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 M × 24-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 (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 M × 24-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

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A[0 – 17]/A[0 – 23]</td>
<td>Output</td>
<td>Tri-stated</td>
<td>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.</td>
</tr>
</tbody>
</table>

Note: The total number of address lines is device-specific.

Table 9-2  External Data Bus Signals

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D[0 – 23]</td>
<td>Input/Output</td>
<td>Tri-stated</td>
<td>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.</td>
</tr>
</tbody>
</table>

Table 9-3  External Bus Control Signals

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA0–AA3</td>
<td>Output</td>
<td>Tri-stated</td>
<td>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, &quot;Address Attribute Registers (AAR0–AAR3),&quot; on page 9-15.</td>
</tr>
<tr>
<td>RAS[0 – 3]</td>
<td>Output</td>
<td>Tri-stated</td>
<td>Row Address Strobe—When defined as RAS (using the BAT bits in the corresponding AAR—see the BAT bits description in Section 9.6.1, &quot;Address Attribute Registers (AAR0–AAR3),&quot; on page 9-15), these signals can be used as RAS for the Dynamic Random Access Memory (DRAM) interface. These signals are tri-statable outputs with programmable polarity.</td>
</tr>
<tr>
<td>RD</td>
<td>Output</td>
<td>Tri-stated</td>
<td>Read Enable—When the DSP is the bus master, RD is an active-low output that is asserted to read external memory on the data bus (D[0 – 23]). Otherwise, RD is tri-stated.</td>
</tr>
</tbody>
</table>
### Table 9-3  External Bus Control Signals (Continued)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WR</td>
<td>Output</td>
<td>Tri-stated</td>
<td><strong>Write Enable</strong>—When the DSP is the bus master, 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.</td>
</tr>
<tr>
<td>BS</td>
<td>Output</td>
<td>Tri-stated</td>
<td><strong>Bus Strobe</strong>—When the DSP is the bus master, 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, BS remains deasserted until the next external bus cycle. NOTE: This signal is not implemented on all devices in the DSP56300 family.</td>
</tr>
</tbody>
</table>
| 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 TA input is ignored. The 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 TA deasserted. In typical operation, TA is:  
  - deasserted at the start of a bus cycle  
  - asserted to enable completion of the bus cycle  
  - deasserted before the next bus cycle  
  
  The current bus cycle completes one clock period after TA is asserted synchronously to CLKOUT. The number of wait states is determined by the 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 TA functionality, the BCR must be programmed to at least one wait state. A zero wait state access cannot be extended by TA deassertion, otherwise improper operation may result. TA can operate synchronously or asynchronously depending on the setting of the TAS bit in the Operating Mode Register (OMR).  
  
  NOTE: Do not use TA functionality while performing DRAM type accesses; otherwise, improper operation may result.  
  
  When the DSP56300 family device is the bus master, but TA is not used for external bus control, TA must be asserted low (pulled down). |
### Signal Description

#### Table 9-3  External Bus Control Signals (Continued)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BR</strong></td>
<td>Output</td>
<td>Output (deasserted)</td>
<td><strong>Bus Request</strong>—An active-low output that is never tri-stated. BR is asserted when the DSP requests bus mastership. BR is deasserted when the DSP no longer needs the bus. 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 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 BR to be asserted under software control, even though the DSP does not need the bus. BR is typically sent to an external bus arbiter that controls the priority, parking, and tenure of each master on the same external bus. BR is only affected by DSP requests for the external bus, never for the internal bus. During hardware reset, BR is deasserted; arbitration is reset to the bus slave state.</td>
</tr>
<tr>
<td><strong>BG</strong></td>
<td>Input</td>
<td>Ignored Input</td>
<td><strong>Bus Grant</strong>—Asserted by an external bus arbitration circuit when the DSP56300 family device becomes the next bus master. BG must be asserted/deasserted synchronous to CLKOUT for proper operation. When BG is asserted, the DSP56300 family device must wait until BB is deasserted before taking bus mastership. When 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.</td>
</tr>
<tr>
<td><strong>BB</strong></td>
<td>Input/Output</td>
<td>Input</td>
<td><strong>Bus Busy</strong>—Indicates that the bus is active. BB must be asserted and deasserted synchronous to CLKOUT. Only after BB is deasserted can a pending bus master become the bus master (and assert BB). Some designs allow a bus master to keep 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 BB uses an “active pull-up” method (that is, BB is driven high and then released and held high by an external pull-up resistor). BB requires an external pull-up resistor.</td>
</tr>
<tr>
<td><strong>BL</strong></td>
<td>Output</td>
<td>Driven high</td>
<td><strong>Bus Lock</strong>—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. 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, BL remains deasserted until the next external indivisible read-modify-write cycle. The only instructions that assert BL automatically are BSET, BCLR, and BCHG when the access is to external memory. An operation can also assert BL by setting the BLH bit in the BCR. This signal is not implemented on all devices in the DSP56300 family.</td>
</tr>
</tbody>
</table>
### External Bus Control Signals (Continued)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>State During Reset</th>
<th>Signal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS</td>
<td>Output</td>
<td>Tri-stated</td>
<td>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.</td>
</tr>
<tr>
<td>BCLK</td>
<td>Output</td>
<td>Tri-stated</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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 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. TA provides dynamic bus access timing. The number of wait states for each external access is determined by the 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 (BS) are asserted in the middle of CLKOUT high phase.

2. Write enable (WR) is asserted with the falling edge of CLKOUT (for a single wait state access). Read enable (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 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](image-url)
Port Operation

Motorola External Memory Interface (Port A)

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 (CAS and RAS), and refresh access generation (CAS before 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
Port Operation

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 (RAS). An in-page access is assumed, and 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 (CAS before 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
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 (BS) are asserted in the middle of CLKOUT high phase.
2. Write (WR) or Read (RD) is asserted with the CLKOUT falling edge.
3. 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. CAS is deasserted before the end of the external access in order to meet the 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 RAS
2. Assertion of the control signals (WR/RD)
3. After RAS precharge time, the assertion of RAS. RAS assertion and CAS timing depend on the number of out-of-page wait states selected by the BRW bits in the DCR.

Figure 9-5. Typical DRAM Connection Diagram
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 \( BR \) and \( BL \), tri-state \( BB \), and ignore \( BG \)). 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 (\( BR \) and \( BG \)) are local arbitration signals between a potential bus master and the arbitration logic; \( BB \) is a system arbitration signal:

- **Bus Request (\( BR \))**—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 (\( BG \))**—Asserted by the bus arbitration controller to signal the requesting device that it is the bus master elect; \( BG \) is valid only when the bus is not busy (that is, \( BB \) is not asserted).

- **Bus Busy (\( BB \))**—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. \( BB \) 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 \( BR \) signals as soon as they need the bus.

2. **Bus Granted by Arbiter**—The arbitration logic designates a bus master-elect by asserting the \( BG \) signal for that device.

3. **Bus Released by Current Master**—The master-elect tests \( BB \) to ensure that the previous master has relinquished the bus. If \( BB \) is deasserted, then the master-elect asserts \( BB \), which designates the device as the new bus master. If a higher priority bus request occurs before the \( BB \) signal is deasserted, then the arbitration logic may replace the current master-elect with the higher priority candidate. However, only one \( BG \) signal may be asserted at one time.

4. **Bus Control Assumed by New Master**—The new bus master begins its bus transfers after asserting \( BB \).

5. **Bus Grant Withdrawn by Arbiter**—The arbitration logic signals the new bus master to relinquish the bus by deasserting \( BG \) at any time.

6. **Bus Released by Current Master**—A DSP56300 core bus master releases its ownership (drives \( BB \) 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 \( BC \) signal and only relinquishes the bus (and deasserts \( BC \)) after completing the entire read-modify-write sequence. When the current bus master releases \( BB \), it first drives the \( BB \) signal high and then the \( BB \) signal is held by the pull-up resistor. The next bus master-elect has received its \( BG \) signal and is waiting for \( BB \) 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 \( BR \) and \( BC \) 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 BR signal only as long as requests for bus transfers are pending or being attempted. If the BCR[BRH] is set, BR remains asserted.

BLH Bit—If the BCR[BLH] bit is cleared, the DSP56300 core asserts its BL signal only during a read-modify-write bus access. If the BCR[BLH] is set, 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 BB are released first. Only in the next cycle is BB released. Therefore, in Slow Bus Release mode, 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 BR and BB), and ignores 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 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 BB asserted after ceasing bus activity, regardless of whether 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](image)
9.5.3 Bus Arbitration Example Cases

The following paragraphs describe various bus arbitration examples.

9.5.3.1 Case 1—Normal

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

9.5.3.2 Case 2—Bus Busy

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

9.5.3.3 Case 3—Low Priority

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

9.5.3.4 Case 4—Default

The arbiter design may specify a default bus master. Such a design asserts $BG$ for the default device whenever no other device requests the bus. Thus, whenever $BB$ is deasserted (that is, the bus is not busy), the default device can take control of the bus by asserting $BB$ without asserting $BR$ first. As long as the bus arbiter leaves $BG$ asserted because no other requests are pending, then the default device continues to assert $BB$ and maintain its bus mastership. This condition is called bus parking and eliminates the need for the default bus master to rearbitrate 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 $BR$ to request bus mastership and the arbiter then asserts $BG$ to the requesting device and $BB$ is deasserted (that is, the bus is not busy), then the requesting device asserts $BB$ and takes control of the bus. If the master device executes a read-modify-write instruction that accesses external memory, then $BB$ remains asserted
until the entire read-modify-write instruction completes execution, even if the bus arbiter deasserts BG. After the execution is complete, the device then drives BB high thereby relinquishing the bus. In DSP56300 family devices in which it is implemented, the 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, 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 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 BR to request bus mastership and the arbiter responds by asserting BG to the requesting device. When BB is deasserted (that is, the bus is not busy), the requesting device asserts BB to assume bus mastership. When the requesting device no longer requires the bus, it deasserts BR, but if no other requests are pending, the bus arbiter leaves BG asserted and 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 rearbitrate 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 AAn/RASn 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.
Port A Control

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

Table 9-4  AAR Bit Definitions

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
### Port A Control

#### Motorola External Memory Interface (Port A)

### Table 9-4  AAR Bit Definitions (Continued)

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>BPAC</td>
<td>0</td>
<td><strong>Bus Packing Enable</strong>&lt;br&gt;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.&lt;br&gt;&lt;br&gt;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, <em>Using the DSP56300 Direct Memory Access Controller</em>, for more information. To prevent improper operation, DMA address + 1 and DMA address + 2 should not cross the AAR bank borders. 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).</td>
</tr>
<tr>
<td>6</td>
<td>BAM</td>
<td>0</td>
<td><strong>Bus Address Multiplexing</strong>&lt;br&gt;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).&lt;br&gt;&lt;br&gt;Note: The BAM bit has no effect in DSP56300 core devices with only eighteen address lines.</td>
</tr>
<tr>
<td>5</td>
<td>BYEN</td>
<td>0</td>
<td><strong>Bus Y Data Memory Enable</strong>&lt;br&gt;Defines whether the <strong>AA/RAS</strong> 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.</td>
</tr>
</tbody>
</table>
Port A Control

### 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.

#### Table 9-4 AAR Bit Definitions (Continued)

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>BXEN</td>
<td>0</td>
<td><strong>Bus X Data Memory Enable</strong>&lt;br&gt;Defines whether the AA/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.</td>
</tr>
<tr>
<td>3</td>
<td>BPEN</td>
<td>0</td>
<td><strong>Bus Program Memory Enable</strong>&lt;br&gt;Defines whether or not the AA/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.</td>
</tr>
<tr>
<td>2</td>
<td>BAAP</td>
<td>0</td>
<td><strong>Bus Address Attribute Polarity</strong>&lt;br&gt;Defines whether the AA/RAS signal is active low or active high. When BAAP is cleared, the AA/RAS signal is active low (useful for enabling memory modules or for DRAM Row Address Strobe). If BAAP is set, the appropriate AA/RAS signal is active high (useful as an additional address bit).</td>
</tr>
<tr>
<td>1 – 0</td>
<td>BAT</td>
<td>0</td>
<td><strong>Bus Access Type</strong>&lt;br&gt;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:&lt;br&gt;00 = Reserved&lt;br&gt;01 = SRAM access&lt;br&gt;10 = DRAM access&lt;br&gt;11 = Reserved&lt;br&gt;When the external access type is defined as DRAM access (BAT[1 – 0] = 10), AA/RAS acts as a Row Address Strobe (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).&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
Table 9-5  Bus Control Register (BCR) Bit Definitions

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>BRH</td>
<td>0</td>
<td><strong>Bus Request Hold</strong>&lt;br&gt;Asserts the BR signal, even if no external access is needed. When BRH is set, the BR signal is always asserted. If BRH is cleared, the BR is asserted only if an external access is attempted or pending.</td>
</tr>
<tr>
<td>22</td>
<td>BLH</td>
<td>0</td>
<td><strong>Bus Lock Hold</strong>&lt;br&gt;Asserts the BL signal, even if no read-modify-write access is occurring. When BLH is set, the BL signal is always asserted. If BLH is cleared, the BL signal is asserted only if a read-modify-write external access is attempted.</td>
</tr>
<tr>
<td>21</td>
<td>BBS</td>
<td>0</td>
<td><strong>Bus State</strong>&lt;br&gt;This read-only bit is set when the DSP is the bus master and is cleared otherwise.</td>
</tr>
<tr>
<td>20 – 16</td>
<td>BDFW</td>
<td>11111 (31 wait states)</td>
<td><strong>Bus Default Area Wait State Control</strong>&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
### Table 9-5  Bus Control Register (BCR) Bit Definitions (Continued)

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 – 13</td>
<td>BA3W</td>
<td>1 (7 wait states)</td>
<td><strong>Bus Area 3 Wait State Control</strong>&lt;br&gt;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.&lt;br&gt;&lt;br&gt;NOTE: Do not program the value of these bits as zero since SRAM memory access requires at least one wait state.&lt;br&gt;&lt;br&gt;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.</td>
</tr>
<tr>
<td>12 – 10</td>
<td>BA2W</td>
<td>111 (7 wait states)</td>
<td><strong>Bus Area 2 Wait State Control</strong>&lt;br&gt;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.&lt;br&gt;&lt;br&gt;NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.&lt;br&gt;&lt;br&gt;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.</td>
</tr>
<tr>
<td>9 – 5</td>
<td>BA1W</td>
<td>11111 (31 wait states)</td>
<td><strong>Bus Area 1 Wait State Control</strong>&lt;br&gt;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.&lt;br&gt;&lt;br&gt;NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.&lt;br&gt;&lt;br&gt;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.</td>
</tr>
<tr>
<td>4 – 0</td>
<td>BA0W</td>
<td>11111 (31 wait states)</td>
<td><strong>Bus Area 0 Wait State Control</strong>&lt;br&gt;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.&lt;br&gt;&lt;br&gt;NOTE: Do not program the value of these bits as zero, since SRAM memory access requires at least one wait state.&lt;br&gt;&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
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 (CAS and RAS) and the refresh access generation (CAS before 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)](image-url)
Table 9-6  DRAM Control Register (DCR) Bit Definitions

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>BRP</td>
<td>0</td>
<td><strong>Bus Refresh Prescaler</strong>&lt;br&gt;Controls a prescaler in series with the refresh clock divider. If BPR is set, a divide-by-64 prescaler is connected in series with the refresh clock divider. If BPR 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 BPR is set) or by one (if BPR is cleared).&lt;br&gt;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 BPR = 0 generates a refresh request every clock cycle, but a refresh access takes at least five clock cycles).&lt;br&gt;When programming the periodic refresh rate, you must consider the RAS time-out period. Hardware support for the RAS time-out restriction does not exist.</td>
</tr>
<tr>
<td>22 – 15</td>
<td>BRF</td>
<td>0</td>
<td><strong>Bus Refresh Rate</strong>&lt;br&gt;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).</td>
</tr>
<tr>
<td>14</td>
<td>BSTR</td>
<td>0</td>
<td><strong>Bus Software Triggered Reset</strong>&lt;br&gt;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 (CAS before 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).</td>
</tr>
<tr>
<td>13</td>
<td>BREN</td>
<td>0</td>
<td><strong>Bus Refresh Enable</strong>&lt;br&gt;Enables/disables the internal refresh counter. When BREN is set, the refresh counter is enabled and a refresh request (CAS before 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 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.&lt;br&gt;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.&lt;br&gt;If BREN is set and a WAIT instruction is executed, periodic refresh is still generated each time the refresh counter reaches zero.&lt;br&gt;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.</td>
</tr>
</tbody>
</table>
### Table 9-6  DRAM Control Register (DCR) Bit Definitions (Continued)

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Name</th>
<th>Reset Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| 12         | BME      | 0           | **Bus Mastership Enable**
Enables/disables interface to a local DRAM for the DSP. When BME is cleared, the RAS and 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 CAS pins are always driven from the DSP. Therefore, DRAM refresh can be performed, even if the DSP is not the bus master. |
| 11         | BPLE     | 0           | **Bus Page Logic Enable**
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 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 RAS assertion and then column address with 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/RAS pins), a page fault occurs. |
| 10         |          | 0           | Reserved. Write to zero for future compatibility. |
| 9 – 8      | BPS      | 0           | **Bus DRAM Page Size**
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:

- 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

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. |
| 7 – 4      |          | 0           | Reserved. Write to zero for future compatibility. |
Port A Control

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

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