

Freescale Semiconductor

Order by APR 25/D

Rev. 0, 4/16/99

Interfacing Fast SRAM to Freescale's DSP56300 Family of Digital Signal Processors

by Phil Brewer

1 Introduction

This application note describes how to interface external Asynchronous Fast Static Random Access Memory (Fast SRAM) to Motorola's DSP56300 family of devices. This document is a supplement to the *DSP56300 24-Bit Digital Signal Processor Family Manual*, and to the user's manuals and technical data sheets for devices in the DSP56300 family.

The intent is to describe methods for interfacing various types of memory to the DSP56300's Memory Expansion Port in order to assist the DSP hardware engineer to fully utilize the processor's resources and generate an optimized memory design. These memory designs use a minimum of additional devices. Taking advantage of the DSP's available control lines makes virtually glueless external memory interfaces possible, thereby reducing the cost and using fewer devices in an application's memory design.

Specifically, this application note describes implementations for asynchronous Fast SRAM using the DSP56303. The DSP56303 is representative of the DSP56300 family and has all the essential family features. Where appropriate, several memory configurations provide a complete set of examples from which the designer can choose.

Contents

1	Introduction 1-1
1.1	DSP56300 Family 1-2
1.2	Application Note Organization 1-2
1.3	Static RAM (SRAM) Types 1-3
1.4	References 1-4
2	Interface Overview2-1
2.1	DSP56300 Control Signals 2-1
2.2	DSP56300 External Memory
	Timing 2-1
2.3	DSP56303 Memory Control
	Registers 2-4
3	32K × 8-bit Memory Based
	Designs3-1
3.1	32K × 24-bit Common Fast SRAM
	Hardware Design 3-2
3.2	32K × 24-bit 'P' Space Fast SRAM
	Example 3-6
3.3	$32K \times 24$ -bit 'P'/'X'/'Y' Fast SRAM
	Example 3-13
3.4	$16K \times 24$ -bit 'P'/'X' and
	16K × 24-bit 'Y' Fast SRAM
	Example
4	128K × 8-bit Memory Based
4	128K × 8-bit Memory Based Designs4-1
4 .1	128K × 8-bit Memory Based Designs4-1 128K × 24-bit Common Fast SRAM
-	128K × 8-bit Memory Based Designs4-1 128K × 24-bit Common Fast SRAM Hardware Design4-2
-	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM Hardware Design 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 4-2
4.1	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 Hardware Design 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5RAM Example
4.1	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5 SRAM Example 4-6 64K × 24-bit 'X' and 4-6
4.1	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5 SRAM Example 4-6 64K × 24-bit 'X' and 64K × 24-bit 'Y' Fast SRAM
4.1 4.2 4.3	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5 SRAM Example 4-6 64K × 24-bit 'X' and 64K × 24-bit 'Y' Fast SRAM Example 4-13
4.1	128K × 8-bit Memory Based Designs
4.1 4.2 4.3	128K × 8-bit Memory Based Designs
4.1 4.2 4.3	128K × 8-bit Memory Based Designs
4.1 4.2 4.3 5 5.1	128K × 8-bit Memory Based Designs
4.1 4.2 4.3	128K × 8-bit Memory Based Designs
4.1 4.2 4.3 5 5.1	128K × 8-bit Memory Based Designs
4.1 4.2 4.3 5 5.1 5.2	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5 SRAM Example 4-6 64K × 24-bit 'Y' Fast SRAM 6 Example 4-13 64K × 24-bit Memory Based 6 Designs 5-1 64K × 24-bit Common Fast SRAM 6 Hardware Design 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6
4.1 4.2 4.3 5 5.1	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 5 SRAM Example 4-6 64K × 24-bit 'Y' Fast SRAM 6 Example 4-13 64K × 24-bit Memory Based 6 Designs 5-1 64K × 24-bit Common Fast SRAM 6 Hardware Design 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6 32K × 24-bit 'X' and 5-6
4.1 4.2 4.3 5 5.1 5.2	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 4-2 128K × 24-bit 'P'/'X'/Y' Fast 4-6 64K × 24-bit 'X' and 4-6 64K × 24-bit 'Y' Fast SRAM 4-13 64K × 24-bit Memory Based 5-1 64K × 24-bit Common Fast SRAM 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6 32K × 24-bit 'X' and 32K × 24-bit 'Y'
4.1 4.2 4.3 5 5.1 5.2	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 4-6 SRAM Example 4-6 64K × 24-bit 'Y' Fast SRAM 4-13 Example 4-13 64K × 24-bit Memory Based 4-13 Designs 5-1 64K × 24-bit Common Fast SRAM 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6 32K × 24-bit 'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-14
4.1 4.2 4.3 5 5.1 5.2	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 4-6 SRAM Example 4-6 64K × 24-bit 'Y' Fast SRAM 4-13 Example 4-13 64K × 24-bit Memory Based 4-13 Designs 5-1 64K × 24-bit Common Fast SRAM 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6 32K × 24-bit 'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-14 DSP Memory Space
4.1 4.2 4.3 5 5.1 5.2	128K × 8-bit Memory Based Designs 4-1 128K × 24-bit Common Fast SRAM 4-2 128K × 24-bit 'P'/'X'/'Y' Fast 4-6 SRAM Example 4-6 64K × 24-bit 'Y' Fast SRAM 4-13 Example 4-13 64K × 24-bit Memory Based 4-13 Designs 5-1 64K × 24-bit Common Fast SRAM 5-2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-6 32K × 24-bit 'X' and 32K × 24-bit 'Y' Fast SRAM Example 5-14





1.1 DSP56300 Family

The Motorola DSP56300 family of DSPs uses a programmable 24-bit fixed-point core. This core is a high-performance, single-clock-cycle-per-instruction engine that provides almost twice the performance of Motorola's popular DSP56000 family core while retaining code compatibility.

The DSP56300 family core consists of a Peripheral/Memory Expansion Port (Port A), External Memory and Peripheral DRAM controller, Data Arithmetic Logic Unit (Data ALU), Address Generation Unit (AGU), Instruction Cache Controller, Program Control Unit, on-chip concurrent six-channel DMA controller, PLL Clock Generator, On-Chip Emulation (OnCETM) module, JTAG Test Access Port (TAP) compatible with the IEEE 1149.1 Standard, and a Peripheral and Memory Expansion Bus. The main features of the core include:

- Object code compatibility with the DSP56000 core
- Modified Harvard architecture with 24-bit instruction and 24-bit data width
- Fully pipelined 24 × 24-bit parallel Multiplier-Accumulator (MAC)
- 56-bit parallel barrel shifter
- 16-bit arithmetic mode of operation
- Highly parallel instruction set
- Position Independent Code (PIC) instruction-set support
- Unique DSP addressing modes
- On-chip memory-expandable hardware stack
- Nested hardware DO loops
- Fast auto-return interrupts
- On-chip instruction cache
- External Memory and Peripheral Access Attribute select support
- On-chip Phase Lock Loop (PLL)
- Program address tracing support

The main differences between derivatives of the DSP56300 family are the size of the on-chip memory and the types of on-chip peripherals and hardware accelerators.

1.2 Application Note Organization

This document has six sections:

- 1. An overview of the contents of this application note
- 2. A description of the DSP56303 memory expansion port, Port A, including its use and timing characteristics
- 3. Three examples of common memory space configurations based on a common hardware memory design that uses $32K \times 8$ -bit $5.0 \times$
- 4. Two examples of common memory space configurations based on a common hardware design that uses 128K x 8-bit 3.3 V Fast SRAM and two Address Attribute Selectors
- 5. Two examples of common memory space configurations based on a common hardware design that uses 64K x 24-bit 3.3 V Fast SRAM and two Address Attributes



6. A summary of the memory design implementations presented in the previous sections and the advantages of using Fast SRAM.

1.3 Static RAM (SRAM) Types

The following SRAM types were considered for this application note.

- Asynchronous Fast SRAM
 - Asynchronous Fast SRAM 8-bit
 - Fast SRAM 5.0 V
 - Fast SRAM 3.3 V
 - Asynchronous Fast SRAM 16-bit
 - Asynchronous Fast SRAM 24-bit
 - Asynchronous Fast SRAM 32-bit
 - Fast SRAM 5.0 V
 - Fast SRAM 3.3 V
 - Asynchronous Fast SRAM modules
- Cache
 - Dual I/O Synchronous SRAM
 - Burst RAM Synchronous Fast SRAM
 - Pipelined Burst RAM Synchronous Fast SRAM
 - Flow-Through Pipelined Burst RAM Synchronous Fast SRAM
 - Secondary Cache Modules
- Synchronous Fast SRAM (SSRAM)

However, only the most popular memory types and those types requiring little or no supporting hardware (i.e., glue logic) are included in this application note. The examples in this report give designers the insight necessary to implement other memory families and memory types, if needed, with DSP56300 family devices.

To achieve the fastest operation on the external memory bus and, consequently, the fewest number of DSP wait states requires the fastest memory available (i.e., Fast SRAM). Slower SRAM can be substituted for the Fast SRAM resulting in DSP-generated external memory wait states each of which is roughly equivalent to one clock period of the DSP core. Each wait state is 12.5 nS for a DSP core running at 80 MHz.

An asynchronous external access for a DSP56300 family device incurs an automatic one wait state penalty when the address for the external memory device is required to be stable during the entire access. Since asynchronous Fast SRAM and SRAM devices have this address stability requirement, the DSP operates with at least one wait state when using these external memories.



1.4 References

Motorola DSP56300 24-bit Digital Signal Processor Family Manual (DSP56300FM/AD), Motorola, 1995.

DSP56301 Technical Data Sheet (DSP56301/D), Motorola, 1995.

DSP56302 Technical Data Sheet (DSP56302/D), Motorola, 1996.

DSP56303 Technical Data Sheet (DSP56303/D), Motorola, 1996.

Motorola DSP56303 24-bit Digital Signal Processor (DSP56303UM/AD), Motorola, 1996.

Motorola Dynamic RAM's & Memory Modules (DL155/D), Motorola, 1993.

Motorola Fast Static RAM Component and Module Data (DL156/D), Rev 3, Motorola, 1995.

Advanced Micro Devices CMOS Memory Products 1991 Data Book/Handbook, AMD, 1991.

Advanced Micro Devices FLASH Memory Products 1994/1995 Data Book/Handbook, AMD, 1995.

Quality Semiconductor QuickSwitch® Products Databook, Quality Semiconductor, 1995.

Quality Semiconductor Application Note AN-11, Bus Switches Provide 5 V and 3 V Logic Conversion with Zero Delay, 1995.



2 Interface Overview

This section describes how external memory devices are interfaced to Motorola's DSP56300 family of devices using its memory and peripheral expansion port.

2.1 DSP56300 Control Signals

Only the DSP control signals used in the memory implementation examples in this note are described below. Other memory configuration implementations may require other support features, which the DSP56300 family of devices have available to assist the designer. These additional memory control signals are described in the Port A Chapter of the DSP56300 24-Bit Digital Signal Processor Family Manual.

- Read Data Enable (RD)—An active low output signal that is asserted during an external memory or peripheral read access.
- Write Data Enable (WR)—An active low output signal that is asserted during an external memory or peripheral write access.
- Address Bus (A0–A17)—Eighteen address lines that allow the DSP563003 to directly address 256K words of external memory or peripherals. These active high output signals are asserted only during external memory or peripheral read or write accesses. These signal lines maintain state when external memory spaces are not being accessed.
- Data Bus (D0–D23)—Twenty-four data lines on the DSP56303 that are active high bidirectional signals asserted only during external program and data memory accesses. These signal lines maintain state when external memory spaces are not being accessed.
- Address Attribute/Row Address Strobe (AA[0-3]/RAS[0-3])—Four Address Attribute or Row Address Strobe signals. When the Address Attribute, AA, option is selected for these signal lines, they can function as chip selects or additional address lines. When the Row Address Strobe, RAS, option is selected for these signal lines, they can function as Row Address Strobe lines for DRAM interfacing.

2.2 DSP56300 External Memory Timing

The DSP56300 family derives its core clock from one of various sources (see PLL and Clock Generator chapter in the *DSP56300 24-Bit Digital Signal Processor Family Manual* for details). All memory interface timings are derived from the period of the DSP core clock. For example, if the DSP core clock frequency is 80 MHz, then the memory timings are based on a 12.5 nS clock cycle time, and an external memory typically requires less than 12.5 nS access time for one DSP wait state operation. However, these timings are affected by several factors, such as the use of the Phase Lock Loop, the use of an external frequency over or under 4 MHz, the source of the external frequency, and propagation delays in the DSP itself. Any of these factors can cause this value to deviate from 12.5 nS. **Table 2-1** represents expected required memory performance data at an 80 MHz DSP core frequency and various wait states using the DSP56303.



Table 2-1. External Memory Speeds with DSP Wait States

External Clock (MHz)	DF	MF	PDF	ws	Core Clock (MHz)	T _C (nS)	t _{AA} – max (nS)	t _{AW} – min (nS)
4.00	1	20	1	1	80.00	12.5	12.4	17.9
4.00	1	20	1	2	80.00	12.5	24.9	30.4
4.00	1	20	1	3	80.00	12.5	37.4	42.9
4.00	1	20	1	4	80.00	12.5	49.9	55.4
4.00	1	20	1	5	80.00	12.5	62.4	67.9
4.00	1	20	1	6	80.00	12.5	74.9	80.4
4.00	1	20	1	7	80.00	12.5	87.4	92.9
4.00	1	20	1	8	80.00	12.5	99.9	105.4
4.00	1	20	1	9	80.00	12.5	112.4	117.9
4.00	1	20	1	10	80.00	12.5	124.9	130.4
4.00	1	20	1	11	80.00	12.5	137.4	142.9
4.00	1	20	1	12	80.00	12.5	149.9	155.4
4.00	1	20	1	13	80.00	12.5	162.4	167.9
4.00	1	20	1	14	80.00	12.5	174.9	180.4
4.00	1	20	1	15	80.00	12.5	187.4	192.9
4.00	1	20	1	16	80.00	12.5	199.9	205.4
4.00	1	20	1	17	80.00	12.5	212.4	217.9
4.00	1	20	1	18	80.00	12.5	224.9	230.4
4.00	1	20	1	19	80.00	12.5	237.4	242.9
4.00	1	20	1	20	80.00	12.5	249.9	255.4

DF = PLL Division Factor

MF = PLL Multiplication Factor

PDF = PLL Pre-Division Factor

WS = wait states

TC = Clock Cycle Time

 t_{AA} = Data access time (i.e., address and AA valid to input data valid)

 t_{AW} = Data access time (i.e., address and AA valid to \overline{WR} deassertion)



2.2.1 DSP56303 External Memory Bus Asynchronous Read Timing

When reading from external asynchronous memory, the DSP56303 memory read access is controlled by the following steps.

- 1. The required memory address is asserted. The memory address is created by combining the address bus, A0–A17, and the Address Attributes, AA0–AA3.
- 2. After a delay of t_{AR} (i.e., address valid to \overline{RD} assertion time), the Read enable signal, \overline{RD} , is asserted.
- 3. Before a delay of t_{OE} (i.e., \overline{RD} assertion to input data valid), the memory device puts valid data on the data bus.
- 4. The DSP latches the data bus data and deasserts \overline{RD} . The DSP does not require any data hold time, t_{OHZ} , after deassertion of the \overline{RD} signal.

The data access time, t_{AA} (i.e., address and AA valid to input data valid), is the time delay typically used by memory devices to specify data access timing. The t_{AA} for a memory device must be less than or equal to the DSP's t_{AA} time for valid data transfers.

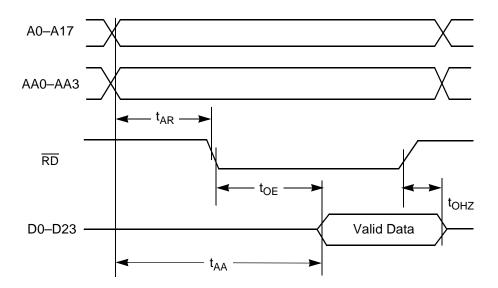


Figure 2-1. External Memory Bus Asynchronous Read Timing

2.2.2 DSP56303 External Memory Bus Asynchronous Write Timing

When writing to external asynchronous memory, the DSP56303 memory write access is controlled by the following steps.

- 1. The memory select address is asserted. The memory select address is created by combining the Address bus, A0–A17, and the Address Attributes AA0–AA3.
- 2. After a delay of t_{AS} —address valid to \overline{WR} assertion time—the Write enable signal, \overline{WR} , is asserted.
- 3. Before a delay of t_{WA} — \overline{WR} assertion to output data valid—the DSP places valid data on the data bus.



- 4. After a delay of t_{DW} —data valid to \overline{WR} deassertion (data setup time)—the DSP deasserts the \overline{WR} signal.
- 5. Then the DSP deasserts the address and address attributes after t_{WR} — \overline{WR} deassertion to address not valid—while holding the data valid for t_{DH} .

The data access time, t_{AW} (i.e., address and AA valid to \overline{WR} deassertion), is typically the critical timing specification for memory devices. The t_{AW} for a memory device must be less than or equal to the DSP's t_{AW} time for valid data transfers.

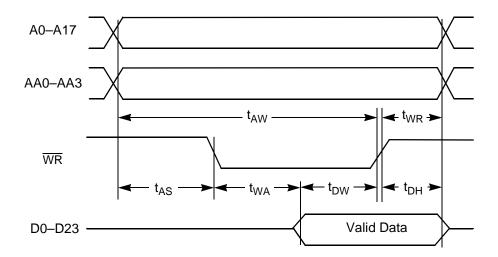


Figure 2-2. External Asynchronous Memory Bus Write Timing

2.3 DSP56303 Memory Control Registers

You must configure the following control registers to access external memory or peripherals properly when using the DSP56303:

- DSP PLL and Clock Generation register
- Bus Control Register
- DRAM Control Register (if DRAM is used)
- Address Attribute Registers



2.3.1 DSP PLL and Clock Generation

You must set the core speed of the DSP for optimum processor and memory performance by configuring the DSP PLL and Clock Generation in the PLL Control (PCTL) register. For detailed information on the PCTL register, see the PLL and Clock Generator chapter in the *DSP56300 24-Bit Digital Signal Processor Family Manual*. The PLL Control register is an X data I/O mapped 24-bit register. The PCTL register can be separated into four sub-functions:

- Frequency Predivider—The input clock frequency can be pre-divided before passing it to the PLL loop frequency multiplier. This frequency predivider has a programmable Division Factor range of 1 to 16. It is set by controlling the values placed in the PCTL register bits 20–23. The Division Factor is the binary value stored in bits 20–23, plus one.
- PLL Loop Frequency Multiplier—The clock frequency output from the predivider is multiplied by the voltage-controlled oscillator (VCO). The Multiplication Factor is set by the value in the PCTL register bits 0–11. The Multiplication Factor is the binary value stored in bits 0–11, plus one.
- Frequency Low-power Divider—The Low-power Divider (LPD) can divide the output frequency of the VCO before it is used by the DSP core. This frequency low power divider has a programmable Division Factor range of 1 to 128. It is set by controlling the values placed in the PCTL register bits 12–14. The low-power division factor is 2ⁿ, where n is the value in PCTL bits 12–14.
- Frequency Control Bits—The following five control bits control the input frequency source, the PLL during Stop mode, the activation of the PLL VCO, and the external availability of the core clock:
 - Crystal frequency is less than 200 kHz, Bit 15
 - Disable XTAL drive output, Bit 16
 - PLL runs during STOP mode, Bit 17
 - Enable PLL operation, Bit 18
 - Disable core clock output, Bit 19

The operating core frequency of the chip is set by the control bits in the PCTL register as follows:

$$F_{CORE} = \frac{F_{EXTAL} \times MF}{PDF \times DF}$$

where

- F_{CORE} is the DSP core frequency.
- F_{EXTAL} is the external input frequency source present on the EXTAL pin.
- PDF is the Predivider Factor defined by the PD0–PD3 bits in PCTL.
- MF is the PLL Multiplication Factor defined by the MF0–MF11 bits in PCTL.
- DF is the Division Factor defined by the DF0-DF2 bits in PCTL.



2.3.2 Bus Control Register (BCR)

The Bus Control Register (BCR) is a 24-bit X data I/O register that controls the external bus wait states generated for each Address Attribute area 0–3 and assigns a default value to all memory areas not covered by an Address Attribute area. Each area can have up to 31 wait states. Select the correct number of wait states for each memory configuration using this register.

- Wait states for Address Attribute area 0, allowing 0–31 wait states, Bits 0–4
- Wait states for Address Attribute area 1, allowing 0–31 wait states, Bits 5–9
- Wait states for Address Attribute area 2, allowing 0–7 wait states, Bits 10–12
- Wait states for Address Attribute area 3, allowing 0–7 wait states, Bits 13–15
- Wait states for address areas not specified by areas 0–3, allowing 0–31 wait states, Bits 16–20
- The bus state status, Bit 21
- Enable Bus Lock Hold, Bit 22
- Enable Bus Request Hold, Bit 23

2.3.3 Address Attribute Control Registers (AAR0–AAR3)

Four 24-bit Address Attribute Control registers in the X data I/O memory space control the activity of the AA0–AA3/RAS0–RAS3 pins. Each AA/RAS pin is asserted if the address and memory space of the appropriate AARx matches the requested external memory address and address space.

- Specify external memory access type; select from Synchronous SRAM, Asynchronous SRAM, and DRAM accesses, Bits 0–1
- Pull the AA pin high, Bit 2
- Activate the AA pin during external program space accesses, Bit 3
- Activate the AA pin during external X data space accesses, Bit 4
- Activate the AA pin during external Y data space accesses, Bit 5
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7
- Specify the number of address bits to compare, allowing the use of 0–12 address bits, Bits 8–11
- Specify the most significant portion of the address to compare, Bits 12–23



2.3.4 Operating Mode Register (OMR)

The Operating Mode Register (OMR) is a 24-bit I/O register that selects the operating mode of the DSP, external memory controls, and stack extension controls. The following flags are applicable to memory interfacing:

- The DSP operating mode is specified by MA–MD, Bits 0–3.
- The External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4.
- The Memory Switch mode bit reconfigures internal memory spaces, Bit 7.
- The Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11.
- The Bus Release Timing bit selects between a fast and slow bus release of the BB pin, Bit 12.
- The Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14.

2.3.5 Status Register (SR)

The Status Register (SR) is a 24-bit I/O register that selects and monitors the results of arithmetic computations and the current state of the DSP. The following flags are applicable to memory interfacing:

- Sixteen-bit Compatibility mode enables full compatibility with object code written for the DSP56000 family, Bit 13.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19.





3 32K \times 8-bit Memory Based Designs

Using one hardware memory design based on three $32K \times 8$ -bit 5.0 V memories, the DSP56303's Memory Expansion Port allows the $32K \times 24$ -bit memory bank to be logically configured for use in various memory space arrangements.

Configuring and using one Memory Expansion Port Address Attribute Control Register, the 32K memory bank can accommodate seven different memory space arrangements:

- 1. 32K × 24-bit 'P' Space Fast SRAM
- 2. $32K \times 24$ -bit 'X' Space Fast SRAM
- 3. $32K \times 24$ -bit 'Y' Space Fast SRAM
- 4. 32K × 24-bit 'P'/'X' Space Fast SRAM
- 5. 32K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. 32K × 24-bit 'X'/'Y' Space Fast SRAM
- 7. 32K × 24-bit 'P'/'X'/'Y' Space Fast SRAM

Configuring and using two Memory Expansion Port Address Attribute Control Registers, the 32K memory bank can accommodate thirteen different memory space arrangements:

- 1. $32K \times 24$ -bit 'P' Space Fast SRAM
- 2. 32K × 24-bit 'X' Space Fast SRAM
- 3. $32K \times 24$ -bit 'Y' Space Fast SRAM
- 4. $32K \times 24$ -bit 'P'/'X' Space Fast SRAM
- 5. 32K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. 32K × 24-bit 'X'/'Y' Space Fast SRAM
- 7. $32K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM
- 8. $16K \times 24$ -bit 'P'/'X' and $16K \times 24$ -bit 'Y' Space Fast SRAM
- 9. $16K \times 24$ -bit 'P'/'Y' and $16K \times 24$ -bit 'X' Space Fast SRAM
- 10. $16K \times 24$ -bit 'P' and $16K \times 24$ -bit 'X'/'Y' Space Fast SRAM
- 11. $16K \times 24$ -bit 'P' and $16K \times 24$ -bit 'X' Space Fast SRAM
- 12. 16K × 24-bit 'P' and 16K × 24-bit 'Y' Space Fast SRAM
- 13. $16K \times 24$ -bit 'X' and $16K \times 24$ -bit 'Y' Space Fast SRAM

All of these memory space configurations efficiently use the full capacity of the memory chips in the $32K \times 24$ -bit memory bank.

To illustrate these configurations, the remainder of this chapter presents examples based on one common hardware design using two Address Attribute Selectors that implement three of the most common configurations: $32K \times 24$ -bit 'P' Space Fast SRAM, $32K \times 24$ -bit 'P'/'X' Space Fast SRAM configuration, and a $16K \times 24$ -bit 'P'/'X' and $16K \times 24$ -bit 'Y' Space Fast SRAM configuration (see **Figure 3-4** for a schematic of the hardware design).



3.1 32K × 24-bit Common Fast SRAM Hardware Design

This section describes an asynchronous Fast SRAM $32K \times 24$ -bit memory bank implementation using Motorola's MCM6206D device. **Figure 3-1** displays the block diagram. Memory bank designs of this size and type can use very low-cost 5.0 V memory devices to satisfy the majority of embedded designs.

The memory bank design uses two Address Attribute Selectors. However, the $32K \times 24$ -bit 'P' Space and $32K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM configurations can use a one Address Attribute Selector design with A14 substituted for AA2. The two Address Attribute Selector design demonstrates the ultimate flexibility of the Address Attribute Selectors.

For this common hardware design, the DSP core runs at a maximum of 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements. This 5 V memory device is organized as $32K \times 8$ -bits. Therefore, three memory devices are used to achieve the 24-bit wide bus.

Since the DSP's data bus is not 5 V tolerant, level conversion to and from 3.3 V and 5 V is necessary on the 24-bit data to accommodate the 5 V memory devices. This is accomplished by using three Quality Semiconductor's QS3245 QuickSwitch® 8-bit bus switches. These switches allow the connection of a 3.3 V CMOS logic DSP data bus on one side and 5 V TTL-compatible logic, memory devices on the other side, effectively providing a 3.3 V-to-5 V level conversion without adding any significant (0.25 nS) propagation delay.

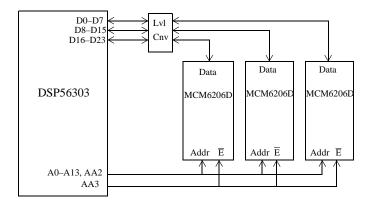


Figure 3-1. 32K × 24-bit Fast SRAM Memory Example

3.1.1 MCM6206D-12 Memory Timing Requirements

For the asynchronous Fast SRAM device to work properly, its timing requirements must be met. Following are the timing requirements for the MCM6206D-12 $32K \times 8$ -bit 12 nS Fast SRAM.

3.1.1.1 MCM6206D-12 Read Cycle Timing

Table 3-1 shows the memory read timing specification values in the memory read cycle timing diagram, **Figure 3-2.**



Read Cycle Parameter	Symbol	Min	Max
Read Cycle Time	t _{AVAV}	12 nS	_
Address Access Time	t _{AVQV}	_	12 nS
Enable Access Time	t _{ELQV}	_	12 nS
Output Enable Access Time	t _{GLQV}	_	6 nS
Enable High to Output High-Z	t _{EHQZ}	0 nS	7 nS
Output Enable High to Output High-Z	t _{GHQZ}	0 nS	6 nS

Table 3-1. MCM6206D-12 Memory Read Timing Specifications

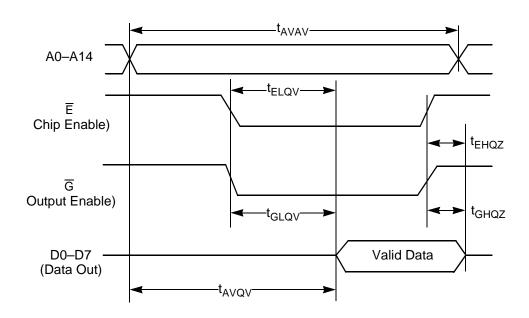


Figure 3-2. MCM6206D-12 Memory Read Cycle Timing Diagram

3.1.1.2 MCM6206D-12 Write Cycle Timing

Table 3-2 shows the memory write timing specification values in the memory write cycle timing diagram, **Figure 3-3**.



Write Cycle Parameter	Symbol	Min	Max
Write Cycle Time	t _{AVAV}	12 nS	_
Address Setup Time	t _{AVWL}	0 nS	_
Address Valid to End off Write	t _{AVWH}	10 nS	_
Write Pulse Width	t _{WLWH}	10 nS	_
Data Valid to End of Write	t _{DVWH}	6 nS	_
Data Hold Time	t _{WHDX}	0 nS	_
Write Recovery Time	t _{WHAX}	0 nS	_

Table 3-2. MCM6206D-12 Memory Write Timing Specifications

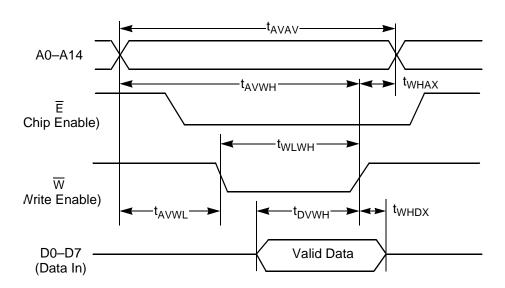


Figure 3-3. MCM6206D-12 Memory Write Cycle Timing Diagram

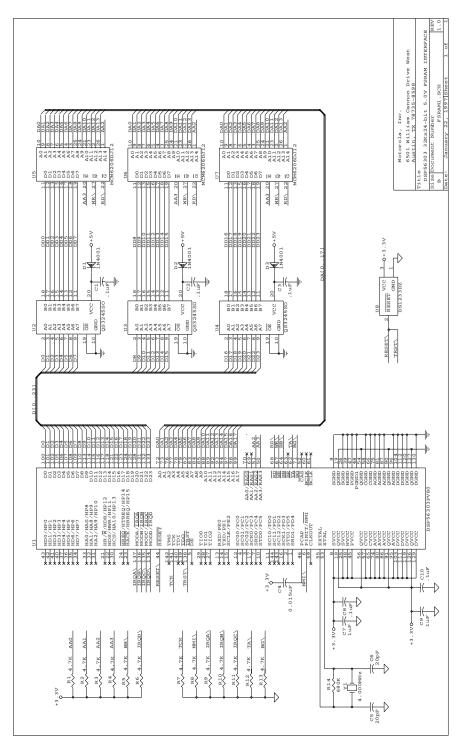


Figure 3-4. 32K × 24-bit Fast SRAM Schematic



3.2 32K × 24-bit 'P' Space Fast SRAM Example

This section describes a $32K \times 24$ -bit 'P' memory space, asynchronous Fast SRAM implementation using Motorola's MCM6206D device. **Figure 3-5** shows the memory map layout, **Figure 3-1** shows the block diagram, and **Example 3-1** shows the example code.

The DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

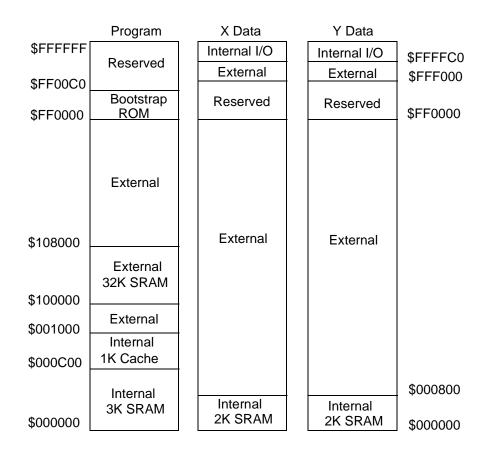


Figure 3-5. 32K × 24-bit 'P' Space Fast SRAM Memory Map

3.2.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $32K \times 24$ -bit 'P' space memory configuration, set up the following DSP control registers.



Set up the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation (PCTL) register. For this example, the DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, bits 20-23 = \$0
- Low-power Divider value = 1, bits 12-14 = \$0
- VCO Multiplication value = 20, bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 3 enables, via Fast SRAM E, external 32K SRAM bank accesses in the address range from \$100000 to \$107FFF during program space requests. Configure the memory address space requirements for the Address Attribute Pin 3 using Address Attribute Register 3 (AAR3). The AAR3 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 0.
- Activate the AA pin during external Y data space accesses, Bit 5 = 0.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0
- Specify the number of address bits to compare, Bits 8-11 = \$9
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR3 is \$100909.



Address Attribute Pin 2 selects, via Fast SRAM A14, address line A14 in the external 32K SRAM bank during accesses in the address range from \$100000 to \$107FFF during program space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 0.
- Activate the AA pin during external Y data space accesses Bit 5 = 0.
- Move the eight least significant bits of address to eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = A
- Specify the most significant portion of the address to compare, Bits 12-23 = \$104

The value loaded into the AAR2 is \$104A0D.

The value loaded into AAR0 and AAR1 is \$000000.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address Attribute area 0 wait states, Bits 0-4 = \$0
- Address Attribute area 1 wait states, Bits 5-9 = \$0
- Address Attribute area 2 wait states, Bits 10–12 = \$1
- Address Attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16-20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 =0

The value loaded into the BCR is \$002400.



Configure the operating mode and external memory controls using the Operating Mode Register. The OMR value combines the following bits for each feature:

- MA–MD bits select the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14 = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 3-1. 32K × 24-bit 'P' Space Fast SRAM Memory Exercise Program

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:13 asram1.asm

```
1
                                                  132,60,3,3,
                                         page
2
3
                               ASRAM1.ASM - Simple program to test 32Kx24-bits of
                                               program memory using a DSP56303
5
6
                               The program uses Internal P:RAM to test External P:RAM
                                               from $100000 - $107FFF @ 1w/s
8
        100000
                                                  $100000
10
                         PMemStart
                                          equ
        108000
                         PMemEnd
11
                                                  $108000
                                          equ
                                                  PMemEnd-PMemStart
12
        008000
                         PMemSize
                                          equ
13
14
                         ;--- Program Specific Storage Locations (X DATA SPACE)
15
                         MEM_FAIL_ADDRESS
        000000
                                                  $000000
16
                                          eau
17
                         MEM_FAIL_WROTE
18
        000001
                                                  $000001
                                          equ
                         MEM_FAIL_READ
19
        000002
                                                  $000002
20
                                          eau
                         MEM_PASS_COUNTER
21
2.2
        000003
                                          equ
                                                  $000003
23
                              ;--- DSP56303 Control Registers (X I/O SPACE)
24
25
                                                  $FFFFFB
                         BCR
        STATA
                                          equ
                                                                  ; Bus Control Register
26
        ОЧЧЧЧЧ
                         PCTL
                                          equ
                                                  SEFFFFD
                                                                   ; PLL Control Register
        844444
                                                                   ; Address Attribute Register #3
27
                         AAR3
                                          equ
                                                  SEFFFF6
28
        FFFFF7
                         AAR2
                                          equ
                                                  $FFFFF7
                                                                   ; Address Attribute Register #2
29
30
                              ;--- PCTL value = 0x0E0013
31
        000000
                         prediv
                                          equ
                                                  0
                                                                   ; Pre-Divider = 1
32
        000000
                         lowdiv
                                                  Ω
                                                                   ; Low Power Divider = 1
                                          equ
33
        000013
                         pllmul
                                                  19
                                                          ; VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
                                          equ
34
        000000
                         crystal
                                                  0
                                                                  ; No, Crystal not less than 200kHz
                                          equ
35
        000000
                         disXTAL
                                                  Ω
                                                                   ; No, do not disable crystal use
                                          equ
36
        020000
                         pllstop
                                                  $020000
                                                                   ; Yes, PLL runs during STOP
                                          equ
37
        040000
                         enpll
                                          equ
                                                  $040000
                                                                   ; Yes, enable PLL operation
38
        080000
                         disclk
                                          equ
                                                  $080000
                                                                   ; Yes, disable CORE clock output
39
        0E0013
                                         equ rediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
                         PCTL_value
40
41
                              ;--- AAR3 value = 0x100909
42
        000001
                         acctype3
                                                             ; External Memory access type = 0x1
                                         equ
        000000
43
                         aahigh3
                                                  0
                                                              Enable AA3 pin to be low when selected
                                          equ
                                                              Yes, Enable AA3 pin on ext 'P' accesses
                         aap3
44
        800000
                                          eau
                                                  $8
                                                             No, Enable AA3 pin on ext 'Y' accesses; No, Enable AA3 pin on ext 'Y' accesses
45
        000000
                         aax3
                                                  Ó
                                          equ
        000000
46
                         aav3
                                         equ
                                                  0
47
        000000
                                                  0
                                                               No, Enable address bus swap
                         aswap3
                                          eau
48
        000000
                         enpack3
                                                              No, Enable packing/unpacking logic
                                                  0
                                          equ
        000900
                                                  $000900
                                                               Compare 9 address bits
49
                         nadd3
                                          equ
                                                              Most significant portion of address, $100000 - $107fff, to compare.
50
        100000
                         msadd3
                                                  $100000
                                          equ
51
                                                               (0001,0000,0xxx,xxxx,xxxx)
52
53
        100909
                          AAR3_value equ acctype3+aahigh3+aap3+aax3+aay3+aswap3+enpack3+nadd3+msadd3
54
                              ;--- AAR2 value = 0x104A0D
55
        000001
                                                             ; External Memory access type = 0x1
56
                         acctype2
                                          equ
                                                  1
        000004
57
                         aahigh2
                                                  $4
                                                             ; Enable AA2 pin to be high when selected
                                          equ
                                                             ; Yes, Enable AA2 pin on ext 'P' accesses ; No, Enable AA2 pin on ext 'X' accesses ; No, Enable AA2 pin on ext 'Y' accesses
        800000
58
                         aap2
                                          equ
                                                  $8
59
        000000
                         aax2
                                          equ
                                                  Ω
60
        000000
                         aay2
                                          equ
                                                  0
                                                              No, Enable address bus swap
61
        000000
                         aswap2
                                          equ
                                                  0
62
        000000
                         enpack2
                                                  Ω
                                                              No, Enable packing/unpacking logic
                                          equ
63
        000A00
                         nadd2
                                                  $000A00
                                                               Compare 10 address bits
                                          equ
64
        104000
                         msadd2
                                                  $104000
                                                              Most significant portion of address,
                                          equ
65
                                                               $104000 - $107fff, to compare.
                                                             ; (0001,0000,01xx,xxxx,xxxx,xxxx)
66
67
        104A0D
                    AAR2_value equ acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
68
69
                              i --- BCR value = 0x002400
70
        000000
                         aaa0ws
                                          equ
                                                  0
                                                                   ; Address Attribute Area 0 \text{ w/s} = 0
71
        000000
                                                  0
                                                                   ; Address Attribute Area 1 w/s = 0
                         aaa1ws
                                          equ
72
        000400
                         aaa2ws
                                                  $000400
                                                                   ; Address Attribute Area 2 w/s = 0
                                          eau
73
        002000
                         aaa3ws
                                                  $002000
                                                                   ; Address Attribute Area 3 w/s = 1
                                          equ
                                                                   ; Default Address Area w/s = 0
74
        000000
                         defws
                                          equ
```



```
75
       000000
                      busss
                                      equ
                                                            ; Bus state status = 0
76
       000000
                      enblh
                                                            ; Enable Bus Lock Hold = 0
                                      equ
77
       000000
                      enbrh
                                                            ; Enable Bus Request Hold = 0
                                      eau
78
       002400
                      BCR_value
                                             aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
                                     equ
79
80
       P:000100
                      org p:$100
81
                                                            ; Keep the program in internal RAM
82
83
                      memtst
84
85
                          ; Initialization Section
       P:000100 08F4BD
                                     #PCTL_value,x:PCTL
                                                            ; Set PLL Control Register
86
                              movep
                 0E0013
87
       P:000102
                 05F43A
                                      #$004000.OMR
                              movec
                                                            ; Disable Address Attribute Priority
                 004000
       P:000104
88
                 05F439
                              movec
                                      #$080000,SR
                                                            ; Enable 1K Cache
                 080000
89
       P:000106
                 08F4BB
                              movep
                                      #BCR_value,x:BCR
                                                            ; Set external wait states
                 002400
90
       P:000108
                 08F4B7
                                      #AAR2_value,x:AAR2
                                                            ; Set Address Attribute Reg2
                              movep
                 104A0D
91
       P:00010A
                 08F4B6
                              movep
                                      #AAR3_value,x:AAR3
                                                            ; Set Address Attribute Reg3
                 100909
93
       P:00010C 05F420
                                      #-1,m0
                                                            ; Set LINEAR addressing mode
                              move
                 FFFFFF
94
       P:00010E
                                      #-1,m3
                              move
                 FFFFFF
96
       P:000110
                 20001B
                              clr
       P:000111
                 000000
                              nop
98
       P:000112
                 570000
                              move
                                      b,x:MEM_FAIL_ADDRESS
                                                            ; Initialize Failed Address -> $000000
                                      b,x:MEM_FAIL_WROTE
                                                            ; Initialize Expected Data -> $000000
99
       P:000113
                 570100
                              move
100
       P:000114
                 570200
                                      b,x:MEM_FAIL_READ
                                                            ; Initialize Data Read -> $000000
                              move
101
       P:000115 570300
                                      b,x:MEM PASS COUNTER ; Initialize Pass Counter -> $000000
                              move
102
103
                           main
104
                           ;--- fill P:memory with initial pattern
105
106
       P:000116 63F400
                                      107
                              move
                 000138
108
       P:000118 000000
                              nop
       P:000119 000000
109
                              nop
110
111
       P:00011A 07DB84
                              move
                                      p:(r3)+,x0
                                                            ; Get the Write Pattern for P:MEM
112
113
       P:00011B 70F400
                              move
                                      #PMemSize,n0
                                                            ; Get memory size
                 008000
114
115
       P:00011D 60F400
                                      #PMemStart,r0
                                                            ; Get starting address for fill
                              move
                 100000
116
117
       P:00011F 06D820
                              rep
                                                            ; Fill RAM with first pattern data
118
       P:000120 075884
                                     x0,p:(r0)+
                              move
119
120
121
                      ;--- Check for expected pattern data in each RAM location ---
122
                       ;--- and then replace with a new data pattern.
                      :-- ... This provides an address check. Since erroneous --- :-- ... addressing will cause the data to be written into ---
123
124
125
                       ;--- ...incorrect locations and this will be evident in
                      ;--- ...the next read pass.
126
127
       P:000121 063890
                                     #PATTN,test_Pm
128
                             DOR
                                                           ; Start Pattern Test Loop
                 00000D
129
       P:000123
                 60F400
                              move
                                      #PMemStart,r0
                                                            ; Get starting address of Test Memory
                 100000
       P:000125
                                      x0,a
130
                 200041
                              t.fr
                                                            ; Save the last test pattern -> a
       P:000126 07DB84
                                     P:(r3)+,x0
131
                                                            ; Get the next test pattern -> x0
                              move
132
133
                                                     ; Test this pattern through external RAM
       P:000127 06D810
                              DOR
                                     n0,next_loc
134
                                                            ; Test all external RAM locations
                 000006
135
       P:000129
                 07E085
                              move
                                      P:(r0),x1
                                                            ; Read RAM location
                                     x1,a
136
       P:00012A
                 200065
                              cmp
                                                            ; Read data = last test pattern?
                                                            ; No, error if compare fails
137
       P:00012B 052409
                                      <ERR
                              bne
138
       P:00012C 075884
                              move
                                      x0,P:(r0)+
                                                            ; Yes, Write next test pattern -> RAM
139
       P:00012D 000000
                              nop
140
                      next_loc
141
```



```
142
       P:00012E 000000
                               nop
                                                                 ; Time to start next test pattern
143
144
                        test_Pm
145
                                                        ; One Pass Complete
146
                                ; All test patterns have been tried and passed in external RAM
147
148
       P:00012F 518300
                                        x:MEM_PASS_COUNTER,b0
                                move
149
       P:000130 000009
                                inc
                                                                 ; Update pass loop counter
       P:000131
                 000000
150
                                nop
151
                                        b0,x:MEM_PASS_COUNTER
       P:000132 510300
                                move
152
153
       P:000133 050FC3
                                bra
                                        main
                                                                 ; Do it all over again
154
155
156
                                                ; ERR--handles error messaging with user
157
158
                                                ; Expected Data --> a
                                                ; Read Data --> x1
; Address of failure --> r0
159
160
161
162
163
       P:000134 600000
                                move
                                        r0,x:MEM_FAIL_ADDRESS ; Save off address of failure
                                                             ; Save off expected data
164
       P:000135 560100
                                move
                                        a,x:MEM_FAIL_WROTE
165
       P:000136 450200
                                        x1,x:MEM_FAIL_READ
                                                                ; Save off data read
                                move
166
167
       P:000137 050C00
                                                                 ; Dynamically HALT here
                                bra
168
169
170
                                        ; Memory Test Patterns
171
172
       P:000138 PATT
                                                $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
                                                173
       P:00013D
                                        dc
174
       P:000141
                                        dc
175
       P:000145
                                        dc
176
       P:000149
                                        dc
177
       P:00014D
                                        dc
178
       P:000151
                                        dc
179
       P:000155
                                        dc
180
       P:000159
                                        dc
       P:00015D
                                                $FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
$FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
181
                                        da
       P:000161
182
                                        dc
       P:000165
                                                $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
$FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
183
                                        dc
       P:000169
184
                                        da
185
       P:00016D
                                        dc
                                                $FEDCBA, $123456, $012345, $EDCBA9
186
187
                  000038
                                PATTN
                                       equ
                                                *-PATT-1
188
189
                                        end
                                                memtst
```

⁰ Errors

⁰ Warnings



3.3 32K × 24-bit 'P'/'X'/'Y' Fast SRAM Example

This section describes a $32K \times 24$ -bit shared 'P'/'X'/'Y' memory space, asynchronous Fast SRAM implementation using Motorola's MCM6206D device. **Figure 3-7** shows the memory map layout, **Figure 3-1** shows the block diagram, and **Example 3-2** shows the example code. A shared memory space means that data written to one address in one memory space can be accessed by the same address in another memory space, (e.g., writing \$012345 to P:\$100000 could be read by X:\$100000, Y:\$100000 or P:\$100000).

The DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

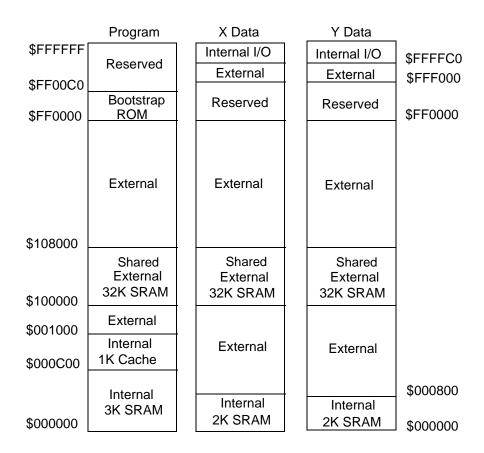


Figure 3-6. 32K × 24-bit 'P'/'X'/'Y' Fast SRAM Memory Map

3.3.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $32K \times 24$ -bit 'P'/'X'/'Y' space memory configuration, set up the following DSP control registers.



You must set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation (PCTL) register. For this example, the DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, Bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 3 enables, via Fast SRAM E, external 32K SRAM bank accesses in the address range from \$100000 to \$107FFF during program space requests. Configure the memory address space requirements for the Address Attribute Pin 3 using Address Attribute Register 3 (AAR3). The AAR3 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$9
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR3 is \$100939.



Address Attribute Pin 2 selects, via Fast SRAM A14, address line A14 in the external 32K SRAM bank during accesses in the address range from \$104000 to \$107FFF during program space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using the Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, AAR Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = A
- Specify the most significant portion of the address to compare, Bits 12-23 = \$104

The value loaded into the AAR2 is \$104A3D.

The value loaded into AAR0 and AAR1 is \$000000.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address Attribute area 0 wait states, Bits 0-4 = \$0
- Address Attribute area 1 wait states, Bits 5-9 = \$0
- Address Attribute area 2 wait states, Bits 10–12 = \$1
- Address Attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16-20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 = 0

The value loaded into the BCR is \$002400.



Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value is combines the following bits for each feature:

- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14 = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 3-2. 32K × 24-bit 'P'/'X'/'Y' Space Fast SRAM Memory Exercise Program

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:18 asram2.asm

```
1
                                                  132,60,3,3,
                                         page
2
3
                                       ASRAM2.ASM - Simple program to test 32Kx24-bits of
4
                                                       Program/X-Data/Y-Data memory using a DSP56303
5
                                       The program uses Internal P:RAM to test External P/X/Y:RAM
                                                       from $100000 - $107FFF @ 1w/s
8
10
        100000
                                                 $100000
                        PMemStart
                                         equ
11
        108000
                        PMemEnd
                                                 $108000
                                         equ
12
        008000
                        PMemSize
                                                 PMemEnd-PMemStart
                                         equ
13
14
                         :--- Program Specific Storage Locations (X DATA SPACE)
                        MEM_FAIL_ADDRESS
15
        000000
                                                 $000000
16
                                         equ
                        MEM FAIL WROTE
17
        000001
18
                                                 $000001
                                         equ
                        MEM_FAIL_READ
19
        000002
2.0
                                                 $000002
                                         equ
                        MEM PASS COUNTER
21
22
        000003
                                                 $000003
                                         eau
23
                        ;--- DSP56303 Control Registers (X I/O SPACE)
24
                                                                ; Bus Control Register
25
        STATA
                        BCR
                                         equ
                                                 SFFFFFB
26
        Сяяяяя
                        PCTL
                                                 SEFFEED
                                                                  ; PLL Control Register
                                         equ
27
        FFFFF6
                        AAR3
                                                 $FFFFF6
                                                                  ; Address Attribute Register #3
2.8
        FFFFF7
                        AAR 2
                                         equ
                                                 $FFFFF7
                                                                  ; Address Attribute Register #2
29
30
                        ;--- PCTL value = 0x0E0013
                        prediv
31
        000000
                                         equ
                                                                  ; Pre-Divider = 1
32
        000000
                        lowdiv
                                                 0
                                                                  ; Low Power Divider = 1
                                         equ
                        pllmul
33
        000013
                                                 19
                                                          ; VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
                                         equ
34
        000000
                                                 0
                                                                  ; No, Crystal not less than 200kHz
                        crystal
                                         equ
35
        000000
                        disXTAL
                                                 0
                                                                  ; No, do not disable crystal use
                                         equ
36
        020000
                        pllstop
                                         equ
                                                 $020000
                                                                  ; Yes, PLL runs during STOP
37
        040000
                        enpll
                                         equ
                                                 $040000
                                                                  ; Yes, enable PLL operation
38
        080000
                                                 $080000
                                                                  ; Yes, disable CORE clock output
                        disclk
                                         equ
39
        0E0013
                         PCTL_value
                                                 prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
                                         equ
40
41
                        i --- AAR3 value = 0x100939
42
        000001
                                                          ; External Memory access type = 0x1
                    acctype3
                                     equ 1
        000000
                                                           Enable AA3 pin to be low when selected
43
                    aahigh3
                                     equ 0
44
        800000
                                                            Yes, Enable AA3 pin on ext 'P' accesses
                     aap3
                                     equ $8
                                                          ; Yes, Enable AA3 pin on ext 'X' accesses
45
        000010
                    aax3
                                     equ $10
46
        000020
                    aay3
                                     equ $20
                                                           Yes, Enable AA3 pin on ext 'Y' accesses
47
                                                          ; No, Enable address bus swap
        000000
                    aswap3
                                     equ 0
48
        000000
                     enpack3
                                     equ 0
                                                            No, Enable packing/unpacking logic
                    nadd3
49
        000900
                                     equ $000900
                                                            Compare 9 address bits
                                                           Most significant portion of address, $100000 - $107fff, to compare.
50
        100000
                    msadd3
                                     equ $100000
51
                                     ; (0001,0000,0xxx,xxxxx,xxxxx)
equ acctype3+aahigh3+aap3+aax3+aay3+aswap3+enpack3+nadd3+msadd3
52
53
        100939
                    AAR3_value
54
                     ;--- AAR2 value = 0x104A3D
55
        000001
                    acctype2
                                                          ; External Memory access type = 0x1
56
                                     eau
                                             1
57
        000004
                                                          ; Enable AA2 pin to be high when selected
                    aahigh2
                                     equ
                                             $4
                                                          Yes, Enable AA2 pin on ext 'P' accesses; Yes, Enable AA2 pin on ext 'X' accesses
        000008
58
                     aap2
                                     equ
                                             $8
        000010
59
                     aax2
                                     equ
                                             $10
                                                          ; Yes, Enable AA2 pin on ext 'Y' accesses
        000020
60
                     aay2
                                     equ
                                             $20
        000000
61
                     aswap2
                                     equ
                                             Λ
                                                          ; No, Enable address bus swap
        000000
                     enpack2
                                                           No, Enable packing/unpacking logic
62
                                     equ
                                             0
                                             $000A00
63
        00A00
                    nadd2
                                     equ
                                                           Compare 10 address bits
                                                            Most significant portion of address,
64
        104000
                    msadd2
                                             $104000
                                     equ
65
                                                            $104000 - $107fff, to compare.
                                                           (0001,0000,01xx,xxxx,xxxx,xxxx)
66
67
        104A3D
                    AAR2_value
                                     equ acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
68
69
                     i--- BCR value = 0 \times 002400
70
        000000
                    aaa0ws
                                     equ
                                                          ; Address Attribute Area 0 w/s = 0
71
        000000
                     aaa1ws
                                     equ
                                                          ; Address Attribute Area 1 w/s = 0
72
        000400
                                             $000400
                     aaa2ws
                                     equ
                                                          ; Address Attribute Area 2 w/s = 1
        002000
                    aaa3ws
                                     equ
                                             $002000
                                                          ; Address Attribute Area 3 w/s = 1
```



```
74
       000000
                      defws
                                     equ
                                                            ; Default Address Area w/s = 0
75
       000000
                                             0
                                                           ; Bus state status = 0
                      busss
                                     equ
76
                                             0 ; Enable Bus Lock Hold = 0
0 ; Enable Bus Request Hold = 0
       000000
                      enblh
                                     eau
77
       000000
                      enbrh
                                     equ
78
                      BCR_value
                                           aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
       002400
                                     equ
79
80
       P:000100
                                     org p:$100 ;Keep the program in internal RAM
81
82
83
                      memtst
84
85
                      ; Initialization Section
       P:000100 08F4BD
                                     #PCTL_value,x:PCTL; Set PLL Control Register
86
                              movep
                 0E0013
87
       P:000102 05F43A
                              movec #$004000,OMR
                                                            ; Disable Address Attribute Priority
                 004000
       P:000104 05F439
88
                              movec #$080000,SR
                                                            ; Enable 1K Cache
                 080000
       P:000106 08F4BB
89
                                     #BCR_value,x:BCR
                                                            ; Set external wait states
                              movep
                 002400
90
       P:000108 08F4B7
                                                            ; Set Address Attribute Reg2
                              movep
                                     #AAR2 value,x:AAR2
                 104A3D
91
       P:00010A 08F4B6
                              movep
                                     #AAR3_value,x:AAR3
                                                            ; Set Address Attribute Reg3
                 100939
92
       P:00010C 05F420
                                                            ; Set LINEAR addressing mode
93
                              move
                                     #-1,m0
                 777777
       P:00010E 05F423
94
                              move
                                     #-1,m3
                 777777
95
96
       P:000110 20001B
                             clrb
97
       P:000111 000000
                              nop
       P:000112 570000
                                     b,x:MEM_FAIL_ADDRESS ; Initialize Failed Address -> $000000
98
                              move
                                    b,x:MEM_FAIL_WROTE ; Initialize Expected Data -> $000000
b,x:MEM_FAIL_READ ; Initialize Data Read -> $000000
99
       P:000113 570100
                             move
100
       P:000114 570200
                             move
       P:000115 570300
                                    b,x:MEM_PASS_COUNTER ; Initialize Pass Counter -> $000000
101
                              move
102
103
                      main
104
105
                       ;--- fill P:memory with initial pattern
106
                      ;-----
       P:000116 63F400
                             move #PATT,r3 ; r3 points to Test Patterns
107
                 000138
       P:000118 000000
108
                             nop
109
       P:000119 000000
                             nop
110
       P:00011A 07DB84
111
                              move
                                     p:(r3)+,x0
                                                    ; Get the Write Pattern for P:MEM
112
                                     #PMemSize,n0 ; Get memory size
113
       P:00011B 70F400
                             move
                 008000
114
115
       P:00011D 60F400
                             move
                                    #PMemStart,r0 ; Get starting address for fill
                 100000
116
117
       P:00011F 06D820
                                                    ; Fill RAM with first pattern data
                              rep
118
       P:000120 075884
                                     x0,p:(r0)+
                             move
119
120
121
                       ;--- Check for expected pattern data in each RAM location ---
                       ;--- and then replace with a new data pattern.
122
123
                       i--- ... This provides an address check. Since erroneous
124
                       ;--- ...addressing will cause the data to be written into ---
                       ;--- ...incorrect locations and this will be evident in ---
125
126
                       ;--- ...the next read pass.
127
       P:000121 063890
                             DOR #PATTN,test_Pm ; Start Pattern Test Loop
128
                 00000D
       P:000123 60F400
129
                             move #PMemStart,r0 ; Get starting address of Test Memory
                 100000
130
       P:000125 200041
                                                   ; Save the last test pattern -> a
                             tfr
                                     x0.a
       P:000126 07DB84
                                     P:(r3)+,x0; Get the next test pattern -> x0
131
                             move
132
                      ; Test this pattern through external RAM
133
       P:000127 06D810
134
                              DOR
                                     n0,next_loc ; Test all external RAM locations
                 000006
135
       P:000129 07E085
                              move
                                     P:(r0),x1
                                                    ; Read RAM location
                                     x1,a ; Read data = last test pattern? <ERR ; No, error if compare fails
136
       P:00012A 200065
                             cmp
                             bne
137
       P:00012B 052409
                                     x0,P:(r0)+ ; Yes, Write next test pattern -> RAM
138
       P:00012C 075884
                             move
139
       P:00012D 000000
                              nop
                      next_loc
140
```



0

Warnings

```
141
142
        P:00012E 000000
                                                            ; Time to start next test pattern
                                nop
143
144
                         test_Pm
145
146
                                           One Pass Complete
147
                         ; All test patterns have been tried and passed in external RAM
        P:00012F 518300
                                 move x:MEM_PASS_COUNTER,b0
148
        P:000130 000009
                                                           ; Update pass loop counter
149
                                  inc
                                          b
150
        P:000131 000000
                                 nop
        P:000132 510300
                                         b0,x:MEM_PASS_COUNTER
151
                                  move
152
        P:000133 050FC3
153
                                 bra
                                          main
                                                           ; Do it all over again
154
155
    ;-----
156
                                 ; ERR -- handles error messaging with user
157
158
                                  ; Expected Data --> a
159
                                  ; Read Data --> x1
                                  ; Address of failure --> r0
160
161
162
                                  ERR
163
        P:000134 600000
                                  move
                                          r0,x:MEM_FAIL_ADDRESS ; Save off address of failure
                                          a,x:MEM_FAIL_WROTE ; Save off expected data x1,x:MEM_FAIL_READ ; Save off data read
164
        P:000135 560100
                                  move
165
        P:000136 450200
                                  move
166
167
        P:000137 050C00
                                  bra
                                                                    ; Dynamically HALT here
168
169
170
                                  ; Memory Test Patterns
171
                                                   $000000,$FFFFFFF,$AAAAAA,$555555,$2BAD2C
$800000,$400000,$200000,$100000
$080000,$040000,$020000,$010000
$008000,$004000,$002000,$001000
172
        P:000138
                                  PATT
                                          dc
173
        P:00013D
                                          dc
174
        P:000141
                                          dc
175
        P:000145
                                          dc
                                                   $000800,$000400,$000200,$000100
$000080,$000040,$000020,$000010
176
        P:000149
                                          dc
177
        P:00014D
                                          dc
178
        P:000151
                                                   $000008,$000004,$000002,$000001
                                          dc
179
                                                   $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
        P:000155
                                          dc
                                                   $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
$FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
180
        P:000159
                                          dc
        P:00015D
181
                                          dc
                                                   $FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
$FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
        P:000161
182
                                          dc
        P:000165
183
                                          dc
184
                                                   $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
        P:000169
                                          dc
185
        P:00016D
                                          dc
                                                   $FEDCBA, $123456, $012345, $EDCBA9
186
        000038
                                  PATTN
187
                                          equ
                                                   *-PATT-1
188
189
                                          end
                                                   memtst
0
     Errors
```



3.4 $16K \times 24$ -bit 'P'/'X' and $16K \times 24$ -bit 'Y' Fast SRAM Example

This section describes a $16K \times 24$ -bit shared 'P'/'X' and $16K \times 24$ -bit 'Y' memory space, asynchronous Fast SRAM implementation using Motorola's MCM6206D device. **Figure 3-9** shows the memory map layout; **Figure 3-1** shows the block diagram; and **Example 3-3** shows the example code.

The DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

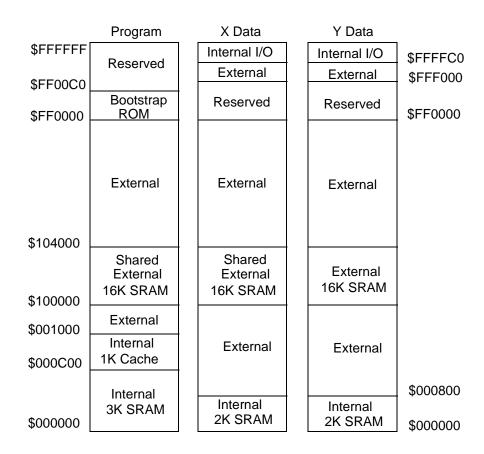


Figure 3-7. $16K \times 24$ -bit 'P'/'X' and $16K \times 24$ -bit 'Y' Memory Map

3.4.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $16K \times 24$ -bit 'P'/'X' and $16K \times 24$ -bit 'Y' space memory configuration, set up the following DSP control registers.



Set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation register (PCTL). For this example, the DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- External Frequency = 4.000 MHz
- Predivider value = 1, Bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 3 enables, via Fast SRAM E, external 32K SRAM bank accesses in the address range from \$100000 to \$103FFF during program, X data and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 3 using Address Attribute Register 3 (AAR3). The AAR3 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8–11 = \$A
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR3 is \$100A39.



Address Attribute Pin 2 selects, via Fast SRAM A14, address line A14 in the external 32K SRAM bank during accesses in the address range from \$100000 to \$103FFF to differentiate program/X data space requests from Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using the Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 0.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = A
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR2 is \$100A1D.

The value loaded into AAR0 and AAR1 is \$000000.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address Attribute area 0 wait states, Bits 0-4 = \$0
- Address Attribute area 1 wait states, Bits 5-9 = \$0
- Address Attribute area 2 wait states, Bits 0–12]= \$1
- Address Attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16-20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 = 0

The value loaded into the BCR regiser is \$002400



Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value combines the following bits for each feature:

- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14. = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 3-3. 16K 'P'/'X' and 16K 'Y' Space Fast SRAM Memory Exercise Program

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:22 asram3.asm

```
1
                                                132,60,3,3,
                                       page
2
3
                            ;
                                  ASRAM3.ASM
                                              - Simple program to test 16Kx24-bits of Program/X-Data
                                                 and 16Kx24-bits of Y-Data memory using a DSP56303
5
6
                                  The program uses Internal P:RAM to test External P/X & Y:RAM
                                                 from $100000 - $103FFF @ 1w/s
8
        100000
10
                        PMemStart
                                                $100000
                                        equ
        104000
11
                        PMemEnd
                                                $104000
                                        equ
12
        004000
                        PMemSize
                                                PMemEnd-PMemStart
                                        equ
13
14
                        ; --- Program Specific Storage Locations (X DATA SPACE)
15
                        P_MEM_FAIL_ADDRESS
        000000
                                                $000000
16
                                        eau
17
                        P_MEM_FAIL_WROTE
18
        000001
                                                $000001
                                        equ
19
                        P_MEM_FAIL_READ
        000002
                                                $000002
20
                                        eau
21
2.2
                        Y_MEM_FAIL_ADDRESS
23
        000003
                                                $00003
                        Y_MEM_FAIL_WROTE
24
        000004
25
                                                $000004
2.6
                        Y_MEM_FAIL_READ
        000005
2.7
                                                $000005
28
29
                        MEM_PASS_COUNTER
30
        000006
                                                $000006
31
32
                        ;--- DSP56303 Control Registers (X I/O SPACE)
33
        FFFFFB
                        BCR
                                                $FFFFFB
                                                               ; Bus Control Register
                                        equ
34
        FFFFFD
                        PCTL
                                                $FFFFFD
                                                                ; PLL Control Register
                                        equ
35
        FFFFF6
                        AAR3
                                                $FFFFF6
                                                                ; Address Attribute Register #3
                                        equ
36
        FFFFF7
                        AAR2
                                                $FFFFF7
                                                                ; Address Attribute Register #2
                                        equ
37
38
                        ;--- PCTL value = 0x0E0013
39
        000000
                        prediv
                                                                ; Pre-Divider = 1
                                        equ
                                                0
        00000
                                                                ; Low Power Divider = 1
40
                        lowdiv
                                                0
                                        equ
        000013
                                                19
                                                        ; VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
41
                        pllmul
                                        equ
42
        000000
                        crystal
                                                0
                                                               ; No, Crystal not less than 200kHz
                                        equ
43
        000000
                        disXTAL
                                                0
                                                                ; No, do not disable crystal use
                                        equ
                                                                ; Yes, PLL runs during STOP
                                                $020000
44
        020000
                       pllstop
                                        eau
45
        040000
                        enpll
                                                $040000
                                                                ; Yes, enable PLL operation
                                        equ
        080000
                        disclk
                                                $080000
                                                                ; Yes, disable CORE clock output
46
                                        equ
47
        0E0013
                        PCTL_value
                                                prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
                                        equ
48
                        ;--- AAR3 value = 0x100A39
49
                                                        ; External Memory access type = 0x1
        000001
50
                    acctype3
                                    equ
        000000
                                            0
51
                                                        ; Enable AA3 pin to be low when selected
                    aahiqh3
                                    equ
                                                          Yes, Enable AA3 pin on ext 'P'
        000008
52
                    aap3
                                    equ
                                            $8
                                                                                          accesses
                                                          Yes, Enable AA3 pin on ext 'X' accesses
53
        000010
                    aax3
                                            $10
                                    equ
                                                          Yes, Enable AA3 pin on ext 'Y' accesses
        000020
54
                    aay3
                                    equ
                                            $20
                                                          No, Enable address bus swap
55
        000000
                    aswap3
                                    equ
                                            Λ
        000000
                                                          No, Enable packing/unpacking logic
56
                    enpack3
                                    equ
                                            0
        000A00
                                            $000A00
                                                          Compare 10 address bits
57
                    nadd3
                                    equ
58
        100000
                    msadd3
                                    equ
                                            $100000
                                                          Most significant portion of address,
59
                                                          $100000 - $103fff, to compare.
                                                          (0001,0000,00xx,xxxx,xxxx,xxxx)
60
61
        100A39
                    AAR3_value
                                    equ acctype3+aahigh3+aap3+aax3+aay3+aswap3+enpack3+nadd3+msadd3
62
63
                    ;--- AAR2 value = 0x100A1D
64
        000001
                    acctype2
                                                        ; External Memory access type = 0x1
                                    equ
65
        000004
                    aahigh2
                                            $4
                                                        ; Enable AA2 pin to be high when selected
                                    equ
        800000
                    aap2
                                            $8
                                                          Yes, Enable AA2 pin on ext 'P' accesses
66
                                    equ
                                                          Yes, Enable AA2 pin on ext 'X' accesses
                    aax2
67
        000010
                                            $10
                                    equ
68
        000000
                    aay2
                                    equ
                                            Ò
                                                          No, Enable AA2 pin on ext 'Y' accesses
        000000
                                                        ; No, Enable address bus swap
69
                    aswap2
                                    equ
        000000
70
                    enpack2
                                    equ
                                                          No, Enable packing/unpacking logic
71
                    nadd2
                                            $000A00
                                                          Compare 10 address bits
        000A00
                                    equ
72
        100000
                    msadd2
                                            $100000
                                                          Most significant portion of address,
                                    eau
73
                                                        ; $100000 - $103fff, to compare.
                                                          (0001,0000,00xx,xxxx,xxxx,xxxx)
```



```
75
       100A1D
                  AAR2_value
                                 eau
                                          acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
76
77
                           i --- BCR value = 0x002400
78
       000000
                   aaa0ws
                                                            ; Address Attribute Area 0 w/s = 0
                                eau
79
       000000
                                          0
                   aaa1ws
                                                            ; Address Attribute Area 1 w/s = 0
                                  equ
                                                            ; Address Attribute Area 2 w/s = 1
80
       000400
                   aaa2ws
                                          $000400
                                  equ
                                                            ; Address Attribute Area 3 w/s = 1
       002000
                                          $002000
81
                   aaa3ws
                                  equ
       000000
                                                            ; Default Address Area w/s = 0
82
                   defws
                                          0
                                  equ
       000000
                                          0
8.3
                   busss
                                  equ
                                                            ; Bus state status = 0
84
       000000
                   enblh
                                          Ω
                                                            ; Enable Bus Lock Hold = 0
                                  equ
                                                             ; Enable Bus Request Hold = 0
85
       000000
                   enbrh
                                  equ
                                          0
86
       002400
                   BCR_value
                                  equ
                                          aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
87
88
29
       P:000100
                                org p:$100
                                                           ;Keep the program in internal RAM
90
91
                   memtst
92
93
                   ; Initialization Section
       P:000100 08F4BD
94
                                        #PCTL_value,x:PCTL
                                                                    ; Set PLL Control Register
                            movep
                 0E0013
95
       P:000102
                 05F43A
                            movec
                                        #$004000,OMR
                                                           ; Disable Address Attribute Priority
                 004000
       P:000104
                 05F439
                                        #$080000,SR
96
                            movec
                                                                    ; Enable 1K Cache
                 080000
       P:000106
                 08F4BB
                                        #BCR_value,x:BCR
97
                            movep
                                                                    ; Set external wait states
                 002400
       P:000108
98
                 08F4B7
                            movep
                                        #AAR2_value,x:AAR2
                                                                    ; Set Address Attribute Reg2
                 100A1D
99
       P:00010A 08F4B6
                                        AAR3_value,x:AAR3
                                                                    ; Set Address Attribute Reg3
                            movep
                 100A39
100
       P:00010C 05F420
                                                                    ; Set LINEAR addressing mode
101
                            move
                                        \#-1.m0
                 FFFFFF
102
       P:00010E 05F423
                                        \#-1,m3
                            move
                 444444
103
       P:000110 20001B
104
                            clr
                                        b
       P:000111
105
                 000000
                            nop
                 570000
                                  \texttt{b,x:P\_MEM\_FAIL\_ADDRESS: Initialize P:Failed Address -> \$000000}
106
       P:000112
                          move
                                  b,x:P_MEM_FAIL_WROTE ; Initialize P:Expected Data -> $000000
b,x:P_MEM_FAIL_READ ; Initialize P:Data Read -> $000000
107
       P:000113
                 570100
                          move
108
                 570200
       P:000114
                           move
                                  b,x:Y_MEM_FAIL_ADDRESS; Initialize Y:Failed Address -> $000000
109
       P:000115
                 570300
                          move
                 570400
                                  b,x:Y_MEM_FAIL_WROTE ; Initialize Y:Expected Data -> $000000
110
       P:000116
                           move
111
       P:000117
                 570500
                           move
                                  b,x:Y_MEM_FAIL_READ
                                                         ; Initialize Y:Data Read -> $000000
112
       P:000118 570600
                          move
                                  b,x:MEM_PASS_COUNTER ; Initialize Pass Counter -> $000000
113
114
                          main
115
; ---
     ______
116
                                  ;--- fill P:memory with initial pattern
117
: ____
       P:000119 63F400
118
                             move
                                        #PATT,r3
                                                           ; r3 points to Test Patterns
                 00014A
119
       P:00011B 000000
                              nop
120
       P:00011C 000000
                              nop
121
       P:00011D 07DB84
122
                              move
                                          p:(r3)+,x0
                                                             ; Get the Write Pattern for P/X:MEM
123
       P:00011E 07DB86
                                         p:(r3)+,y0
                                                             ; Get the Write Pattern for Y:MEM
                              move
124
125
       P:00011F 70F400
                                          #PMemSize,n0
                                                             ; Get memory size
                              move
                 004000
126
       P:000121 60F400
127
                                          #PMemStart,r0
                                                            ; Get starting address for fill
                              move
                 100000
128
129
       P:000123 06D820
                                          n0
                                                             ; Fill P/X:RAM with first data pattern
                              rep
                                          x0,p:(r0)+
       P:000124 075884
130
                              move
131
       P:000125 60F400
                                          #PMemStart,r0
132
                              move
                 100000
       P:000127
133
                                          n0
                 06D820
                                                             ; Fill Y:RAM with first data pattern
                              rep
134
       P:000128 4E5800
                              move
                                         y0,y:(r0)+
135
136
137
                              ;--- Check for expected pattern data in each RAM location ---
                              ;--- and then replace with a new data pattern.
138
139
                              ;--- ... This provides an address check. Since erroneous
140
                              ;--- ...addressing will cause the data to be written into ---
                              ;--- ...incorrect locations and this will be evident in
141
```



```
142
                              ;--- ...the next read pass.
143
       P:000129 063890
144
                              DOR
                                         #PATTN,test_Pm ; Start Pattern Test Loop
                 000013
145
       P:00012B 60F400
                                         #PMemStart,r0
                                                            ; Get starting address of Test Memory
                              move
                 100000
146
       P:00012D 200041
                              tfr
                                         x0,a
                                                            ; Save the last P/X test pattern -> a
       P:00012E 200059
                                                            ; Save the last Y test pattern -> b
147
                                         y0,b
                              t.fr
                                                            ; Get the next P/X test pattern -> x0
       P:00012F 07DB84
                                          P:(r3)+,x0
148
                              move
       P:000130 07DB86
149
                                                            ; Get the next Y test pattern -> y0
                              move
                                         P:(r3)+,y0
150
                              ; Test this pattern through external RAM
151
       P:000131 06D810
                              DOR
152
                                         n0,next_loc
                                                            ; Test all external RAM locations
                 A00000
153
       P:000133 07E085
                              move
                                         P:(r0),x1
                                                            ; Read P/X:RAM location
                                         <PERR
154
       P:000134 200065
                              cmp
                                                            ; Read data = last test pattern?
155
       P:000135 05240D
                                                            ; No, error if compare fails
                              bne
156
       P:000136 076084
                              move
                                         x0,P:(r0)
                                                            ; Yes, Write next test pattern ->
P/X:RAM
157
158
       P:000137 4FE000
                                         Y:(r0),y1
                                                           ; Read Y:RAM location
                              move
                                                            ; Read data = last test pattern?
159
       P:000138 20007D
                              cmp
                                                            ; No, error if compare fails
160
       P:000139 05240D
                              bne
                                          <YERR
       P:00013A 4E5800
                                         y0,Y:(r0)+
                                                            ; Yes, Write next test pattern ->
161
                             move
Y:RAM
162
       P:00013B 000000
                              nop
163
                          next_loc
164
165
       P:00013C 000000
                                                             ; Time to start next test pattern
                            nop
166
167
                          test_Pm
168
                                              One Pass Complete
169
                             ; All test patterns have been tried and passed in external RAM
170
                             move x:MEM_PASS_COUNTER,b0
171
       P:00013D 518600
       P:00013E 000009
P:00013F 000000
172
                              inc
                                                                    ; Update pass loop counter
173
                             nop
                                        b0,x:MEM_PASS_COUNTER
174
       P:000140 510600
                              move
175
176
       P:000141 050F98
                             bra
                                        main
                                                                    ; Do it all over again
177
178 ;-----
179
                             ; PERR -- handles P/X:RAM error messaging with user
180
181
                              ; Expected Data --> a
182
                              ; Read Data --> x1
                              ; Address of failure --> r0
183
184
185
       P:000142 600000
P:000143 560100
186
                              move r0,x:P\_MEM\_FAIL\_ADDRESS
                                                                   ; Save off address of failure
                                         a,x:P_MEM_FAIL_WROTE
                                                                   ; Save off expected data ; Save off data read
187
                                 move
188
       P:000144 450200
                                 move
                                        x1,x:P_MEM_FAIL_READ
189
190
       P:000145 050C00
                                                                    ; Dynamically HALT here
191
192
193
                                 YERR -- handles Y:RAM error messaging with user
194
195
                              ; Expected Data --> b
196
                              ; Read Data --> y1
197
                              ; Address of failure --> r0
198
199
                              YERR
200
       P:000146 600300
                                        r0,x:Y_MEM_FAIL_ADDRESS
                                                                   ; Save off address of failure
                                 move
       P:000147 570400
P:000148 470500
                                        b,x:Y_MEM_FAIL_WROTE ; Save off expected data y1,x:Y_MEM_FAIL_READ ; Save off data read
201
                                 move
202
                                 move
203
       P:000149 050C00
204
                                 bra
                                                                    ; Dynamically HALT here
205
206
207
208
                              ; Memory Test Patterns
209
       P:00014A
210
                              PATT
                                          dс
                                                     $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
211
       P:00014F
                                          dc
                                                     $800000,$400000,$200000,$100000
212
       P:000153
                                         dc
                                                     $080000,$040000,$020000,$010000
213
       P:000157
                                          dc
                                                     $008000,$004000,$002000,$001000
214
       P:00015B
                                          dc
                                                     $000800,$000400,$000200,$000100
215
       P:00015F
                                                     $000080,$000040,$000020,$000010
216
       P:000163
                                                     $000008,$000004,$000002,$000001
                                          dc
       P:000167
                                                     $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
217
```



Errors Warnings

218	P:00016B		dc	\$F7FFFF,\$FBFFFF,\$FDFFFF,\$FEFFFF
219	P:00016F		dc	\$FF7FFF,\$FFBFFF,\$FFDFFF,\$FFEFFF
220	P:000173		dc	\$FFF7FF,\$FFFBFF,\$FFFDFF,\$FFFEFF
221	P:000177		dc	\$FFFF7F,\$FFFFBF,\$FFFFDF,\$FFFFEF
222	P:00017B		dc	\$FFFFF7,\$FFFFFB,\$FFFFFD,\$FFFFFE
223	P:00017F		dc	\$FEDCBA, \$123456, \$012345, \$EDCBA9
224				
225	P:000183		dc	\$000000,\$FFFFFF,\$AAAAAA,\$555555,\$2BAD2C
226	P:000188		dc	\$800000,\$400000,\$200000,\$100000
227	P:00018C		dc	\$080000,\$040000,\$020000,\$010000
228	P:000190		dc	\$008000,\$004000,\$002000,\$001000
229	P:000194		dc	\$000800,\$000400,\$000200,\$000100
230	P:000198		dc	\$000080,\$000040,\$000020,\$000010
231	P:00019C		dc	\$000008,\$000004,\$000002,\$000001
232	P:0001A0		dc	\$7FFFFF,\$BFFFFF,\$DFFFFF,\$EFFFFF
233	P:0001A4		dc	\$F7FFFF,\$FBFFFF,\$FDFFFF,\$FEFFFF
234	P:0001A8		dc	\$FF7FFF,\$FFBFFF,\$FFDFFF,\$FFEFFF
235	P:0001AC		dc	\$FFF7FF,\$FFFBFF,\$FFFDFF,\$FFFEFF
236	P:0001B0		dc	\$FFFF7F,\$FFFFBF,\$FFFFDF,\$FFFFEF
237	P:0001B4		dc	\$FFFFF7,\$FFFFFB,\$FFFFFD,\$FFFFFE
238	P:0001B8		dc	\$FEDCBA, \$123456, \$012345, \$EDCBA9
239				
240	000038	PATTN	equ	((*-PATT)/2)-1
241			-	
242			end	memtst





4 128K × 8-bit Memory Based Designs

This section describes how to implement several different DSP memory space Fast SRAM designs using three $128K \times 8$ -bit 3.3 V memories with a Motorola DSP56303 device.

Using one hardware memory design based on three $128K \times 8$ -bit 3.3 V memories, the DSP56303's Memory Expansion Port allows the $128K \times 24$ -bit memory bank to be logically configured for use in various memory space arrangements. The configuration is accomplished by programmatically changing the Memory Expansion Port's Address Attribute Control Registers.

Configuring and using one Memory Expansion Port Address Attribute Control Register, the 128K memory bank can accommodate seven different memory space arrangements.

- 1. 128K × 24-bit 'P' Space Fast SRAM
- 2. 128K × 24-bit 'X' Space Fast SRAM
- 3. 128K × 24-bit 'Y' Space Fast SRAM
- 4. 128K × 24-bit 'P'/'X' Space Fast SRAM
- 5. 128K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. 128K × 24-bit 'X'/'Y' Space Fast SRAM
- 7. 128K × 24-bit 'P'/'X'/'Y' Space Fast SRAM

Configuring and using two Memory Expansion Port Address Attribute Control Registers, the 128K memory bank can accommodate thirteen different memory space arrangements.

- 1. 128K × 24-bit 'P' Space Fast SRAM
- 2. 128K × 24-bit 'X' Space Fast SRAM
- 3. 128K × 24-bit 'Y' Space Fast SRAM
- 4. 128K × 24-bit 'P'/'X' Space Fast SRAM
- 5. 128K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. 128K × 24-bit 'X'/'Y' Space Fast SRAM
- 7. $128K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM
- 8. $64K \times 24$ -bit 'P'/'X' and $64K \times 24$ -bit 'Y' Space Fast SRAM
- 9. 64K × 24-bit 'P'/'Y' and 64K × 24-bit 'X' Space Fast SRAM
- 10. $64K \times 24$ -bit 'P' and $64K \times 24$ -bit 'X'/'Y' Space Fast SRAM
- 11. $64K \times 24$ -bit 'P' and $64K \times 24$ -bit 'X' Space Fast SRAM
- 12. 64K × 24-bit 'P' and 64K × 24-bit 'Y' Space Fast SRAM
- 13. $64K \times 24$ -bit 'X' and $64K \times 24$ -bit 'Y' Space Fast SRAM

All of these memory space configurations efficiently use the full capacity of the memory chips in the $128K \times 24$ -bit memory bank.



The memory configuration examples in the remainder of this chapter are based on one common hardware design that uses two Address Attribute Selectors and implements two of the most common configurations: $128K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM configuration and a $64K \times 24$ -bit 'Y' Space Fast SRAM configuration (see **Figure 4-4** for a schematic of the hardware design).

4.1 128K × 24-bit Common Fast SRAM Hardware Design

This section describes an asynchronous Fast SRAM $128K \times 24$ -bit memory bank implementation using Motorola's MCM6926 device (see **Figure 4-1**). Memory bank designs of this size and type allow for future application expandability in both size and speed. The 3.3 V devices also require less hardware, providing a glueless memory interface with the DSP.

The memory bank design is implemented using two Address Attribute Selectors. However, for the $128K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM configuration a one Address Attribute Selector design can be used—A16 substituted for AA2. The two Address Attribute Selector design demonstrates the ultimate flexibility of the Address Attribute Selectors.

For this common hardware design, the DSP core runs at a maximum of 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one-wait state external memory requirements. This 3.3 V device is organized as $128K \times 8$ -bits. Therefore, three memory devices are used to achieve the 24-bit wide bus.

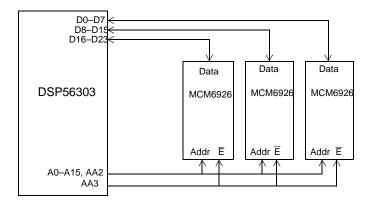


Figure 4-1 128K × 24-bit Fast SRAM Memory Example

4.1.1 MCM6926-12 Memory Timing Requirements

For the asynchronous Fast SRAM device to work properly, its timing requirements must be met. The sections that follow give the timing requirements for the MCM6926-12 $128K \times 8$ -bit 12 nS Fast SRAM.

4.1.1.1 Read Cycle Timing

Table 4-1 shows the memory read timing values used in the memory read cycle timing diagram, **Figure 4-2.**



Read Cycle Parameter	Symbol	Min	Max
Read Cycle Time	t _{AVAV}	12 nS	_
Address Access Time	t _{AVQV}	_	12 nS
Enable Access Time	t _{ELQV}	_	12 nS
Output Enable Access Time	t _{GLQV}	_	6 nS
Enable High to Output High-Z	t _{EHQZ}	0 nS	6 nS
Output Enable High to Output High-Z	t _{GHQZ}	0 nS	6 nS

Table 4-1. MCM6926-12 Memory Read Timing Specifications

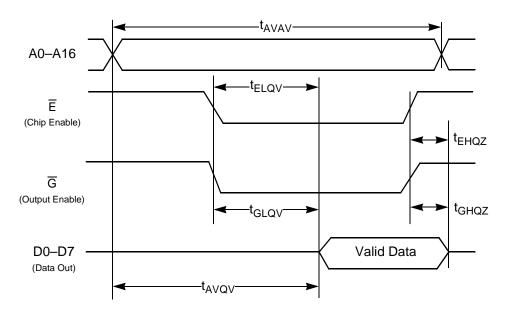


Figure 4-2. MCM6926 Memory Read Cycle Timing Diagram

4.1.1.2 Write Cycle Timing

Table 4-2 shows the memory write timing values used in the memory write cycle timing diagram, **Figure 4-3**.



	Table 4-2.	MCM6926-12 Memory	Write	Timing	Specifications
--	------------	-------------------	-------	--------	----------------

Write Cycle Parameter	Symbol	Min	Max
Write Cycle Time	t _{AVAV}	12 nS	_
Address Setup Time	t _{AVWL}	0 nS	_
Address Valid to End off Write	t _{AVWH}	10 nS	_
Write Pulse Width	t _{WLWH}	10 nS	_
Data Valid to End of Write	t _{DVWH}	6 nS	_
Data Hold Time	t _{WHDX}	0 nS	_

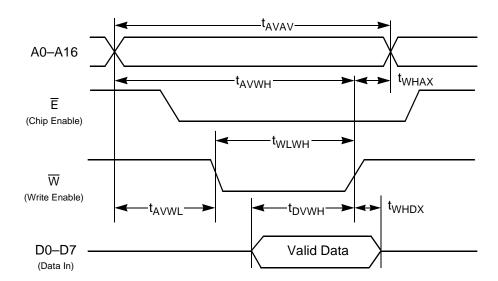


Figure 4-3. MCM6926 Memory Write Cycle Timing Diagram



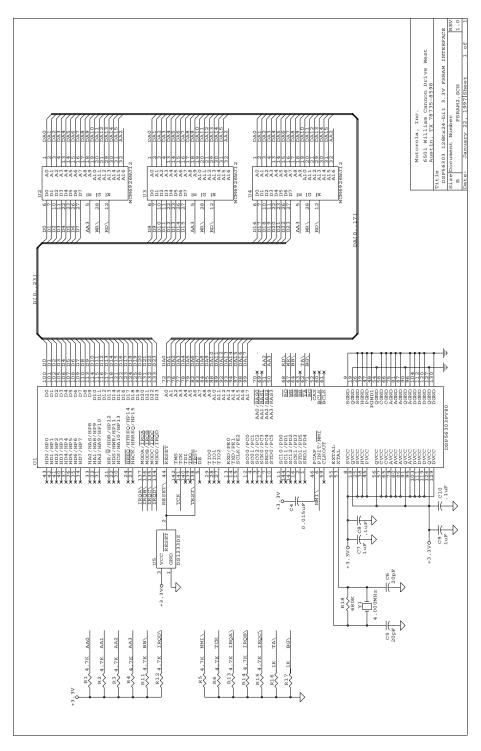


Figure 4-4. 128K × 24-bit Fast SRAM Schematic



4.2 $128K \times 24$ -bit 'P'/'X'/'Y' Fast SRAM Example

This section describes a $128K \times 24$ -bit shared 'P'/'X'/'Y' memory space, asynchronous Fast SRAM implementation using Motorola's MCM6926 device (see **Figure 4-5** for the memory map layout, **Figure 4-1** for the block diagram, and **Example 4-1** for the example code). In a shared memory space, data written to one address in one memory space can be accessed by the same address in another memory space (e.g., writing \$012345 to P:\$100000 could be read by X:\$100000, Y:\$100000 or P:\$100000).

The DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

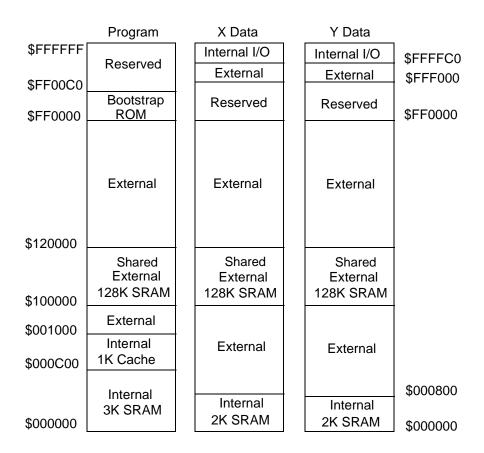


Figure 4-5. 128K × 24-bit 'P'/'X'/'Y' Fast SRAM Memory Map

4.2.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $128K \times 24$ -bit 'P'/'X'/'Y' space memory configuration, set up the following DSP control registers.



Set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation register (PCTL). For this example, the DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, Bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 3 enables, via Fast SRAM \overline{E} , the external 128K SRAM bank accesses in the address range from \$100000 to \$11FFFF during program, X data, and Y data space requests. Configure the memory address space requirements for Address Attribute Pin 3 using Address Attribute Register 3 (AAR3). The AAR3 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$7
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR3 is \$100739.



Address Attribute Pin 2 selects, via Fast SRAM A16, address line A16 in the external 128K SRAM bank during accesses in the address range from \$110000 to \$11FFFF during program, X data, and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using the Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 1.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$8
- Specify the most significant portion of the address to compare, Bits 12-23 = \$110

The value loaded into the AAR2 is \$11083D.

The value loaded into AAR0 and AAR1 is \$000000.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address attribute area 0 wait states, Bits 0-4 = \$0
- Address attribute area 1 wait states, Bits 5-9 = \$0
- Address attribute area 2 wait states, Bits 10-12 = \$1
- Address attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16-20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 =0

The value loaded into the BCR is \$002400.



Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value combines the following bits for each feature:

- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit, reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14. = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 4-1. 128K × 24-bit 'P'/'X'/'Y' Space Fast SRAM Memory Exercise

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:25 asram4.asm

```
1
                                                132,60,3,3,
                                       page
2
3
                              ASRAM4.ASM - Simple program to test 128Kx24-bits of unified
                                              Program/X-Data/Y-Data memory using a DSP56303
5
6
                              The program uses Internal P:RAM to test External P/X/Y:RAM
                                              from $100000 - $11FFFF @ 1w/s
8
        100000
10
                        PMemStart
                                             $100000
                                    equ
        120000
11
                        PMemEnd
                                             $120000
                                    equ
12
        020000
                        PMemSize
                                             PMemEnd-PMemStart
                                    equ
13
14
                        ; --- Program Specific Storage Locations (X DATA SPACE)
15
                        MEM_FAIL_ADDRESS
        000000
                                             $000000
16
                                    eau
17
                        MEM_FAIL_WROTE
18
        000001
                                             $000001
                                    equ
19
                        MEM FAIL READ
        000002
                                             $000002
20
                                    eau
                        MEM_PASS_COUNTER
21
                                             $000003
2.2
        000003
                                    equ
23
24
                        ;--- DSP56303 Control Registers (X I/O SPACE)
25
                        BCR
                                          SFFFFFB
        STATA
                                    equ
                                                        ; Bus Control Register
2.6
        ОЧЧЧЧЧ
                        PCTL
                                    equ
                                          SEFFEFD
                                                         ; PLL Control Register
2.7
        844444
                        AAR3
                                    equ
                                          SFFFFF6
                                                         ; Address Attribute Register #3
28
        FFFFF7
                        AAR2
                                    equ
                                          $FFFFF7
                                                         ; Address Attribute Register #2
29
30
                        ;--- PCTL value = 0x0E0013
31
        000000
                        prediv
                                    equ
                                          0
                                                         ; Pre-Divider = 1
32
        000000
                        lowdiv
                                          0
                                                          Low Power Divider = 1
                                    equ
                                                         ; VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
33
        000013
                        pllmul
                                          19
                                    equ
34
        000000
                                          0
                                                         ; No, Crystal not less than 200kHz
                        crystal
                                    equ
35
        000000
                        disXTAL
                                                          No, do not disable crystal use
                                    equ
                                          0
36
        020000
                        pllstop
                                          $020000
                                                         ; Yes, PLL runs during STOP
                                    equ
37
        040000
                        enpll
                                    equ
                                          $040000
                                                          Yes, enable PLL operation
38
        080000
                        disclk
                                    equ
                                          $080000
                                                          Yes, disable CORE clock output
39
        0E0013
                                          prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
                        PCTL_value equ
40
41
                        ; --- AAR3 value = 0x100739
42
        000001
                        acctype3
                                                         ; External Memory access type = 0x1
                                    equ
        000000
                                          0
43
                        aahigh3
                                                         ; Enable AA3 pin to be low when selected
                                    equ
                                                          Yes, Enable AA3 pin on ext 'P'
44
        800000
                        aap3
                                    eau
                                          $8
                                                                                           accesses
                                                          Yes, Enable AA3 pin on ext 'X' accesses
45
        000010
                        aax3
                                    equ
                                          $10
                                                          Yes, Enable AA3 pin on ext 'Y' accesses
        000020
46
                        aav3
                                          $20
                                    equ
                                                          No, Enable address bus swap
No, Enable packing/unpacking logic
47
        000000
                        aswap3
                                          0
                                    equ
48
        000000
                        enpack3
                                          0
                                    equ
        000700
                                          $000700
49
                                                          Compare 7 address bits
                        nadd3
                                    equ
                                                          Most significant portion of address, $100000 - $11ffff, to compare.
        100000
50
                                          $100000
                        msadd3
                                    equ
51
                                                           (0001,000x,xxxx,xxxx,xxxx)
52
53
        100739
                        AAR3_value equ acctype3+aahigh3+aax3+aax3+aaya3+enpack3+nadd3+msadd3
54
55
                        ;--- AAR2 value = 0x11083D
        000001
                                                         ; External Memory access type = 0x1
                                         1
56
                        acctype2
                                    equ
        000004
57
                        aahigh2
                                          $4
                                                         ; Enable AA2 pin to be high when selected
                                    equ
58
        800000
                        aap2
                                    equ
                                          $8
                                                          Yes, Enable AA2 pin on ext 'P' accesses
                                                          Yes, Enable AA2 pin on ext 'X' accesses
59
        000010
                        aax2
                                    equ
                                          $10
                                                          Yes, Enable AA2 pin on ext 'Y' accesses
60
        000020
                        aay2
                                          $20
                                    equ
61
        000000
                        aswap2
                                    equ
                                          0
                                                          No, Enable address bus swap
62
        000000
                        enpack2
                                          0
                                                          No, Enable packing/unpacking logic
                                    equ
                        nadd2
63
        000800
                                          $000800
                                                          Compare 8 address bits
                                    equ
64
        110000
                        msadd2
                                          $110000
                                                          Most significant portion of address,
                                    equ
65
                                                           $110000 - $11ffff, to compare.
                                                          (0001,0001,xxxx,xxxx,xxxx)
66
67
        11083D
                        AAR2_value equ acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
68
69
                        ;--- BCR value = 0 \times 002400
70
        000000
                        aaa0ws
                                    equ
                                          0
                                                         ; Address Attribute Area 0 \text{ w/s} = 0
71
        000000
                                          0
                                                         ; Address Attribute Area 1 w/s = 0
                        aaa1ws
                                    equ
72
        000400
                        aaa2ws
                                          $000400
                                                        ; Address Attribute Area 2 w/s = 1
                                    eau
73
                                                         ; Address Attribute Area 3 w/s = 1
        002000
                                          $002000
                        aaa3ws
                                    equ
74
        000000
                        defws
                                          0
                                                         ; Default Address Area w/s = 0
                                    equ
```



```
75
        000000
                       busss
                                    equ
                                                        ; Bus state status = 0
76
        000000
                        enblh
                                                       ; Enable Bus Lock Hold = 0
                                    equ
77
        000000
                        enbrh
                                          0
                                                        ; Enable Bus Request Hold = 0
                                    eau
78
        002400
                       BCR_value equ aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
79
80
                                org p:$100 ;Keep the program in internal RAM
        P:000100
81
82
83
                       memtst.
84
85
                        ; Initialization Section
       P:000100 08F4BD
                                 movep #PCTL_value,x:PCTL
                                                               ; Set PLL Control Register
86
                  0E0013
87
        P:000102 05F43A
                                 movec #$004000.OMR
                                                                ; Disable Address Attribute Priority
                  004000
88
       P:000104 05F439
                                 movec #$080000,SR
                                                                ; Enable 1K Cache
                  080000
89
        P:000106 08F4BB
                                 movep #BCR_value,x:BCR
                                                                ; Set external wait states
                  002400
90
        P:000108 08F4B7
                                 movep #AAR2_value,x:AAR2
                                                                ; Set Address Attribute Reg2
                  11083D
91
        P:00010A 08F4B6
                                  movep #AAR3_value,x:AAR3
                                                                ; Set Address Attribute Reg3
                  100739
93
        P:00010C 05F420
                                          #-1,m0
                                                                ; Set LINEAR addressing mode
                                 move
                  FFFFFF
94
        P:00010E 05F423
                                  move \#-1,m3
                  FFFFFF
96
        P:000110 20001B
                                  clr b
        P:000111 000000
                                  nop
98
        P:000112 570000
                                  move b,x:MEM_FAIL_ADDRESS ; Initialize Failed Address -> $000000
                                 move b,x:MEM_FAIL_WROTE ; Initialize Expected Data -> $000000 move b,x:MEM_FAIL_READ ; Initialize Data Read -> $000000 move b,x:MEM_PASS_COUNTER ; Initialize Pass Counter -> $000000
99
        P:000113 570100
100
        P:000114 570200
        P:000115 570300
101
102
103
                       main
104
                        ;--- fill P:memory with initial pattern
105
106
                                move #PATT,r3 ; r3 points to Test Patterns
       P:000116 63F400
107
                  000138
108
        P:000118 000000
                                   nop
       P:000119 000000
109
                                   nop
110
111
       P:00011A 07DB84
                                   move p:(r3)+,x0; Get the Write Pattern for P:MEM
112
113
       P:00011B 70F400
                                    move #PMemSize,n0 ; Get memory size
                  020000
114
115
       P:00011D 60F400
                                   move #PMemStart,r0 ; Get starting address for fill
                  100000
116
        P:00011F 06D820
117
                                   rep n0
                                                       ; Fill RAM with first pattern data
118
        P:000120 075884
                                   move x0,p:(r0)+
119
120
121
                        ;--- Check for expected pattern data in each RAM location ---
122
                        ;--- and then replace with a new data pattern.
                        :-- ... This provides an address check. Since erroneous --- :-- ... addressing will cause the data to be written into ---
123
124
125
                        ;--- ...incorrect locations and this will be evident in
                        ;--- ...the next read pass.
126
127
       P:000121 063890
                                  DOR #PATTN,test_Pm ; Start Pattern Test Loop
128
                  00000D
129
       P:000123 60F400
                                   move #PMemStart,r0
                                                               ; Get starting address of Test Memory
                  100000
       P:000125 200041
P:000126 07DB84
                                    tfr x0,a
130
                                                                ; Save the last test pattern -> a
                                   move P:(r3)+,x0
131
                                                                ; Get the next test pattern -> x0
132
133
                        ; Test this pattern through external RAM
       P:000127 06D810
                                                               ; Test all external RAM locations
134
                                   DOR n0,next_loc
                  000006
       P:000129 07E085
                                                             ; Read RAM location
; Read data = last test pattern?
135
                                    move P:(r0),x1
                                    cmp x1,a
136
       P:00012A 200065
                                                              ; No, error if compare fails
137
       P:00012B 052409
                                   bne <ERR
138
       P:00012C 075884
                                    move x0,P:(r0)+
                                                               ; Yes, Write next test pattern -> RAM
139
        P:00012D 000000
140
                       next_loc
141
```

128K × 8-bit Memory Based Designs

```
142
        P:00012E 000000
                                 nop
                                                                    ; Time to start next test pattern
143
144
                         test_Pm
145
146
                                           One Pass Complete
                         ; All test patterns have been tried and passed in external RAM
147
148
        P:00012F 518300
                                      move x:MEM_PASS_COUNTER,b0
        P:000130 000009
149
                                                                    ; Update pass loop counter
                                      inc
                                            b
        P:000131
                   000000
150
                                      nop
151
                                      move b0,x:MEM_PASS_COUNTER
        P:000132 510300
152
153
        P:000133 050FC3
                                      bra main
                                                                    ; Do it all over again
154
                          ;-----
155
156
                            ERR -- handles error messaging with user
157
158
                          ; Expected Data --> a
                          ; Read Data --> x1
; Address of failure --> r0
159
160
161
162
163
        P:000134 600000
                                      move r0,x:MEM_FAIL_ADDRESS
                                                                            ; Save off address of failure
164
        P:000135 560100
                                      move a,x:MEM_FAIL_WROTE
                                                                            ; Save off expected data
165
        P:000136 450200
                                      move x1,x:MEM_FAIL_READ
                                                                            ; Save off data read
166
167
        P:000137 050C00
                                                                             ; Dynamically HALT here
168
169
170
                          ; Memory Test Patterns
171
                                          $000000,$FFFFFFF,$AAAAAA,$555555,$2BAD2C
$800000,$400000,$200000,$100000
$080000,$040000,$020000,$010000
$008000,$004000,$002000,$001000
$000800,$000400,$000200,$000100
172
        P:000138
                         PATT
173
        P:00013D
                                  dc
174
        P:000141
                                  dc
175
        P:000145
                                  dc
176
        P:000149
                                  dc
                                          $000080,$000040,$000020,$000010
$000008,$000004,$000002,$000001
177
        P:00014D
                                  dc
178
        P:000151
                                  dc
179
        P:000155
                                          $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
                                  dc
180
        P:000159
                                          $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
                                  dc
        P:00015D
                                          $FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
$FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
181
                                  dc
        P:000161
182
                                  dc
183
        P:000165
                                          $FFFF7F,$FFFFBF,$FFFFFFF
                                  dc
        P:000169
                                          $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
184
                                  dc
185
        P:00016D
                                          $FEDCBA, $123456, $012345, $EDCBA9
                                  dc
186
187
        000038
                         PATTN equ
                                          *-PATT-1
188
189
                                  end memtst
```

0 Errors 0 Warnings



4.3 $64K \times 24$ -bit 'X' and $64K \times 24$ -bit 'Y' Fast SRAM Example

This section describes a $64K \times 24$ -bit 'X' and $64K \times 24$ -bit 'Y' memory space, asynchronous Fast SRAM implementation using Motorola's MCM6926 device (see **Figure 4-6** for the memory map layout, **Figure 4-1** for the block diagram, and **Example 4-2** for the example code).

The DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

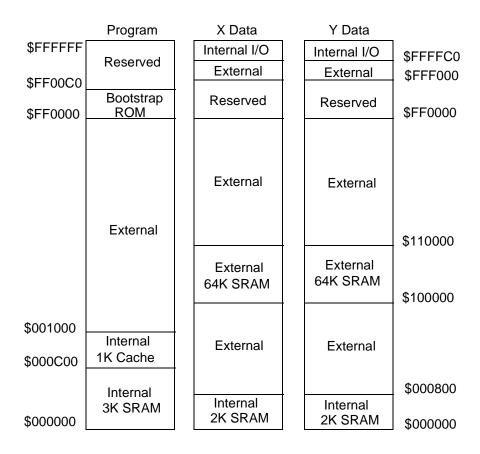


Figure 4-6. $64K \times 24$ -bit 'X' and $64K \times 24$ -bit 'Y' Memory Map

4.3.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $64K \times 24$ -bit 'X' and $64K \times 24$ -bit 'Y' space memory configuration, set up the following DSP control registers.



Set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation register (PCTL). For this example, the DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, Bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 3 enables, via Fast SRAM \overline{E} , the external 128K SRAM bank accesses in the address range from \$100000 to \$10FFFF during X data and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 3 using Address Attribute Register 3 (AAR3). The AAR3 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 0.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$8
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR3 is \$100831.



Address Attribute Pin 2 selects, via Fast SRAM A16, the external 128K SRAM bank during accesses in the address range from \$100000 to \$10FFFF to differentiate X data space requests from Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using the Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 0.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin during external Y data space accesses, Bit 5 = 0.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$8
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR2 is \$100815.

The value loaded into AAR0 and AAR1 is \$000000.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address attribute area 0 wait states, Bits 0-4 = \$0
- Address attribute area 1 wait states, Bits 5-9 = \$0
- Address attribute area 2 wait states, Bits 10–12 = \$1
- Address attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16-20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 =0

The value loaded into the BCR is \$002400.



Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value combines the following bits for each feature:

- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14 = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR register value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.

The value loaded into the SR is \$080000, which is the value loaded during reset.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 4-2. 64K 'X' and 64K 'Y' Space Fast SRAM Memory Exercise

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:29 asram5.asm

```
132,60,3,3,
                                       page
2
3
                              ASRAM5.ASM
                                           - Simple program to test 64Kx24-bits of X-Data
4
                                             and 64Kx24-bits of Y-Data memory using a DSP56303
5
                              The program uses Internal P:RAM to test External X: & Y:RAM
                                             from $100000 - $10FFFF @ 1w/s
8
10
        100000
                                            $100000
                        MemStart
                                    equ
11
        110000
                        MemEnd
                                            $110000
                                    equ
12
        010000
                        MemSize
                                            MemEnd-MemStart
                                    equ
13
14
                        :--- Program Specific Storage Locations (X DATA SPACE)
                        X_MEM_FAIL_ADDRESS
15
16
        000000
                                            $000000
                                    eau
                        X MEM FAIL WROTE
17
        000001
18
                                            $000001
                                    equ
19
                        X_MEM_FAIL_READ
        000002
2.0
                                            $000002
                                    equ
21
22
                        Y_MEM_FAIL_ADDRESS
        000003
                                            $000003
23
                                    equ
24
25
                        Y_MEM_FAIL_WROTE
        000004
                                            $000004
                                    equ
26
                        Y_MEM_FAIL_READ
        000005
27
                                    equ
                                            $000005
2.8
29
                        MEM_PASS_COUNTER
30
        000006
                                    equ
                                            $000006
31
32
                        ;--- DSP56303 Control Registers (X I/O SPACE)
33
        FFFFFB
                        BCR
                                            $FFFFFB
                                                          Bus Control Register
                                    equ
                                                       ;
34
        FFFFFD
                        PCTL
                                            $FFFFFD
                                                        ; PLL Control Register
                                    equ
35
        FFFFF6
                        AAR3
                                            $FFFFF6
                                                        ; Address Attribute Register #3
                                    equ
36
        FFFFF7
                        AAR2
                                    equ
                                            $FFFFF7
                                                        ; Address Attribute Register #2
37
38
                        ;--- PCTL value = 0x0E0013
39
        000000
                        prediv
                                                        ; Pre-Divider = 1
                                            0
                                    equ
40
        000000
                                                        ; Low Power Divider = 1
                        lowdiv
                                            0
                                    equ
                                            19
41
        000013
                        pllmul
                                                          VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
                                    equ
        000000
                                                          No, Crystal not less than 200kHz
42
                        crystal
                                            0
                                    equ
                                                        ; No, do not disable crystal use
43
        000000
                        disXTAL
                                    eau
                                            $020000
44
        020000
                        pllstop
                                                          Yes, PLL runs during STOP
                                    equ
                                                          Yes, enable PLL operation
45
        040000
                                            $040000
                        enpll
                                    equ
46
        080000
                        disclk
                                            $080000
                                                          Yes, disable CORE clock output
                                    eau
                        PCTL_value equ prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
47
        0E0013
48
49
                        i --- AAR3 value = 0x100831
50
        000001
                        acctype3
                                    equ
                                            1
                                                        ; External Memory access type = 0x1
        000000
                        aahigh3
                                            0
                                                        ; Enable AA3 pin to be low when selected
51
                                    equ
52
        000000
                                            Ω
                                                          No, Enable AA3 pin on ext 'P' accesses
                        aap3
                                    equ
                                                          Yes, Enable AA3 pin on ext 'X' accesses
        000010
53
                                            $10
                        aax3
                                    equ
                                                          Yes, Enable AA3 pin on ext 'Y' accesses
        000020
54
                        aay3
                                    equ
                                            $20
55
        000000
                        aswap3
                                    equ
                                            0
                                                          No, Enable address bus swap
        000000
56
                        enpack3
                                    equ
                                            Λ
                                                          No, Enable packing/unpacking logic
                                            $000800
57
        000800
                        nadd3
                                    equ
                                                          Compare 8 address bits
                                                          Most significant portion of address,
58
        100000
                        msadd3
                                    equ
                                            $100000
59
                                                          $100000 - $10ffff, to compare.
                                                          (0001,0000,xxxx,xxxx,xxxx,xxxx)
60
61
        100831
                    AAR3_value
                                  equ
                                       acctype3+aahigh3+aap3+aax3+aay3+aswap3+enpack3+nadd3+msadd3
62
63
                        ;--- AAR2 value = 0x100815
64
        000001
                        acctype2
                                            1
                                                        ; External Memory access type = 0x1
                                    equ
65
        000004
                        aahigh2
                                                          Enable AA2 pin to be high when selected
                                    equ
                                            $4
66
        000000
                        aap2
                                            0
                                                          No, Enable AA2 pin on ext 'P' accesses
                                    equ
67
        000010
                        aax2
                                    equ
                                            $10
                                                          Yes, Enable AA2 pin on ext 'X' accesses
        000000
                                            0
                                                          No, Enable AA2 pin on ext 'Y' accesses
68
                        aay2
                                    equ
                                                          No, Enable address bus swap
69
        000000
                        aswap2
                                            0
                                    equ
70
        000000
                        enpack2
                                                          No, Enable packing/unpacking logic
                                    equ
71
        000800
                        nadd2
                                            $000800
                                                          Compare 8 address bits
                                    eau
72
        100000
                        msadd2
                                            $100000
                                                        ; Most significant portion of address,
                                    equ
                                                          $100000 - $10ffff, to compare.
73
```



```
74
                                                   ; (0001,0000,xxxx,xxxx,xxxx,xxxx)
75
       100815
                 AAR2_valu equ
                                  acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
76
77
                     i --- BCR value = 0x002400
78
       000000
                    aaa0ws
                                                   ; Address Attribute Area 0 w/s = 0
                                      0
                              equ
79
       000000
                                      0
                                                  ; Address Attribute Area 1 w/s = 0
                    aaa1ws
                               equ
       000400
                                      $000400
                                                  ; Address Attribute Area 2 w/s = 1
; Address Attribute Area 3 w/s = 1
80
                    aaa2ws
                              eau
81
       002000
                    aaa3ws
                                      $002000
                              equ
       000000
                    defws
                                                  ; Default Address Area w/s = 0
82
                              equ
                                      0
       000000
                                      Ω
83
                    husss
                              equ
                                                   ; Bus state status = 0
                                                  ; Enable Bus Lock Hold = 0
       000000
                    enblh equ
                                      0
84
                                                  ; Enable Bus Request Hold = 0
85
       000000
                    enbrh
                              equ
                                      Λ
                    BCR_value equ
86
       002400
                                      aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
87
88
                      89
       P:000100
                        org p:$100
                                                     ;Keep the program in internal RAM
90
91
                     memtst
92
                     ; Initialization Section
93
94
       P:000100 08F4BD
                         movep
                                #PCTL_value,x:PCTL
                                                      ; Set PLL Control Register
                0E0013
       P:000102
                05F43A
95
                         movec #$004000,OMR
                                                       ; Disable Address Attribute Priority
                004000
       P:000104
                05F439
                         movec #$080000,SR
96
                                                       ; Enable 1K Cache
                080000
       P:000106
97
                08F4BB
                         movep #BCR_value,x:BCR
                                                      ; Set external wait states
                002400
98
       P:000108 08F4B7
                         movep #AAR2_value,x:AAR2
                                                      ; Set Address Attribute Reg2
                100815
99
       P:00010A 08F4B6
                         movep #AAR3_value,x:AAR3
                                                      ; Set Address Attribute Reg3
                100831
100
       P:00010C 05F420
                                \#-1,m0
                                                      ; Set LINEAR addressing mode
101
                         move
                FFFFFF
       P:00010E 05F423
102
                                #-1,m3
                         move
                FFFFFF
103
       P:000110 20001B
104
                         clr
                                  h
       P:000111
105
                000000
                         nop
       P:000112
                570000
                                b,x:X_MEM_FAIL_ADDRESS; Initialize X:Failed Address -> $000000
106
                         move
                                b,x:X_MEM_FAIL_WROTE ; Initialize X:Expected Data -> $000000
b,x:X_MEM_FAIL_READ ; Initialize X:Data Read -> $000000
                570100
107
       P:000113
                         move
108
       P:000114
                570200
                         move
109
       P:000115 570300
                         move
                                 b,x:Y_MEM_FAIL_ADDRESS ; Initialize Y:Failed Address -> $000000
110
       P:000116 570400
P:000117 570500
                         move
                                111
                         move
                                b,x:MEM_PASS_COUNTER ; Initialize Pass Counter -> $000000
112
       P:000118 570600
                         move
113
114
                             main
115
                             ;--- fill external memory with initial pattern
116
117
                             P:000119 63F400
                                    #PATT,r3
118
                                                       ; r3 points to Test Patterns
                             move
                00014A
119
       P:00011B 000000
                             nop
120
       P:00011C 000000
                             nop
121
       P:00011D 07DB84
122
                             move
                                    p:(r3)+,x0
                                                        ; Get the Write Pattern for X:MEM
123
       P:00011E 07DB86
                                    p:(r3)+,y0
                                                        ; Get the Write Pattern for Y:MEM
                             move
124
125
       P:00011F 70F400
                                    #MemSize,n0
                                                       ; Get memory size
                             move
                 010000
126
       P:000121 60F400
127
                                    #MemStart,r0
                                                       ; Get starting address for fill
                            move
                100000
128
       P:000123 06D820
                                    n0
                                                        ; Fill X:RAM with first data pattern
                             rep
129
       P:000124 445800
                                    x0,x:(r0)+
                             move
130
       P:000125 60F400
                                    #MemStart.r0
131
                             move
                100000
       P:000127 06D820
132
                             rep
                                    n0
                                                        ; Fill Y:RAM with first data pattern
                                    y0,y:(r0)+
       P:000128 4E5800
133
                             move
134
135
136
                             :--- Check for expected pattern data in each RAM location ---
137
                             ;--- and then replace with a new data pattern.
138
                             ;--- ... This provides an address check. Since erroneous ---
139
                             ;--- ...addressing will cause the data to be written into ---
140
                             ;--- ...incorrect locations and this will be evident in
                             ;--- ...the next read pass.
141
```



```
142
143
       P:000129 063890
                           DOR
                                 #PATTN,test_m
                                                          ; Start Pattern Test Loop
144
       P:00012B 60F400
                                                          ; Get starting address of Test Memory
                                    #MemStart,r0
                            move
                 100000
145
       P:00012D 200041
                            tfr
                                    x0,a
                                                          ; Save the last X test pattern -> a
                                                          ; Save the last Y test pattern -> b ; Get the next X test pattern -> x0
       P:00012E 200059
146
                            tfr
                                    y0,b
                                    P:(r3)+,x0
       P:00012F 07DB84
147
                            move
                                    P:(r3)+,y0
                                                           ; Get the next Y test pattern -> y0
       P:000130 07DB86
148
                             move
149
                              ; Test this pattern through external RAM
150
       P:000131 06D810
                            DOR
151
                                    n0,next_loc
                                                           ; Test all external RAM locations
                 A00000
152
       P:000133 45E000
                             move
                                    X:(r0),x1
                                                           ; Read X:RAM location
153
       P:000134 200065
                                    x1,a
                                                           ; Read data = last test pattern?
                             cmp
154
       P:000135 05240D
                            bne
                                    <XERR
                                                           ; No, error if compare fails
155
                                    x0,X:(r0)
       P:000136 446000
                                                           ; Yes, Write next test pattern -> X:RAM
                             move
156
       P:000137 4FE000
157
                             move
                                    Y:(r0),y1
                                                           ; Read Y:RAM location
158
       P:000138
                 20007D
                                    y1,b
                                                           ; Read data = last test pattern?
                             cmp
                                                           ; No, error if compare fails
159
       P:000139
                 05240D
                                    <YERR
160
       P:00013A
                 4E5800
                             move
                                    y0,Y:(r0)+
                                                           ; Yes, Write next test pattern -> Y:RAM
161
       P:00013B 000000
                            nop
162
                 next_loc
163
       P:00013C 000000
164
                            nop
                                                           ; Time to start next test pattern
165
166
                 test_m
167
168
                                              One Pass Complete
                             ; All test patterns have been tried and passed in external RAM
169
170
       P:00013D 518600
                            move x:MEM_PASS_COUNTER,b0
171
       P:00013E 000009
                                                           ; Update pass loop counter
                            inc
172
       P:00013F 000000
                            nop
173
       P:000140 510600
                            move b0,x:MEM PASS COUNTER
174
175
       P:000141 050F98
                           bra main
                                                           ; Do it all over again
176
177
178
                             ; XERR -- handles X:RAM error messaging with user
179
180
                              ; Expected Data --> a
181
                              ; Read Data --> x1
182
                              ; Address of failure --> r0
183
184
                              XERR
                                    r0,x:X_MEM_FAIL_ADDRESS
185
       P:000142 600000
                              move
                                                                    ; Save off address of failure
       P:000143 560100
186
                              move
                                     a,x:X_MEM_FAIL_WROTE
                                                                     ; Save off expected data
       P:000144 450200
                                    x1,x:X_MEM_FAIL_READ
187
                                                                    ; Save off data read
                              move
188
189
       P:000145 050C00
                              bra
                                                                     ; Dynamically HALT here
190
191
192
                                  YERR -- handles Y:RAM error messaging with user
193
                              ; Expected Data --> b
194
195
                              ; Read Data --> yl
                              ; Address of failure --> r0
196
197
198
                              YERR
                                                                ; Save off address of failure ; Save off expected data
199
       P:000146 600300
                                      r0,x:Y MEM FAIL ADDRESS
                              move
       P:000147 570400
                                    b,x:Y_MEM_FAIL_WROTE
200
                              move
201
       P:000148 470500
                                    y1,x:Y_MEM_FAIL_READ
                                                                    ; Save off data read
                              move
202
       P:000149 050C00
203
                              bra
                                                                     ; Dynamically HALT here
204
205
206
207
                              ; Memory Test Patterns
208
209
       P:00014A
                              PATT
                                      dc
                                              $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
210
       P:00014F
                                      dc
                                              $800000,$400000,$200000,$100000
211
       P:000153
                                      dc
                                              $080000,$040000,$020000,$010000
212
       P:000157
                                      dc
                                              $008000,$004000,$002000,$001000
213
       P:00015B
                                      dc
                                              $000800,$000400,$000200,$000100
                                              $000080,$000040,$000020,$000010
$000008,$000004,$000002,$000001
214
       P:00015F
215
       P:000163
                                      dc
                                              $7FFFFF,$BFFFFF,$DFFFFF,$EFFFFF
216
       P:000167
```



128K × 8-bit Memory Based Designs

```
217
           P:00016B
                                                           dc
                                                                       $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
                                                                      $F'FFFF',$FBFFFF,$FFDFFF',$FFEFFF
$FF7FFF',$FFFBFF',$FFFDFF',$FFFEFF
$FFF7F',$FFFFBF',$FFFFDF',$FFFFEFF
$FFFF7',$FFFFFB,$FFFFD',$FFFFFE
$FEDCBA,$123456',$012345',$EDCBA9
218
           P:00016F
                                                          dc
219
           P:000173
                                                           dc
220
           P:000177
                                                          dc
221
           P:00017B
                                                           dc
222
           P:00017F
                                                           dc
223
224
           P:000183
                                                           dc
                                                                       $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
                                                                       $800000,$400000,$200000,$100000
$080000,$040000,$020000,$010000
           P:000188
225
                                                           dc
226
           P:00018C
                                                          dc
227
           P:000190
                                                                       $008000,$004000,$002000,$001000
                                                           dc
           P:000194
                                                                       $000800,$000400,$000200,$000100
$000080,$000040,$000020,$000010
$000008,$000004,$000002,$000001
228
                                                           dc
                                                          dc
229
           P:000198
           P:00019C
230
                                                          dc
                                                                       $7FFFFF,$BFFFFF,$DFFFFFF,$EFFFFF
$F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
231
           P:0001A0
                                                           dc
232
           P:0001A4
                                                           dc
                                                                       $FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
$FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
           P:0001A8
233
                                                           dc
234
           P:0001AC
                                                           dc
                                                                       $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
$FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
235
           P:0001B0
                                                           dc
236
           P:0001B4
                                                           dc
237
           P:0001B8
                                                           dc
                                                                       $FEDCBA, $123456, $012345, $EDCBA9
238
239
           000038
                                               PATTN
                                                           equ
                                                                       ((*-PATT)/2)-1)
240
241
                                               end
                                                           memtst
```

0 Errors 0 Warnings



5 64K × 24-bit Memory Based Designs

This section describes how to implement several different DSP memory space Fast SRAM configuration designs using one $64K \times 24$ -bit 3.3 V memory with a Motorola DSP56303 device.

Using one hardware memory design based on one $64K \times 24$ -bit 3.3 V memory, the DSP56303's Memory Expansion Port allows the memory bank to be logically configured for use in various memory space arrangements.

Configuring and using one Memory Expansion Port Address Attribute Control Register, the memory bank can accommodate seven different memory space arrangements:

- 1. $64K \times 24$ -bit 'P' Space Fast SRAM
- 2. $64K \times 24$ -bit 'X' Space Fast SRAM
- 3. 64K × 24-bit 'Y' Space Fast SRAM
- 4. $64K \times 24$ -bit 'P'/'X' Space Fast SRAM
- 5. 64K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. $64K \times 24$ -bit 'X'/'Y' Space Fast SRAM
- 7. $64K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM

Configuring and using two Memory Expansion Port Address Attribute Control Registers, the memory bank can accommodate thirteen different memory space arrangements:

- 1. $64K \times 24$ -bit 'P' Space Fast SRAM
- 2. $64K \times 24$ -bit 'X' Space Fast SRAM
- 3. $64K \times 24$ -bit 'Y' Space Fast SRAM
- 4. $64K \times 24$ -bit 'P'/'X' Space Fast SRAM
- 5. 64K × 24-bit 'P'/'Y' Space Fast SRAM
- 6. $64K \times 24$ -bit 'X'/'Y' Space Fast SRAM
- 7. $64K \times 24$ -bit 'P'/'X'/'Y' Space Fast SRAM
- 8. $32K \times 24$ -bit 'P'/'X' and $32K \times 24$ -bit 'Y' Space Fast SRAM
- 9. $32K \times 24$ -bit 'P'/'Y' and $32K \times 24$ -bit 'X' Space Fast SRAM
- 10. $32K \times 24$ -bit 'P' and $32K \times 24$ -bit 'X'/'Y' Space Fast SRAM
- 11. $32K \times 24$ -bit 'P' and $32K \times 24$ -bit 'X' Space Fast SRAM
- 12. $32K \times 24$ -bit 'P' and $32K \times 24$ -bit 'Y' Space Fast SRAM
- 13. $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' Space Fast SRAM



The memory configuration examples presented in the remainder of this section are based on one common hardware design that uses two Address Attributes and implements two of the most common configurations: $32K \times 24$ -bit 'P'/'X' and $32K \times 24$ -bit 'Y' Space Fast SRAM configuration and a $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' Space Fast SRAM configuration (see **Figure 5-4** for a schematic of the hardware design).

5.1 64K × 24-bit Common Fast SRAM Hardware Design

This section describes an asynchronous Fast SRAM 64K × 24-bit memory bank implementation using GSI Technology's GS71024T device (see **Figure 5-1**). Memory bank designs using single memory devices allow for compact embedded designs.

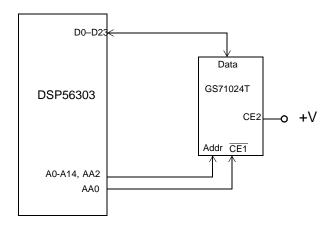


Figure 5-1. 64K × 24-bit Fast SRAM Memory Example

The memory bank design is implemented using two Address Attribute Selectors, AA2 and AA0. AA2 splits the 64K address space of the memory bank into two 32K address spaces. AA0 enables the memory bank.

For this common hardware design, the DSP core runs at a maximum of 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements. This device is organized as $64K \times 24$ -bits. Therefore, one memory device satisfies the 24-bit wide general-purpose DSP bus.

Since the memory device operates at 3.3 V, no voltage level conversion is required.

5.1.1 GS71024T-12 Memory Timing Requirements

For the asynchronous Fast SRAM device to work properly, its timing requirements must be met. Following are the timing requirements for the GS71024T-12 $64K \times 24$ -bit 12 nS Fast SRAM.



5.1.1.1 Read Cycle Timing

Table 5-1 contains the memory read timing specification values used in the memory read cycle timing diagram, **Figure 5-2**.

Read Cycle Parameter	Symbol	Min	Max
Read Cycle Time	t _{AVAV}	12 nS	_
Address Access Time	t _{AVQV}	_	12 nS
Enable Access Time	t _{ELQV}	_	12 nS
Output Enable Access Time	t _{GLQV}	_	6 nS
Enable High to Output High-Z	t _{EHQZ}	_	6 nS
Output Enable High to Output High-Z	t _{GHQZ}	_	6 nS

Table 5-1. GS71024T-12 Memory Read Timing Specifications

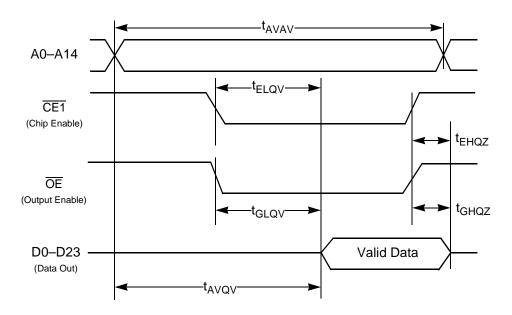


Figure 5-2. GS71024T-12 Memory Read Cycle Timing Diagram



5.1.1.2 Write Cycle Timing

Table 5-2 shows the memory write timing values in the memory write cycle timing diagram, **Figure 5-3**.

Write Cycle Parameter	Symbol	Min	Max
Write Cycle Time	t _{AVAV}	12 nS	_
Address Setup Time	t _{AVWL}	0 nS	_
Address Valid to End off Write	t _{AVWH}	8 nS	_
Write Pulse Width	t _{WLWH}	8 nS	_
Data Valid to End of Write	t _{DVWH}	3 nS	_
Data Hold Time	t _{WHDX}	0 nS	_

Table 5-2. GS71024T-12 Memory Write Timing Specifications

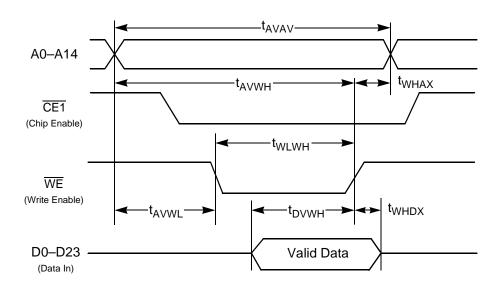


Figure 5-3. GS71024T-12 Memory Write Cycle Timing Diagram

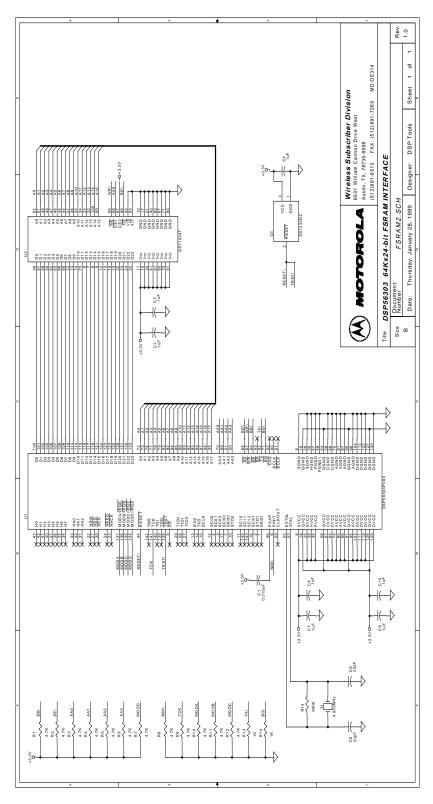


Figure 5-4. 128K × 32-bit Fast SRAM Schematic



5.2 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y' Fast SRAM Example

This section describes a $64K \times 24$ -bit shared 'P'/'X' and $32K \times 24$ -bit 'Y' memory space, asynchronous Fast SRAM implementation using GSI's GS71024T device (see **Figure 5-5** for memory map layout, **Figure 5-1** for the block diagram, and **Example 5-1** for the example code). In a shared memory space, data written to one address in one memory space can be accessed by the same address in another memory space (e.g., writing \$012345 to P:\$100000 could be read back by X:\$100000 or P:\$100000).

The DSP core runs at 80 MHz, and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

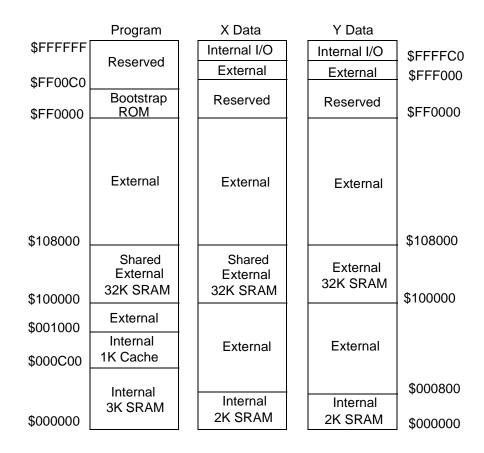


Figure 5-5. $32K \times 24$ -bit 'P'/'X' and $32K \times 24$ -bit 'Y' Fast SRAM Memory Map

5.2.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $32K \times 24$ -bit 'P'/'X' and $32K \times 24$ -bit 'Y' space memory configuration, set up the following DSP control registers.



Set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation register (PCTL). For this example, the DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:

- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is \$0E0013.

Address Attribute Pin 0 enables, via Fast SRAM CE1, external 64K SRAM bank accesses in the address range from \$100000 to \$107FFF during Program/X data, and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 0 using Address Attribute Register 0 (AAR0). The AAR0 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, bits 0-1 = \$1
- Pull the AA pin high when selected, Bit 2 = 0
- Activate the AA pin during external program space accesses, Bit 3 = 1
- Activate the AA pin during external X data space accesses, Bit 4 = 1
- Activate the AA pin during external Y data space accesses, Bit 5 = 1
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0
- Specify the number of address bits to compare, Bits 8-11 = \$9
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR0 is \$100939.



Address Attribute Pin 2 selects, via Fast SRAM A15, the external 64K SRAM bank during accesses in the address range from \$100000 to \$107FFF between program/X data space requests and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1
- Pull the AA pin high when selected, Bit 2= 1
- Activate the AA pin during external program space accesses, Bit 3 = 0
- Activate the AA pin during external X data space accesses, Bit 4 = 0
- Activate the AA pin during external Y data space accesses, Bit 5 = 1
- Move the eight least significant bits of the address to eight most significant bits of the external address bus, Bit 6 = 0
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0
- Number of address bits to compare, Bits 8–11 = \$9
- Most significant portion of address to compare, Bits 12–23 = \$100

The value loaded into the AAR2 register is \$100925.

Select the proper number of wait states must for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address attribute area 0 wait states, Bits 0-4 = \$1
- Address attribute area 1 wait states, Bits 5-9 = \$1
- Address attribute area 2 wait states, Bits 10–12 = \$1
- Address attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16–20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 =0

The value loaded into the BCR is \$002421.



Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value combines the following bits for each feature:

- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14 = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 5-1. 32K × 24-bit 'P'/'X' and 32K × 24-bit 'Y Space Fast SRAM Memory Exercise

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 08:05:33 asram6.asm

```
1
                                                  132,60,3,3,
                                         page
2
                               ASRAM6.ASM - Simple program to test 32Kx24-bit of Program/X-Data and 32Kx24-bits of Y-Data memory
3
5
                                               using a DSP56303
6
                                The program uses Internal P:RAM to test:
8
                                        24-bit External P:/X: & Y:RAM from $100000 - $107FFF @ 1w/s
10
11
12
        100000
                         PMemStart
                                                  $100000
                                          equ
13
        108000
                         PMemEnd
                                                  $108000
                                          equ
        008000
14
                         PMemSize
                                          eau
                                                  PMemEnd-PMemStart
15
        100000
                         YMemStart
                                                  $100000
16
                                          eau
17
        108000
                         YMemEnd
                                                  $108000
                                          equ
18
        008000
                         YMemSize
                                                  YMemEnd-YMemStart
                                          equ
19
20
                         :--- Program Specific Storage Locations (X DATA SPACE)
                         P_MEM_FAIL_ADDRESS
21
2.2
        000000
                                                  $000000
23
                         P_MEM_FAIL_WROTE
        000001
24
                                                  $000001
                                          equ
25
                         P_MEM_FAIL_READ
        000002
2.6
                                                  $000002
                                          equ
27
28
                         Y_MEM_FAIL_ADDRESS
29
        000003
                                                  $000003
30
                         Y_MEM_FAIL_WROTE
31
        00004
                                                  $000004
                                          equ
32
                         Y_MEM_FAIL_READ
33
        000005
                                                  $000005
34
35
                         MEM_PASS_COUNTER
36
        000009
                                                  $000009
                                          equ
37
38
                         old_p_pattern
39
        00000A
                                                  $00000A
                                                                   ; Last written pattern to P:/X:RAM
                                          equ
40
                         old_y_pattern
        00000B
                                                  $00000B
41
                                          equ
                                                                   ; Last written pattern to Y:RAM
42
43
                         ;--- DSP56303 Control Registers (X I/O SPACE)
44
        FFFFFB
                         BCR
                                          $FFFFFB
                                                          ; Bus Control Register
                                 eau
45
        FFFFFD
                         PCTL
                                          $FFFFFD
                                                            PLL Control Register
                                 equ
        FFFFF6
                         AAR3
                                          $FFFFF6
46
                                                           ; Address Attribute Register #3
                                 equ
47
                         AAR2
                                          $FFFFF7
                                                            Address Attribute Register #2
        FFFFF7
                                 equ
48
                                          $FFFFF8
                                                           ; Address Attribute Register #1
        FFFFF8
                         AAR1
                                 equ
49
                                          SFFFFF9
                                                           ; Address Attribute Register #0
        FFFFF9
                         AAR 0
                                 equ
50
51
                         i--- PCTL value = 0 \times 0 \times 0 \times 0 \times 13
        000000
                         prediv equ
                                          0
                                                           ; Pre-Divider = 1
52
        000000
                                                            Low Power Divider = 1
53
                                          Ω
                         lowdiv
                                 equ
        000013
                                                            VCO Mult = 20; (19+1)*4.00MHz=80.00MHz
54
                                          19
                         pllmul
                                 equ
55
        000000
                                                            No, Crystal not less than 200 \mathrm{kHz}
                         crystal equ
                                          Λ
                                                           ; No, do not disable crystal use
        000000
56
                         disXTAL equ
                                          $020000
57
        020000
                         pllstop equ
                                                            Yes, PLL runs during STOP
58
        040000
                         enpll
                                 equ
                                          $040000
                                                            Yes, enable PLL operation
                         disclk equ
59
        080000
                                          $080000
                                                           ; Yes, disable CORE clock output
60
        0E0013
                         PCTL_value
                                     equ
                                            \verb|prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk||
61
62
                         ;--- AAR2 value = 0x100925
63
        000001
                         acctype2
                                                           ; External Memory access type = 0x1
                                      equ
                                              1
64
        000004
                         aahigh2
                                              $4
                                                            Enable AA2 pin to be high when selected
                                      equ
65
        000000
                         aap2
                                              0
                                                           ; No, Enable AA2 pin on ext 'P' accesses
                                      equ
                                                            No, Enable AA2 pin on ext 'X' accesses
Yes, Enable AA2 pin on ext 'Y' accesses
        000000
                         aax2
                                              0
66
                                      equ
67
        000020
                         aay2
                                              $20
                                      equ
68
        000000
                         aswap2
                                      equ
                                              0
                                                             No, Enable address bus swap
        000000
69
                         enpack2
                                                            No, Enable packing/unpacking logic
                                      equ
70
                                              $000900
        000800
                         nadd2
                                                             Compare 9 address bits
                                      equ
                                                            Most significant portion of address,
71
        100000
                         {\tt msadd2}
                                              $100000
                                      equ
72
                                                             $100000 - $107fff, to compare.
73
                                                             (0001,0000,0xxx,xxxx,xxxx,xxxx)
74
        100925
                         AAR2_value equ acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
```



```
76
                       ; --- AAR0 value = 0x100939
77
        000001
                       acctype0
                                                       ; External Memory access type = 0x1
                                    eau
                                           1
78
                       aahigh0
                                           $0
        000004
                                                       ; Enable AAO pin to be high when selected
                                    equ
                                                       ; Yes, Enable AAO pin on ext 'Y' accesses
; Yes, Enable AAO pin on ext 'X' accesses
79
        000008
                       aap0
                                           $8
                                    equ
80
        000010
                       aax0
                                           $10
                                    equ
                                                       ; Yes, Enable AAO pin on ext 'Y' accesses
; No, Enable address bus swap
        000020
                                           $20
81
                       aav0
                                    equ
        000000
82
                       aswap0
                                           0
                                    equ
        000000
                       enpack0
                                                       ; No, Enable packing/unpacking logic
83
                                    equ
                                           0
                                           $000900
84
        000900
                                                       ; Compare 9 address bits
                       nadd0
                                    equ
                       msadd0
                                           $100000
       108000
85
                                                       ; Most significant portion of address,
                                    equ
                                                         $100000 - $107fff, to compare.
86
                                                         (0001,0000,0xxx,xxxx,xxxx,xxxx)
87
       100939
                       AARO_value equ acctype0+aahigh0+aap0+aax0+aay0+aswap0+enpack0+nadd0+msadd0
88
89
90
                       i --- BCR value = 0x002421
                       aaa0ws
91
       000001
                                    equ
                                           $1
                                                       ; Address Attribute Area 0 w/s = 1
92
        000020
                       aaalws
                                    equ
                                           $20
                                                       ; Address Attribute Area 1 w/s = 1
                       aaa2ws
                                           $000400
93
        000400
                                                       ; Address Attribute Area 2 w/s = 1
                                    equ
94
        002000
                       aaa3ws
                                           $002000
                                                       ; Address Attribute Area 3 w/s = 1
                                    equ
95
        000000
                                           0
                                                       ; Default Address Area w/s = 0
                       defws
                                    equ
96
        000000
                       busss
                                    equ
                                           0
                                                       ; Bus state status = 0
97
        000000
                       enblh
                                    equ
                                           0
                                                       ; Enable Bus Lock Hold = 0
98
        000000
                                           0
                                                       ; Enable Bus Request Hold = 0
                       enbrh
                                    equ
99
        002421
                       BCR_value equ aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
100
101
       P:000100
                                        p:$100 ;Keep the program in internal RAM
102
103
104
                       memtst
105
106
                       ; Initialization Section
                                      #PCTL_value,x:PCTL
       P:000100 08F4BD
                                                               ; Set PLL Control Register
107
                               movep
                  0E0013
108
       P:000102 05F43A
                                       #$004000,OMR
                                                               ; Disable Address Attribute Priority
                               movec
                  004000
       P:000104 05F439
109
                                       #$080000,SR
                                                               ; Enable 1K Cache
                               movec
                  080000
       P:000106 08F4BB
110
                                       #BCR value,x:BCR
                                                               ; Set external memory wait states
                               movep
                  002421
       P:000108 08F4B9
111
                                       #AAR0 value,x:AAR0
                                                               ; Set Address Attribute Reg0
                               movep
                  108930
       P:00010C 08F4B7
112
                                                               ; Set Address Attribute Reg2
                               movep
                                       #AAR2_value,x:AAR2
                  100825
113
       P:000110 05F420
114
                               move
                                       \#-1.m0
                                                               ; Set LINEAR addressing mode
                  FFFFFF
115
       P:000112 05F422
                               move
                                       \#-1,m2
                  FFFFFF
116
       P:000114 05F423
                                       \#-1,m3
                               move
                  FFFFFF
117
       P:000116 05F424
                               move
                                       \#-1,m4
                  FFFFFF
       P:000118 05F425
                               move
                                       \#-1,m5
                  FFFFFF
119
120
       P:00011A 20001B
                               clr b
121
       P:00011B 000000
                               nop
                 570000
       P:00011C
122
                               move b,x:P_MEM_FAIL_ADDRESS ; Initialize P/X:Failed Address -> $000000
123
       P:00011D 570100
                               move b,x:P_MEM_FAIL_WROTE
                                                               ; Initialize P/X:Expected Data -> $000000
       P:00011E 570200
124
                               move b,x:P_MEM_FAIL_READ
                                                               ; Initialize P/X:Data Read -> $000000
125
126
       P:00011F 570300
                               move b,x:Y MEM FAIL ADDRESS
                                                               ; Initialize Y:Failed Address -> $000000
       P:000120 570400
                               move b,x:Y_MEM_FAIL_WROTE
127
                                                               ; Initialize Y: Expected Data -> $000000
       P:000121 570500
128
                               move b,x:Y MEM FAIL READ
                                                               ; Initialize Y:Data Read -> $000000
129
       P:000125 570900
130
                               move b,x:MEM PASS COUNTER
                                                               ; Initialize Pass Counter -> $000000
131
132
                       main
133
                               Fill memory spaces with Test Patterns
134
135
                                             ______
       P:000126 65F400
                               move #PATT,r5
                                                               ; r5 points to Test Patterns
136
                  00018C
       P:000128 000000
137
                               nop
138
       P:000129 000000
                               nop
139
140
                       ; Fill P/X:RAM
       P:00012A 07DD84
                                       p:(r5)+,x0
141
                               move
                                                               ; Get the Write Pattern
142
       P:00012B 70F400
                               move
                                       #PMemSize,n0
                                                               ; Get memory size
                  008000
```



```
143
       P:00012D 60F400
                             move
                                     #PMemStart,r0
                                                            ; Get starting address for fill
                100000
144
145
       P:00012F 06D820
                                                            ; Fill P:RAM with first data pattern
                             rep
       P:000130 075884
146
                                     x0,p:(r0)+
                             move
147
148
       P:000131 070A04
                             move
                                     x0,p:old_p_pattern
                                                           ; Save the written pattern
149
                      ; Fill Y:RAM
150
151
       P:000132 07DD84
                                                            ; Get the Next Write Pattern
                                     p:(r5)+,x0
                             move
       P:000133 72F400
                                     #YMemSize,n2
                                                            ; Get memory size
152
                              move
                008000
       P:000135 62F400
153
                             move
                                     #YMemStart,r2
                                                            ; Get starting address for fill
                100000
154
       P:000137 06DA20
155
                             rep
                                     n2
                                                            ; Fill Y:RAM with data pattern
156
       P:000138 4C5A00
                             move
                                     x0,y:(r2)+
157
158
       P:000139 070B04
                             move
                                     x0,p:old_y_pattern ; Save the written pattern
159
160
161
                            Cycle through each memory space with a pattern
162
163
                      ;--- Check for expected pattern data in each RAM location ---
164
                      ;--- and then replace with a new data pattern.
165
166
                      ;--- ... This provides an address check. Since erroneous
                      ;--- ...addressing will cause the data to be written into ---
167
168
                       ;--- ...incorrect locations and this will be evident in
169
                      ;--- ...the next read pass.
170
171
       P:000149 063890
                             DOR #PATTN,test_mem ; Start Pattern Test Loop
                 000031
172
                      ; Do a pass through P/X:RAM
173
       P:00014B 60F400
                             move #PMemStart,r0
                                                     ; Get starting address of Test P: Memory
                 100000
174
       P:00014D 078A0E
                             move p:old_p_pattern,a ; Get the last P test pattern -> a
       P:00014E 07DD84
175
                             move p:(r5)+,x0
                                                      ; Get the next P test pattern -> x0
176
177
                      ; Test this pattern through external P/X:RAM
       P:00014F 06D810
                                                      ; Test all external P/X:RAM locations
178
                             DOR n0,next_p_loc
                000007
179
       P:000151 07E085
                                                      ; Read P:RAM location
                             move p:(r0),x1
180
       P:000152 200065
                              cmp x1,a
                                                      ; Read data = last test pattern?
181
       P:000153 05244D
                              bne
                                   <PERR
                                                      ; No, error if compare fails
182
       P:000154 075884
                             move x0,p:(r0)+
                                                     ; Yes, Write next test pattern -> P:RAM
183
184
       P:000155 070A04
                             move x0,p:old_p_pattern; Save the newly written test pattern
185
       P:000156 000000
                             nop
186
                      next_p_loc
187
188
                      ; Do a pass through Y:RAM
189
       P:000157 62F400
                             move #YMemStart,r2
                                                      ; Get starting address of Test Y: Memory
                 100000
190
       P:000159 078B0E
                             move p:old_y_pattern,a ; Get the last Y test pattern -> a
                             move p:(r5)+,x0
191
       P:00015A 07DD84
                                                      ; Get the next Y test pattern -> x0
192
193
                      ; Test this pattern through external Y:RAM
       P:00015B 06DA10
194
                             DOR n2,next_y_loc
                                                      ; Test all external Y:RAM locations
                000007
195
       P:00015D 4DE200
                             move y:(r2),x1
                                                      ; Read Y:RAM location
196
       P:00015E 200065
                                  x1,a
                                                     ; Read data = last test pattern?
                             cmp
                                                     ; No, error if compare fails
; Yes, Write next test pattern -> Y:RAM
197
       P:00015F
                052445
                                    <YERR
                             bne
198
       P:000160 4C5A00
                             move x0,y:(r2)+
199
200
       P:000161 070B04
                             move x0,p:old_y_pattern ; Save the newly written test pattern
       P:000162 000000
201
                             nop
202
                      next_y_loc
       P:00017A 000000
                             nop
203
                                                       ; Time to start next test pattern
204
                      test_mem
205
206
                                     One Pass Complete
                      ; All test patterns have been tried and passed in external RAM
207
208
       P:00017B 518900
                             move x:MEM_PASS_COUNTER,b0
209
       P:00017C 000009
                             inc
                                   b
                                                        ; Update pass loop counter
210
       P:00017D 000000
                             nop
211
       P:00017E 510900
                             move b0,x:MEM_PASS_COUNTER
212
213
       P:00017F 050F47
                             bra main
                                                        ; Do it all over again
214
```



```
215
216
                               PERR -- handles P/X:RAM error messaging with user
217
218
                           ; Expected Data --> a
                           ; Read Data --> x1
; Address of failure --> r0
219
220
221
2.2.2
                                   move r0,x:P_MEM_FAIL_ADDRESS ; Save off address of failure move a,x:P_MEM_FAIL_WROTE ; Save off expected data
        P:000180 600000
223
224
        P:000181 560100
                                                                       ; Save off data read
        P:000182 450200
                                   move x1,x:P_MEM_FAIL_READ
225
226
        P:000183 050C00
2.27
                                   bra
                                                                       ; Dynamically HALT here
228
229
                           ;-----
230
                               YERR -- handles Y:RAM error messaging with user
231
232
                           ; Expected Data --> b
233
                           ; Read Data --> x1
                           ; Address of failure --> r2
234
235
236
237
        P:000184 620300
                                   move r2,x:Y_MEM_FAIL_ADDRESS ; Save off address of failure
                                   move b,x:Y_MEM_FAIL_WROTE ; Save off expected data move x1,x:Y_MEM_FAIL_READ ; Save off data read
238
        P:000185 570400
239
        P:000186 450500
                                   move x1,x:Y_MEM_FAIL_READ
240
241
        P:000187 050C00
                                   bra
                                                                        ; Dynamically HALT here
242
243
                           ; Memory Test Patterns
244
245
        P:00018C
                          PATT
                                                      $000000, $FFFFFF, $AAAAAA, $555555, $2BAD2C
                                                      $800000,$400000,$200000,$100000
$080000,$040000,$020000,$010000
$008000,$004000,$002000,$001000
$000800,$000400,$000200,$000100
246
        P:000191
                                          dc
247
        P:000195
                                          dc
248
        P:000199
                                          dc
249
        P:00019D
                                          dc
                                                      $000080,$000040,$000020,$000010
$000008,$000004,$000002,$000001
        P:0001A1
250
                                          dc
251
        P:0001A5
                                          dc
                                                      $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
252
        P:0001A9
                                          dc
                                                      $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
253
        P:0001AD
                                          dc
                                                      $FF7FFF,$FFBFFF,$FFFFFFF,$FFFFFFF
254
        P:0001B1
                                          da
255
        P:0001B5
                                          dc
                                                      $FFFF7F,$FFFFBF,$FFFFFFF
256
        P:0001R9
                                          dc
257
        P:0001BD
                                                      $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
                                          dc
258
        P:0001C1
                                                      $FEDCBA, $123456, $012345, $EDCBA9
                                          dc
259
260
        P:0001C5
                                          dc
                                                      $000000,$FFFFFFF,$AAAAAA,$555555,$2BAD2C
$800000,$400000,$200000,$100000
261
        P:0001CA
                                          dc
262
        P:0001CE
                                          dc
                                                      $080000,$040000,$020000,$010000
                                                      $008000,$004000,$002000,$001000
263
        P:0001D2
                                          dc
264
        P:0001D6
                                          dc
                                                      $000800,$000400,$000200,$000100
265
        P:0001DA
                                                      $000080,$000040,$000020,$000010
                                          dc
266
        P:0001DE
                                          dc
                                                      $000008,$000004,$000002,$000001
267
        P:0001E2
                                          dc
                                                      $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
                                                      $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
$FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
        P:0001E6
268
269
        P:0001EA
                                          dc
270
        P:0001EE
                                                      $FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
271
        P:0001F2
                                          dc
                                                      $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
272
        P:0001F6
                                                      $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
273
        P:0001FA
                                          dc
                                                      $FEDCBA, $123456, $012345, $EDCBA9
274
275
        P:0001FE
                                          dc
                                                      $000000, $FFFFFF, $AAAAAA, $555555, $2BAD2C
276
        P:000203
                                                      $800000,$400000,$200000,$100000
                                          dc
                                                      $080000,$040000,$020000,$010000
$008000,$004000,$002000,$001000
277
        P:000207
                                          dc
278
        P:00020B
                                          dc
                                                      $000800,$000400,$000200,$000100
$000080,$000040,$000020,$000010
279
        P:00020F
                                          dc
280
        P:000213
                                          dc
281
                                                      $000008,$000004,$000002,$000001
        P:000217
                                          dc
                                                      $7FFFFF,$BFFFFFF,$DFFFFF,$EFFFFF
282
        P:00021B
                                          da
                                                      $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
$FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
        P:00021F
283
                                          dc
        P:000223
284
                                          dc
                                                      $FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
$FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
        P:000227
285
                                          dc
286
        P:00022B
                                          dc
                                                      $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
2.87
        P:00022F
                                          dc
288
        P:000233
                                          dc
                                                      $FEDCBA, $123456, $012345, $EDCBA9
289
290
        000038
                          PATTN
                                          equ
                                                      ((*-PATT)/3)-1
291
292
                                           end
                                                     memtst
0
      Errors
      Warnings
```



5.3 $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' Fast SRAM Example

This section describes a $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' memory space, asynchronous Fast SRAM implementation using GSI's GS71024T device (see **Figure 5-6** for memory map layout, **Figure 5-1** for the block diagram, and **Example 5-2** for the example code).

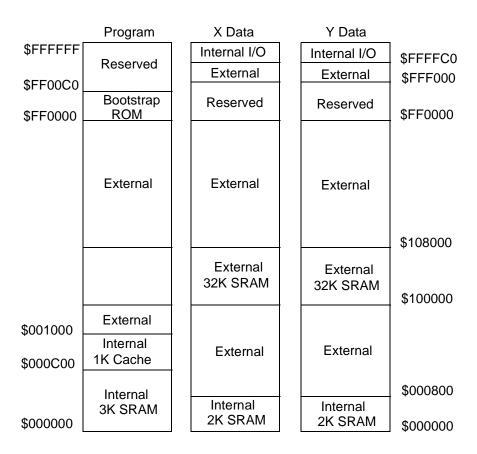


Figure 5-6. $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' Memory Map

The DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The asynchronous Fast SRAM requires an access time equal to or less than 12.4 nS to satisfy the DSP's one wait state external memory requirements.

5.3.1 DSP56303 Port A Timing Requirements and Register Settings

For optimal use of the $32K \times 24$ -bit 'X' and $32K \times 24$ -bit 'Y' space memory configuration, set up the following DSP control registers.

Set the core speed of the DSP for optimum processor and memory performance using the DSP PLL and Clock Generation register (PCTL). In this example, the DSP core runs at 80 MHz and the input frequency source is a 4.000 MHz crystal. The PCTL register value combines the following bits for each feature:



- Desired Core Frequency = 80 MHz
- Given the External Frequency = 4.000 MHz
- Predivider value = 1, Bits 20-23 = \$0
- Low-power Divider value = 1, Bits 12-14 = \$0
- VCO Multiplication value = 20, Bits 0–11 = \$013
- Crystal less than 200 kHz, Bit 15 = 0
- Disable XTAL drive output, Bit 16 = 0
- PLL runs during STOP, Bit 17 = 1
- Enable PLL operation, Bit 18 = 1
- Disable core clock output, Bit 19 = 1

The value loaded into the PCTL register is be \$0E0013.

Address Attribute Pin 0 enables, via Fast SRAM $\overline{\text{CET}}$, the external 64K SRAM bank accesses in the address range \$100000 to \$107FFF during X data and Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 0 using Address Attribute Register 0 (AAR0). The AAR0 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 0.
- Activate the AA pin during external program space accesses, Bit 3 = 0.
- Activate the AA pin during external X data space accesses, Bit 4 = 1.
- Activate the AA pin to during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$9
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR0 is \$100931.



Address Attribute Pin 2 selects, via Fast SRAM A15, the external 64K SRAM bank during accesses in the address range \$100000 to \$107FFF to differentiate X data requests from Y data space requests. Configure the memory address space requirements for the Address Attribute Pin 2 using Address Attribute Register 2 (AAR2). The AAR2 value combines the following bits for each feature:

- Specify the external memory access type as asynchronous SRAM, Bits 0-1 = \$1.
- Pull the AA pin high when selected, Bit 2 = 1.
- Activate the AA pin during external program space accesses, Bit 3 = 0.
- Activate the AA pin during external X data space accesses, Bit 4 = 0.
- Activate the AA pin during external Y data space accesses, Bit 5 = 1.
- Move the eight least significant bits of the address to the eight most significant bits of the external address bus, Bit 6 = 0.
- Enable the internal packing/unpacking logic during external DMA accesses, Bit 7 = 0.
- Specify the number of address bits to compare, Bits 8-11 = \$9
- Specify the most significant portion of the address to compare, Bits 12-23 = \$100

The value loaded into the AAR2 is \$100925.

Select the proper number of wait states for the memory configuration using the Bus Control Register (BCR). The BCR value combines the following bits for each feature:

- Address attribute area 0 wait states, Bits 0-4 = \$1
- Address attribute area 1 wait states, Bits 5-9 = \$1
- Address attribute area 2 wait states, Bits 10–12 = \$1
- Address attribute area 3 wait states, Bits 13–15 = \$1
- Default address area wait states, Bits 16–20 = \$0
- Bus state status, Bit 21 = 0
- Enable Bus Lock Hold, Bit 22 = 0
- Enable Bus Request Hold, Bit 23 =0

The value loaded into the BCR is \$002421.

Configure the operating mode and external memory controls using the Operating Mode Register (OMR). The OMR value combines the following bits for each feature:



- MA–MD bits specify the DSP operating mode, Bits 0-3 = \$0.
- External Bus Disable Bit disables the external bus controller for power conservation when external memory is not used, Bit 4 = \$0.
- Memory Switch Mode bit reconfigures internal memory spaces, Bit 7 = \$0.
- Transfer Acknowledge Synchronize Select bit selects the synchronization method for the Transfer Acknowledge (TA) pin, Bit 11 = \$0.
- Bus Release Timing bit selects between a fast and slow bus release of the \overline{BB} pin, Bit 12 = \$0.
- Address Attribute Priority Disable bit allows the Address Attribute pins, AA0–AA3, to be used in any combination, Bit 14 = \$1.
- All other OMR bits are selected for their defaults of \$000000.

The value loaded into the OMR is \$004000.

Configure the memory mode of the DSP using the Status Register (SR). The SR value combines the following bits for each feature:

- Sixteen-Bit Compatibility mode enables full compatibility to object code written for the DSP56000 family of DSPs, Bit 13 = \$0.
- Instruction Cache Enable bit enables the instruction cache controller and changes the last 1K of internal program memory into cache memory, Bit 19 = \$1.
- All other Status Register bits are selected for their defaults of \$000000.

The value loaded into the SR is \$080000, which is the value loaded during reset.



Example 5-2. 32K × 24-bit 'X' and 32K × 24-bit 'Y' Space Fast SRAM Memory Exercise

Motorola DSP56300 Assembler Version 6.0.1.6 97-01-25 09:27:34 asram7.asm

```
1
                                         page
                                                  132,60,3,3,
2
                                ASRAM7.ASM
                                               Simple program to test
                                               32Kx24-bits of X-Data and 32Kx24-bits of Y-Data
5
                                               memory using a DSP56303
                                The program uses Internal P:RAM to test:
8
                                       24-bit External X: & Y:RAM from $100000 - $107FFF @ 1w/s
                                                  $100000
10
        100000
                         XMemStart
                                          equ
11
        108000
                         XMemEnd
                                          equ
                                                  $108000
        008000
12
                         XMemSize
                                                  XMemEnd-XMemStart
                                          equ
13
14
        100000
                         YMemStart
                                          equ
                                                  $100000
        108000
                         YMemEnd
                                                  $108000
15
                                          equ
16
        008000
                         YMemSize
                                                  YMemEnd-YMemStart
                                          equ
17
18
                         ;--- Program Specific Storage Locations (X DATA SPACE)
19
                         X_MEM_FAIL_ADDRESS
20
        000003
                                                  $000003
                                          equ
                         X_MEM_FAIL_WROTE
21
22
        000004
                                                  $000004
                                          equ
                         X_MEM_FAIL_READ
23
        000005
24
                                                  $000005
25
26
                         Y_MEM_FAIL_ADDRESS
27
        000006
                                                  $000006
2.8
                         Y_MEM_FAIL_WROTE
29
        000007
                                                  $000007
30
                         Y_MEM_FAIL_READ
        800000
31
                                                  $000008
32
33
                         MEM_PASS_COUNTER
34
        00000C
                                                  $0000C
35
36
                         old_x_pattern
37
        00000E
                                          equ
                                                  $00000E
                                                                   ; Last written pattern to X:RAM
38
                         old_y_pattern
39
        00000F
                                                  $00000F
                                                                   ; Last written pattern to Y:RAM
                                          equ
40
41
                         ;--- DSP56303 Control Registers (X I/O SPACE)
42
        FFFFFB
                         BCR
                                                  $FFFFFB
                                                                   ; Bus Control Register
                                          equ
                                                                   ; PLL Control Register
43
        FFFFFD
                         PCTL
                                          eau
                                                  $FFFFFD
44
        FFFFF6
                         AAR3
                                                  $FFFFF6
                                                                   ; Address Attribute Register #3
                                          equ
45
        FFFFF7
                         AAR2
                                                  $FFFFF7
                                                                   ; Address Attribute Register #2
                                          equ
46
                         AAR1
                                                                     Address Attribute Register #1
        FFFFF8
                                                  $FFFFF8
                                          eau
47
        FFFFF9
                         AAR0
                                                  $FFFFF9
                                                                   ; Address Attribute Register #0
                                          equ
48
                         i--- PCTL value = 0 \times 0 \times 0 \times 0 \times 13
49
        000000
50
                         prediv
                                                  0
                                                                   ; Pre-Divider = 1
                                          equ
        000000
                                                                    Low Power Divider = 1
51
                         lowdiv
                                          equ
                                                  0
                                                                     VCO Mult = 20; (19+1)*4.00MHz = 80.00 MHz
52
        000013
                         pllmul
                                                  19
                                          equ
                                                                   ; No, Crystal not less than 200kHz
        000000
53
                                                  0
                         crystal
                                          equ
                         disXTAL
        000000
                                                                    No, do not disable crystal use
54
                                          equ
                                                  Λ
                                                  $020000
55
        020000
                         pllstop
                                          equ
                                                                   ; Yes, PLL runs during STOP
        040000
                                                  $040000
                                                                   ; Yes, enable PLL operation
56
                         enpll
                                          equ
57
        080000
                         disclk
                                          equ
                                                  $080000
                                                                   ; Yes, disable CORE clock output
58
        0E0013
                         PCTL_value
                                          equ
                                               prediv+lowdiv+pllmul+crystal+disXTAL+pllstop+enpll+disclk
59
60
                         ;--- AAR2 value = 0x100925
        000001
61
                         acctype2
                                          equ
                                                  1
                                                                   ; External Memory access type = 0x1
62
        000004
                         aahigh2
                                                  $4
                                                                   ; Enable AA2 pin to be high when selected
                                          equ
63
        000000
                         aap2
                                                  0
                                                                     No, Enable AA2 pin on ext 'P' accesses
                                          equ
                                                                     No, Enable AA2 pin on ext 'X' accesses
64
        000000
                         aax2
                                                  0
                                          equ
65
        000020
                         aay2
                                                  $20
                                                                     Yes, Enable AA2 pin on ext 'Y' accesses
                                          equ
66
        000000
                         aswap2
                                          equ
                                                  0
                                                                     No, Enable address bus swap
67
        000000
                         enpack2
                                          equ
                                                                     No, Enable packing/unpacking logic
                                                  $000900
                                                                     Compare 9 address bits
68
        000900
                         nadd2
                                          equ
                                                                    Most significant portion of address, $100000 - $107fff, to compare.
69
        100000
                         msadd2
                                                  $100000
                                          equ
70
71
                                                                     (0001,0000,0xxx,xxxx,xxxx,xxxx)
72
        100925
                     AAR2_value
                                         acctype2+aahigh2+aap2+aax2+aay2+aswap2+enpack2+nadd2+msadd2
                                   equ
```



```
74
                        ;--- AAR0 value = 0x100931
        000001
75
                        acctype0
                                                        ; External Memory access type = 0x1
                                    equ
76
        000004
                        aahigh0
                                                        ; Enable AAO pin to be low when selected
                                      eau
77
                        aap0
                                                       ; No, Enable AAO pin on ext 'P' accesses
        000000
                                              0
                                      equ
                                                       ; Yes, Enable AAO pin on ext 'X' accesses
; Yes, Enable AAO pin on ext 'Y' accesses
78
        000010
                        aax0
                                              $10
                                      equ
79
        000000
                        aav0
                                      equ
                                              $20
                                                        ; No, Enable address bus swap
; No, Enable packing/unpacking logic
80
        000000
                        aswap0
                                              0
                                      equ
81
        000000
                        enpack0
                                      equ
        000900
                                              $000900
                        nadd0
                                                        ; Compare 9 address bits
82
                                      equ
                        msadd0
                                              $100000
83
        100000
                                                        ; Most significant portion of address,
                                      equ
84
                                                          $100000 - $107fff, to compare.
                                                        ; (0001,0000,0xxx,xxxx,xxxx,xxxx)
85
                   AARO_value equ acctype0+aahigh0+aap0+aax0+aay0+aswap0+enpack0+nadd0+msadd0
       100931
86
87
88
                        ;--- BCR value = 0 \times 002421
89
       000001
                        aaa0ws
                                      equ
                                                        ; Address Attribute Area 0 w/s = 1
                                              $20 ; Address Attribute Area 1 w/s = 1
$000400 ; Address Attribute Area 2 w/s = 1
$002000 ; Address Attribute Area 3 w/s = 1
90
        000020
                        aaa1ws
                                      equ
91
        000400
                        aaa2ws
                                      equ
92
        002000
                        aaa3ws
                                      equ
93
        000000
                        defws
                                              Ω
                                                        ; Default Address Area w/s = 0
                                      equ
94
        000000
                                              0
                                                       ; Bus state status = 0
                        busss
                                      equ
                                           0
                                                      ; Enable Bus Lock Hold = 0
; Enable Bus Request Hold = 0
95
        000000
                        enblh
                                      equ
96
        000000
                        enbrh
                                      equ
97
        002421
                        BCR_value
                                      equ
                                           aaa0ws+aaa1ws+aaa2ws+aaa3ws+defws+busss+enblh+enbrh
98
99
100
       P:000100
                                       org p:$100 ; Keep the program in internal RAM
102
                memtst
103
104
                        ; Initialization Section
105
       P:000100 08F4BD
                                              #PCTL_value,x:PCTL ; Set PLL Control Register
                                     movep
                  0E0013
106
       P:000102 05F43A
                                              #$004000,OMR
                                                                    ; Disable Address Attribute Priority
                                      movec
                  004000
107
       P:000104 05F439
                                              #$080000.SR
                                      movec
                                                                    ; Enable 1K Cache
                  080000
       P:000106 08F4BB
108
                                              #BCR_value,x:BCR
                                                                    ; Set external memory wait states
                                      movep
                  002421
109
       P:000108 08F4B9
                                              #AARO value,x:AARO ; Set Address Attribute Reg0
                                      movep
                  100915
       P:00010C 08F4B7
111
                                      movep
                                              #AAR2 value,x:AAR2 ; Set Address Attribute Reg2
                  100925
112
113
       P:000110 05F420
                                               #-1,m0
                                                                    ; Set LINEAR addressing mode
                                      move
                  FFFFFF
114
115
       P:000112 05F421
                                      move
                                               \#-1,m1
                  FFFFFF
116
       P:000114 05F422
                                               \#-1,m2
                                      move
                  FFFFFF
117
       P:000116 05F423
                                               \#-1,m3
                  FFFFFF
       P:000118 05F424
                                               \#-1,m4
                                      move
                  FFFFFF
119
       P:00011A 05F425
                                               \#-1,m5
                                      move
                  FFFFFF
        P:00011C 20001B
121
                                      clr
        P:00011D 000000
122
                                      nop
123
124
        P:000121 570300
                                               b,x:X_MEM_FAIL_ADDRESS ;Initialize X:Failed Address -> $000000
                                      move
                                              125
        P:000122 570400
                                      move
126
        P:000123 570500
                                               b,x:X_MEM_FAIL_READ
                                      move
127
128
        P:000124 570600
                                               b,x:Y_MEM_FAIL_ADDRESS ;Initialize Y:Failed Address -> $000000
                                      move
        P:000125 570700
                                                                   ;Initialize Y:Expected Data -> $000000
129
                                               b,x:Y_MEM_FAIL_WROTE
                                      move
       P:000126 570800
130
                                               b,x:Y MEM FAIL READ
                                                                    ;Initialize Y:Data Read -> $000000
                                      move
131
       P:00012A 570C00
                                              b,x:MEM_PASS_COUNTER ;Initialize Pass Counter -> $000000
132
                                      move
133
134
                        main
135
136
                               Fill memory spaces with Test Patterns
137
138
       P:00012B 65F400
                                      move #PATT,r5
                                                                   ; r5 points to Test Patterns
                  0001A9
       P:00012D 000000
139
                                      nop
140
       P:00012E 000000
141
```



```
142
                       ; Fill X:RAM
143
       P:000137 07DD84
                                     p:(r5)+,x0
                                                             ; Get the Next Write Pattern
                             move
                                      #XMemSize,n1
144
       P:000138 71F400
                              move
                                                            ; Get memory size
                 008000
145
       P:00013A 61F400
                                      #XMemStart,r1
                                                            ; Get starting address for fill
                              move
                 100000
146
       P:00013C 06D920
147
                                                             ; Fill X:RAM with data pattern
                              rep
                                     n1
       P:00013D 445900
                                     x0,x:(r1)+
148
                              move
149
       P:00013E 070E04
150
                                                            ; Save the written pattern
                              move
                                     x0,p:old_x_pattern
151
152
                      ; Fill Y:RAM
153
       P:00013F 07DD84
                                     p:(r5)+,x0
                                                             ; Get the Next Write Pattern
                              move
154
       P:000140 72F400
                              move
                                      #YMemSize,n2
                                                             ; Get memory size
                 008000
       P:000142 62F400
155
                              move
                                      #YMemStart,r2
                                                             ; Get starting address for fill
                 100000
156
157
       P:000144 06DA20
                                                             ; Fill Y:RAM with data pattern
                              rep
                                     x0,y:(r2)+
158
       P:000145 4C5A00
                              move
159
160
       P:000146 070F04
                              move
                                     x0,p:old_y_pattern
                                                            ; Save the written pattern
161
162
163
                            Cycle through each memory space with a pattern
164
165
166
                       ; --- Check for expected pattern data in each RAM location ---
167
                       ;--- and then replace with a new data pattern.
168
                       ;--- ... This provides an address check. Since erroneous ---
169
                       ;--- ...addressing will cause the data to be written into ---
170
                       ;--- ...incorrect locations and this will be evident in
                       ;--- ...the next read pass.
171
172
173
174
                      ; Do a pass through X:RAM
       P:000164 61F400
175
                             move #XMemStart,r1
                                                         ; Get starting address of Test X: Memory
                 100000
176
       P:000166 078E0E
                             move p:old_x_pattern,a
move p:(r5)+,x0
                                                        ; Get the last X test pattern -> a
177
       P:000167 07DD84
                                                         ; Get the next X test pattern -> x0
178
179
                       ; Test this pattern through external X:RAM
                                                        ; Test all external X:RAM locations
       P:000168 06D910
180
                            DOR n1,next_x_loc
                 000007
181
       P:00016A 45E100
                              move x:(r1),x1
                                                         ; Read X:RAM location
                             cmp x1,a
bne <XERR
182
       P:00016B 200065
                                                         ; Read data = last test pattern?
183
       P:00016C 052451
                                                         ; No, error if compare fails
184
       P:00016D 445900
                              move x0,x:(r1)+
                                                         ; Yes, Write next test pattern -> X:RAM
185
                                                         ; Point to next memory location
186
       P:00016E 070E04
                             move x0,p:old_x_pattern ; Save the newly written test pattern
187
       P:00016F 000000
                              nop
188
                     next_x_loc
189
                       ; Do a pass through Y:RAM
190
                           move #YMemStart,r2
       P:000170 62F400
191
                                                         ; Get starting address of Test Y: Memory
                 100000
192
       P:000172 078F0E
                              move p:old_y_pattern,a ; Get the last Y test pattern -> a
193
       P:000173 07DD84
                              move p:(r5)+,x0
                                                         ; Get the next Y test pattern -> x0
194
195
                      ; Test this pattern through external Y:RAM
196
       P:000174 06DA10
                             DOR n2,next_y_loc
                                                        ; Test all external Y:RAM locations
                 000007
197
       P:000176 4DE200
                              move y:(r2),x1
                                                         ; Read Y:RAM location
198
       P:000177 200065
                              cmp x1,a
                                                        ; Read data = last test pattern?
       P:000178 052449
199
                                                         ; No, error if compare fails
                              bne
                                    <YERR
                              move x0,y:(r2)+
200
       P:000179 4C5A00
                                                        ; Yes, Write next test pattern -> Y:RAM
; Point to next memory location
201
       P:00017A 070F04
P:00017B 000000
                             move x0,p:old_y_pattern ; Save the newly written test pattern
202
203
                             nop
204
                      next_y_loc
205
       P:000193 000000
206
                             nop
                                                         ; Time to start next test pattern
207
208
                      test_mem
209
210
                                      One Pass Complete
211
                         All test patterns have been tried and passed in external RAM
                          move x:MEM_PASS_COUNTER,b0
212
       P:000194 518C00
       P:000195 000009
213
                                   b
                                                         ; Update pass loop counter
                              inc
       P:000196 000000
214
                              nop
```



```
215
        P:000197 510C00
                                  move b0,x:MEM_PASS_COUNTER
216
        P:000198 050F13
217
                                  bra main
                                                                    ; Do it all over again
218
219
220
221
2.2.2
                           ; XERR -- handles X:RAM error messaging with user
223
224
                           ; Expected Data --> a
                           ; Read Data --> x1
; Address of failure --> r1
225
226
2.27
228
                          XERR
        P:00019D 610300
P:00019E 560400
P:00019F 450500
229
                                    move r1,x:X_MEM_FAIL_ADDRESS ; Save off address of failure
230
                                   move a,x:X_MEM_FAIL_WROTE ; Save off expected data move x1,x:X_MEM_FAIL_READ ; Save off data read
231
                                   move x1,x:X_MEM_FAIL_READ
232
233
        P:0001A0 050C00
                                   bra
                                                                        ; Dynamically HALT here
234
235
236
                                YERR -- handles Y:RAM error messaging with user
237
238
                           ; Expected Data --> b
239
                           ; Read Data --> x1
240
                           ; Address of failure --> r2
241
242
243
        P:0001A1 620600
                                   move r2,x:Y_MEM_FAIL_ADDRESS ; Save off address of failure
                                   move b,x:Y_MEM_FAIL_WROTE ; Save off expected data move x1,x:Y_MEM_FAIL_READ ; Save off data read
        P:0001A2 570700
244
245
        P:0001A3 450800
246
247
        P:0001A4 050C00
                                                                        ; Dynamically HALT here
                                    bra
248
249
250
                           ; Memory Test Patterns
251
252
        P:0001A9
                                           $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
                           PATT
253
                                    dc
254
        P:0001AE
                                           $800000,$400000,$200000,$100000
$080000,$040000,$020000,$010000
                                    dc
        P:0001B2
255
                                    dc
        P:0001B6
                                           $008000,$004000,$002000,$001000
256
                                    dc
257
        P:0001BA
                                           $000800,$000400,$000200,$000100
                                    dc
258
                                           $000080,$000040,$000020,$000010
        P:0001BE
                                    dc
259
        P:0001C2
                                    dc
                                           $000008,$000004,$000002,$000001
260
        P:0001C6
                                    dc
                                           $7FFFFF,$BFFFFF,$DFFFFF,$EFFFFF
$F7FFFF,$FBFFFFF,$FDFFFF,$FEFFFF
261
        P:0001CA
                                    dc
                                           $FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
$FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
262
        P:0001CE
                                    dc
263
        P:0001D2
                                    dc
264
        P:0001D6
                                    dc
                                           $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
265
        P:0001DA
                                    dc
                                           $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
266
        P:0001DE
                                    dc
                                           $FEDCBA, $123456, $012345, $EDCBA9
267
                                           $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
$800000,$400000,$200000,$100000
        P:0001E2
268
269
        P:0001E7
                                    dc
270
        P:0001EB
                                           $080000,$040000,$020000,$010000
                                    dc
                                           $008000,$004000,$002000,$001000
$000800,$000400,$000200,$000100
271
        P:0001EF
                                    dc
272
        P:0001F3
273
        P:0001F7
                                           $000080,$000040,$000020,$000010
                                    dс
                                           $000008,$000004,$000002,$000001
$7FFFFF,$BFFFFFF,$DFFFFF,$EFFFFF
$F7FFFFF,$FBFFFFF,$FDFFFF,$FEFFFF
274
        P:0001FB
                                    dc
275
        P:0001FF
                                    dc
276
        P:000203
                                    dc
                                           $FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
$FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
277
        P:000207
                                    dc
278
        P:00020B
                                    dc
279
        P:00020F
                                           $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
                                    dc
                                           $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
280
        P:000213
                                    dc
281
        P:000217
                                    dc
                                           $FEDCBA, $123456, $012345, $EDCBA9
282
        P:00021B
                                    dc
                                           $000000,$FFFFFF,$AAAAAA,$555555,$2BAD2C
283
                                           $800000,$400000,$200000,$100000
284
        P:000220
                                    dc
285
        P:000224
                                           $080000,$040000,$020000,$010000
                                    dc
                                           $008000,$004000,$002000,$001000
286
        P:000228
                                    dc
        P:00022C
                                           $000800,$000400,$000200,$000100
2.87
                                    dc
288
        P:000230
                                    dc
                                           $000080,$000040,$000020,$000010
289
        P:000234
                                    dc
                                           $000008,$000004,$000002,$000001
290
        P:000238
                                    dc
                                           $7FFFFF,$BFFFFFF,$DFFFFFF,$EFFFFF
                                           $F7FFFF,$FBFFFF,$FDFFFF,$FEFFFF
$FF7FFF,$FFBFFF,$FFDFFF,$FFEFFF
291
        P:00023C
                                    dc
292
        P:000240
                                    dc
293
        P:000244
                                    dc
                                           $FFF7FF,$FFFBFF,$FFFDFF,$FFFEFF
294
        P:000248
                                    dc
                                           $FFFF7F,$FFFFBF,$FFFFDF,$FFFFEF
295
        P:00024C
                                           $FFFFF7,$FFFFFB,$FFFFFD,$FFFFFE
```



$\sigma 4K \times 24$ -bit Memory Based Designs

296	P:000250		dc	\$FEDCBA,\$123456,\$012345,\$EDCBA9
297				
298	P:000254		dc	\$000000,\$FFFFFFF,\$AAAAAA,\$555555,\$2BAD2C
299	P:000259		dc	\$800000,\$400000,\$200000,\$100000
300	P:00025D		dc	\$080000,\$040000,\$020000,\$010000
301	P:000261		dc	\$008000,\$004000,\$002000,\$001000
302	P:000265		dc	\$000800,\$000400,\$000200,\$000100
303	P:000269		dc	\$000080,\$000040,\$000020,\$000010
304	P:00026D		dc	\$000008,\$000004,\$000002,\$000001
305	P:000271		dc	\$7FFFFF,\$BFFFFF,\$DFFFFF,\$EFFFFF
306	P:000275		dc	\$F7FFFF,\$FBFFFF,\$FDFFFF,\$FEFFFF
307	P:000279		dc	\$FF7FFF,\$FFBFFF,\$FFDFFF,\$FFEFFF
308	P:00027D		dc	\$FFF7FF,\$FFFBFF,\$FFFDFF,\$FFFEFF
309	P:000281		dc	\$FFFF7F,\$FFFFBF,\$FFFFDF,\$FFFFEF
310	P:000285		dc	\$FFFFF7,\$FFFFFB,\$FFFFFD,\$FFFFFE
311	P:000289		dc	\$FEDCBA, \$123456, \$012345, \$EDCBA9
312				
313	000038	PATTN	equ	((*-PATT)/4)-1
314			_	
315			end	memtst
Λ	Frrore			

⁰ Errors 0 Warnings



6 DSP Memory Space Configurations

This section summarizes the memory design implementations presented in the previous sections.

As illustrated in Sections 3, 4 and 5, a common hardware memory design using the DSP56303's Memory Expansion Port allows the memory bank to be logically configured for use in various memory space arrangements by simply setting up the Memory Expansion Port's Address Attribute Control Registers. Configuring the Memory Expansion Port's Address Attribute Control Registers, the memory bank can accommodate thirteen different memory space arrangements if two Address Attribute control lines are used:

- 1. Memory Bank 'P' Space Fast SRAM
- 2. Memory Bank 'X' Space Fast SRAM
- 3. Memory Bank 'Y' Space Fast SRAM
- 4. Memory Bank 'P'/'X' Space Fast SRAM
- 5. Memory Bank 'P'/'Y' Space Fast SRAM
- 6. Memory Bank 'X'/'Y' Space Fast SRAM
- 7. Memory Bank 'P'/'X'/'Y' Space Fast SRAM
- 8. Memory Bank/2 'P'/'X' and Memory Bank/2 'Y' Space Fast SRAM
- 9. Memory Bank/2 'P'/'Y' and Memory Bank/2 'X' Space Fast SRAM
- 10. Memory Bank/2 'P' and Memory Bank/2 'X'/'Y' Space Fast SRAM
- 11. Memory Bank/2 'P' and Memory Bank/2 'X' Space Fast SRAM
- 12. Memory Bank/2 'P' and Memory Bank/2 'Y' Space Fast SRAM
- 13. Memory Bank/2 'X' and Memory Bank/2 'Y' Space Fast SRAM

6.1 Fast SRAM Advantages

Fast SRAM provides the designer with a high performance memory implementation allowing flexible memory configuration capabilities for the DSP56300 family. Fast SRAM allows the DSP to access external memory at its maximum speed with the minimum number of wait states (i.e., one wait state). For Fast SRAM speed selection, the critical timing specification used is typically based on the Fast SRAM's data access time, t_{AA} . However, all timing specifications must be met and should always be reviewed for compliance.

Fast SRAM allows the designer a high degree of memory configuration flexibility. Using the DSP's Address Attribute lines, a 24-bit Fast SRAM memory bank can be configured to satisfy various memory system implementations:

- 'P' memory space
- 'X' memory space
- 'Y' memory space
- Unified 'P' & 'X' memory space



- Unified 'P' & 'Y' memory space
- Unified 'X' & 'Y' memory space
- Unified 'P' & 'X' with separate 'Y' memory spaces
- Unified 'P' & 'Y' with separate 'X' memory spaces
- Separate 'P' with unified 'X' and 'Y' memory spaces
- Unified 'P', 'X' and 'Y' memory spaces
- Separate 'P' and 'X' memory spaces
- Separate 'P' and 'Y' memory spaces
- Separate 'X' and 'Y' memory spaces
- Separate 'P', 'X' and 'Y' memory spaces









How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor Technical Information Center, CH370 1300 N. Alma School Road Chandler, Arizona 85224 +1-800-521-6274 or +1-480-768-2130 support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH Technical Information Center Schatzbogen 7 81829 Muenchen, Germany +44 1296 380 456 (English) +46 8 52200080 (English) +49 89 92103 559 (German) +33 1 69 35 48 48 (French) support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd. Headquarters ARCO Tower 15F 1-8-1, Shimo-Meguro, Meguro-ku, Tokyo 153-0064 Japan 0120 191014 or +81 3 5437 9125 support.japan@freescale.com

Asia/Pacific

Freescale Semiconductor Hong Kong Ltd. Technical Information Center 2 Dai King Street Tai Po Industrial Estate Tai Po, N.T., Hong Kong +800 2666 8080 support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

