

Chapter 9

External Memory Interface (Port A)

The external memory expansion port, Port A, can be used either for memory expansion or for memory-mapped I/O. External memory is easily and quickly retrieved through the use of DMA or simple MOVE commands. For more information on Port A programming see application note AN1751D, *DSP563xx Port A Programming*. Several features make Port A versatile and easy to use, resulting in a low part-count connection with fast or slow static memories, dynamic memories, I/O devices and multiple bus master system. The Port A data bus is 24 bits wide with a separate 18-bit or 24-bit address bus.

External memory is divided into three possible $16\text{ M} \times 24\text{-bit}$ spaces: X data, Y data, and program memory. Each space or all spaces can access a given external memory. Access type and attributes are under software control. See the memory map in **Chapter 11, *Operating Modes and Memory Spaces*** for memory space that is not accessible through Port A. An internal wait state generator can be programmed to statically insert up to 31 wait states for access to slower memory or I/O devices. A Transfer Acknowledge ($\overline{\text{TA}}$) signal allows an external device to dynamically control the number of wait states inserted into a bus access operation. The bus arbitration allows multiple potential masters of the Port A bus. One DSP56300 processor can use the Port A bus to access external devices while other potential masters perform internal operations that do not require the Port A bus. See the memory map in the device-specific user's manual for memory space that is not accessible.

Note: The AA lines can operate as memory-mapped chip selects or address lines to external devices, depending upon the mode selected. Some DSP56300 family devices have eighteen address lines. For these processors, if all four Address Attribute (AA) lines are used as address lines, the total addressable external memory per space (X data, Y data, and program) is $4\text{ M} \times 24\text{-bit}$. If all four AA lines are used, then the memory must always be selected, because no AA lines are available for chip select. As a result, an external read or write outside the 4M range could still go to the external memory (depending on the settings of the AA registers).

9.1 Signal Description

Table 9-1 through **Table 9-3** show the signals that the external memory interface uses for controlling and transferring data.

Table 9-1 External Address Bus Signals

Signal Name	Type	State During Reset	Signal Description
A[0 – 17]/ A[0 – 23]	Output	Tri-stated	Address Bus —When the DSP is the bus master, A[0 – 17]/A[0 – 23] are active-high outputs that specify the address for external program and data memory accesses. Otherwise, the signals are tri-stated. To minimize power dissipation, A[0 – 17]/A[0 – 23] do not change state when external memory spaces are not being accessed.
Note: The total number of address lines is device-specific.			

Table 9-2 External Data Bus Signals

Signal Name	Type	State During Reset	Signal Description
D[0 – 23]	Input/Output	Tri-stated	Data Bus —When the DSP is the bus master, D[0 – 23] are active-high, bidirectional input/outputs that provide the bidirectional data bus for external program and data memory accesses. Otherwise, D[0 – 23] are tri-stated.

Table 9-3 External Bus Control Signals

Signal Name	Type	State During Reset	Signal Description
AA0-AA3	Output	Tri-stated	Address Attribute —When defined as AA, these signals can be used as chip selects or additional address lines. Unlike address lines, these lines are deasserted between external accesses. For information about asserting AA signals simultaneously, see Section 9.6.1 , "Address Attribute Registers (AAR0–AAR3)," on page 9-15.
$\overline{\text{RAS}}[0 – 3]$			Row Address Strobe —When defined as $\overline{\text{RAS}}$ (using the BAT bits in the corresponding AAR—see the BAT bits description in Section 9.6.1 , "Address Attribute Registers (AAR0–AAR3)," on page 9-15), these signals can be used as $\overline{\text{RAS}}$ for the Dynamic Random Access Memory (DRAM) interface. These signals are tri-statable outputs with programmable polarity.
$\overline{\text{RD}}$	Output	Tri-stated	Read Enable —When the DSP is the bus master, $\overline{\text{RD}}$ is an active-low output that is asserted to read external memory on the data bus (D[0 – 23]). Otherwise, $\overline{\text{RD}}$ is tri-stated.

Table 9-3 External Bus Control Signals (Continued)

Signal Name	Type	State During Reset	Signal Description
\overline{WR}	Output	Tri-stated	Write Enable —When the DSP is the bus master, \overline{WR} is an active-low output that is asserted to write external memory on the data bus (D[0 – 23]). Otherwise, the signal is tri-stated.
\overline{BS}	Output	Tri-stated	Bus Strobe —When the DSP is the bus master, \overline{BS} is asserted for half a clock cycle at the start of a bus cycle to provide an “early bus start” signal for a bus controller. If the external bus is not used during an instruction cycle, \overline{BS} remains deasserted until the next external bus cycle. NOTE: This signal is not implemented on all devices in the DSP56300 family.
\overline{TA}	Input	Ignored Input	Transfer Acknowledge —If the DSP56300 family device is the bus master and there is no external bus activity, or the DSP56300 family device is not the bus master, the \overline{TA} input is ignored. The \overline{TA} input is a Data Transfer Acknowledge (DTACK) function that can extend an external bus cycle indefinitely. Any number of wait states (that is, 1, 2, ..., infinity) may be added to the wait states inserted by the BCR by keeping \overline{TA} deasserted. In typical operation, \overline{TA} is: <ul style="list-style-type: none"> ■ deasserted at the start of a bus cycle ■ asserted to enable completion of the bus cycle ■ deasserted before the next bus cycle <p>The current bus cycle completes one clock period after \overline{TA} is asserted synchronously to CLKOUT. The number of wait states is determined by the \overline{TA} input or by the Bus Control Register (BCR), whichever is longer. The BCR can be used to set the minimum number of wait states in external bus cycles. To use the \overline{TA} functionality, the BCR must be programmed to at least one wait state. A zero wait state access cannot be extended by \overline{TA} deassertion, otherwise improper operation may result. \overline{TA} can operate synchronously or asynchronously depending on the setting of the TAS bit in the Operating Mode Register (OMR).</p> <p>NOTE: Do not use \overline{TA} functionality while performing DRAM type accesses; otherwise, improper operation may result.</p> <p>When the DSP56300 family device is the bus master, but \overline{TA} is not used for external bus control, \overline{TA} must be asserted low (pulled down).</p>

Table 9-3 External Bus Control Signals (Continued)

Signal Name	Type	State During Reset	Signal Description
\overline{BR}	Output	Output (deasserted)	Bus Request —An active-low output that is never tri-stated. \overline{BR} is asserted when the DSP requests bus mastership. \overline{BR} is deasserted when the DSP no longer needs the bus. \overline{BR} may be asserted or deasserted independent of whether the DSP56300 family device is a bus master or not. Bus “parking” allows bus access without asserting \overline{BR} (see the descriptions of bus “parking” in Section 9.5.3.4 and Section 9.5.3.6). The Bus Request Hold (BRH) bit in the Bus Control Register (BCR) allows \overline{BR} to be asserted under software control, even though the DSP does not need the bus. \overline{BR} is typically sent to an external bus arbiter that controls the priority, parking, and tenure of each master on the same external bus. \overline{BR} is only affected by DSP requests for the external bus, never for the internal bus. During hardware reset, \overline{BR} is deasserted; arbitration is reset to the bus slave state.
\overline{BG}	Input	Ignored Input	Bus Grant —Asserted by an external bus arbitration circuit when the DSP56300 family device becomes the next bus master. \overline{BG} must be asserted/deasserted synchronous to CLKOUT for proper operation. When \overline{BG} is asserted, the DSP56300 family device must wait until \overline{BB} is deasserted before taking bus mastership. When \overline{BG} is deasserted, bus mastership is typically given up at the end of the current bus cycle. This may occur in the middle of an instruction that requires more than one external bus cycle for execution.
\overline{BB}	Input/Output	Input	Bus Busy —Indicates that the bus is active. \overline{BB} must be asserted and deasserted synchronous to CLKOUT. Only after \overline{BB} is deasserted can a pending bus master become the bus master (and assert \overline{BB}). Some designs allow a bus master to keep \overline{BB} asserted after ceasing bus activity. This is called “bus parking” and allows the current bus master to reuse the bus without re-arbitration until another device requires the bus (see Section 9.5.3.4 and Section 9.5.3.6). Deassertion of \overline{BB} uses an “active pull-up” method (that is, \overline{BB} is driven high and then released and held high by an external pull-up resistor). \overline{BB} requires an external pull-up resistor.
\overline{BL}	Output	Driven high	Bus Lock —Asserted at the start of an external divisible read-modify-write bus cycle, remains asserted between the read and write cycles, and is deasserted at the end of the write bus cycle. This provides an “early bus start” signal for the bus controller. \overline{BL} may be used to “resource lock” an external multi-port memory for secure semaphore updates. Early deassertion provides an “early bus end” signal useful for external bus control. If the external bus is not used during an instruction cycle, \overline{BL} remains deasserted until the next external indivisible read-modify-write cycle. The only instructions that assert \overline{BL} automatically are BSET, BCLR, and BCHG when the access is to external memory. An operation can also assert \overline{BL} by setting the BLH bit in the BCR. This signal is not implemented on all devices in the DSP56300 family.

Table 9-3 External Bus Control Signals (Continued)

Signal Name	Type	State During Reset	Signal Description
CAS	Output	Tri-stated	Column Address Strobe —When the DSP is the bus master, DRAM uses CAS to strobe the column address. Otherwise, if the Bus Mastership Enable (BME) bit in the DRAM Control Register (DCR) is cleared, the signal is tri-stated.
BCLK	Output	Tri-stated	Bus Clock —When the DSP is the bus master, BCLK is an active-high output. BCLK is active as a sampling signal when the program Address Trace Mode is enabled (by setting the ATE bit in the OMR). When BCLK is active and synchronized to CLKOUT by the internal PLL, BCLK precedes CLKOUT by one-fourth of a clock cycle. The BCLK rising edge can be used to sample the internal Program Memory access on the address lines. NOTE: The address trace functionality described here is not practical above 80 MHz, so it does not apply in DSP56300 chips with a clock that runs above 80 MHz.
$\overline{\text{BCLK}}$	Output	Tri-stated	Bus Clock —When the DSP is the bus master, $\overline{\text{BCLK}}$ is an active-low output that is the inverse of the BCLK signal. Otherwise, the signal is tri-stated.

9.2 Port Operation

External bus timing is defined by the operation of the Address Bus, Data Bus, and Bus Control pins as described in the previous sections. The DSP56300 core external ports interface with a wide variety of memory and peripheral devices, high speed SRAMs and DRAMs, and slower memory devices. The $\overline{\text{TA}}$ control signal and the Bus Control Register (BCR) described in **Section 9.6.2** control the external bus timing. The BCR provides constant bus access timing through the insertion of wait states. $\overline{\text{TA}}$ provides dynamic bus access timing. The number of wait states for each external access is determined by the $\overline{\text{TA}}$ input or by the BCR, whichever specifies the longest time.

The external memory address is defined by the Address Bus (A[0 – 17]/A[0 – 23]) and the memory Address Attribute signals (AA[0 – 3]). The Address Attribute signals have the same timing as the Address Bus and may be used as additional address lines. The Address Attribute signals are also used to generate Chip Select (CS) signals for the appropriate memory chips. These CS signals change the memory chips from low power Standby mode to Active mode and begin the access time. This allows slower memories to be used since the Address Attribute signals are address-based rather than read or write enable-based.

9.2.1 SRAM Support

The DSP56300 core can interface easily with SRAMs. Because the address must remain stable during the entire bus cycle, however, at least one wait state must be inserted

regardless of the speed of the SRAM. **Figure 9-1** shows an SRAM access timing example (for detailed timing information, see the specific technical data sheet for the device used in the design). **Figure 9-2** shows a typical DSP56300 family device-to-SRAM connection.

SRAM access consists of the following steps:

1. Address Bus (A[0 – 17]/A[0 – 23]), Address Attributes (AA[0 – 3]), and Bus Strobe (\overline{BS}) are asserted in the middle of CLKOUT high phase.
2. Write enable (\overline{WR}) is asserted with the falling edge of CLKOUT (for a single wait state access). Read enable (\overline{RD}) is asserted in the middle of CLKOUT low phase.
3. For a write operation, data is driven in the middle of CLKOUT high phase. For a read operation, data is sampled in the middle of CLKOUT last low phase of the external access.

For accessing slower memories, wait states (from the BCR or by the \overline{TA} signal) postpone the disappearance of the external address and increase memory access time. In any case, SRAM access requires at least one wait state.

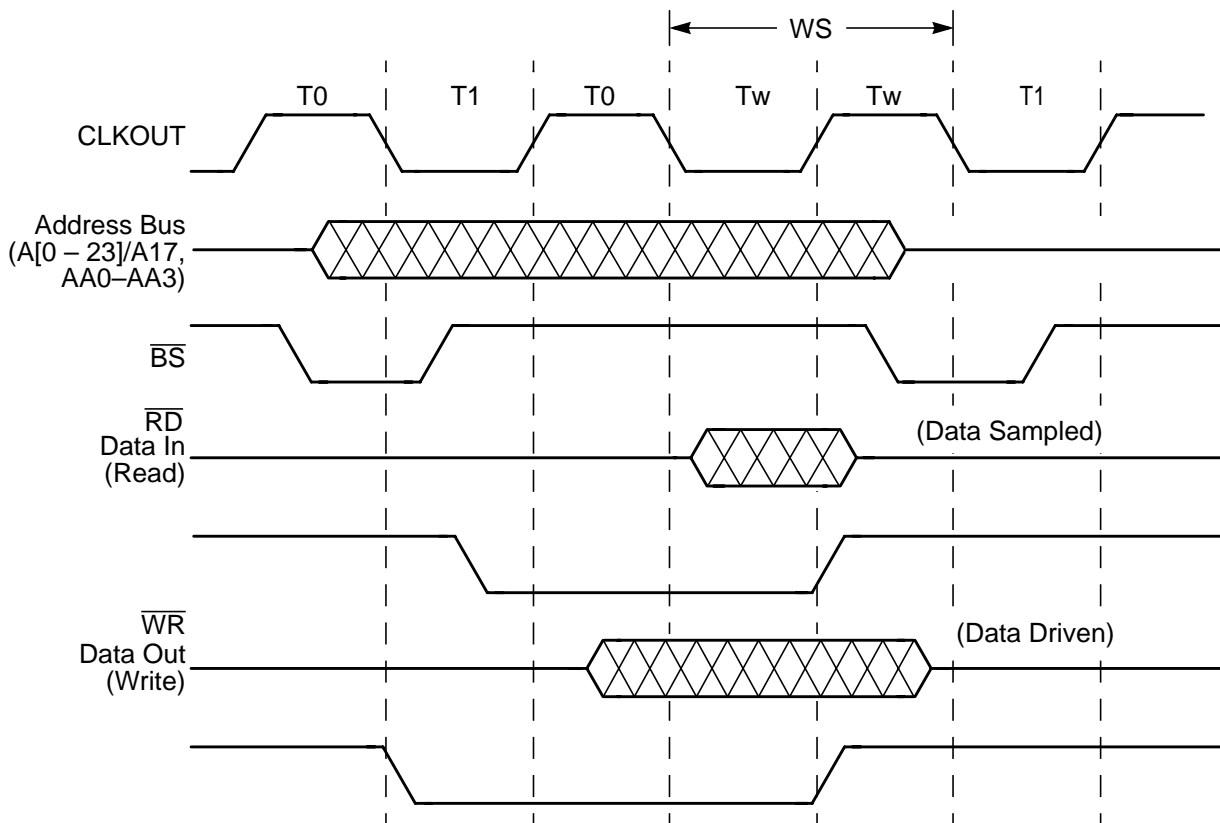


Figure 9-1. SRAM Access with One Wait State Example

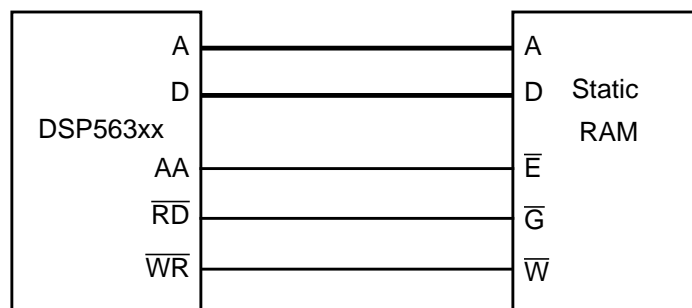


Figure 9-2. Example SRAM Connection Diagram

Note: The assertion of \overline{WR} depends on the number of wait states programmed in the BCR. If one wait state is programmed, \overline{WR} is asserted with the falling edge of CLKOUT. If two or three wait states are programmed, \overline{WR} assertion is delayed by half a clock cycle (half CLKOUT cycle). If four or more wait states are programmed, \overline{WR} assertion is delayed by a full clock cycle. This feature enables the connection of slow external devices that require long address setup time before write assertion in order to prevent false writes.

9.2.2 DRAM Support

DRAMs are becoming the preferred external memory choice for many reasons, including:

- Low cost per bit due to dynamic storage cell density
- Increasing packaging density due to multiplexed address and control pins
- Improved price-performance relative to SRAMs due to Fast Access mode (Page mode)
- Commodity pricing due to high-volume production

Port A bus control signals are an efficient interface to DRAM devices in both random read/write cycles and Fast Access mode (Page mode). An on-chip DRAM controller controls the page hit circuit, address multiplexing (row address and column address), control signal generation (\overline{CAS} and \overline{RAS}), and refresh access generation (\overline{CAS} before \overline{RAS}) for a large variety of DRAM module sizes and different access times. The DRAM controller operation and programming is described in **Section 9.6.3**, "DRAM Control Register," on page 9-21.

External bus timing is controlled by the DRAM Control Register (DCR) described in **Section 9.6.3**. The DCR controls insertion of wait states to provide constant bus access timing. The external memory address is defined by the Address Bus (A[0 – 23]/A[0 – 17]). The “n” low order address bits are multiplexed inside the DSP56300 core, and the new 24-bit address is driven to the external bus. The address multiplexing enables a

glueless interface to DRAMs by simply connecting the “n” low order bits to the memory address pins. When the BAT bits in the corresponding AAR are programmed, an Address Attribute signal can function as a Row Address Strobe ($\overline{\text{RAS}}$). An in-page access is assumed, and $\overline{\text{RAS}}$ is therefore kept asserted until one of the following events occurs:

- An out-of-page access is detected
- An access to another bank of dynamic memory is attempted
- A refresh access is attempted ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$)
- A write to one of the following registers is detected:
 - BCR
 - DCR
 - AAR3
 - AAR2
 - AAR1
 - AAR0
- A loss of bus mastership is detected while the BME bit in the DCR register is cleared
- WAIT or STOP instruction is detected
- Hardware or software reset is detected

Figure 9-3 and **Figure 9-4** show DRAM in-page access timing examples. For detailed timing information, see the technical data sheet for the device used in the design. **Figure 9-5** shows a typical DSP56300 family device-to-DRAM connection.

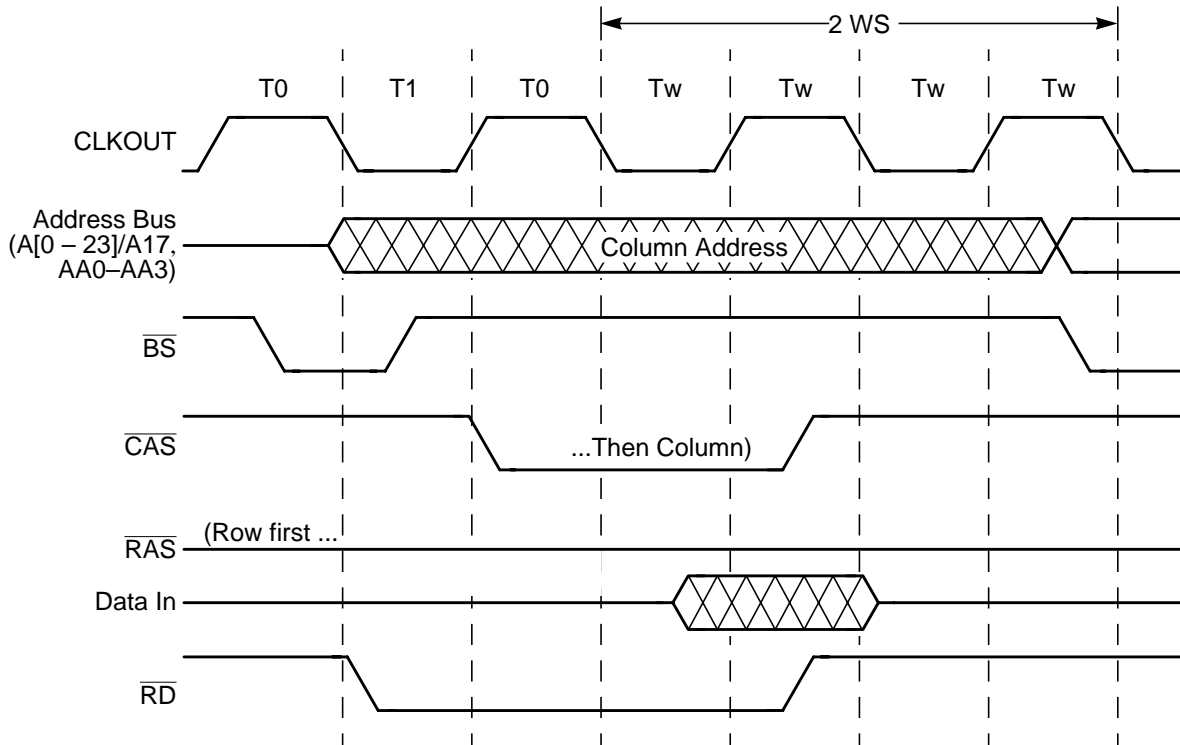


Figure 9-3. DRAM Read Access (In-Page) with Two Wait States

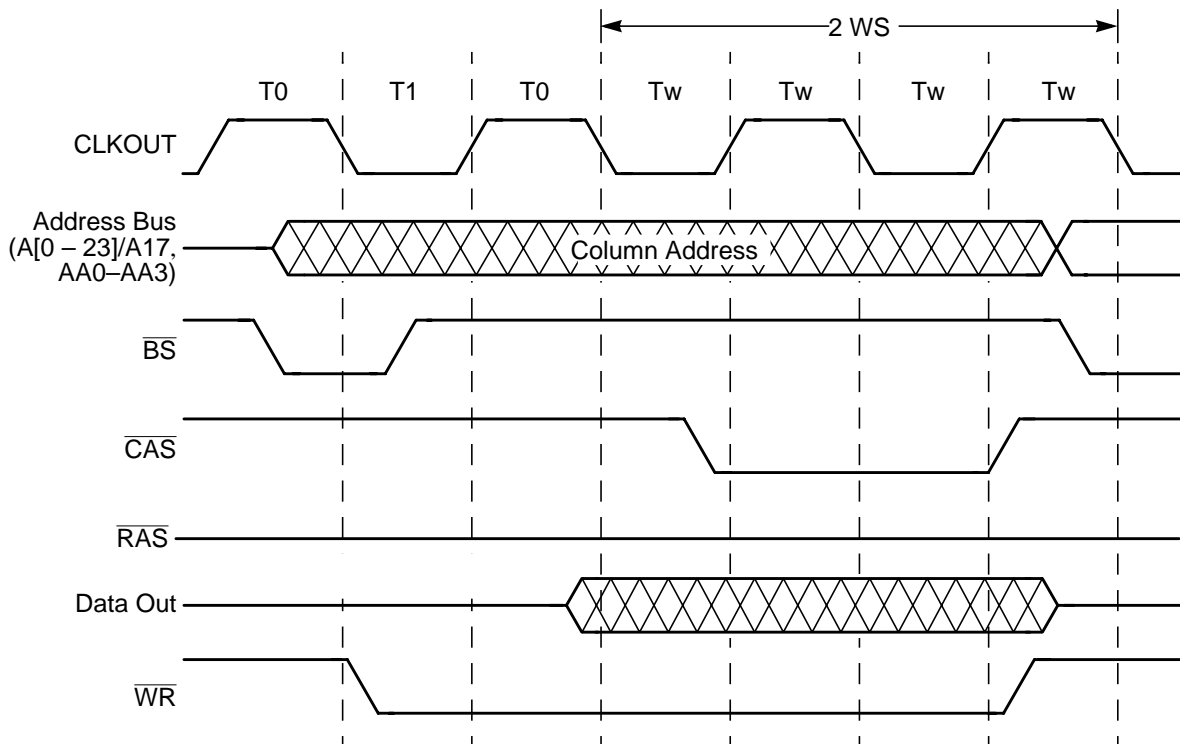


Figure 9-4. DRAM Write Access (In-Page) with Two Wait States Example

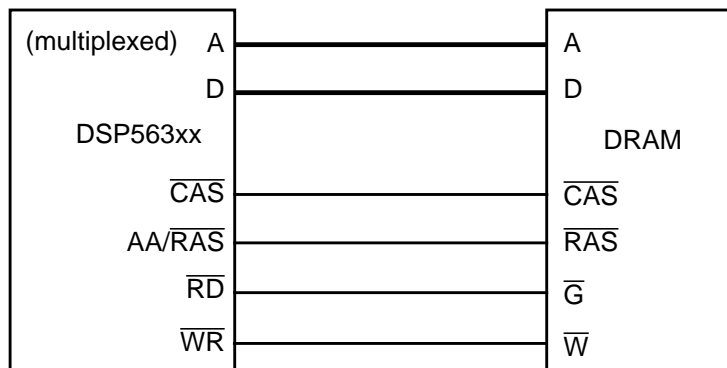


Figure 9-5. Typical DRAM Connection Diagram

9.2.2.1 DRAM In-Page Access

A DRAM in-page access consists of the following steps:

1. Column address (a subset of A[0 – 23]/A17, as determined by the BPS bits in the DCR) and Bus Strobe (\overline{BS}) are asserted in the middle of CLKOUT high phase.
2. Write (\overline{WR}) or Read (\overline{RD}) is asserted with the CLKOUT falling edge.
3. \overline{CAS} assertion timing depends on the number of in-page wait states selected by the DCR[BCW] bits and on the access purpose (read or write). (See **Figure 9-3** and **Figure 9-4** for examples of DRAM in-page read and write accesses using two wait states).
4. \overline{CAS} is deasserted before the end of the external access in order to meet the \overline{CAS} precharge timing.

Note: In all cases, DRAM access requires at least one wait state.

9.2.2.2 DRAM Out-of-Page Access

An out-of-page access consists of the following steps:

1. Deassertion of \overline{RAS}
2. Assertion of the control signals ($\overline{WR}/\overline{RD}$)
3. After \overline{RAS} precharge time, the assertion of \overline{RAS} . \overline{RAS} assertion and \overline{CAS} timing depend on the number of out-of-page wait states selected by the BRW bits in the DCR.

9.3 Port A Disable

In applications sensitive to power consumption, Port A may not be required because the memory that is used resides in the processor. A special feature of the Port A controller allows you to reduce the power consumption significantly by setting the EBD bit in the Operating Mode Register (OMR) to disable the Port A controller. This causes the DSP56300 device to release the bus (that is, deassert \overline{BR} and \overline{BL} , tri-state \overline{BB} , and ignore \overline{BG}). With the controller disabled, no external DMA accesses or refresh accesses can be performed.

Note: To prevent improper operation when OMR[EBD] is set, do not access external memory, and always clear Refresh Enable (BREN—DCR[13]) to prevent any external DRAM refresh attempts.

9.4 Bus Handshake and Arbitration

Bus transactions are governed by a single bus master. Bus arbitration determines which device becomes the bus master. The arbitration logic implementation is system-dependent but must result in, at most, one device becoming the bus master (even if multiple devices request bus ownership). The arbitration signals permit simple implementation of a variety of bus arbitration schemes (for example, fairness, priority, etc.). The system designer must provide the external logic to implement the arbitration scheme.

9.5 Bus Arbitration Signals

There are three bus arbitration signals. Two of them (\overline{BR} and \overline{BG}) are local arbitration signals between a potential bus master and the arbitration logic; \overline{BB} is a system arbitration signal:

- Bus Request (\overline{BR})—Asserted by a device to request use of the bus; it is held asserted until the device no longer needs the bus. This includes time when it is the bus master as well as when it is not the bus master.
- Bus Grant (\overline{BG})—Asserted by the bus arbitration controller to signal the requesting device that it is the bus master elect, \overline{BG} is valid only when the bus is not busy (that is, \overline{BB} is not asserted).
- Bus Busy (\overline{BB})—This signal is driven by the current bus master and controls the hand-over of bus ownership by the bus master at the end of bus possession. \overline{BB} is an active pull-up signal (that is, it is driven high before release and then held high by an external pull-up resistor).

9.5.1 The Arbitration Protocol

The bus is arbitrated by a central bus arbiter, using individual request/grant lines to each bus master. The arbitration protocol can operate in parallel with bus transfer activity so that the bus can be handed over without much performance penalty. The arbitration sequence occurs as follows:

1. **Bus Requested by Device**—All candidates for bus ownership assert their respective $\overline{\text{BR}}$ signals as soon as they need the bus.
2. **Bus Granted by Arbiter**—The arbitration logic designates a bus master-elect by asserting the $\overline{\text{BG}}$ signal for that device.
3. **Bus Released by Current Master**—The master-elect tests $\overline{\text{BB}}$ to ensure that the previous master has relinquished the bus. If $\overline{\text{BB}}$ is deasserted, then the master-elect asserts $\overline{\text{BB}}$, which designates the device as the new bus master. If a higher priority bus request occurs before the $\overline{\text{BB}}$ signal is deasserted, then the arbitration logic may replace the current master-elect with the higher priority candidate. However, only one $\overline{\text{BG}}$ signal may be asserted at one time.
4. **Bus Control Assumed by New Master**—The new bus master begins its bus transfers after asserting $\overline{\text{BB}}$.
5. **Bus Grant Withdrawn by Arbiter**—The arbitration logic signals the new bus master to relinquish the bus by deasserting $\overline{\text{BG}}$ at any time.
6. **Bus Released by Current Master**—A DSP56300 core bus master releases its ownership (drives $\overline{\text{BB}}$ high and then releases the bus) after completing the current external bus access (except for the cases described in the following note). If an instruction is executing a read-modify-write external access, a DSP56300 core master asserts the $\overline{\text{BL}}$ signal and only relinquishes the bus (and deasserts $\overline{\text{BL}}$) after completing the entire read-modify-write sequence. When the current bus master releases $\overline{\text{BB}}$, it first drives the $\overline{\text{BB}}$ signal high and then the $\overline{\text{BB}}$ signal is held by the pull-up resistor. The next bus master-elect has received its $\overline{\text{BG}}$ signal and is waiting for $\overline{\text{BB}}$ to be deasserted before claiming ownership.

Note: The three packing accesses, the two accesses of a read-modify-write instruction (BSET, BCLR, BCHG), and the up-to-four fetch burst accesses are treated as one access from an arbitration point of view (that is, the bus mastership is not released during the execution of these accesses).

The DSP56300 core has two control bits (BRH and BLH) and one status bit (BBS), in the Bus Control Register (BCR) to permit software control of the $\overline{\text{BR}}$ and $\overline{\text{BL}}$ signals and to verify whether the device is the bus master. See **Section 9.6.2** for more information about the BCR.

- **BRH Bit**—If the BCR[BRH] bit is cleared, the DSP56300 core asserts its $\overline{\text{BR}}$ signal only as long as requests for bus transfers are pending or being attempted. If the BCR[BRH] is set, $\overline{\text{BR}}$ remains asserted.
- **BLH Bit**—If the BCR[BLH] bit is cleared, the DSP56300 core asserts its $\overline{\text{BL}}$ signal only during a read-modify-write bus access. If the BCR[BLH] is set, $\overline{\text{BL}}$ remains asserted (even when not a bus master).
- **BBS Bit**—This read-only bit in the BCR is set when the DSP is the bus master and cleared when it is not.

The DSP56300 core uses the OMR[BRT] bit control bit to enable Fast or Slow Bus Release mode. In Fast Bus Release mode, all Port A pins are tri-stated in the same cycle. In Slow Bus Release mode an extra cycle is added and all Port A pins except $\overline{\text{BB}}$ are released first. Only in the next cycle is $\overline{\text{BB}}$ released. Therefore, in Slow Bus Release mode, $\overline{\text{BB}}$ is guaranteed to be the last pin that is tri-stated. This may be useful in systems where a possibility for contention exists. A more detailed explanation (including timing diagrams) is provided in the appropriate technical data sheet.

Note: During the execution of WAIT and STOP instructions, the DSP56300 releases the bus (that is, deasserts $\overline{\text{BR}}$ and $\overline{\text{BB}}$), and ignores $\overline{\text{BG}}$.

9.5.2 Arbitration Scheme

Bus arbitration is implementation-dependent. **Figure 9-6** illustrates a common bus arbitration scheme. The arbitration logic determines device priorities and assigns bus ownership depending on those priorities. For example, an implementation may hold $\overline{\text{BG}}$ asserted for the current bus owner if none of the other devices are requesting the bus. As a consequence, the current bus master may keep $\overline{\text{BB}}$ asserted after ceasing bus activity, regardless of whether $\overline{\text{BR}}$ is asserted or deasserted. This situation is called “bus parking” and allows the current bus master to use the bus repeatedly without re-arbitration until some other device requests the bus.

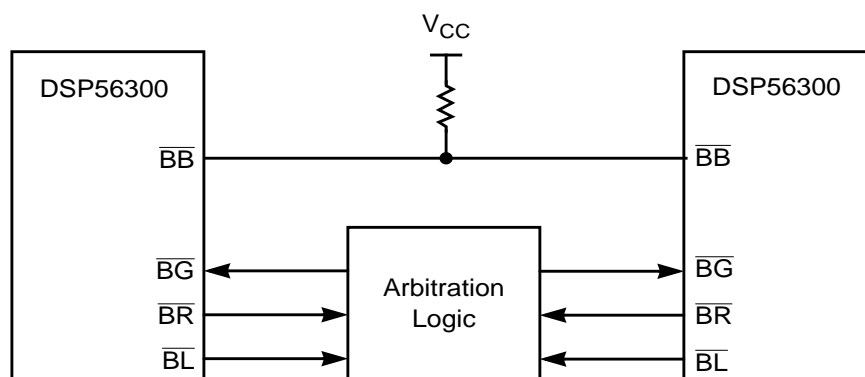


Figure 9-6. Example Bus Arbitration Scheme

9.5.3 Bus Arbitration Example Cases

The following paragraphs describe various bus arbitration examples.

9.5.3.1 Case 1—Normal

The \overline{BB} signal is high, indicating that no device is controlling the bus (that is, the bus is not busy). A device requests mastership by asserting \overline{BR} . The arbiter then asserts the \overline{BG} signal for the requesting devices. Since \overline{BB} is high, indicating that the bus is not busy, the requesting device asserts \overline{BB} and takes control of the bus.

9.5.3.2 Case 2—Bus Busy

The \overline{BB} signal is asserted indicating that a device is already the bus master. If a second device requests mastership by asserting \overline{BR} , the arbiter responds by asserting the \overline{BG} signal for the requesting device. However, since the bus is busy (i.e., \overline{BB} is already asserted by the current master), the requesting device cannot assert \overline{BB} until the current master drives \overline{BB} high to release the bus. After the first master drives \overline{BB} high, the requesting device can then assert \overline{BB} and take control of the bus.

9.5.3.3 Case 3—Low Priority

If multiple devices assert \overline{BR} at the same time, the arbiter grants the bus to the device with the highest priority. The arbiter withholds the assertion of \overline{BG} for a lower priority device until the \overline{BR} for the higher priority device is deasserted. The lower device cannot take control of the bus until the higher priority device deasserts \overline{BR} , the arbiter asserts \overline{BG} to the lower priority device, and the current master deasserts \overline{BB} .

9.5.3.4 Case 4—Default

The arbiter design may specify a default bus master. Such a design asserts \overline{BG} for the default device whenever no other device requests the bus. Thus, whenever \overline{BB} is deasserted (that is, the bus is not busy), the default device can take control of the bus by asserting \overline{BB} without asserting \overline{BR} first. As long as the bus arbiter leaves \overline{BG} asserted because no other requests are pending, then the default device continues to assert \overline{BB} and maintain its bus mastership. This condition is called bus parking and eliminates the need for the default bus master to re-arbitrate for the bus during its next external access.

9.5.3.5 Case 5—Bus Lock during Read-Modify-Write Instructions

Typically, if a device asserts \overline{BR} to request bus mastership and the arbiter then asserts \overline{BG} to the requesting device and \overline{BB} is deasserted (that is, the bus is not busy), then the requesting device asserts \overline{BB} and takes control of the bus. If the master device executes a read-modify-write instruction that accesses external memory, then \overline{BB} remains asserted

until the entire read-modify-write instruction completes execution, even if the bus arbiter deasserts \overline{BG} . After the execution is complete, the device then drives \overline{BB} high thereby relinquishing the bus. In DSP56300 family devices in which it is implemented, the \overline{BL} signal can be used to ensure that a multiport memory can only be written by one master at a time.

Note: During external read-modify-write instruction execution, \overline{BL} is asserted.

9.5.3.6 Case 6—Bus Parking

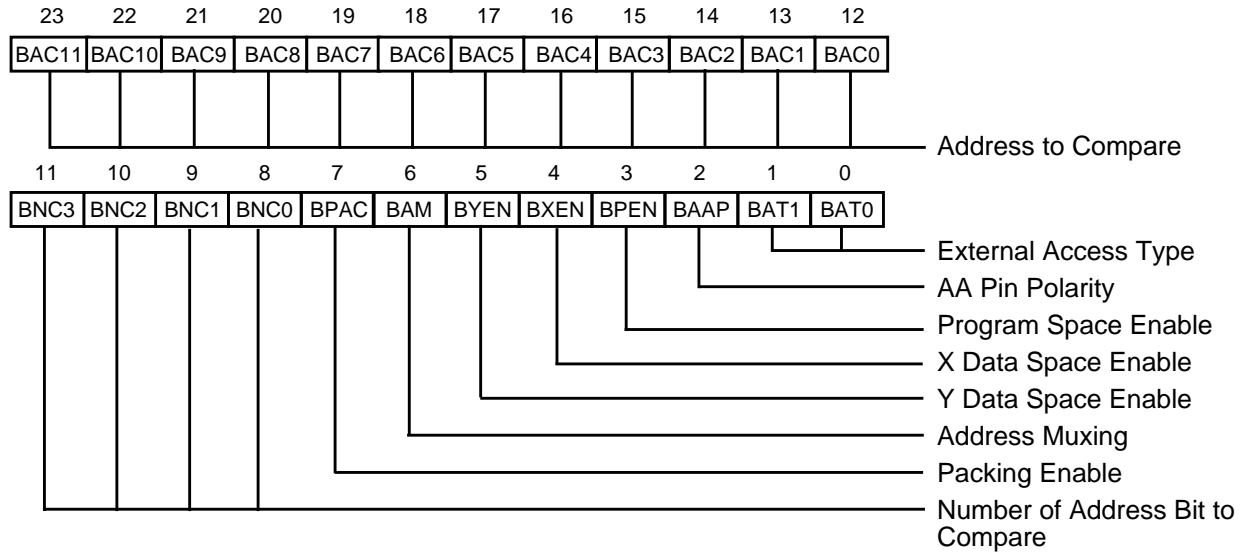
As described in **Section 9.5.3.4**, bus parking is a strategy that permits a device to take control of the bus without asserting \overline{BR} . In addition to designs which use a default bus master device, an arbiter design may allow the last bus master to retain control of the bus until mastership is requested by another device. In such a design, a device asserts \overline{BR} to request bus mastership and the arbiter responds by asserting \overline{BG} to the requesting device. When \overline{BB} is deasserted (that is, the bus is not busy), the requesting device asserts \overline{BB} to assume bus mastership. When the requesting device no longer requires the bus, it deasserts \overline{BR} , but if no other requests are pending, the bus arbiter leaves \overline{BG} asserted and \overline{BB} remains asserted for that device (that is, the last device maintains its bus mastership). Thus, the last device to control the bus is parked on the bus. This eliminates the need for the last bus master to re-arbitrate for the bus during its next external access.

9.6 Port A Control

Port A control consists of four Address Attribute Registers (AAR0–AAR3), the Bus Control Register (BCR), and the DRAM Control Register (DCR).

9.6.1 Address Attribute Registers (AAR0–AAR3)

The four Address Attribute Registers (AAR0–AAR3) are 24-bit read/write registers that control the activity of the AA[0–3]/RAS[0–3] pins. The associated AA_n/\overline{RAS}_n pin is asserted if the address defined by the BAC bits in the associated AAR matches the exact number of external address bits defined by BNC bits, and the external address space (X data, Y data, or program) is enabled by the AAR. All AARs are disabled (that is, all the AAR bits are cleared) during hardware reset. The AAR bits are shown in **Figure 9-7** and described in this section. All AAR bits are read/write control bits.



- Notes:
1. A priority mechanism exists among the four AAR control registers in order to resolve selection conflicts. AAR3 has the highest priority and AAR0 has the lowest priority (for example if the external address matches the address and the space that is specified is in both AAR1 and AAR2, the external access type is selected according to AAR2). The priority mechanism allows continuous partitioning of the external address space.
 2. When a selection conflict occurs, that is the external address matches the address and the space that is specified in more than one AAR, the assertion of the lower priority AA/RAS pin(s) is programmable. When the OMR[APD] bit is cleared (see Chapter 6), only one AA/RAS pin of higher priority is asserted. When the OMR[APD] bit is set, the lower priority AA/RAS pin(s) are asserted in addition to the high-priority AA/RAS pin. AAR of higher priority defines the external memory access type (memory type, wait states, and so on). The lower-priority AA/RAS pin(s) associated with DRAM memory type (BAT[1 – 0] = 10) are not activated. This allows glueless support of Long Move (move L:) instruction to/from external memory as shown in **Figure 9-8**.

Figure 9-7. Address Attribute Registers (AAR0–AAR3)

Table 9-4 AAR Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23 – 12	BAC	0	Bus Address to Compare Defines the upper 12 bits of the 24-bit address with which to compare the external address to decide whether to assert the corresponding AA/RAS signal. This is also true when 16-bit compatibility mode is in use. The BNC[3 – 0] bits define the number of address bits to compare.
11 – 8	BNC	0	Bus Number of Address Bits to Compare Defines the number of bits (from the BAC bits) that are compared to the external address. The BAC bits are always compared to the Most Significant Portion of the external address (for example, if BNC[3 – 0] = 0011, then the BAC[11 – 9] bits are compared to the 3 MSBs of the external address). If no bits are specified (that is, BNC[3 – 0] = 0000), the AA signal is activated for the entire 16 M-word space identified by the space enable bits (BPEN, BXEN, BYEN), but only when the address is external to the internal memory map. The combinations BNC[3 – 0] = 1111, 1110, 1101 are reserved.

Table 9-4 AAR Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
20	BPAC	0	<p>Bus Packing Enable Defines whether the internal packing/unpacking logic is enabled. When the BPAC bit is set, packing is enabled. In this mode each DMA external access initiates three external accesses to 8-bit wide external memory (the addresses for these accesses are DAB, then DAB + 1 and then DAB + 2). Packing to a 24-bit word (or unpacking from a 24-bit word to three 8-bit words) is done automatically by the expansion port control hardware. The external memory should reside in the eight Least Significant Bits (LSBs) of the external data bus, and the packing (or unpacking for external write accesses) is done in “Little Endian” order (that is, the low byte is stored in the lowest of the three memory locations and is transferred first; the middle byte is stored/transferred next; and the high byte is stored/transferred last). When this bit is cleared, the expansion port control logic assumes a 24-bit wide external memory.</p> <p>NOTE: The BPAC bit is used only for DMA accesses and not core accesses. To ensure sequential external accesses, the DMA address should advance three steps at a time in two-dimensional mode with a row length of one and an offset size of three. Refer to Motorola application note, APR23/D, <i>Using the DSP56300 Direct Memory Access Controller</i>, for more information.</p> <p>To prevent improper operation, DMA address + 1 and DMA address + 2 should not cross the AAR bank borders.</p> <p>Arbitration is not allowed during the packing access (that is, the three accesses are treated as one access with respect to arbitration, and bus mastership is not released during these accesses)</p>
6	BAM	0	<p>Bus Address Multiplexing Defines whether the eight LSBs of the address appear on address lines A0–A7 (Least Significant Portion of the external address bus) or on address lines A16–A23 (Most Significant Portion of the external address bus). When BAM is set, the eight LSBs appear on address lines A16–A23. When BAM is cleared, the eight LSBs appear normally on address lines A0–A7. This feature enables you to connect an external peripheral to the MSBs of the address, thus decreasing the load on the Least Significant Portion of the external address and enabling a more efficient interface to external memories. BAM is ignored during DRAM access (BAT[1 – 0] = 10).</p> <p>NOTE: The BAM bit has no effect in DSP56300 core devices with only eighteen address lines.</p>
5	BYEN	0	<p>Bus Y Data Memory Enable Defines whether the AA/RAS pin and logic should be activated during external Y data space accesses. When set, BYEN enables the comparison of the external address to the BAC bits during external Y data space accesses. If BYEN is cleared, no address comparison is performed during external Y data space accesses.</p>

Table 9-4 AAR Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
4	BXEN	0	Bus X Data Memory Enable Defines whether the AA/ $\overline{\text{RAS}}$ pin and logic should be activated during external X data space accesses. When set, BXEN enables the comparison of the external address to the BAC bits during external X data space accesses. If BXEN is cleared, no address comparison is performed during external X data space accesses.
3	BPEN	0	Bus Program Memory Enable Defines whether or not the AA/ $\overline{\text{RAS}}$ pin and logic should be activated during external program space accesses. When set, BPEN enables the comparison of the external address to the BAC bits during external program space accesses. If BPEN is cleared, no address comparison is performed during external program space accesses.
2	BAAP	0	Bus Address Attribute Polarity Defines whether the AA/ $\overline{\text{RAS}}$ signal is active low or active high. When BAAP is cleared, the AA/ $\overline{\text{RAS}}$ signal is active low (useful for enabling memory modules or for DRAM Row Address Strobe). If BAAP is set, the appropriate AA/ $\overline{\text{RAS}}$ signal is active high (useful as an additional address bit).
1 – 0	BAT	0	Bus Access Type Define the type of external memory (DRAM or SRAM) to access for the area defined by the BAC[11 – 0], BYEN, BXEN, and BPEN bits. The encoding of BAT[1 – 0] is: 00 = Reserved 01 = SRAM access 10 = DRAM access 11 = Reserved When the external access type is defined as DRAM access (BAT[1 – 0] = 10), AA/ $\overline{\text{RAS}}$ acts as a Row Address Strobe ($\overline{\text{RAS}}$) signal. Otherwise, it acts as an Address Attribute signal. External accesses to the default area are always executed as if BAT[1 – 0] = 01 (that is, SRAM access). NOTE: If Port A is used for external accesses, the BAT bits in AAR0 – AAR3 must be initialized to the SRAM access type (that is, BAT = 01) or to the DRAM access type (that is, BAT = 10). To ensure proper operation of Port A, this initialization must occur even for an AAR register that is not used during a Port A access. At reset the BAT bits are initialized to 00.

9.6.2 Bus Control Register

The Bus Control Register (BCR) is a 24-bit read/write register that controls the external bus activity and Bus Interface Unit operation. All BCR bits except bit 21, BBS, are read/write bits. The BCR bits are shown in **Figure 9-8**.

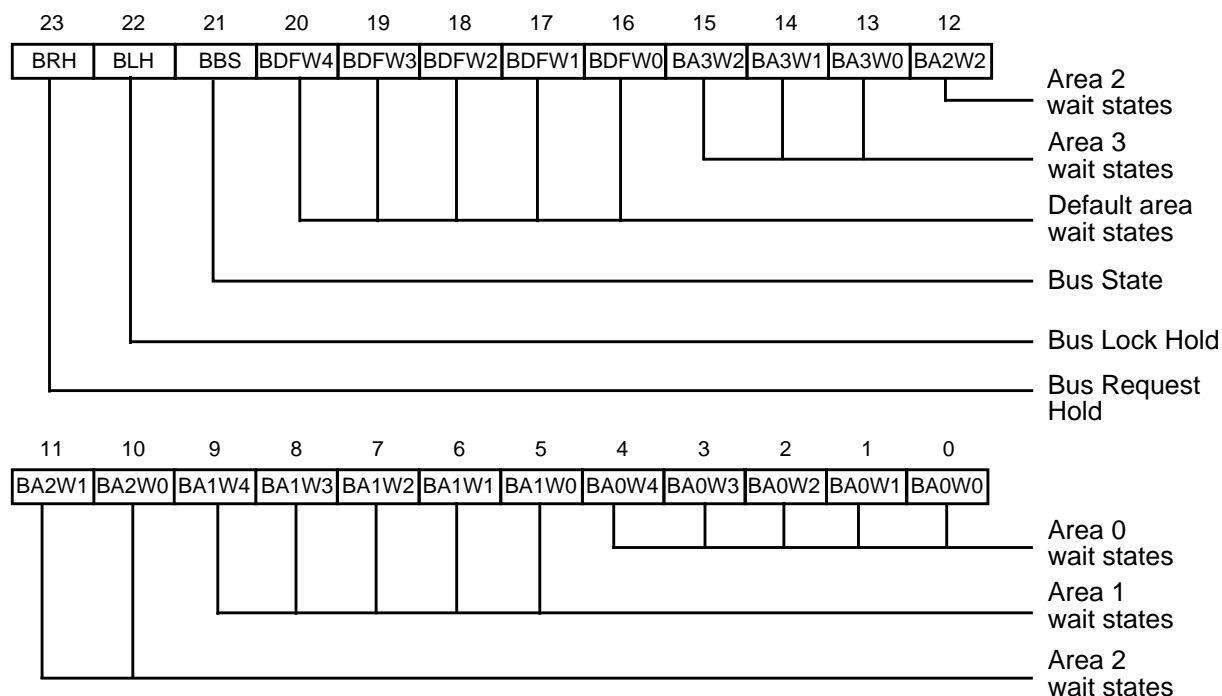


Figure 9-8. Bus Control Register (BCR)

Table 9-5 Bus Control Register (BCR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23	BRH	0	Bus Request Hold Asserts the \overline{BR} signal, even if no external access is needed. When BRH is set, the \overline{BR} signal is always asserted. If BRH is cleared, the \overline{BR} is asserted only if an external access is attempted or pending.
22	BLH	0	Bus Lock Hold Asserts the \overline{BL} signal, even if no read-modify-write access is occurring. When BLH is set, the \overline{BL} signal is always asserted. If BLH is cleared, the \overline{BL} signal is asserted only if a read-modify-write external access is attempted.
21	BBS	0	Bus State This read-only bit is set when the DSP is the bus master and is cleared otherwise.
20 – 16	BDFW	11111 (31 wait states)	Bus Default Area Wait State Control Defines the number of wait states (one through 31) inserted into each external access to an area that is not defined by any of the AAR registers. The access type for this area is SRAM only. These bits should not be programmed as zero since SRAM memory access requires at least one wait state. When four through seven wait states are selected, one additional wait state is inserted at the end of the access. When selecting eight or more wait states, two additional wait states are inserted at the end of the access. These trailing wait states increase the data hold time and the memory release time and do not increase the memory access time.

Table 9-5 Bus Control Register (BCR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
15 – 13	BA3W	1 (7 wait states)	<p>Bus Area 3 Wait State Control Defines the number of wait states (one through seven) inserted in each external SRAM access to Area 3 (DRAM accesses are not affected by these bits). Area 3 is the area defined by AAR3.</p> <p>NOTE: Do not program the value of these bits as zero since SRAM memory access requires at least one wait state.</p> <p>When four through seven wait states are selected, one additional wait state is inserted at the end of the access. This trailing wait state increases the data hold time and the memory release time and does not increase the memory access time.</p>
12 – 10	BA2W	111 (7 wait states)	<p>Bus Area 2 Wait State Control Defines the number of wait states (one through seven) inserted into each external SRAM access to Area 2 (DRAM accesses are not affected by these bits). Area 2 is the area defined by AAR2.</p> <p>NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.</p> <p>When four through seven wait states are selected, one additional wait state is inserted at the end of the access. This trailing wait state increases the data hold time and the memory release time and does not increase the memory access time.</p>
9 – 5	BA1W	11111 (31 wait states)	<p>Bus Area 1 Wait State Control Defines the number of wait states (one through 31) inserted into each external SRAM access to Area 1 (DRAM accesses are not affected by these bits). Area 1 is the area defined by AAR1.</p> <p>NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.</p> <p>When four through seven wait states are selected, one additional wait state is inserted at the end of the access. When selecting eight or more wait states, two additional wait states are inserted at the end of the access. These trailing wait states increase the data hold time and the memory release time and do not increase the memory access time.</p>
4 – 0	BA0W	11111 (31 wait states)	<p>Bus Area 0 Wait State Control Defines the number of wait states (one through 31) inserted in each external SRAM access to Area 0 (DRAM accesses are not affected by these bits). Area 0 is the area defined by AAR0.</p> <p>NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.</p> <p>When selecting four through seven wait states, one additional wait state is inserted at the end of the access. When selecting eight or more wait states, two additional wait states are inserted at the end of the access. These trailing wait states increase the data hold time and the memory release time and do not increase the memory access time.</p>

9.6.3 DRAM Control Register

The DRAM controller is an efficient interface to dynamic RAM devices in both random read/write cycles and Fast Access mode (Page mode). An on-chip DRAM controller controls the page hit circuit, the address multiplexing (row address and column address), the control signal generation ($\overline{\text{CAS}}$ and $\overline{\text{RAS}}$) and the refresh access generation ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) for a variety of DRAM module sizes and access times. The on-chip DRAM controller configuration is determined by the DRAM Control Register (DCR). The DRAM Control Register (DCR) is a 24-bit read/write register that controls and configures the external DRAM accesses. The DCR bits are shown in **Figure 9-9**.

Note: To prevent improper device operation, you must guarantee that all the DCR bits except BSTR are not changed during a DRAM access.

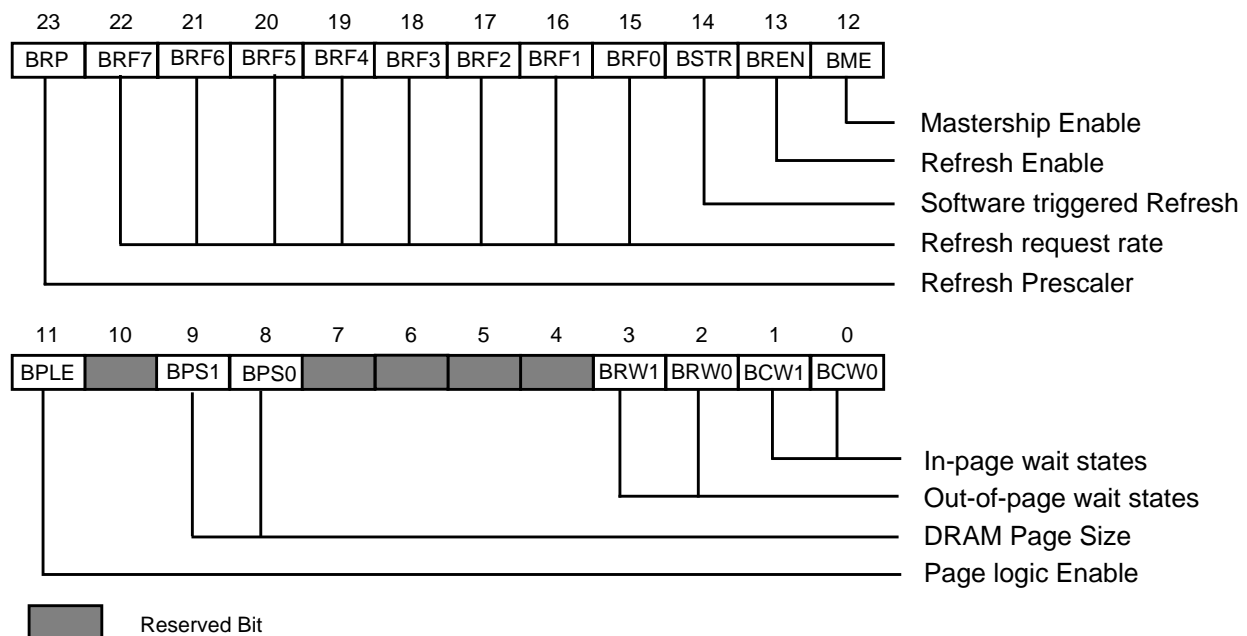


Figure 9-9. DRAM Control Register (DCR)

Table 9-6 DRAM Control Register (DCR) Bit Definitions

Bit Number	Bit Name	Reset Value	Description
23	BRP	0	<p>Bus Refresh Prescaler Controls a prescaler in series with the refresh clock divider. If BRP is set, a divide-by-64 prescaler is connected in series with the refresh clock divider. If BRP is cleared, the prescaler is bypassed. The refresh request rate (in clock cycles) is the value written to BRF[7 – 0] bits + 1, multiplied by 64 (if BRP is set) or by one (if BRP is cleared).</p> <p>NOTE: Refresh requests are not accumulated and, therefore, in a fast refresh request rate not all the refresh requests are served (for example, the combination BRF[7 – 0] = \$00 and BRP = 0 generates a refresh request every clock cycle, but a refresh access takes at least five clock cycles).</p> <p>When programming the periodic refresh rate, you must consider the $\overline{\text{RAS}}$ time-out period. Hardware support for the $\overline{\text{RAS}}$ time-out restriction does not exist.</p>
22 – 15	BRF	0	<p>Bus Refresh Rate Controls the refresh request rate. The BRF[7 – 0] bits specify a divide rate of 1–256 (BRF[7 – 0] = \$00–\$FF). A refresh request is generated each time the refresh counter reaches zero if the refresh counter is enabled (BRE = 1).</p>
14	BSTR	0	<p>Bus Software Triggered Reset Generates a software-triggered refresh request. When BSTR is set, a refresh request is generated and a refresh access is executed to all DRAM banks (the exact timing of the refresh access depends on the pending external accesses and the status of the BME bit). After the refresh access ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) is executed, the DRAM controller hardware clears the BSTR bit. The refresh cycle length depends on the BRW[1 – 0] bits (a refresh access is as long as the out-of-page access).</p>
13	BREN	0	<p>Bus Refresh Enable Enables/disables the internal refresh counter. When BREN is set, the refresh counter is enabled and a refresh request ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) is generated each time the refresh counter reaches zero. A refresh cycle occurs for all DRAM banks together (that is, all pins that are defined as $\overline{\text{RAS}}$ are asserted together). When this bit is cleared, the refresh counter is disabled and a refresh request may be software triggered by using the BSTR bit.</p> <p>In a system in which DSPs share the same DRAM, the DRAM controller of more than one DSP may be active, but it is recommended that only one DSP have its BREN bit set and that bus mastership is requested for a refresh access.</p> <p>If BREN is set and a WAIT instruction is executed, periodic refresh is still generated each time the refresh counter reaches zero.</p> <p>If BREN is set and a STOP instruction is executed, periodic refresh is not generated and the refresh counter is disabled. The contents of the DRAM are lost.</p>

Table 9-6 DRAM Control Register (DCR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
12	BME	0	<p>Bus Mastership Enable</p> <p>Enables/disables interface to a local DRAM for the DSP. When BME is cleared, the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ pins are tri-stated when mastership is lost. Therefore, you must connect an external pull-up resistor to these pins. In this case (BME = 0), the DSP DRAM controller assumes a page fault each time the mastership is lost. A DRAM refresh requires a bus mastership. If the BME bit is set, the RAS and $\overline{\text{CAS}}$ pins are always driven from the DSP. Therefore, DRAM refresh can be performed, even if the DSP is not the bus master.</p>
11	BPLE	0	<p>Bus Page Logic Enable</p> <p>Enables/disables the in-page identifying logic. When BPLE is set, it enables the page logic (the page size is defined by BPS[1 – 0] bits). Each in-page identification causes the DRAM controller to drive only the column address (and the associated $\overline{\text{CAS}}$ signal). When BPLE is cleared, the page logic is disabled, and the DRAM controller always accesses the external DRAM in out-of-page accesses (for example, row address with $\overline{\text{RAS}}$ assertion and then column address with $\overline{\text{CAS}}$ assertion). This mode is useful for low power dissipation. Only one in-page identifying logic exists. Therefore, during switches from one DRAM external bank to another DRAM bank (the DRAM external banks are defined by the access type bits in the AARs, different external banks are accessed through different AA/$\overline{\text{RAS}}$ pins), a page fault occurs.</p>
10		0	Reserved. Write to zero for future compatibility.
9 – 8	BPS	0	<p>Bus DRAM Page Size</p> <p>Defines the size of the external DRAM page and thus the number of the column address bits. The internal page mechanism works according to these bits only if the page logic is enabled (by the BPLE bit). The four combinations of BPS[1 – 0] enable the use of many DRAM sizes (1 M bit, 4 M bit, 16 M bit, and 64 M bit). The encoding of BPS[1 – 0] is:</p> <p>00 = 9-bit column width, 512 01 = 10-bit column width, 1 K 10 = 11-bit column width, 2 K 11 = 12-bit column width, 4 K</p> <p>When the row address is driven, all 24 bits of the external address bus are driven [for example, if BPS[1 – 0] = 01, when driving the row address, the 14 MSBs of the internal address (XAB, YAB, PAB, or DAB) are driven on address lines A0–A13, and the address lines A[14 – 23] are driven with the 10 MSBs of the internal address. This method enables the use of different DRAMs with the same page size.</p>
7 – 4		0	Reserved. Write to zero for future compatibility.

Table 9-6 DRAM Control Register (DCR) Bit Definitions (Continued)

Bit Number	Bit Name	Reset Value	Description
3 – 2	BRW	0	<p>Bus Row Out-of-page Wait States Defines the number of wait states that should be inserted into each DRAM out-of-page access. The encoding of BRW[1 – 0] is:</p> <p>00 = 4 wait states for each out-of-page access 01 = 8 wait states for each out-of-page access 10 = 11 wait states for each out-of-page access 11 = 15 wait states for each out-of-page access</p>
1 – 0	BCW	0	<p>Bus Column In-page Wait State Defines the number of wait states to insert for each DRAM in-page access. The encoding of BCW[1 – 0] is:</p> <p>00 = 1 wait state for each in-page access 01 = 2 wait states for each in-page access 10 = 3 wait states for each in-page access 11 = 4 wait states for each in-page access</p>