_{23-8} should be used for 16-bit peripherals.

10.4 BOOT MEMORY INTERFACE

This section describes the boot memory interface of all ADSP-21xx processors except the ADSP-2181.

The entire internal program memory, or any portion of it, can be loaded from an external source using a boot sequence. To interface with inexpensive EPROM, the processor loads instructions one byte at a time.

Automatic booting at reset depends on the state of the MMAP pin at the time of processor reset. The boot sequence occurs if the MMAP pin is 0. The boot sequence can also be initiated after reset by software.

The ADSP-2111, ADSP-2171, and ADSP-21msp5x processors, which include a Host Interface Port (HIP), can boot using either the memory interface or the HIP (from a host computer). The state of the BMODE pin determines which method is used: the memory interface if BMODE=0, or the HIP if BMODE=1. Booting through the HIP is described in Chapter 7.

$\overline{BR}$ is recognized during the booting sequence. The bus is granted after completion of loading the current byte.

The ADSP-216x contain on-chip program memory ROM; on these devices, no booting occurs.

10.4.1 Boot Pages

Boot memory is organized into eight pages, each of which can be 8K bytes long. Every fourth byte of a page is an "empty" byte, except the first one, which contains the page length. Each set of three bytes between successive empty bytes contains an instruction. The page length is read first and then bytes are loaded from the top of the page downwards. This results in shorter booting times for shorter pages.

The length of the boot page is given as:

page length = (number of 24-bit PM words / 8) – 1

That is, a page length of 0 causes the boot address generator to generate byte addresses for 8 words which reside in 32 sequential ROM locations.

The PROM Splitter utility, part of the ADSP-2100 Family Development Software tools, calculates the proper page length for your program and orders the bytes of your program as shown in Figure 10.16 (on the next page).

10 Memory Interface

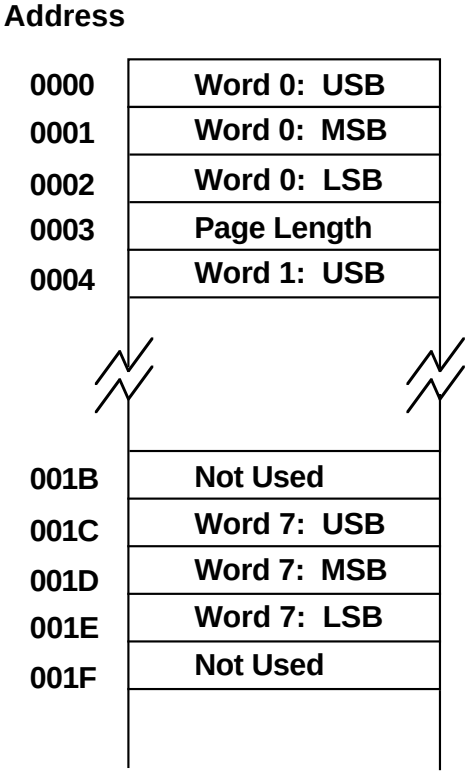


Figure 10.16 EPROM Contents

10.4.2 Powerup Boot & Software Reboot

Upon a hardware or software reset, the boot sequence occurs if the MMAP pin is a logical 0. The boot sequence on reset always loads boot page 0. After reset, boot loading can occur under program control from any one of up to 8 different boot pages. The boot page select field (BPAGE) in the memory-mapped System Control Register (see Figure 10.17) specifies which boot page is to be loaded. To boot from a specific boot page, set BPAGE to the desired page number and, in the same memory-mapped register, set the boot force bit (BFORCE). When the boot force bit is set, the software-forced booting sequence starts. Except for the page selection and (possibly) the number of wait states, there is no difference between a software-forced boot sequence and a reset boot sequence.

Tables 9.2–9.7 in the System Interface chapter show the state of the processor control registers after a reset and after a software reboot. Essentially, the processor's control state is saved, but stacks are cleared and execution starts at the restart vector, at program memory location 0x0000.

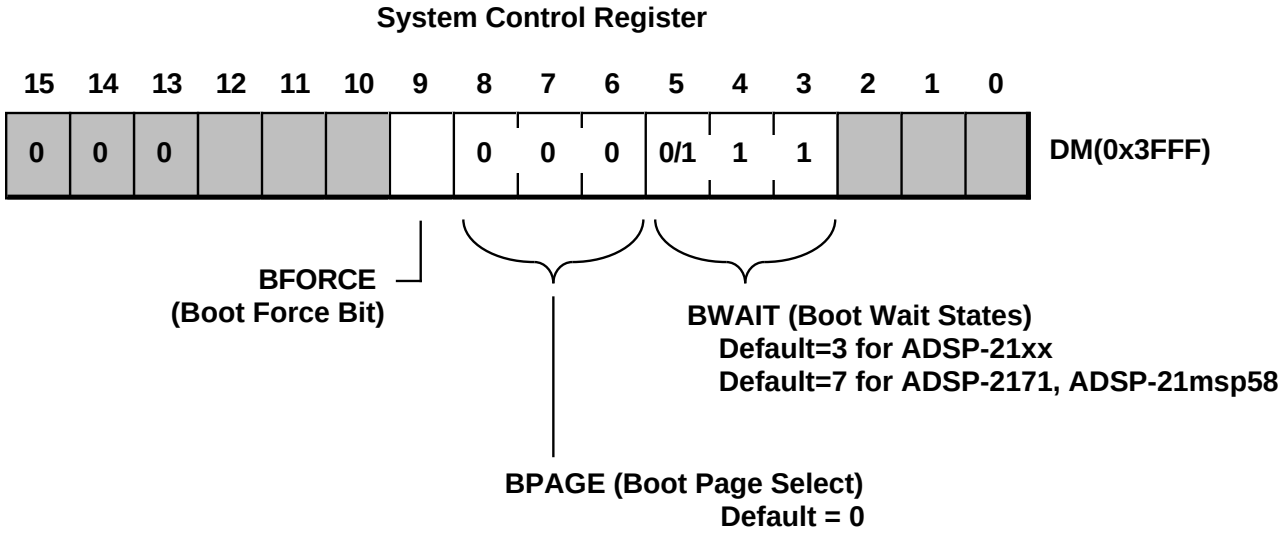


Figure 10.17 Boot Control Fields In System Control Register

10.4.3 Boot Memory Access

The processor can boot its internal memory from a single byte-wide CMOS EPROM, such as the 27C64 and 27C512. A low-cost, commodity-grade EPROM with an industry-standard access time can be used. The number of wait states for the boot memory access is selected in the BWAIT field of the System Control Register (see Figure 10.17). This field can be set to any value from 0 to 7 in order to generate 0 to 7 wait states. The default value at reset is 3 wait states on the ADSP-2101, ADSP-2105, ADSP-2111, and ADSP-2115. BWAIT defaults to 7 wait states on the ADSP-2171 and ADSP-21msp58.

Timing of the boot memory access is identical to that of external program memory or external data memory accesses, except that the active strobe is $\overline{\text{BMS}}$ rather than $\overline{\text{PMS}}$ or $\overline{\text{DMS}}$. To address eight pages of 8K bytes each, 16 bits are needed. The least significant 14 bits are output on the 14-bit address bus, and the most significant 2 bits are output on the 2 MSBs of the data bus during a boot memory access. Data is read from the middle eight bits of the data bus.

10.4.4 Boot Loading Sequence

The order in which the processor loads data into its internal memory during a boot operation is unimportant in most applications. The boot loading sequence is explained in this section for those instances in which the order is relevant, for instance when a latch is providing data rather than an EPROM.

To execute the boot operation, the boot address generator generates the appropriate byte addresses and loads internal program memory with the contents of the EPROM. The internal program memory is loaded beginning with the high addresses. For example, assume that eight 24-bit words are loaded into the processor during the booting process. The first word written into program memory is written to address 0x0007. The last word loaded is written to internal program memory address 0x0000.

The boot address is made up of several values, as shown in Figures 10.18 and 10.19: the 3-bit page number (from BPAGE in the system control register); the 8-bit page length, which is always read first (from the fourth byte of the page); a 3-bit word counter value; and a 2-bit code whose value determines which byte of the word is being addressed.

The last 24-bit word (instruction or data value) is loaded into the processor first. The byte loading order is: upper byte, lower byte, middle byte. The word pointer is then decremented. This addresses the second-to-last 24-bit word in the EPROM.

For example, to boot from page 0 the shortest allowable page (with eight 24-bit words corresponding to a page length of 0), the following addresses would be generated (see Figure 10.20):

1. The first address generated is 0x0003 which reads the page length.

2. The next address generated in this example is address 0x001C. This is the upper byte of the last word.

3. The byte code is then updated to specify the lower byte (the final two bits are 10) and the address generated is 0x001E.

4. The byte address changes again, this time to address the middle byte (the two bit code is 01) and the address generated is 0x001D.

5. Once all three bytes are loaded, the word counter is decremented. The three succeeding byte addresses generated are 0x0018, 0x001A, and 0x0019.

6. The word counter is decremented again and the next set of byte addresses generated is 0x0014, 0x0016, and 0x0015. This process continues until word 0 is loaded.

The contents of the EPROM, the byte addresses, and the order of addresses generated is shown in Figure 10.20.

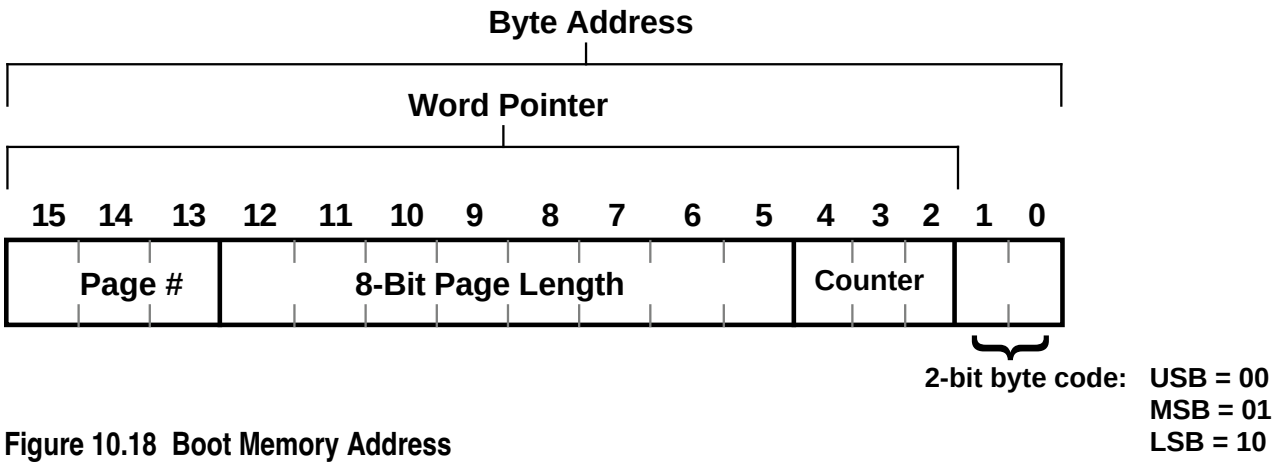


Figure 10.18 Boot Memory Address

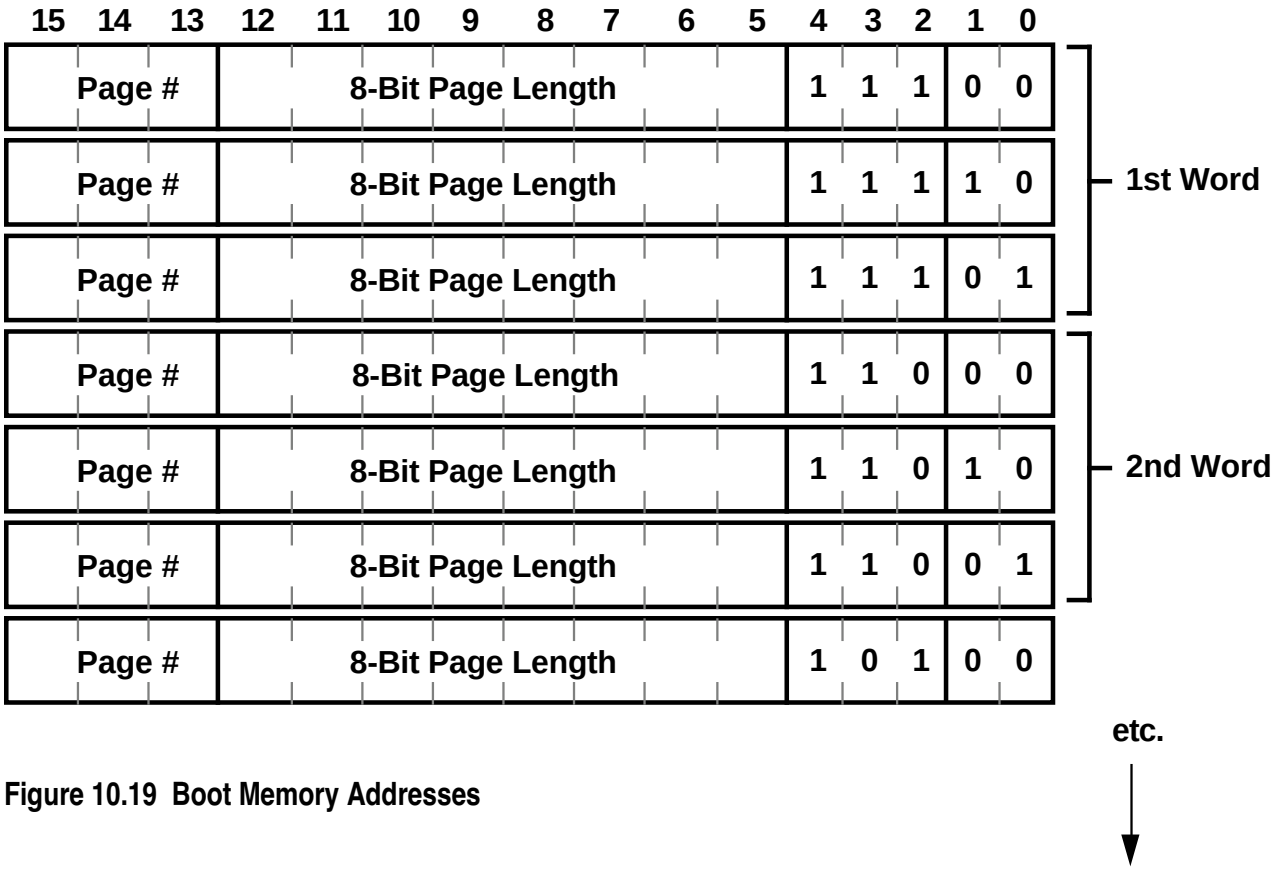


Figure 10.19 Boot Memory Addresses

10 Memory Interface

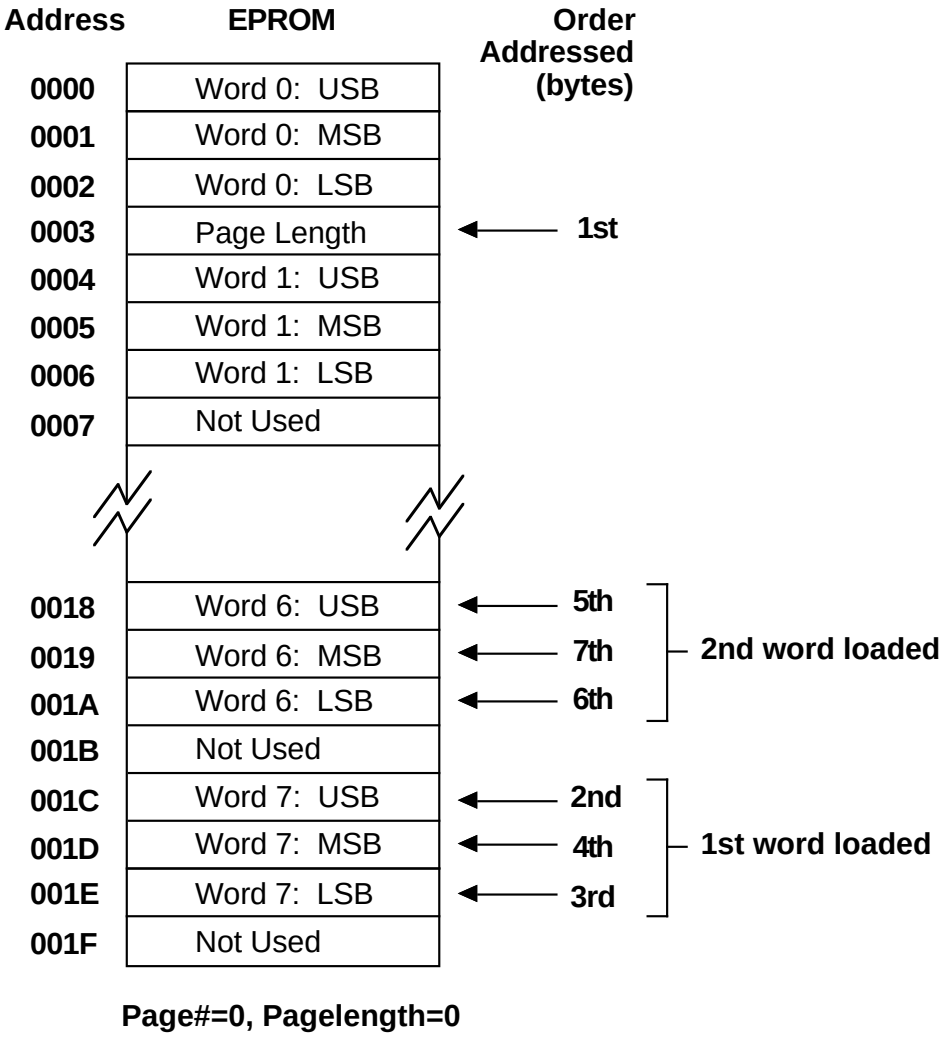


Figure 10.20 Example of Boot Loading Order (with Page#=0, Pagelength=0)

10.5 BUS REQUEST / GRANT

This section describes the bus request and grant feature of all ADSP-21xx processors, including the ADSP-2181.

The ADSP-21xx can relinquish control of its data and address buses to an external device. The external device requests the bus by asserting (low) the bus request signal, $\overline{BR}$. $\overline{BR}$ is an asynchronous input. If the ADSP-21xx is not performing an external access, it responds to the active $\overline{BR}$ input in the following processor cycle by:

- tristating the data and address buses and the $\overline{xMS}$, $\overline{RD}$, $\overline{WR}$ output drivers,
- asserting the bus grant ($\overline{BG}$) signal, and
- halting program execution (unless Go Mode is enabled).

If Go Mode is enabled, the ADSP-21xx continues to execute instructions from its internal memory. It will not halt program execution until it encounters an instruction that requires an external access. (An "external access" may be either a memory device access or, on the ADSP-2181, a memory overlay access, BDMA access, or I/O space access.)

If Go Mode is not enabled, the ADSP-21xx always halts before granting the bus. The processor's internal state is not affected by granting the bus, and the serial ports and host interface port (on the ADSP-2111, ADSP-2171, ADSP-21msp5x) remain active during a bus grant, whether or not the processor core halts.

If the ADSP-21xx is performing an external access when the $\overline{BR}$ signal is asserted, it will not grant the buses until the cycle after the access completes. The sequence of events is illustrated in Figure 10.21. The entire instruction does not need to be completed when the bus is granted. If a single instruction requires two external accesses, the bus will be granted between the two accesses. The second access is performed after $\overline{BR}$ is removed.

When the $\overline{BR}$ input is released, the ADSP-21xx releases the $\overline{BG}$ signal, reenables the output drivers and continues program execution from the point where it stopped. $\overline{BG}$ is always deasserted in the same cycle that the removal of $\overline{BR}$ is recognized. Refer to the data sheet for exact timing relationships.

The bus request feature operates at all times, including when the processor is booting and when $\overline{RESET}$ is active. During $\overline{RESET}$, $\overline{BG}$ is asserted in the same cycle that $\overline{BR}$ is recognized. During booting, the bus is granted after completion of loading of the current byte (including any wait states). Using bus request during booting is one way to bring the booting operation under control of a host computer.

10 Memory Interface

The ADSP-2171 and ADSP-2181 processors have an additional feature, the Bus Grant Hung ($\overline{BGH}$) output, which lets them operate in a multiprocessor system with a minimum number of wasted cycles. The $\overline{BGH}$ pin asserts when the ADSP-21xx is ready to execute an instruction but is stopped because the external bus is granted to another device. The other device can release the bus by deasserting bus request. Once the bus is released, the ADSP-21xx deasserts $\overline{BG}$ and $\overline{BGH}$ and executes the external access. Figure 10.22 shows timing for the $\overline{BGH}$ signal.

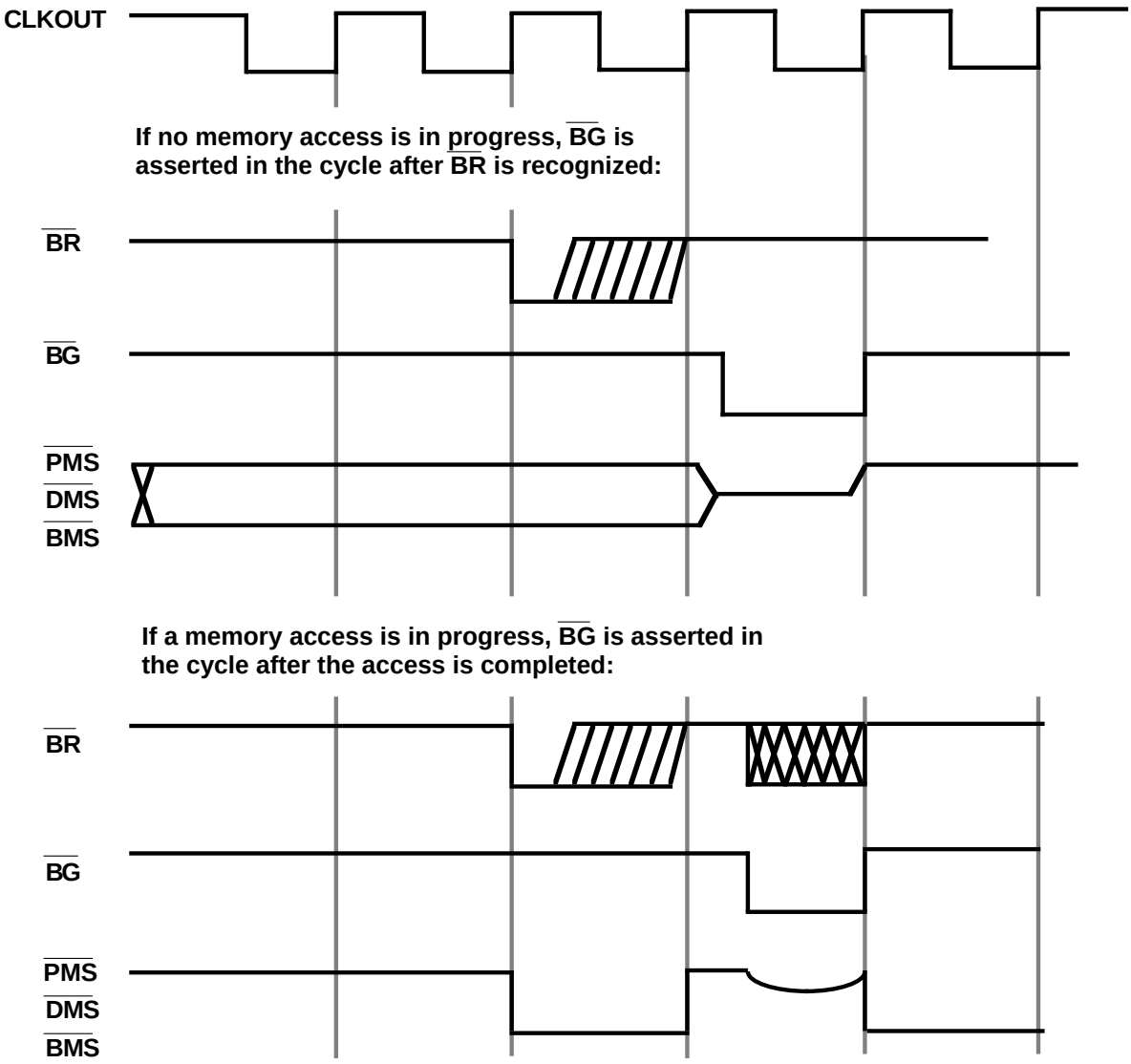


Figure 10.21 Bus Request (with and without external access)

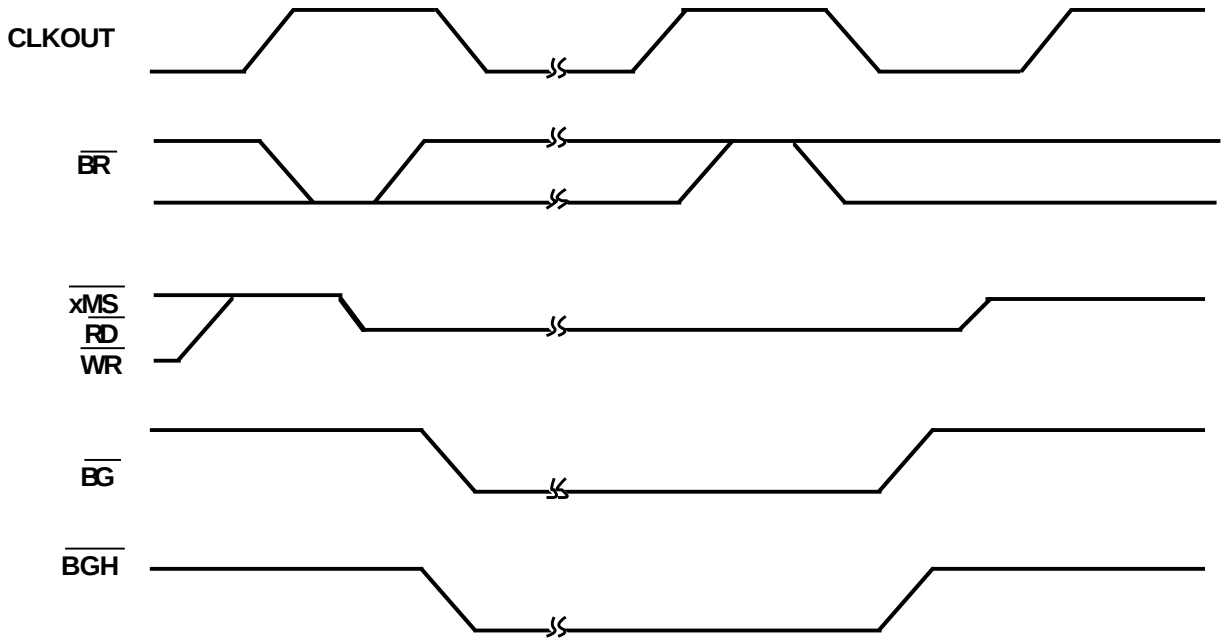


Figure 10.22 Bus Grant Hung (BGH) Timing (ADSP-2171, ADSP-2181 only)

10.6 ADSP-2181 MEMORY INTERFACES

The ADSP-2181 has the same modified Harvard architecture for internal memory as the other processors of the ADSP-2100 family. In this architecture, Data Memory stores data values and Program Memory stores both instructions and data. The ADSP-2181 has as its full base memory on-chip: 16K x 24-bit words of internal program memory RAM and 16K x 16-bit words of internal data memory RAM.

There are four separate memory spaces: data memory, program memory, byte memory, and I/O memory. To provide external access to these memory spaces, the ADSP-2181 extends the internal address and data buses off-chip and provides the PMS, DMS, BMS, and IOMS select lines. The PMS, DMS, BMS, and IOMS signals indicate which memory space is being accessed.

The composite memory space (and its CMS select line) lets a single off-chip memory be accessed as multiple memory spaces. The Composite Memory Select register lets you define which memory spaces are selected by the CMS signal.

10 Memory Interface

Figure 10.23 shows the external memory buses and control signals in an ADSP-2181 system. Two control lines determine the direction of external memory transfers: $\overline{RD}$ is active low signaling a read and $\overline{WR}$ is active low for a write operation. Typically, you would connect $\overline{RD}$ to $\overline{OE}$ (Output Enable) and $\overline{WR}$ to $\overline{WE}$ (Write Enable) of your memory.

Internal memory accesses do not drive any external signals: $\overline{PMS}$, $\overline{DMS}$, $\overline{BMS}$, $\overline{IOMS}$, $\overline{RD}$, and $\overline{WR}$ remain high (deasserted), and the address and data buses are tristated.

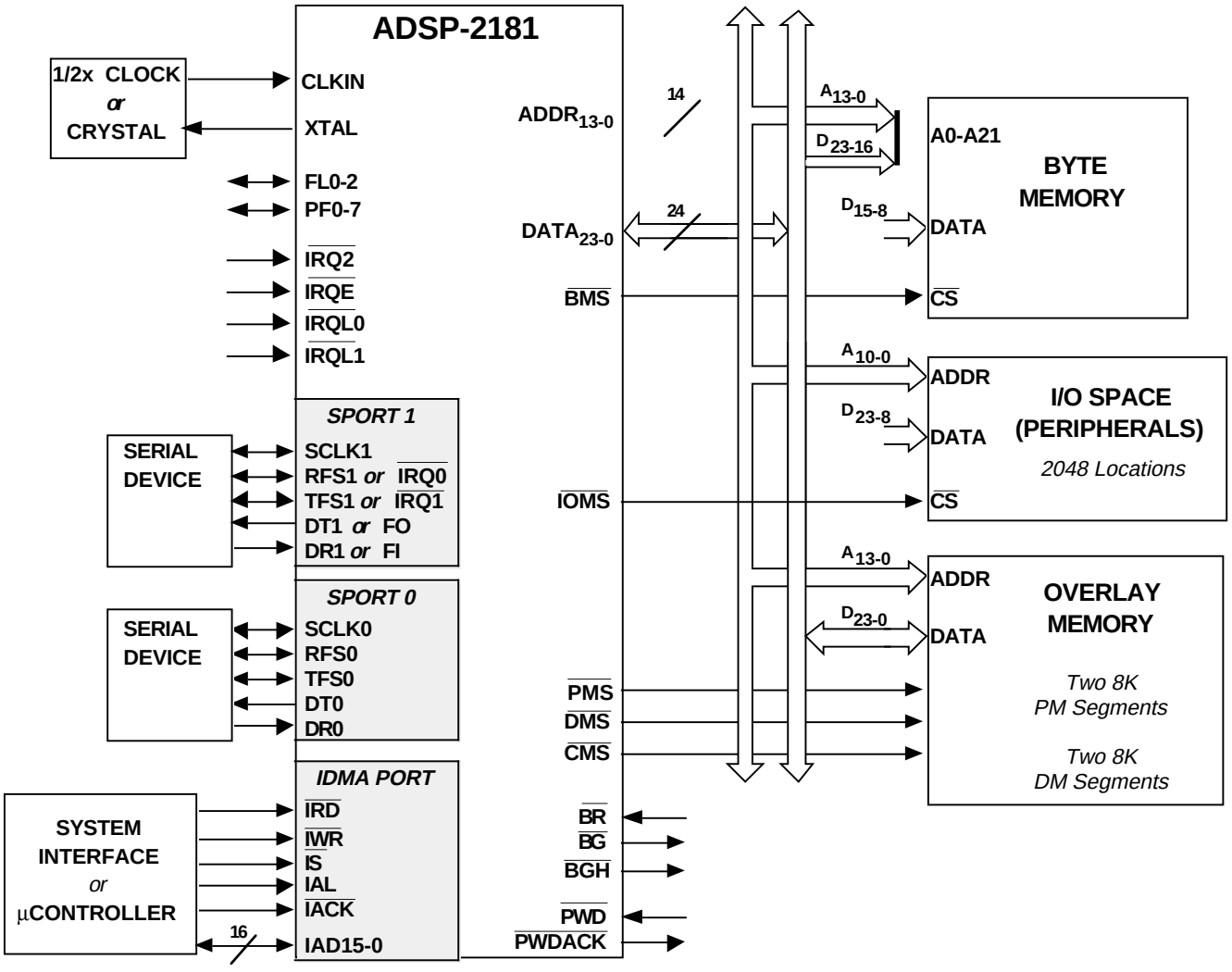


Figure 10.23 ADSP-2181 System Diagram

Unlike other processors of the ADSP-2100 family, the ADSP-2181 supports several additional memory interfacing features. These features include:

- **External Overlay Memory** in 8K segments: these segments can be swapped for the upper 8K of internal program memory or lower 8K of data memory.
- **I/O Memory** space: this memory space is for peripheral I/O, has 2K (16-bit wide) locations, and has four user-assignable waitstate ranges.
- **Byte Memory & Byte Memory DMA (BDMA)**: this memory space can address up to 4M bytes. The byte memory interface supports booting from and runtime access to inexpensive 8-bit memories. The DMA feature lets you define the number of memory locations the DSP will transfer to/from internal memory in the background while continuing foreground processing.
- **Internal Direct Memory Access (IDMA) Port**: this port supports booting from and runtime access to host systems (for example, PC Bus Interface ASICs). The DMA feature of this port lets you define the number of memory locations the DSP will transfer to/from internal memory in the background while continuing foreground processing.

For complete information on the BDMA port, including booting, and IDMA port, refer to the *DMA Ports* chapter of this manual.

The ADSP-2181 uses a half-instruction-rate clock input from which it generates a full-instruction-rate internal clock. For example, from a 16.67 MHz clock input (CLKIN) the ADSP-2181 generates a 33.33 MHz instruction rate clock. All timing diagrams for the processor use the full-instruction-rate output clock (CLKOUT) as a reference.

All external memories may have automatic wait state generation associated with them. The number of wait states—each equal to one instruction cycle—is programmable.

10.6.1 ADSP-2181 Program Memory Interface

The ADSP-2181 processor addresses its 16K of internal program memory as well as two 8K external program memory overlays. All program memory is 24 bits wide. Up to two accesses to internal program memory can be completed per instruction cycle; this lets the DSP complete all operations in a single cycle. The PWAIT field of the System Control Register (shown in Figure 10.24) sets the number of waitstates for each access to program memory overlays. PWAIT defaults (after reset) to seven.

10 Memory Interface

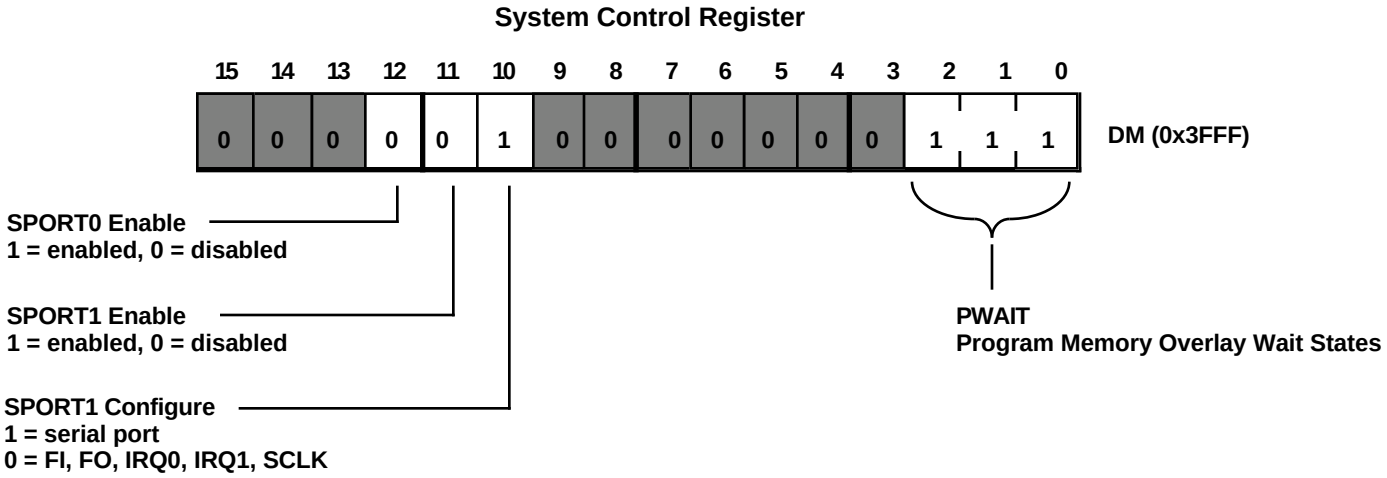


Figure 10.24 PWAIT Field in System Control Register

The on-chip program memory and overlays can hold instructions and data intermixed in any combination. The ADSP-21xx linker determines where to place relocatable code and data segments. You may specify absolute address placement for any module or data structure, including the code for the restart and interrupt vector locations. The restart vector is at program memory address 0x0000.

The ADSP-2181's MMAP pin lets you select from two program memory configurations. The MMAP pin also controls whether the ADSP-2181 boots after $\overline{\text{RESET}}$ is released. Figure 10.25 shows the MMAP options and the resulting memory maps for program memory.

The program memory overlay select register (PMOVLAY) lets you choose a memory overlay to map from address PM(0x2000) to address PM(0x3FFF). The memory mapped to this space and corresponding PMOVLAY register values are shown in Figure 10.25. Table 10.3 shows how PMOVLAY relates to the addressing of memory locations (with address line A13).

PMOVLAY	*Memory*	*A13*	*A12:0*
0	Internal	—	—
1	External overlay 1	0	13 LSBs of address between 0x2000 and 0x3FFF
2	External overlay 2	1	13 LSBs of address between 0x2000 and 0x3FFF

Table 10.3 PMOVLAY and Program Memory Overlay Addressing

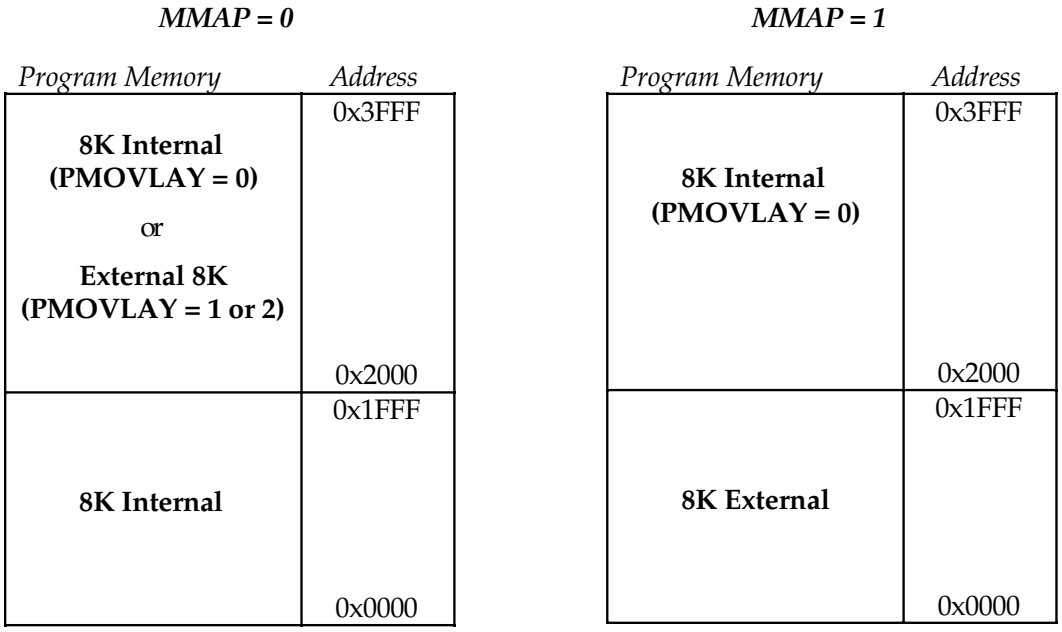


Figure 10.25 ADSP-2181 Program Memory Map

The following example instructions demonstrate how to use the PMOVLAY register.

```
PMOVLAY=DM(0x1234);  {type 3 instruction, PMOVLAY is loaded
}
                     { with the contents of address
DM(0x1234)}

PMOVLAY=2;           {type 7 instruction, PMOVLAY is loaded }
                     { with the value 2. }

PMOVLAY=AX0;         {PMOVLAY is loaded from AX0 register.}

AX0=PMOVLAY;         {AX0 is loaded from PMOVLAY register.}
```

If you are using a system design that sets MMAP=1, note that the first 8K is used to support a single segment of external memory. This allows an external ROM-based system to operate properly. In this mode, the external program memory address always has A13 set to 0 and 8K of internal PM is available. Set PMOVLAY=0 and MMAP=1. This mode is available on other

10 Memory Interface

ADSP-2100 family processors.

Figure 10.26 shows a memory design that provides full external program and data memory overlays for an ADSP-2181 processor, assuming that MMAP=0. The important points to note about this design are:

- Three 32K x 8-bit SRAMs are required for full external program and data memory overlays; glue logic is *not* required.
- Four control lines are required for read ($\overline{RD}$), write ($\overline{WR}$), chip select ($\overline{CMS}$), and data/program memory select ($\overline{PMS}$ or $\overline{DMS}$).
- Composite Memory Select (CMSSEL) is configured to assert the $\overline{CMS}$ control line when Program Memory Select ($\overline{PMS}$) or Data Memory Select ($\overline{DMS}$) are asserted.
- The order of overlays stored in this design (from lowest address to highest) is PM Overlay 1, PM Overlay 2, DM Overlay 1, and DM Overlay 2. Address line 13 (A13) of the ADSP-2181 selects between overlay 1 or 2. Figure 10.27 shows a memory map of this design.

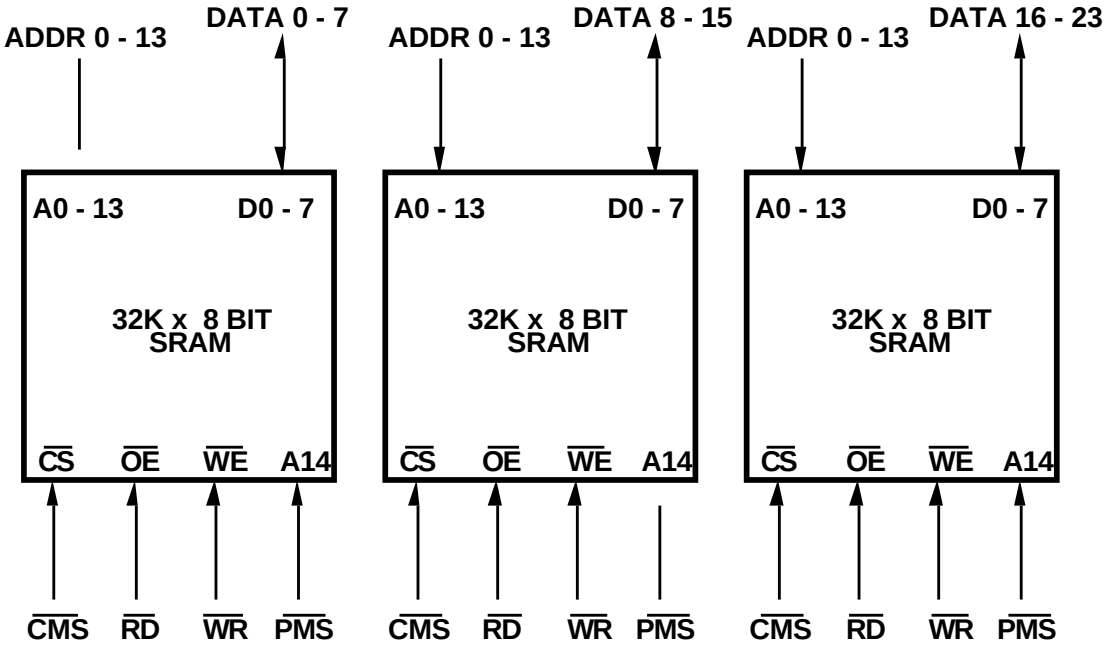


Figure 10.26 Example Program and Data Memory Overlay Design

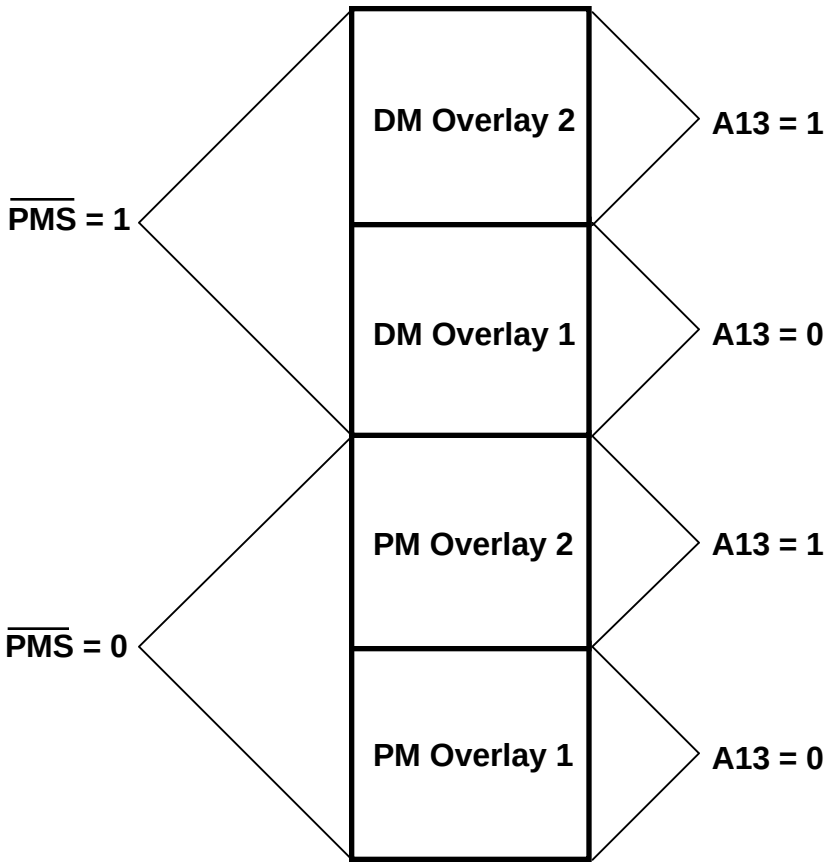


Figure 10.27 Memory Overlay Addressing For Example Design

There are some restrictions on using program memory overlays:

- The ADSP-2181's program sequencer does *not* consider the value in the PMOVLAY register. Switching pages during operations that are sensitive to the current PMOVLAY register value can result in program execution errors. For example, if your program is performing a loop operation on one of the external overlays and the program changes to another external or internal overlay, an incorrect loop operation could occur.

- The contents of the PMOVLAY register are *not* automatically saved and restored on the processors status stack when the processor responds to an interrupt. If your program uses overlays, you must save and restore the

contents of PMOVLAY as part of your interrupt service routine.

10.6.2 ADSP-2181 Data Memory Interface

The ADSP-2181 addresses 16K x 16-bit wide internal data memory and two 8K x 16-bit wide external data memory overlays. All accesses to internal data memory are completed in a single processor instruction cycle. The DWAIT field of the Waitstate Control Register (shown in Figure 10.28) sets the number of waitstates for each access to data memory overlays. Figure 10.29 shows the data memory map of the ADSP-2181.

The processor's memory-mapped control/status registers are mapped into the top locations of internal data memory, addresses 0x3FE0-0x3FFF. Most of the ADSP-2181's control registers correspond to those found on other ADSP-21xx processors. Note that the ADSP-2181's System Control Register does not have the boot memory control fields found on other ADSP-21xx processors. Also note that the Waitstate Control Register

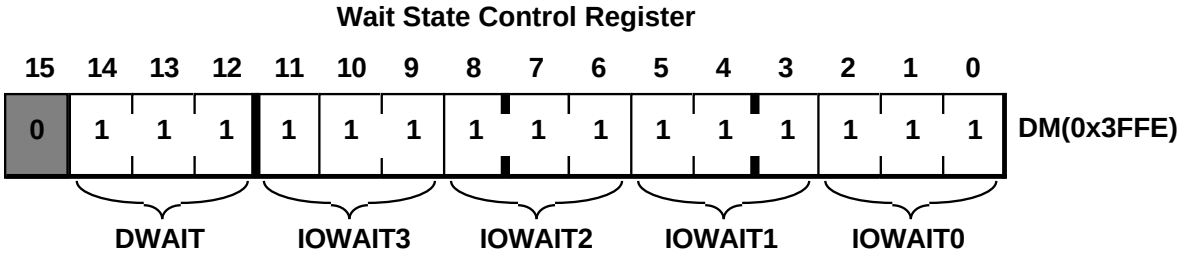


Figure 10.28 ADSP-2181 Wait State Control Register

Data Memory	*Address*
32 Memory-Mapped Control Registers	0x3FFF 0x3FE0
Internal 8160 words	0x3FDF 0x2000
8K Internal (DMOVLAY=0) or External 8K (DMOVLAY=1,2)	0x1FFF 0x0000

Figure 10.29 ADSP-2181 Data Memory Map

includes four fields for the ADSP-2181's I/O memory space.

The Data Memory overlay select (DMOVLAY) register lets you choose a memory overlay to map from address DM(0x0000) to address DM(0x1FFF). The DMOVLAY register is unique to the ADSP-2181. The memory mapped to this space and corresponding DMOVLAY contents are shown in Figure 10.29. Table 10.4 shows how DMOVLAY relates to memory addressing (address line A13).

DMOVLAY	*Memory*	*A13*	*A12:0*
0	Internal	—	—
1	External overlay 1	0	13 LSBs of address between 0x0000 and 0x1FFF
2	External overlay 2	1	13 LSBs of address between 0x0000 and 0x1FFF

Table 10.4 DMOVLAY and Data Memory Overlay Addressing

The following example instructions demonstrate how to use the DMOVLAY register:

```
DMOVLAY=DM(0x1234);  {type 3 instruction, DMOVLAY is loaded
}
                     { with the contents of address
DM(0x1234)}

DMOVLAY=2;           {type 7 instruction, DMOVLAY is loaded }
                     { with the value 2. }

DMOVLAY=AX0;         {DMOVLAY is loaded from AX0 register.}

AX0=DMOVLAY;         {AX0 is loaded from DMOVLAY register.}
```

For an example memory design that provides full external program and data memory overlays for an ADSP-2181 processor, see the previous section "Program Memory Interface."

Two control lines indicate the direction of external transfers. Memory read ($\overline{RD}$) is active low signaling a read and memory write ($\overline{WR}$) is active low for a write operation. Typically, you would connect $\overline{DMS}$ to $\overline{CE}$ (Chip Enable),

$\overline{RD}$ to $\overline{OE}$ (Output Enable) and $\overline{WR}$ to $\overline{WE}$ (Write Enable) of your memory.

10.6.3 ADSP-2181 Byte Memory Interface

The ADSP-2181's byte memory space is 8 bits wide and can address up to 4M bytes of program code or data. This memory space takes the place of the boot memory space found on other ADSP-2100 family processors. Unlike boot memory space, byte memory has read/write access through the ADSP-2181's BDMA port.

Byte memory space consists of 256 pages, each containing 16K x 8-bit wide locations. This memory can be written and read in four different formats: 24-bit, 16-bit, 8-bit MSB alignment, and 8-bit LSB alignment.

Each read/write to byte memory consists of data (on data bus lines 15:8) and address (on address bus lines 13:0 plus data lines 23:16). The 22-bit byte memory address lets you access up to 4M bytes of ROM or RAM.

For complete information on the ADSP-2181's byte memory and BDMA port, refer to the *DMA Ports* chapter of this manual.

10.6.4 ADSP-2181 I/O Memory Space

The ADSP-2181 has a dedicated I/O Memory Space instead of the memory-mapped I/O used on other ADSP-21xx processors. The I/O memory space consists of 2048 locations with four associated programmable waitstate regions. Figure 10.30 shows the Wait State

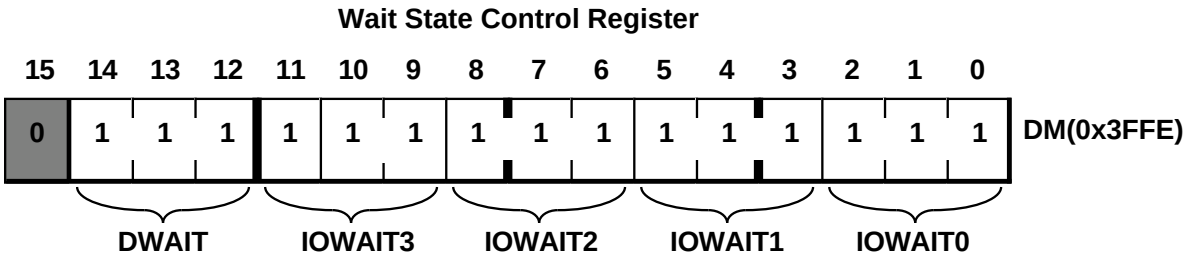


Figure 10.30 ADSP-2181 Waitstate Control Register

Control Register and the IOWAIT0-3 bit fields that control I/O memory waitstate regions.

The Wait State Control Register is divided into the following fields:

- **IOWAIT0**. This 3-bit field sets the number of waitstates (0-7) for accesses to I/O memory addresses 0x000–0x1FF.
- **IOWAIT1**. This 3-bit field sets the number of waitstates (0-7) for accesses to I/O memory addresses 0x200–0x3FF.
- **IOWAIT2**. This 3-bit field sets the number of waitstates (0-7) for accesses to I/O memory addresses 0x400–0x5FF.
- **IOWAIT3**. This 3-bit field sets the number of waitstates (0-7) for accesses to I/O memory addresses 0x600–0x7FF.
- **DWAIT**. This 3-bit field sets the number of waitstates (0-7) for accesses to external program and data memory overlays.

Note: The PWAIT field of the System Control Register sets the number of waitstates for access to external program memory overlays.

When you connect a parallel I/O device to the ADSP-2181 as shown in

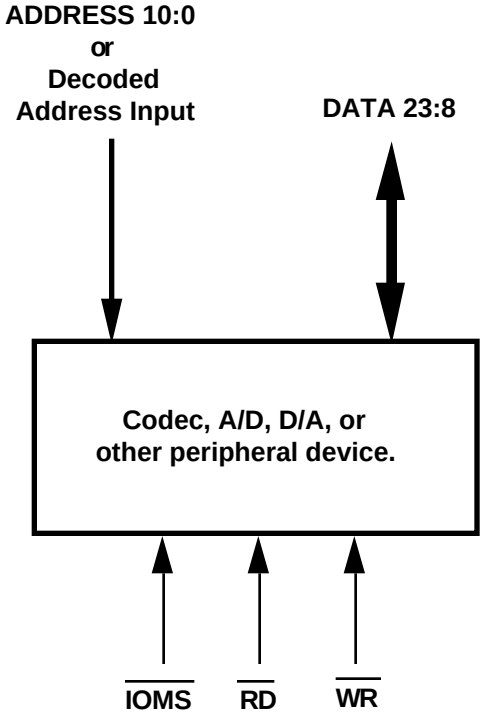


Figure 10.31 I/O Memory Space Peripheral Connection Example

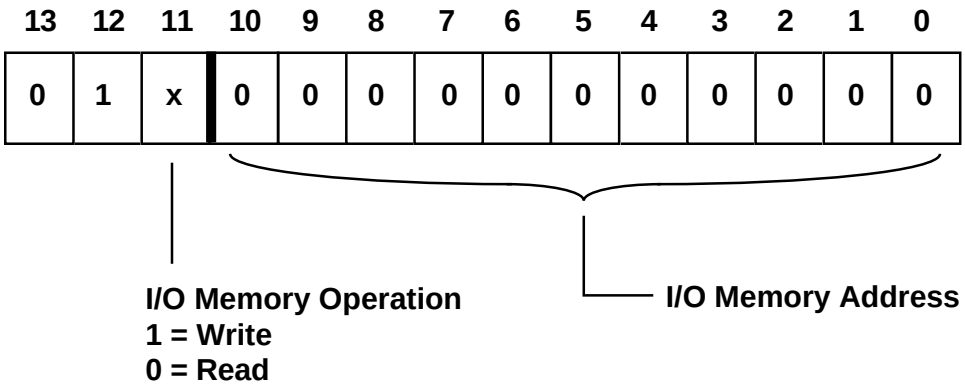


Figure 10.32 I/O Memory Address Word

Figure 10.31, the address sent to the device appears on the external address bus as shown in Figure 10.32.

Host interfaces can use the additional communications channel provided by the ADSP-2181's I/O memory space. If your system bus interface ASIC uses a set of data registers for passing control information from the system bus and must also pass large amounts of sample data, map the control registers as I/O memory peripherals and transfer the sample data using IDMA. This combination of the I/O memory and IDMA channels reduces system bus transfer rate limitations.

Note: As with other ADSP-2100 Family processors, on the ADSP-2181 you can define memory-mapped I/O ports with the assembler's .PORT directive. On the ADSP-2181, this directive defines memory-mapped I/O ports in external program memory overlays or data memory overlays. If you want to use this feature, you must make sure at runtime that you are on the correct program memory overlay or data memory overlay when accessing the port; the assembler and linker will not flag errors in .PORT accesses related to overlays because the issue is resolved at runtime. The "IO" keyword does not work with the .PORT directive; to assign symbolic

labels to I/O memory addresses, use a `#define` macro. The best use of the .PORT directive is in porting non-ADSP-2181 applications to the ADSP-2181; otherwise, use I/O memory space for memory-mapped I/O.

10.6.5 ADSP-2181 Composite Memory Select

The ADSP-2181 has a programmable memory select signal, Composite Memory Select ($\overline{CMS}$). This signal lets you generate a memory select for devices mapped to more than one memory space, with the same timing as other individual memory select signals ($\overline{PMS}$, $\overline{DMS}$, $\overline{BMS}$, and $\overline{IOMS}$).

Based on the value of CMSSEL in the Programmable Flag & Composite Select Control register (see Figure 10.33), the ADSP-2181 asserts $\overline{CMS}$

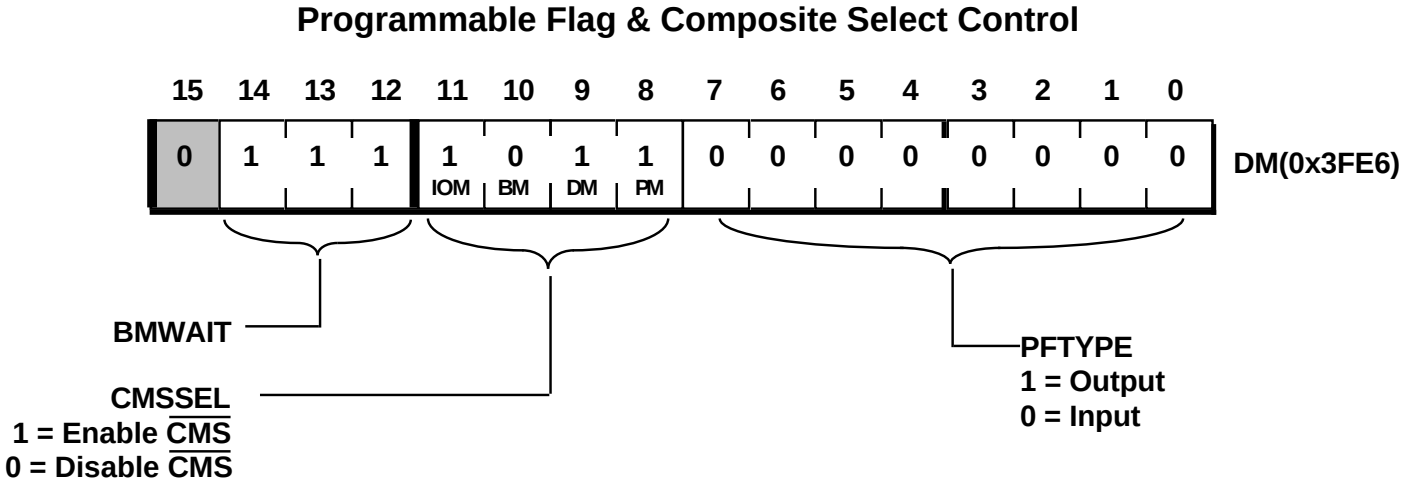


Figure 10.33 CMSSEL Selection for $\overline{CMS}$ Signal

when the corresponding memory select signal (or signals) are asserted. Each $\overline{xMS}$ signal can be individually enabled. After reset, CMSSEL is initialized to enable $\overline{PMS}$, $\overline{DMS}$, and $\overline{IOMS}$ (with $\overline{BMS}$ disabled).

Figure 10.26 (earlier in this chapter) shows an example of how to use the

$\overline{CMS}$ signal. In this system the $\overline{CMS}$ line drives the chip select for all three SRAMs. This lets you use three 32K x 8-bit SRAMs, with no glue logic, for complete program and data memory overlays.

10.6.6 External Memory Read – Overlays & I/O Memory

External memory reads may access either PM overlays, DM overlays, or I/O memory space. These read operations occur in the following sequence (see Figure 10.34):

1) The ADSP-2181 executes a read from an external memory address; the address is driven on the address bus and $\overline{PMS}$, $\overline{DMS}$, $\overline{BMS}$, or $\overline{IOMS}$, and $\overline{RD}$ is asserted. ($\overline{CMS}$ may also be asserted, depending how it is configured.)

2) The external peripheral drives the data onto the data bus.

3) The ADSP-2181 reads the data and deasserts $\overline{RD}$.

$\overline{WR}$ remains high (deasserted) throughout the external memory read operation.

Note that ADSP-2181 internal memory accesses do not drive any

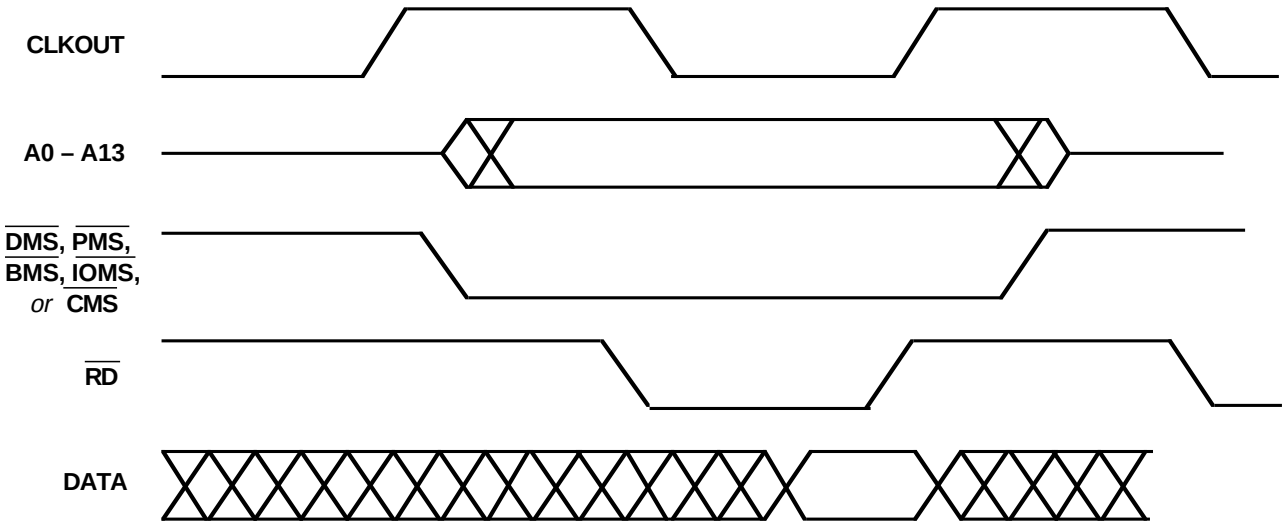


Figure 10.34 External Memory Read Timing

external signals: $\overline{PMS}$, $\overline{DMS}$, $\overline{IOMS}$, $\overline{BMS}$, $\overline{CMS}$, $\overline{RD}$, and $\overline{WR}$ remain high (deasserted), and the address and data buses are tristated.

10.6.7 External Memory Write – Overlays & I/O Memory

External memory writes may access either PM overlays, DM overlays, or I/O memory space. These read operations occur in the following sequence (see Figure 10.35):

1) The ADSP-2181 executes a write to an external memory address; the address is driven on the address bus, data is driven on the data bus, and $\overline{PMS}$, $\overline{DMS}$, $\overline{BMS}$, or $\overline{IOMS}$, and $\overline{WR}$ is asserted. ($\overline{CMS}$ may also be asserted, depending how it is configured.)

2) The external peripheral stores the data.

3) The ADSP-2181 stops driving the address and data buses and deasserts $\overline{WR}$.

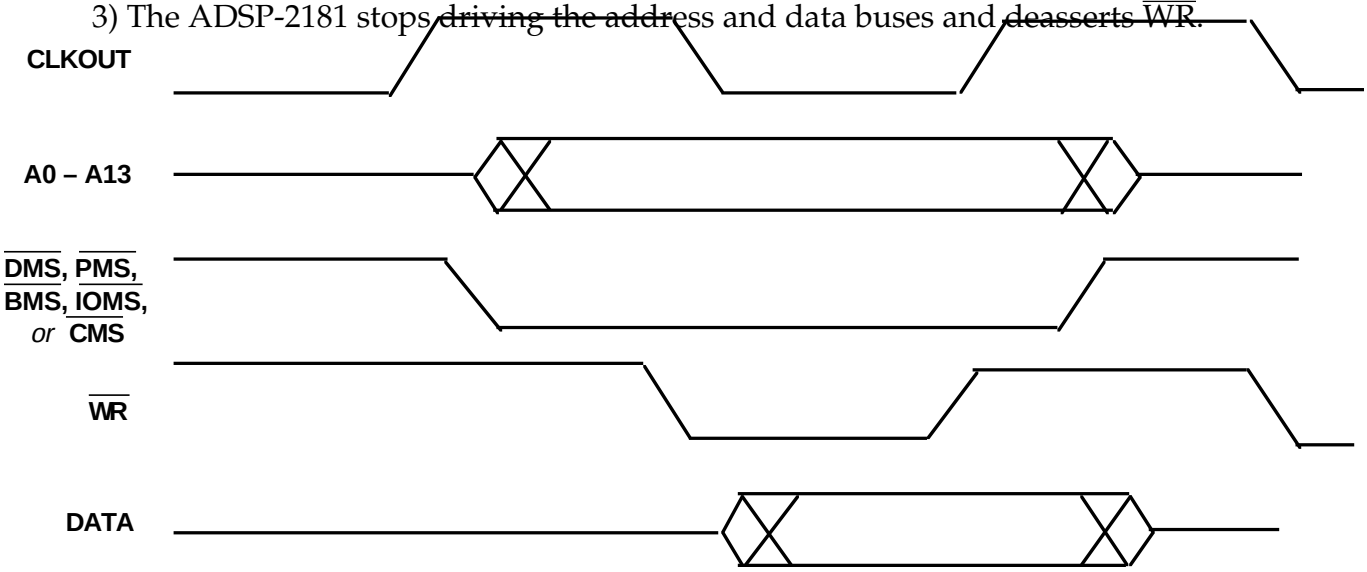


Figure 10.35 External Memory Write Timing

$\overline{RD}$ remains high (deasserted) throughout the external memory write operation.

10.7 MEMORY INTERFACE SUMMARY (ALL PROCESSORS)

Table 10.5 summarizes the states of the memory interface pins for various combinations of program memory and data memory accesses. Table 10.6 summarizes the states of the memory interface and control pins during reset, booting (ADSP-21xx boot memory booting, not ADSP-2181 byte memory booting), and bus grant.

10 Memory Interface

Access	$\overline{PMS}$	$\overline{DMS}$	$\overline{BMS}$	$\overline{RD}$	$\overline{WR}$	*Address*	*Data*
Internal program memory only	high	high	high	high	high	tristated	tristated
Internal data memory only	high	high	high	high	high	tristated	tristated
Internal program memory, external data memory	high	low	high	low (for read)	low (for write)	DM address	DM data
Internal data memory, external program memory	low	high	high	low (for read)	low (for write)	PM address	PM data
External boot memory	high	high	low	low (for read)	high	Boot address	Boot data, Boot page address

Table 10.5 Pin States During Memory Accesses

Operation	*Address*	*Data*	$\overline{PMS}$ $\overline{DMS}$ $\overline{BMS}$	$\overline{RD}$ $\overline{WR}$	*CLKOUT*	*SPORTs*	$\overline{BG}$
Reset	tristated	tristated	high	high	active	tristated	high
Booting* after Reset	active	active	$\overline{BMS}$ active $\overline{PMS}$, $\overline{DMS}$ high	$\overline{RD}$ active $\overline{WR}$ high	active	tristated	high
Reboot*	active	active	$\overline{BMS}$ active $\overline{PMS}$, $\overline{DMS}$ high	$\overline{RD}$ active $\overline{WR}$ high	active	active	high
$\overline{BR}$ Asserted during Normal Operation, Booting*, or Go Mode	tristated	tristated	tristated	tristated	active	active	low
$\overline{BR}$ Asserted during Reset	tristated	tristated	tristated	tristated	active	tristated	low

Table 10.6 Pin States During Reset, Booting*, and Bus Grant

* ADSP-21xx boot memory booting, not ADSP-2181 byte memory booting.