2 EXPANSION PORT

2.1 INTRODUCTION

The expansion port of the DSP56300 Core has the following features/functions:

- Interrupt And Mode Control
- Clock and Phase-Locked Loop (PLL)
- On-chip Emulator Interface (OnCE)/JTAG Interface
- Memory Expansion Port (Port A)
- Emulation Port

Port A is the memory expansion port that can be used either for memory expansion or for memory-mapped I/O. A number of features make port A versatile and easy to use. These features provide 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 bit wide with a separate 24 bit address bus capable of sustained rate of one memory access per clock cycle for data space accesses (using synchronous static memory). External memory is divided into three 16M X 24 bit spaces - X, Y and P. An internal wait state generator can be programmed to insert up to 31 wait state if access to slower memory or I/O device is required. A bus wait signal allows an external device to control the number of wait states inserted in a bus access operation. Bus arbitration signals allow an external device use of the bus while internal operations continue using the internal memory.

2.2 EXPANSION PORT SIGNAL DESCRIPTION

2.2.1 Interrupt And Mode Control

$\overline{\text{RESET}}$

(Reset) - Active low, Schmitt trigger input. $\overline{\text{RESET}}$ is internally synchronized to the clock out (CLKOUT). When asserted, the chip is placed in the reset state and the internal phase generator is reset. The Schmitt trigger input allows a slowly rising input (such as a capacitor charging) to reliably reset the chip. If $\overline{\text{RESET}}$ is negated synchronous to the clock out (CLKOUT), exact start-up timing is guaranteed, allowing

multiple processors to start-up synchronously and operate together in "lock-step". When the $\overline{\text{RESET}}$ pin is negated, the initial chip operating mode is latched from the MODA, MODB, MODC and MODD pins.
$\overline{\text{RESET}}$ pin can tolerate 5V.

MODA/$\overline{\text{IRQA}}$ (Mode Select A/External Interrupt Request A) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODA/$\overline{\text{IRQA}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQA}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQA}}$ to exit the wait state. If the processor is in the STOP standby state and $\overline{\text{IRQA}}$ is asserted, the processor will exit the STOP state.
MODA/$\overline{\text{IRQA}}$ pin can tolerate 5V.

MODB/$\overline{\text{IRQB}}$ (Mode Select B/External Interrupt Request B) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODB/$\overline{\text{IRQB}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQB}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQB}}$ to exit the wait state.
MODB/$\overline{\text{IRQB}}$ pin can tolerate 5V.

MODC/$\overline{\text{IRQC}}$ (Mode Select C/External Interrupt Request C) - Active low Schmitt trigger input, internally synchronized to the clock out (CLKOUT). MODC/$\overline{\text{IRQC}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQC}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQC}}$ to exit the wait state.
MODC/$\overline{\text{IRQC}}$ pin can tolerate 5V.

MODD/$\overline{\text{IRQ}}$D (Mode Select D/External Interrupt Request D) - Active low Schmitt

trigger input, internally synchronized to the clock out (CLKOUT). MODD/$\overline{\text{IRQD}}$ selects the initial chip operating mode during hardware reset and becomes a level sensitive or negative edge triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, MODC and MODD select one of 16 initial chip operating modes, latched into the operating mode register (OMR) when the $\overline{\text{RESET}}$ pin is negated. If $\overline{\text{IRQD}}$ is asserted synchronous to the clock out (CLKOUT), multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQD}}$ to exit the wait state.
MODD/$\overline{\text{IRQD}}$ pin can tolerate 5V.

2.2.2 Clock and Phase-Locked Loop (PLL)

EXTAL (External Clock/Crystal Input) - This input connects the internal crystal oscillator input to an external crystal or an external clock.

XTAL (Crystal Output) - This output connects the internal crystal oscillator output to an external crystal. If an external clock is used, XTAL should not be connected.

PCAP (PLL capacitor) - This input connects the off-chip capacitor for PLL filter. One terminal of the capacitor is connected to PCAP while the other terminal is connected to PVCC.

CLKOUT (Clock Output) - This output pin provides an output clock synchronized to the internal core clock phase.

NOTE 1: If PLL is enabled and both the multiplication and division factors are equal to one, then CLKOUT is also synchronized to EXTAL.

NOTE 2: If PLL is disabled, CLKOUT frequency and the chip frequency is half the frequency of EXTAL.

PINIT/$\overline{\text{NMI}}$ (PLL Initial/Non Maskable Interrupt) - During the assertion of hardware reset, PINIT/$\overline{\text{NMI}}$ is configured as PINIT and its value is written into the PEN bit of the PLL control register and determines whether the PLL is enabled or disabled. After hardware reset negation and during normal instruction processing the PINIT/$\overline{\text{NMI}}$ Schmitt trigger input pin is configured as $\overline{\text{NMI}}$, a negative edge triggered, non maskable interrupt request, internally synchronized to the clock out (CLKOUT).
PINIT/$\overline{\text{NMI}}$ pin can tolerate 5V.

2.2.3 On-Chip Emulator Interface (OnCE)/JTAG Interface

$\overline{\text{DE}}$ (Debug Event) - This open drain bidirectional active low pin provides, as an input, a means of entering the debug mode of operation from an external command controller, and as an output, a means of acknowledging that the chip has entered the debug mode. This pin when asserted as an input causes the DSP56300 core to finish the current instruction being executed, save the instruction pipeline information, enter the debug mode and wait for commands to be entered from the debug serial input line. This pin is asserted as an output for three clock cycles when the chip enters the debug mode as a result of a debug request or as a result of meeting a breakpoint condition.

$\overline{\text{DE}}$ pin can tolerate 5V.

TCK (Test Clock) - The test clock input TCK pin is the test clock used to synchronize the JTAG test logic

TCK pin can tolerate 5V.

TDI (Test Data Input) - The test data input TDI pin is the serial input for test instructions and data. TDI is sampled on the rising edge of TCK and it has an internal pullup resistor.

TDI pin can tolerate 5V.

TDO (Test data output) - The test data output TDO pin is the serial output for test instructions and data. TDO is three-stateable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.

TMS (Test Mode Select) - The test mode select input (TMS) pin is used to sequence the test controller’s state machine. The TMS is sampled on the rising edge of TCK and it has an internal pullup resistor

TMS pin can tolerate 5V.

$\overline{\text{TRST}}$ (Test Reset) - This active low Schmitt trigger input pin $\overline{\text{TRST}}$ is used to asynchronously initialize the test controller. The $\overline{\text{TRST}}$ has an internal pullup resistor

$\overline{\text{TRST}}$ pin can tolerate 5V.

2.2.4 Expansion Port (Port A)

A0-A23 (Address Bus) - three-state. Active high outputs when a bus master, three-stated otherwise, specify the address for external program and data memory accesses. To minimize power dissipation, A0–A23 do not change state when external memory spaces are not being accessed. A0–A23 are three-stated during hardware reset.

D0-D23 (Data Bus) - three-state, active high, bidirectional input/outputs when a bus master. These pins provide the bidirectional data bus for external program and data memory accesses. D0–D23 are in the high impedance state when not a bus master, or when there is no external bus activity. They are also three-stated during hardware reset.

AA(3:0)/$\overline{\text{RAS}}$(3:0)(Address Attribute or Row Address Strobe) - three-state outputs with a programmable polarity. When defined as Address Attribute these signals can be used as chip selects or additional address lines. When defined as $\overline{\text{RAS}}$ these signals can be used as Row Address Strobe for DRAM interface. The AA/$\overline{\text{RAS}}$ pins are three stated during hardware reset.

$\overline{\text{RD}}$ (Read Enable) - three-state. Active low output when bus master, three-stated otherwise. $\overline{\text{RD}}$ is asserted to read external memory on the data bus (D0–D23). $\overline{\text{RD}}$ is three-stated during hardware reset.

$\overline{\text{WR}}$ (Write Enable) - three-state. Active low output when bus master, three-stated otherwise. $\overline{\text{WR}}$ is asserted to write external memory on the data bus (D0–D23). $\overline{\text{WR}}$ is three-stated during hardware reset.

$\overline{\text{TA}}$ (Transfer Acknowledge) - active low input. If the DSP56300 core is the bus master and there is no external bus activity or the DSP56300 core is not the bus master, the $\overline{\text{TA}}$ input is ignored. The $\overline{\text{TA}}$ input is a synchronous/asynchronous (according to TAS bit in the OMR register) “DTACK” function which can extend an external bus cycle indefinitely. Any number of wait states (1, 2,..., infinity) may be added to the wait states inserted by the BCR by keeping $\overline{\text{TA}}$ negated. In typical operation, $\overline{\text{TA}}$ is negated at the start of a bus cycle, is asserted to enable completion of the bus cycle and is negated before the next bus cycle. The current bus cycle completes one clock period after $\overline{\text{TA}}$ is asserted synchronous to CLKOUT. The number of wait states is determined by the $\overline{\text{TA}}$ 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. If $\overline{\text{TA}}$ is tied low (asserted) and no wait states are specified in the BCR register, zero wait states will be inserted into external bus cycles.

NOTE1 In order to use the $\overline{\text{TA}}$ functionality the BCR must be programmed to at least one wait state, a zero wait state access can not be extended by negating $\overline{\text{TA}}$, Otherwise improper operation may result.

NOTE2 $\overline{\text{TA}}$ functionality may not be used while performing DRAM type accesses, Otherwise improper operation may result.

$\overline{\text{BR}}$ (Bus Request) - active low output, never three-stated. $\overline{\text{BR}}$ is asserted when the CPU or DMA is requesting bus mastership. $\overline{\text{BR}}$ is negated when the CPU or DMA no longer needs the bus. $\overline{\text{BR}}$ may be asserted or negated independent of whether the DSP56300 Core is a bus master or a bus slave. Bus “parking” allows $\overline{\text{BR}}$ to be negated even though the DSP56300 Core is the bus master. See the description of bus “parking” in the $\overline{\text{BB}}$ pin description. The BRH bit in the Bus Control Register (Section 2.5.2) allows $\overline{\text{BR}}$ to be asserted under software control even though the CPU or DMA does not need the bus. $\overline{\mathrm{BR}}$ is typically sent to an external bus arbitrator which controls the priority, parking and tenure of each DSP56300 Core on the same external bus. $\overline{\mathrm{BR}}$ is only affected by CPU or DMA requests for the external bus, never for the internal bus. During hardware reset, $\overline{\mathrm{BR}}$ is negated and the arbitration is reset to the bus slave state.

$\overline{\mathrm{BG}}$

(Bus Grant) – active low input. $\overline{\mathrm{BG}}$ must be asserted/negated synchronous to the clock out (CLKOUT) for proper operation. $\overline{\mathrm{BG}}$ is asserted by an external bus arbitration circuit when the DSP56300 Core may become the next bus master. When $\overline{\mathrm{BG}}$ is asserted, the DSP56300 Core must wait until $\overline{\mathrm{BB}}$ is negated before taking bus mastership. When $\overline{\mathrm{BG}}$ is negated, bus mastership is typically given up at the end of the current bus cycle. This may occur in the middle of an instruction which requires more than one external bus cycle for execution. $\overline{\mathrm{BG}}$ is ignored during hardware reset.

$\overline{\mathrm{BB}}$

(Bus Busy) - bidirectional active low input/output, must be asserted and negated synchronous to the clock out (CLKOUT).This signal indicates that the bus is active. Only after this signal is negated the pending bus master can become the bus master (and then assert it again).The bus master may keep $\overline{\mathrm{BB}}$ asserted after ceasing bus activity regardless of whether $\overline{\mathrm{BR}}$ is asserted or negated, this is called “bus parking” and allows the current bus master to reuse the bus without re-arbitration until other device wants the bus. The negation of $\overline{\mathrm{BB}}$ is done by an “active pull-up” method i.e.$\overline{\mathrm{BB}}$ is driven high and then released and held high by an external pull-up resistor. $\overline{\mathrm{BB}}$ is an active low input during hardware reset.

NOTE: $\overline{\mathrm{BB}}$ requires an external pullup resistor.

$\overline{\mathrm{BL}}$

(Bus Lock) - active low output, never three-stated. Asserted at the start of an external indivisible Read-Modify-Write (RMW) bus cycle and negated at the end of the write bus cycle. $\overline{\mathrm{BL}}$ remains asserted between the read and write bus cycles of the RMW bus sequence. $\overline{\mathrm{BL}}$ may be used to “resource lock” an external multi-port memory for secure semaphore updates. The only instructions which automatically assert $\overline{\mathrm{BL}}$ are BSET, BCLR or BCHG instruction which accesses external memory. $\overline{\mathrm{BL}}$ can also be asserted by setting the BLH bit in the BCR register (see Section 2.5.2). $\overline{\mathrm{BL}}$ is negated during hardware reset.

$\overline{\mathrm{BS}}$

(Bus Strobe) - three-state. Active low output when a bus master, three-stated when not a bus master. Asserted at the start of a bus cycle (for half of a clock cycle) providing an “early bus start” signal for a bus controller. If the external bus is not used during an instruction cycle $\overline{\text{BS}}$ remains negated until the next external bus cycle. $\overline{\text{BS}}$ is three-stated during hardware reset.

$\overline{\text{CAS}}$ (Column Address Strobe) - active low output when bus master and three stated otherwise (if BME bit in DRAM control register is cleared), $\overline{\text{CAS}}$ is used by DRAM memories to strobe column address. $\overline{\text{CAS}}$ is three stated during hardware reset.

BCLK (Bus Clock) - three-state. Active high output when a bus master BCLK is used by synchronous SRAM to sample address, data and control signals. BCLK is active only during SSRAM accesses, when active BCLK is synchronized to CLKOUT by the internal Phase Lock Loop, BCLK precedes CLKOUT by 1/4 of a clock cycle. BCLK is three stated during hardware reset.

2.3 EXPANSION PORT OPERATION

The external bus timing is defined by the operation of the Address Bus, Data Bus and Bus Control pins described in the previous paragraph. The DSP56300 Core external ports are designed to interface with a wide variety of memory and peripheral devices, high speed synchronous static RAMs, high speed static RAMs and dynamic RAMs, as well as slower memory devices. For detailed explanation see the paragraphs on synchronous static RAM support, static RAM support and dynamic RAM support.

2.3.1 Static RAM support

External bus timing is controlled by the $\overline{\text{TA}}$ control signal and by the Bus Control Register (BCR) that is described in Section 2.5.2. Insertion of wait states is controlled by the BCR to provide constant bus access timing, and by $\overline{\text{TA}}$ to provide 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 is longer.
The external memory address is defined by the Address Bus A0-A23 and the Memory Address attribute signals AA(3:0). 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 signals for the appropriate memory chips. These chip select 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.

2.3.1.1 Synchronous Static RAM (SSRAM) Support

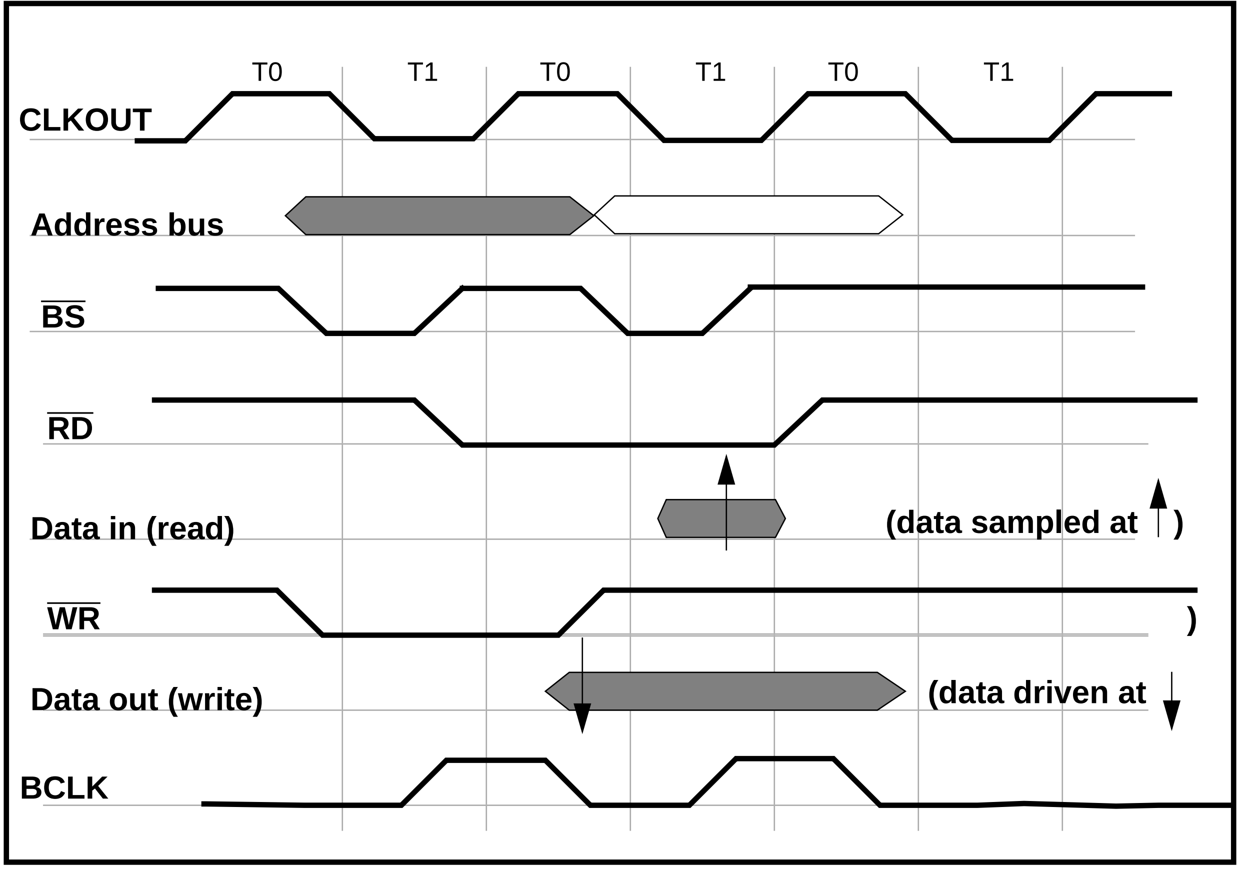
Synchronous Static RAM devices can be easily interfaced to the DSP56300 Core bus timing. The Synchronous Static RAMs internal pipeline fits the DSP56300 Core pipeline, and therefore permits high speed data transfers (each clock cycle, zero wait states) with a reasonable access time. Due to the DSP56300 Core pipeline structure, one cycle stall is inserted after performing fetch from external SSRAM. In such a case, the effective number of stall states in the pipeline will be the number specified in the Bus Control Register (BCR) + 1 (although the external access itself will be performed exactly the same for fetches and for data moves).

Figure 2-3 shows a connection configuration (for a detailed timing information see the specific DSP56300 Core based chip technical data sheet). The synchronous SRAM access is composed from the following steps:

1. The address - A(23:0), address attributes - AA(3:0),bus strobe - $\overline{\text{BS}}$, and write enable - $\overline{\text{WR}}$ are asserted before the rising edge of BCLK. ($\overline{\text{WR}}$ only for write accesses, for read accesses $\overline{\text{RD}}$ is asserted asynchronously to BCLK).
2. Bus strobe is negated before the change of the new address thus enabling the sample of address and control (for devices that do not use BCLK).
3. The address - A(23:0), address attributes - AA(3:0), and write enable - $\overline{\text{WR}}$ are negated (for write accesses) with the falling edge of BCLK (new address and controls are driven if another external SSRAM access is needed).
4. **For write operation**: Data is driven with the falling edge of BCLK
   For read operation: Data is sampled with the leading edge of BCLK.

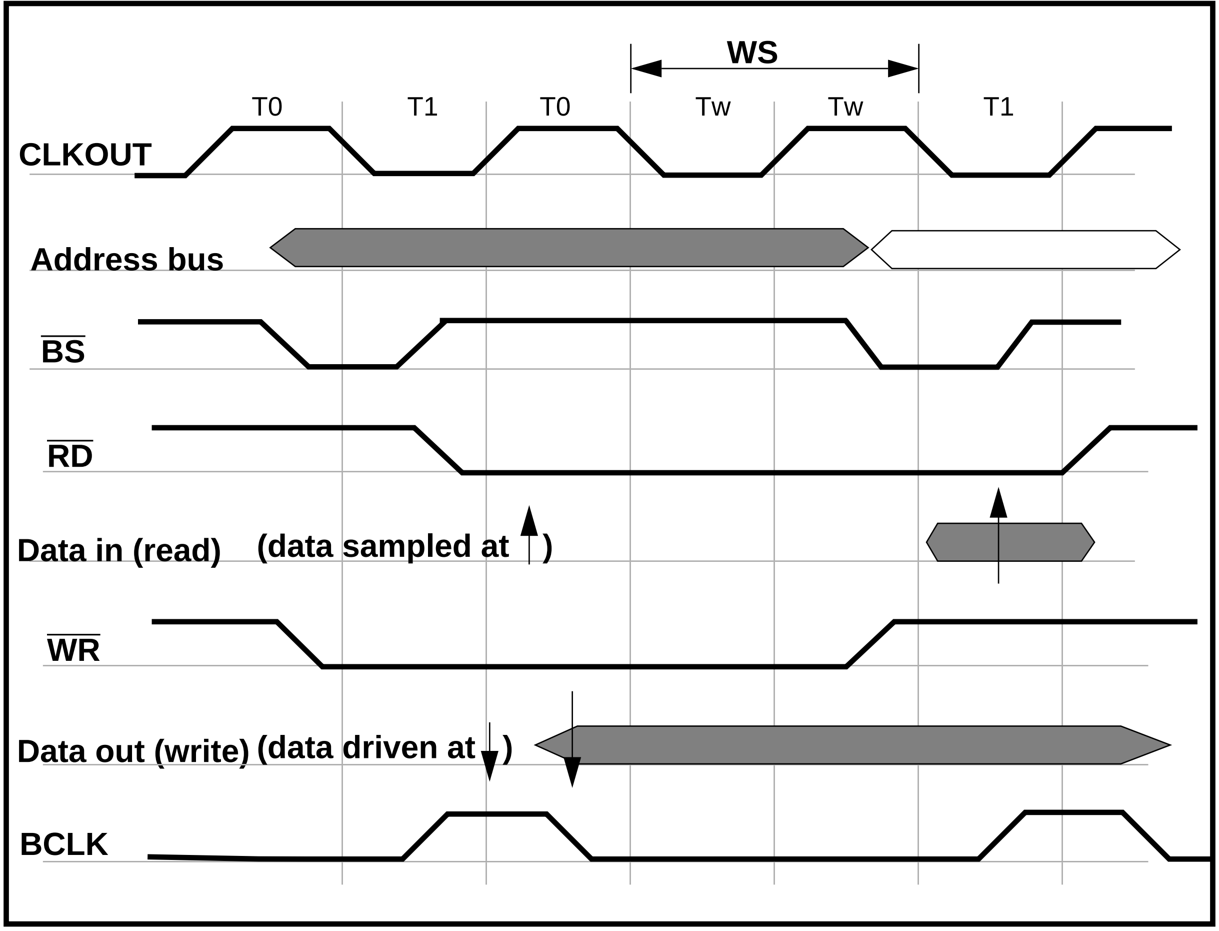
Wait states (from BCR or by $\overline{\text{TA}}$ signal) will postpone the appearance of the next leading edge of BCLK thus increasing memory access time (see Figure 2-2 on page 2-10).

Figure 2-1. Bus operation - zero wait states Sync. SRAM access



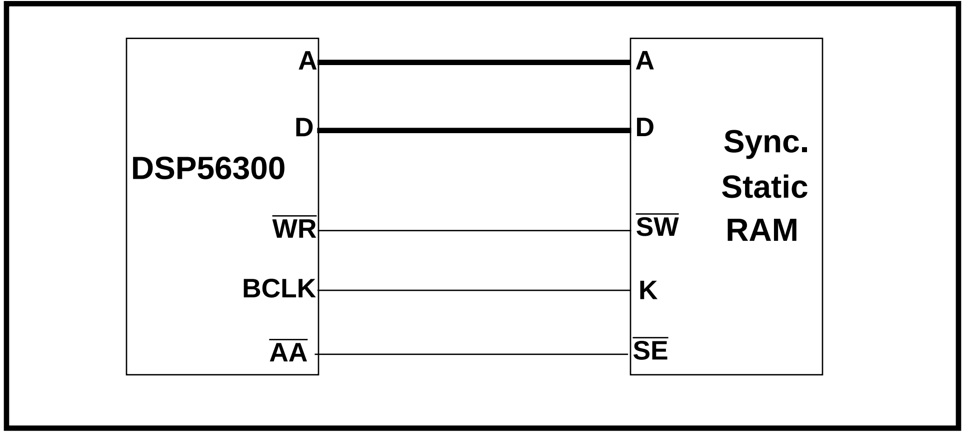
- for detailed timing specification see the specific data sheet

Figure 2-2. Bus operation - one wait state Sync. SRAM access



- for detailed timing specification see the specific data sheet

Figure 2-3. Synchronous Static RAM connection diagram



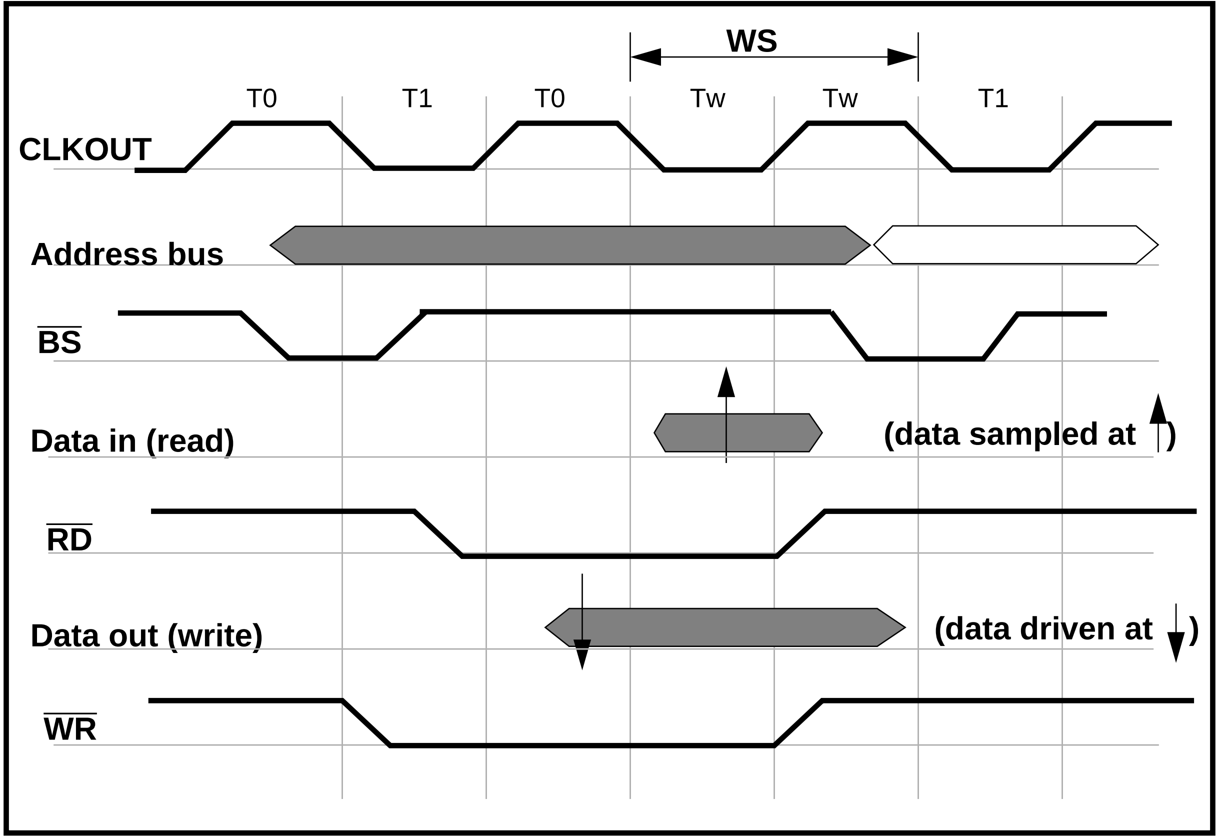
2.3.1.2 Asynchronous Static RAM (SRAM) Support

Static RAMs can be easily interfaced to the DSP56300 Core bus timing. Due to the Static RAM requirement to keep the address stable during the entire bus cycle, at least one wait state must be inserted to the bus operation. The next diagram shows a possible configuration (for a detailed timing information see the specific DSP56300 Core based chip technical data sheet).The static RAM access is composed from the following steps (see also Figure 2-4 on page 2-11)

1. The address - A(23:0), address attributes - AA(3:0),and bus strobe - $\overline{\text{BS}}$ are asserted in the middle of CLKOUT high phase.
2. Write enable - $\overline{\text{WR}}$ is asserted with the falling edge of CLKOUT (for a single wait state access). Read enable - $\overline{\text{RD}}$, is asserted in the middle of CLKOUT low phase.
3. **For write operation**: Data is driven in the middle of CLKOUT high phase. **For read operation**: Data is sampled in the middle of CLKOUT last low phase of the external access.

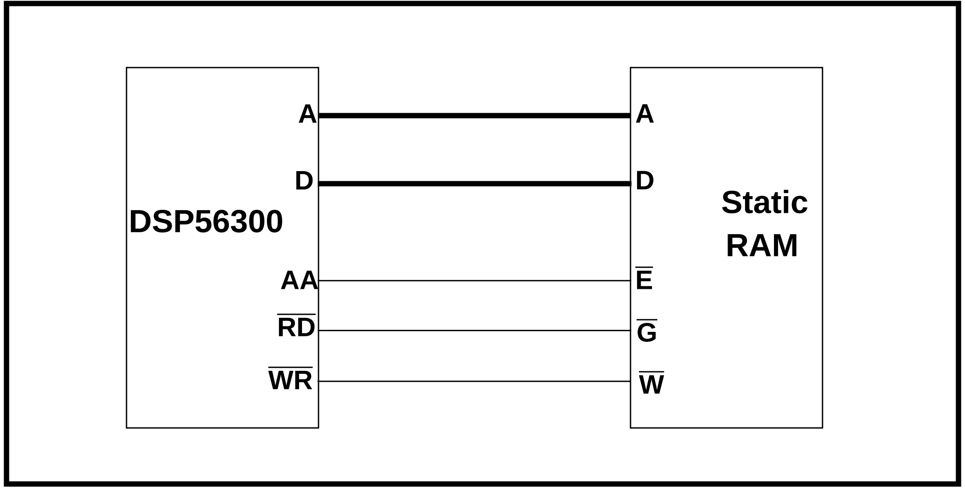
Wait states (from BCR or by $\overline{\text{TA}}$ signal) will postpone the disappearance of the external address thus increasing memory access time. In any case, static RAM access requires at least one wait state.

Figure 2-4. Bus operation one wait state - SRAM access



- for detailed timing specification see the specific data sheet

Figure 2-5. Static RAM connection diagram



NOTE 1: BCLK is negated during asynchronous SRAM access and therefore all signals timing are related to CLKOUT.

NOTE 2: When the external access type is defined as SRAM, the assertion of $\overline{\mathrm{WR}}$ signal depends on the number of wait states programmed in the BCR. If a single wait state is programmed in the BCR, $\overline{\mathrm{WR}}$ signal is asserted with the falling edge of CLKOUT. If the number of wait states programmed is 2 or 3, $\overline{\mathrm{WR}}$ assertion is delayed by half of a clock cycle (half CLKOUT cycle). If the number of wait states programmed is 4 or more, $\overline{\mathrm{WR}}$ 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 write.

2.3.2 Dynamic Memories Support

External bus timing is controlled by the DRAM Control Register (DCR) that is described in Section 2.6.1. Insertion of wait states is controlled by the DCR to provide constant bus access timing.

The external memory address is defined by the Address Bus A0-A23. The n low order address bits are multiplexed inside the DSP56300 Core, and the new 24 bits address is driven to the external bus.The address multiplexing enable glue-less interface to dynamic memories by simply connecting the low order n bits to the memory address pins. The Address Attribute signals function as $\overline{\mathrm{RAS}}$. An in page access is assumed and therefore $\overline{\mathrm{RAS}}$ is kept asserted unless one of the following occurs:

1. An out of page access is detected.
2. An access to another bank of dynamic memory is attempted.
3. A refresh access is attempted ($\overline{\mathrm{CAS}}$ before $\overline{\mathrm{RAS}}$).

4. A write to of the following registers is detected: BCR, DCR, AAR3, AAR2, AAR1, AAR0.
5. A lost of bus mastership is detected while the BME bit in the DCR register is cleared.
6. Wait or stop instruction are detected.
7. Hardware or software reset are detected.

Modern dynamic memory (DRAM) are becoming the preferred choice for a wide variety of computing systems based on

1. Cost per bit due to dynamic storage cell density.
2. Packaging density due to multiplexed address and control pins.
3. Improved price-performance relative to static RAMs due to fast access mode (page mode).
4. Commodity pricing due to high volume production.

Port A bus control signals are designed for 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 ($\overline{\text{CAS}}$ and $\overline{\text{RAS}}$) and refresh access generation ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) for a large variety of DRAM module sizes and different access times. The DRAM controller operation and programming is described in Section 2.6. The next diagram shows a possible configuration (for a detailed timing information see the specific DSP56300 Core based chip technical data sheet). The dynamic RAM access is composed from the following steps (in page access):

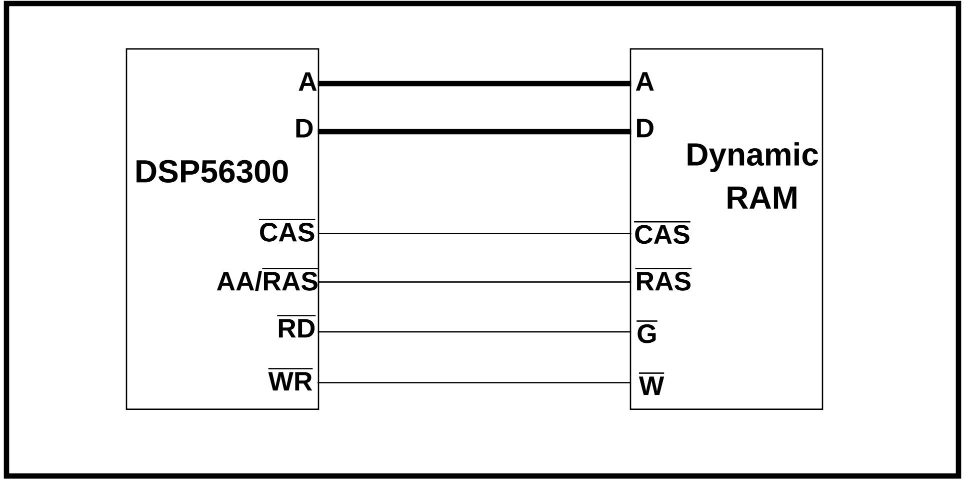
1. The column address - A(23:0), and bus strobe - BS are asserted in the middle of CLKOUT high phase.
2. Write enable - WR, or read enable - RD are asserted with the falling edge of CLKOUT.
3. $\overline{\text{CAS}}$ assertion timing depends on the number of in page wait states selected by BCW bits in DCR register and on the access purpose (read/write). (See Figure 2-7 on page 2-14 for DRAM in page 2 w.s example).
4. $\overline{\text{CAS}}$ is negated before the end of the external access in order to meet the $\overline{\text{CAS}}$ precharge timing.

In any case, DRAM access requires at least one wait state.

Out of page access: The out of page access will start with the negation of $\overline{\text{RAS}}$, the assertion of the control signals ($\overline{\text{WR/RD}}$) and after $\overline{\text{RAS}}$ precharge time the assertion of $\overline{\text{RAS}}$. $\overline{\text{RAS}}$ assertion, and $\overline{\text{CAS}}$ timing, depend on the number of out of page wait states selected by BRW bit in DCR register.

NOTE: The 56300 Core does not support external DRAM devices overlapping, i.e. two or more external DRAM devices, connected to different AA pins, with common addresses.

Figure 2-6. Dynamic RAM connection diagram



- Address line are multiplexed inside the DSP56300 Core

Figure 2-7. Bus operation two wait states - DRAM read access (in-page)

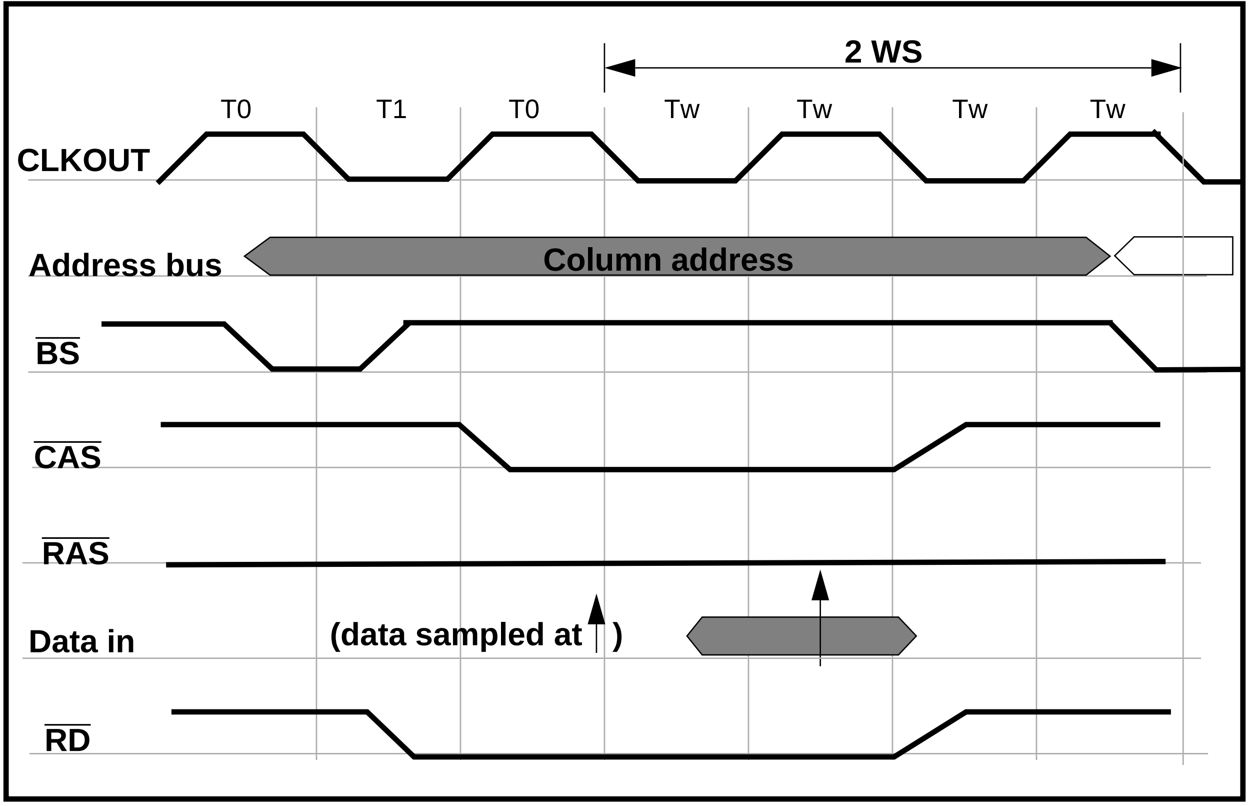
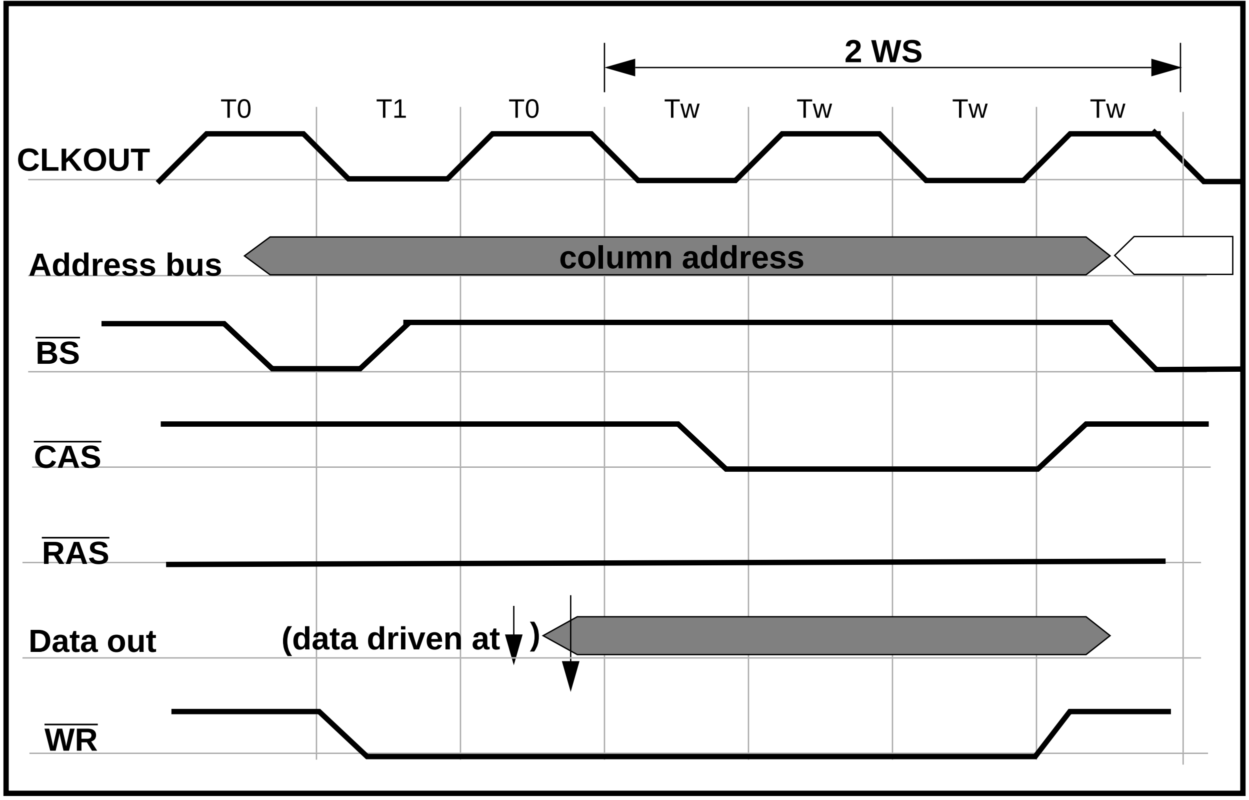


Figure 2-8. Bus operation two wait states - DRAM write access (in-page)



- for detailed timing specification see the specific data sheet

2.3.3 Expansion Port Stalls

In addition to the wait states that are controlled by the BCR, DCR and the $\overline{TA}$ negation there are two more cases where the expansion port controller stalls the DSP56300 pipeline.

2.3.3.1 External Fetch From Synchronous SRAM.

Due to the DSP56300 pipeline any external fetch from Synchronous SRAM will add one cycle stall. This one cycle stall will be inserted even if the synchronous SRAM already needs wait states for access time. The instruction cache enable/disable has no affect on this stall.

2.3.3.2 Non Synchronous SRAM Access Immediately Following Synchronous SRAM access.

Due to the synchronous SRAM pipelined access there is a possibility of contention on the data bus in a case of non synchronous SRAM (DRAM or asynchronous SRAM) access immediately following a synchronous SRAM access. This sequence is automatically detected by the expansion port control hardware, and a one cycle stall is inserted in order to avoid contention.
In case of a default area (always SRAM) access immediately following a synchronous SRAM access, the user should be careful that the SRAM will not be activated in the

second cycle of the synchronous SRAM access, because the select of the default area SRAM is generated externally (not one of the AA signals).

2.3.4 Expansion port Disable

In many application that are sensitive to the power consumption there is no use of the expansion port because all the memory reside inside the chip itself. A special feature of the expansion port controller enables the user to reduce significantly the power consumption of the expansion port controller by setting the EBD bit in the OMR register. If this bit is set the expansion port controller is disabled, the DSP56300 will release the bus i.e.negate $\overline{BR}$ and $\overline{BL}$, tristate $\overline{BB}$, and ignore $\overline{BG}$. Of course no external DMA accesses or refresh accesses can be performed. When EBD is set the user should not attempt to access the external memory, otherwise improper operation will result. Likewise, before EBD bit is set, the user should clear BREN (Refresh Enable - bit 13 in DCR) to prevent a refresh attempt to external DRAM, otherwise improper operation will result.

2.4 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 (e.g. fairness, priority, etc.). External logic must be provided by the system designer to implement the arbitration scheme.

2.4.1 Bus Arbitration Signals

Three signals are provided for bus arbitration. Two of them are considered as local arbitration signals and one as system arbitration signal. The local arbitration signals run between a potential bus master and the arbitration logic. The local signals are $\overline{BR}$ and $\overline{BG}$. $\overline{BB}$ is a system arbitration signal. These signals are described below.

$\overline{BR}$ Bus Request - Asserted by the requesting device to indicate that it wants to use the bus, and 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.

$\overline{BG}$ Bus Grant - 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 (Bus Busy signal - $\overline{BB}$ is described below).

$\overline{\text{BB}}$ Bus Busy - The system arbitration signal $\overline{\text{BB}}$ is monitored by all potential bus masters and is driven by the current bus master. This signal controls the hand-over of bus ownership by the bus master at the end of bus possession. $\overline{\text{BB}}$ is an active pull-up signal i.e.it is driven high before it is released (and then held high by an external pull-up resistor).

2.4.2 The Arbitration Protocol

The bus is arbitrated by a central bus arbitrator, using individual request/grant lines to each bus master. The arbitration protocol can operate in parallel with bus transfer activity so that the bus hand-over can be made without much performance penalty.

The arbitration sequence occurs as follows:

1. All candidates for bus ownership assert their respective $\overline{\text{BR}}$ signals as soon as they need the bus.
2. The arbitration logic designates a bus master-elect by asserting the $\overline{\text{BG}}$ signal for that device.
3. The master-elect tests $\overline{\text{BB}}$ to ensure that the previous master has relinquished the bus. If $\overline{\text{BB}}$ is negated, 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 was negated, then the arbitration logic may replace the current master-elect with the higher priority candidate. However, only one $\overline{\text{BG}}$ signal must be asserted at one time.
4. The new bus master begins its bus transfers after the assertion of $\overline{\text{BB}}$.
5. The arbitration logic signals the current bus master to relinquish the bus by negating $\overline{\text{BG}}$ at any time. An DSP56300 Core bus master releases its ownership (drives $\overline{\text{BB}}$ high and then release it) after completing the current external bus access except for the cases described in **NOTE2**. If an instruction is executing a Read-Modify-Write external access, an DSP56300 Core master asserts the $\overline{\text{BL}}$ signal and will only relinquish the bus (and negate $\overline{\text{BL}}$) after completing the entire Read-Modify-Write sequence. When the current bus master release $\overline{\text{BB}}$, 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 negated before claiming ownership
6. The possession of the bus by the new bus master is done by asserting the $\overline{\text{BB}}$ signal.

The DSP56300 Core has 2 control bits and one status bit, located in the Bus Control Register (BCR - see Section 2.5.2) to permit software control of the $\overline{\text{BR}}$ and $\overline{\text{BL}}$ signals, and to verify when the chip is the bus master. If the BRH bit in the BCR register 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 BRH bit is set, $\overline{\text{BR}}$ will remain asserted. If the BLH bit in the BCR register is cleared, the DSP56300 Core asserts its $\overline{\text{BL}}$ signal only during a read-modify-write bus access. If the BLH bit is set, $\overline{\text{BL}}$ will remain asserted (even when not a bus master).

The DSP56300 core has a control bit located in the Operating Mode Register (BRT in OMR register) that enable fast/slow bus release mode. In fast bus release mode all port A pins are three stated in the same cycle. In slow bus release mode an extra cycle is add, all port A pins except $\overline{BB}$ are released first and only in the next cycle $\overline{BB}$ is released. Therefore, in slow mode it is guaranteed that $\overline{BB}$ is the last pin that is three stated. This may be useful in systems where a possibility of contention exists. More detailed explanation (including timing diagrams) may be found in the data sheet.

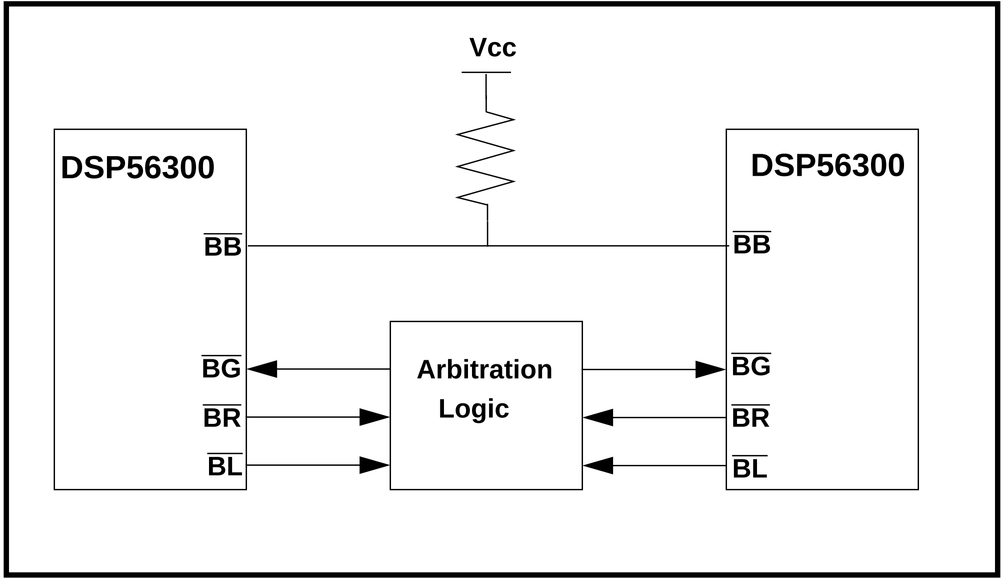
NOTE1 During the execution of WAIT and STOP instructions the DSP56300 will release the bus (i.e.negate $\overline{BR}$ and $\overline{BB}$), and ignore $\overline{BG}$.

NOTE2 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 form an arbitration point of view, i.e.the bus mastership will not be released during the execution of these accesses.

2.4.3 Arbitration Scheme

The bus arbitration scheme is implementation dependent. The diagram in Figure 2-9 on page 2-18 illustrates a common method of implementing the bus arbitration scheme. The arbitration logic determines the device priorities and assigns bus ownership depending on those priorities. An implementation of a bus arbitration scheme may hold $\overline{BG}$ asserted, for example, to the current bus owner if none of the other devices are requesting the bus. As a consequence, the current bus master may keep $\overline{BB}$ asserted after ceasing bus activity, regardless of whether $\overline{BR}$ is asserted or negated. 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 2-9. Bus Arbitration scheme



2.4.4 Bus Arbitration Example Cases

2.4.4.1 Case 1 – Normal

If the device requesting mastership asserts $\overline{\mathrm{BR}}$, the arbiter asserts the requesting devices $\overline{\mathrm{BG}}$ and $\overline{\mathrm{BB}}$ is driven high and then released, indicating the bus is not busy. The requesting device will assert $\overline{\mathrm{BB}}$.

2.4.4.2 Case 2 – Bus Busy

If the device requesting mastership asserts $\overline{\mathrm{BR}}$, the arbiter responds by asserting the requesting devices $\overline{\mathrm{BG}}$, however, the bus is busy because $\overline{\mathrm{BB}}$ is asserted. The requesting device will not assert $\overline{\mathrm{BB}}$ until $\overline{\mathrm{BB}}$ is driven high and then released by the current bus master.

2.4.4.3 Case 3 – Low Priority

If the device requesting mastership asserts $\overline{\mathrm{BR}}$, the arbiter withholds asserting the requesting devices $\overline{\mathrm{BG}}$ because a higher priority device requested the bus. $\overline{\mathrm{BB}}$ of the requesting device will not be asserted.

2.4.4.4 Case 4 – Default

If a device does not request the bus and the arbiter, by design (i.e. default), asserts $\overline{\mathrm{BG}}$ and $\overline{\mathrm{BB}}$ is negated indicating the bus is not busy. The granted device will assert $\overline{\mathrm{BB}}$. If the bus arbiter leaves $\overline{\mathrm{BG}}$ asserted because other requests are not pending, then $\overline{\mathrm{BB}}$ will remain asserted. 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.

2.4.4.5 Case 5 – Bus Lock during RMW

If the device requesting mastership asserts $\overline{\mathrm{BR}}$ and the arbiter asserts the requesting devices $\overline{\mathrm{BG}}$ and $\overline{\mathrm{BB}}$ is negated, then the requesting device will assert $\overline{\mathrm{BB}}$. If a read-modify-write (RMW) instruction which accesses external memory is being executed, and the bus arbiter negates $\overline{\mathrm{BG}}$, then $\overline{\mathrm{BB}}$ will remain asserted until the entire RMW instruction completes execution. $\overline{\mathrm{BB}}$ will then be driven high and released thereby relinquishing the bus. Note that during external RMW instruction execution, $\overline{\mathrm{BL}}$ is asserted. In general, the $\overline{\mathrm{BL}}$ signal can be used to ensure that a multiport memory can only be written by one master at a time.

2.4.4.6 Case 6 – Bus Park

The device requesting mastership asserts $\overline{\mathrm{BR}}$, the arbiter asserts the requesting devices $\overline{\mathrm{BG}}$ and $\overline{\mathrm{BB}}$ is negated indicating the bus is not busy – the requesting device will assert $\overline{\mathrm{BB}}$. When the requesting device no longer requires the bus it will negate $\overline{\mathrm{BR}}$. If the bus arbiter leaves $\overline{\mathrm{BG}}$ asserted because other requests are not pending, then $\overline{\mathrm{BB}}$ will remain asserted. This condition is called bus parking and eliminates the need for the last bus master to rearbitrate for the bus during its next external access.

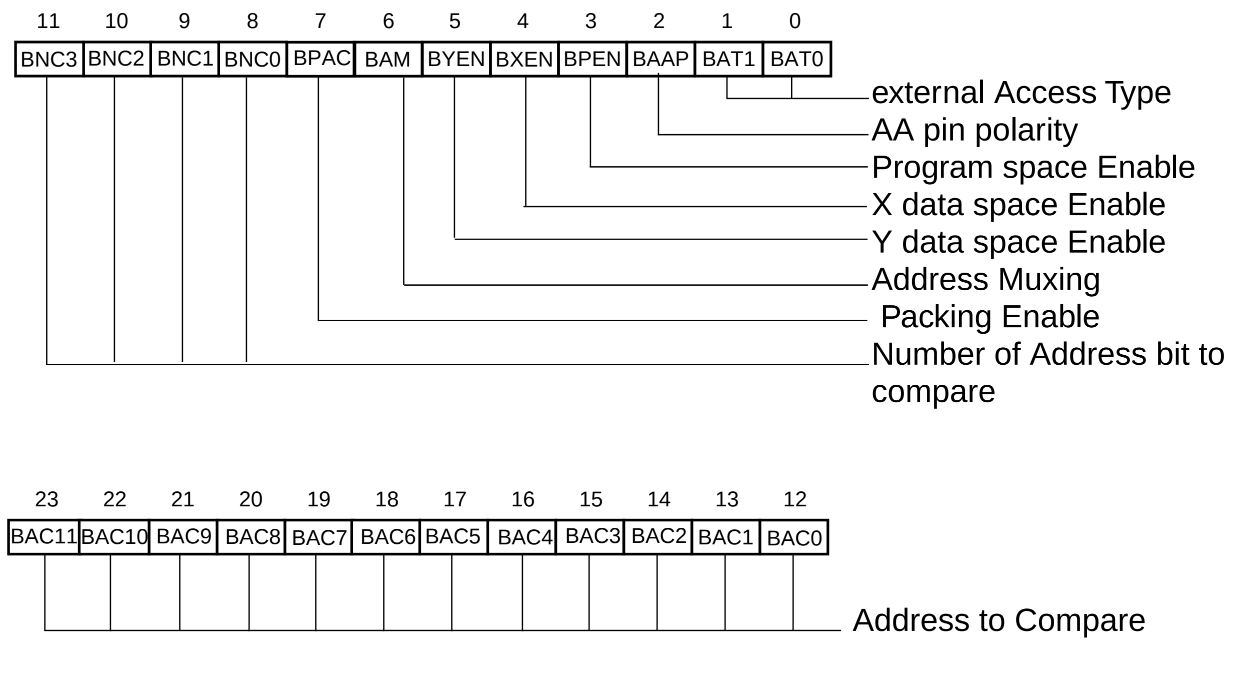
2.5 EXPANSION PORT CONTROL

The expansion port control consist of 4 Address Attribute Registers, DRAM control register and the Bus Control Register.

2.5.1 AA control Registers (one for each AA pin)

The **four** control registers (AAR3, AAR2, AAR1, AAR0) are 24 bit read write registers used to control the activity of the AA3-0/$\overline{\text{RAS}}$3-0 pins. An AA/$\overline{\text{RAS}}$ pin is asserted if the address in his appropriate AAR register (BAC bits) matches the external address (the exact number of address bits that are compared is determined by BNC bits) and if the external access is aimed to a space (X Y or P) that is enabled in the appropriate AAR register. All AAR registers are disabled (all the AAR bits are cleared) during hardware reset. The AAR bits are shown in the following figure and described in the following paragraphs.

Figure 2-10. Address Attribute Registers (AAR3-0)



NOTE 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, (e.g. if the external address matches the address and the space that is specified in both AAR1 and AAR2, the

external access type will be selected according to the AAR2 register) The priority mechanism allows continues partition of the external address space.

NOTE 3 When the AA/$\overline{\text{RAS}}$ pin functions as AA pin, it is negated at the start of the next clock cycle only if there is no external access that use the same AA pin (i.e. the AA pin will be kept asserted in a sequence of two consecutive external accesses that access the same memory bank). This method enables the use of low power standby mode in the external memories (these memories should be accessed first by a dummy access)

NOTE 4 The programmer should guarantee an AAR register is not changed while accessing the memory selected by this AAR, otherwise improper operation may result.

NOTE 5 A write operation to any AAR register will cause the DRAM controller to invalidate the page logic and will force the next DRAM access to be an out of page access.

2.5.1.1 BAT(1:0) - External Access Type and pin definition- bits 1-0

The read/write control bits BAT(1:0) define the external access type (DRAM, SRAM or SSRAM) to the area defined by BAC(11:0),BYEN, BXEN and BPEN bits. The encoding of BAT1 - BAT0 is described in the following table.

BAT1	BAT0	external access type
0	0	Synchronous SRAM access
0	1	Static RAM access
1	0	DRAM access
1	1	Reserved

When the external access type is defined as DRAM access (BAT(1:0) = 10) the AA/$\overline{\text{RAS}}$ pin will act as a $\overline{\text{RAS}}$ pin, otherwise it will act as a AA pin. External accesses to the default area will be always executed as if BAT(1:0) of the default area equals 01, i.e. static RAM access.
BAT(1:0) bits are cleared during hardware reset.

2.5.1.2 BAAP - AA pin Polarity - bit 2

The read/write control bit BAAP defines whether the AA/$\overline{\text{RAS}}$ pin is an active low or an active high pin. When BAAP is cleared the AA/$\overline{\text{RAS}}$ pin is an active low pin (useful for enabling memory modules, or for DRAM row address strobe), if BAAP is set the appropriate AA/$\overline{\text{RAS}}$ pin is a active high pin (useful as additional address bit).
BAAP bit is cleared during hardware reset.

2.5.1.3 BPEN - Program space Enable - bit 3

The read/write control bit BPEN defines whether the AA/$\overline{\text{RAS}}$ pin and logic should be activated during external program space accesses. BPEN when set enables the comparison of the external address to the BAC bits during external program space accesses. If BPEN is cleared no comparison of address is performed, during external Program space accesses.
BPEN bit is cleared during hardware reset.

2.5.1.4 BXEN - X data space Enable - bit 4

The read/write control bit BXEN defines whether the AA pin and logic should be activated during external X data space accesses. BXEN when set enables the comparison of the external address to the BAC bits during external X data space accesses. If BXEN is cleared no comparison of address is performed, during external X data space accesses.
BXEN bit is cleared during hardware reset.

2.5.1.5 BYEN - Y data space Enable - bit 5

The read/write control bit BYEN defines whether the AA pin and logic should be activated during external Y data space accesses. BYEN when set enables the comparison of the external address to the BAC bits during external Y data space accesses. If BYEN is cleared no comparison of address is performed during external Y data space accesses.
BYEN bit is cleared during hardware reset.

2.5.1.6 BAM - Address Muxing - bit 6

The read/write control bit BAM defines whether the 8 least significant bits of the address will appear on A7-A0 pins (LS portion of the external address bus) or on A23-A16 pins (MS portion of the external address bus). When BAM is set, the 8 LS bits will appear on A23-A16 pins. When BAM is cleared, the address will appeared normally and will occupy the entire external address bus (A23-A0). This feature enables to connect an external

peripheral to the most significant bits of the address thus decreasing the load on the LSP of the external address and enables more efficient interface to external memories. BAM is ignored during DRAM access (BAT1=1).
BAM bit is cleared during hardware reset.

2.5.1.7 BPAC- Packing Enable - bit 7

The read/write control bit BPAC enables (when set) the internal packing/unpacking logic. In this mode each DMA external access will initiate three external accesses to an eight bits wide external memory (the address of these accesses will be the original DMA address - the DAB, then DAB+1 and then DAB+2). The packing to a 24 bits word (or the unpacking from 24 bits word to 3 eight bits words) is done automatically by the expansion port control hardware. The external memory should reside in the eight least significant bits of the external data bus, and the packing (or unpacking for external write accesses) is done least significant byte first (i.e.the first data byte is the LS byte the second is the middle byte and the last is the MS byte). When this bit is cleared the expansion port control logic assumes a 24 bit wide external memory.
BPAC bit is cleared during hardware reset.

NOTE 1 BPAC is considered only for DMA accesses, and ignored during core accesses.

NOTE 2 In order to ensure sequential external accesses the DMA address should advance in steps of three. See example of the DMA channel programming in Chapter 8.1 - DMA CONTROLLER PROGRAMMING MODEL

NOTE 3 DMA address +1, and DMA address +2 should not cross the AAR bank borders otherwise improper operation may result.

NOTE 4 Arbitration is not allowed during the packing access, i.e. the three accesses are treated as one access from the arbitration point of view, and the bus mastership will not be released during these accesses.

NOTE 5 Packing Mode is not allowed to Synchronous SRAM with zero wait states, otherwise improper operation may result.

2.5.1.8 BNC(3:0) - Number of address bits to Compare - bits 11-8

The read/write control bits BNC(3:0) defines the number of bits (from the BAC bits) that are compared to the external address. If no bits should be compared (BNC(3:0) = 0000) the AA pin is activated only according to the space enable bits (BPEN, BXEN, BYEN). The combinations BNC(3:0) = 1111, 1110, 1101 are reserved.
BCN(3:0) bit are cleared during hardware reset.

2.5.1.9 BAC(11:0) - Address to compare - bits 23-12

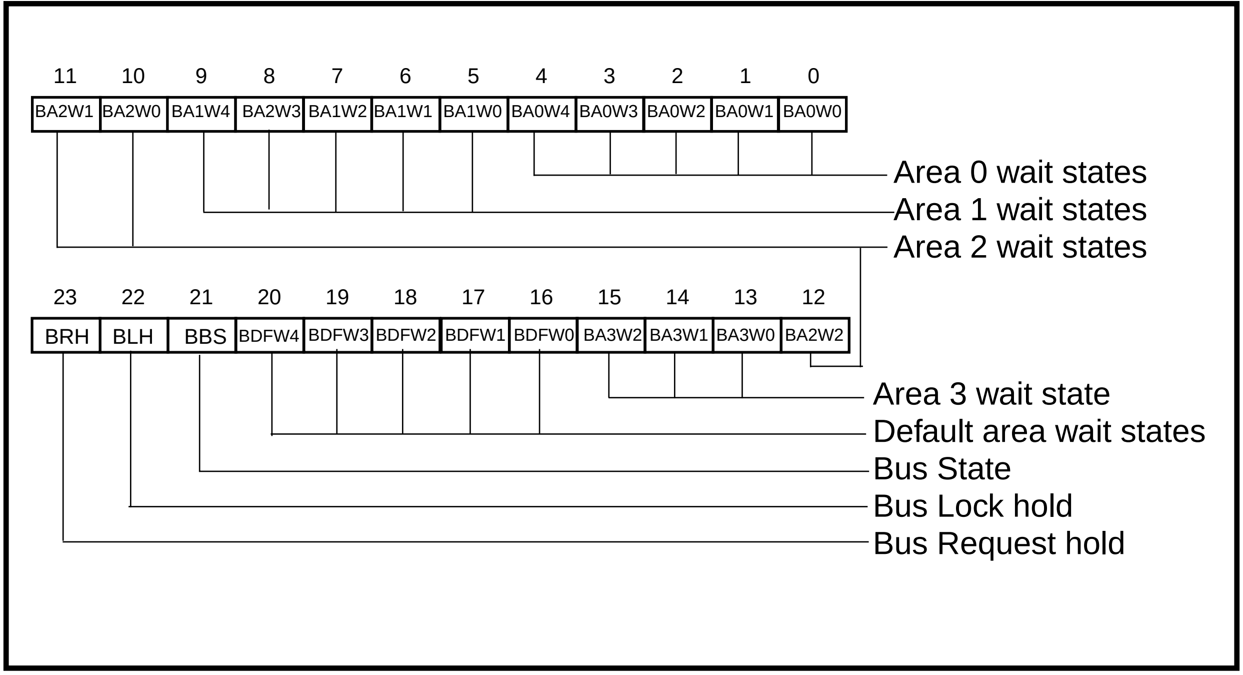
The read/write control bits BAC(11:0) defines the address that should be compared to the

external address in order to decide if to assert the AA/$\overline{\text{RAS}}$ pin. The number of bits that should be compared is defined by the BNC(3:0) bits, BAC bits are always compared to the most significant portion of the external address bus (e.g. if BNC(3:0) = 0011 then BAC(11:9) are compared to the 3 most significant bits of the external address).
BAC(11:0) bits are cleared during hardware reset.

2.5.2 Bus Control Register

The Bus Control Register (BCR) is a 24 bit read write register used to control the external bus activity and Bus Interface Unit operation. The BCR bits are shown in Figure 2-11 on page 2-24 and described in the following paragraphs.

Figure 2-11. Bus Control Register (BCR)



2.5.2.1 BA0W(4:0) - Area 0 Wait control - bits 4-0

The read/write control bits BA0W(4:0) define the number of wait states (0 - 31) inserted in each external SRAM or synchronous SRAM accesses to area 0 (DRAM accesses are not affected by these bits). Area 0 is the area defined by AAR0 register.
For SRAM accesses only, the value of these bits should not be programmed as zero since SRAM memory access requires at least one wait state.
For SRAM accesses only, when selecting 4 to 7 wait states, one additional wait state will be inserted at the end of the access. When selecting 8 or more wait states, two additional wait states will be 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.
BA0W(4:0) bits are set during hardware reset (i.e. 31 wait states).

2.5.2.2 BA1W(4:0) - Area 1 Wait control - bits 9-5

The read/write control bits BA1W(4:0) define the number of wait states (0 - 31) inserted in each external SRAM or synchronous SRAM accesses to area 1 (DRAM accesses are not affected by these bits). Area 1 is the area defined by AAR1 register.
For SRAM accesses only, the value of these bits should not be programmed as zero since SRAM memory access requires at least one wait state.
For SRAM accesses only, when selecting 4 to 7 wait states, one additional wait state will be inserted at the end of the access. When selecting 8 or more wait states, two additional wait states will be 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.
BA1W(4:0) bits are set during hardware reset (i.e. 31 wait states).

2.5.2.3 BA2W(2:0) - Area 2 Wait control - bits 12-10

The read/write control bits BA2W(2:0) define the number of wait states (0 - 7) inserted in each external SRAM or synchronous SRAM accesses to area 2 (DRAM accesses are not affected by these bits). Area 2 is the area defined by AAR2 register.
For SRAM accesses only, the value of these bits should not be programmed as zero since SRAM memory access requires at least one wait state.
For SRAM accesses only, when selecting 4 to 7 wait states, one additional wait state will be 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.
BA2W(2:0) bits are set during hardware reset (i.e. 7 wait states).

2.5.2.4 BA3W(2:0) - Area 3 Wait control - bits 15-13

The read/write control bits BA3W(2:0) define the number of wait states (0 - 7) inserted in each external SRAM or synchronous SRAM accesses to area 3 (DRAM accesses are not affected by these bits). Area 3 is the area defined by AAR3 register.
For SRAM accesses only, the value of these bits should not be programmed as zero since SRAM memory access requires at least one wait state.
For SRAM accesses only, when selecting 4 to 7 wait states, one additional wait state will be 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.
BA3W(2:0) bits are set during hardware reset (i.e. 7 wait states).

2.5.2.5 BDFW(4:0)- Default Area Wait control - bits 20-16

The read/write control bits BDFW(3:0) define the number of wait states (0 - 31) inserted in each external accesses to an area which is not defined by any of the BAAR registers. The access type to this area is SRAM only. The value of these bits should not be programmed as zero since SRAM memory access requires at least one wait state.
When selecting 4 to 7 wait states, one additional wait state will be inserted at the end of the access. When selecting 8 or more wait states, two additional wait states will be 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.
BDFW(4:0) bits are set during hardware reset (e.g. 31 wait states).

2.5.2.6 BBS - Bus State - bit 21

The read only Bus State status bit - BBS is set when the DSP is the bus master and cleared otherwise. BBS bit is cleared during hardware reset.

2.5.2.7 BLH - Bus Lock Hold - bit 22

The read/write control bit Bus Lock Hold - BLH is used to assert the $\overline{\mathrm{BL}}$ pin even if no read-modify-write access is occurring. When BLH is set, $\overline{\mathrm{BL}}$ pin is always asserted. If BLH bit is cleared $\overline{\mathrm{BL}}$ pin is asserted only if a read-modify-write external access is attempted.
BLH bit is cleared during hardware reset.

2.5.2.8 BRH - Bus Request Hold - bit 23

The read/write control bit Bus Request Hold - BRH is used to assert the $\overline{\mathrm{BR}}$ pin even if no external access is needed. When BRH is set $\overline{\mathrm{BR}}$ pin is always asserted, if BRH bit is cleared $\overline{\mathrm{BR}}$ pin is asserted only if external access is attempted or pending.
BRH bit is cleared during hardware reset.

2.5.3 IDentification Register

The IDentification Register (IDR) is a 24 bit read only via programmed register used to identify the different DSP56300 core-based family members. This register specify the chip number and revision and the DSP56300 core revision. The exact number for each DSP56300 core member can be found in the specific part data sheet.

2.6 DRAM CONTROLLER

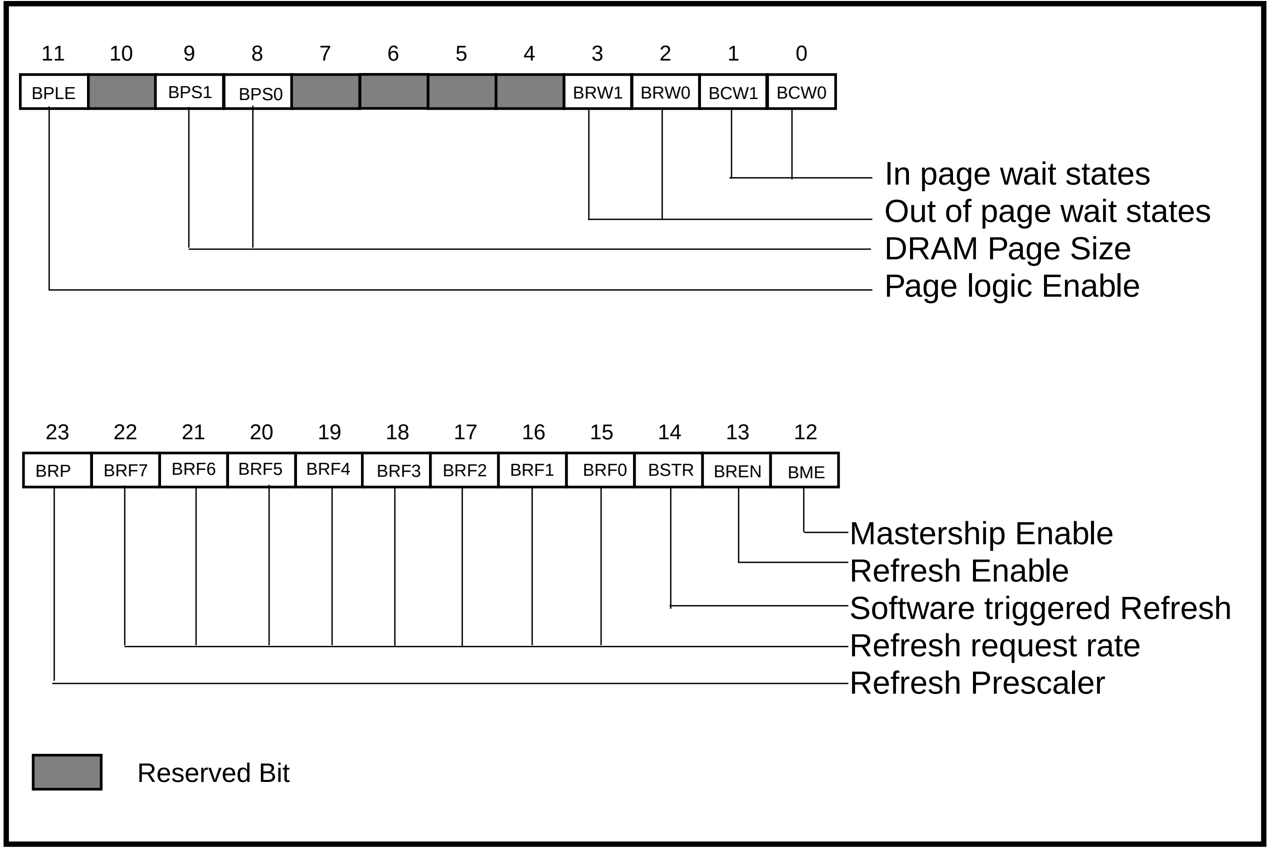
The DRAM controller is designed for efficient interface to dynamic RAM devices in both random read/write cycles and fast access mode (page mode). An on-chip DRAM controller controls the page hit circuit, the address multiplexing (row address and column address), the control signal generation ($\overline{\mathrm{CAS}}$ and $\overline{\mathrm{RAS}}$) and the refresh access generation ($\overline{\mathrm{CAS}}$ before $\overline{\mathrm{RAS}}$) for a large variety of DRAM module sizes and different access times. The on chip DRAM controller configuration is determined by the DRAM Control Register (DCR).

2.6.1 DRAM Control Register

The DRAM Control Register (DCR) is a 24 bit read write register used to control and configure the external DRAM accesses. The DCR bits are shown in Figure on page 2-27 and described in the following paragraphs.

NOTE The programmer must guarantee that all the DCR bits except BSTR are not changed while accessing a DRAM. Otherwise improper operation may result.

Figure 2-12. DRAM Control Register (DCR)



2.6.1.1 BCW(1:0) - In page Wait states - bits 1-0

The read/write control bits - BCW(1:0) define the number of wait states that should be inserted in each DRAM in-page access.
The encoding of BCW1 and BCW0 is described in the following table. BCW(1:0) bits are cleared during hardware reset

BCW1	BCW0	DRAM External access
0	0	1 w.s for each in-page access
0	1	2 w.s for each in-page access
1	0	3 w.s for each in-page access
1	1	4 w.s for each in-page access

2.6.1.2 BRW(1:0) - Out of page Wait states- bits 3-2

The read/write control bits- BRW(1:0) define the number of wait states that should be inserted in each DRAM out of page access.

The encoding of BRW1 and BRW0 is described in the next table. BRW(1:0) bits are cleared during hardware reset.

BRW1	BRW0	DRAM External access
0	0	4 w.s for each out-of-page access
0	1	8 w.s for each out-of--page access
1	0	11 w.s for each out-of-page access
1	1	15 w.s for each out-of-page access

2.6.1.3 BPS(1:0) - DRAM Page Size - bits 9-8

The read/write control bits- BPS(1:0) define the size of the external DRAM page and thereby the number of the column address bits. The internal page mechanism works according to these bits only if the page logic is enabled (by BPLE bit). The four combinations of BPS(1:0) enable the use of many DRAM sizes (1Mbit, 4Mbit, 16Mbit and 64Mbit). The encoding of BPS1 and BPS0 is described in the following table. BPS(1:0) bits are cleared during hardware reset.

BPS1	BPS0	Column address width	DRAM Page size
0	0	9 bits	512
0	1	10 bits	1K
1	0	11 bits	2K
1	1	12 bits	4K

NOTE When driving the row address all the 24 address bits of the external address bus are driven. e.g: if BPS(1:0) = 01, when driving the row address the 14 MS bits of the internal address (XAB, YAB,PAB or DAB) will be driven on A(13:0) pins, and A(23:14) pins will be driven with the 10 MSB of the internal address. This method enables the use of different DRAMs with the same page size.

2.6.1.4 BPLE - Page logic Enable - bit 11

The read/write Page logic Enable - BPLE is used to enable/disable the in-page identifying logic. When this bit is set it enables the page logic (the page size is defined by BPS(1:0) bits), each in-page identification will cause the DRAM controller to drive only the column address (and the associate $\overline{\text{CAS}}$ signal).When this bit is cleared the page logic is disabled, and the DRAM controller will always access the external DRAM in out-of-page accesses (e.g. row address with $\overline{\text{RAS}}$ assertion and then column address with $\overline{\text{CAS}}$ assertion). This mode is useful for low power dissipation.There is only one in-page identifying logic and therefore when switching from one DRAM external bank to another DRAM bank (the DRAM external banks are defined by the access type bits in the AAR registers, different external bank are accessed through different AA/$\overline{\text{RAS}}$ pin) a page fault occurs. BPLE bit is cleared during hardware reset.

2.6.1.5 BME - Mastership Enable - bit 12

The read/write control bit Mastership Enable - BME is used to enable/disable interface to a local DRAM for the DSP. When BME is cleared, the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ pins are three-stated when mastership is lost and therefore the user 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 and a DRAM refresh will require a bus mastership. If BME bit is set the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ pins are always driven from the DSP and therefore DRAM refresh can be performed even if the DSP is not the bus master. BME bit is cleared during hardware reset.

2.6.1.6 BREN - Refresh Enable - bit 13

The read/write control bit Refresh Enable - BRE enables/disables the internal refresh counter. When this bit 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 will occur to all DRAM banks together (all pins that were defined as $\overline{\text{RAS}}$ will be 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. BRE bit is cleared during hardware reset.

NOTE 1 In a system where more than one DSP share the same DRAM, the DRAM controller of more than one DSP may be active, but it is recommended that only one DSP will have its BREN bit set, and bus mastership will be requested for a refresh access.

NOTE 2 If BREN is set and a WAIT instruction is executed, periodic refresh will still be generated each time the refresh counter reaches zero.

NOTE 3 If BREN is set and a STOP instruction is executed, periodic refresh will not be generated and the refresh counter will be disabled.

2.6.1.7 BSTR - Software Triggered Refresh - bit 14

The read/write control/status bit Software triggered refresh - BSTR is used to generate a software triggered refresh request. When this bit is set a refresh request is generated and a refresh access will be executed to all DRAM banks (the exact timing of the refresh access depends on the pending external accesses and on BME bit). After the refresh access ($\overline{\text{CAS}}$ before $\overline{\text{RAS}}$) was executed BSTR bit is cleared by the DRAM controller hardware.The refresh cycle length depends on the BRW(1:0) bits (a refresh access is as long as the out-of-page access). BSTR bit is cleared during hardware reset.

2.6.1.8 BRF(7:0) Refresh rate - bits 22-15

The read/write control bits BRF(7:0) control the refresh request rate.The BRF(7:0) specify a divide rate of 1 (BRF(7:0)= $00) to 256 (BRF(7:0) = $FF). A refresh request will be generated each time the refresh counter reaches zero if the refresh counter is enabled (by setting BRE bit). BRF(7:0) bits are cleared during hardware reset.

2.6.1.9 BRP - Refresh Prescaler - bit 23

The read/write control bit - BRP 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 BRP is set) or by 1 (if BRP is cleared). BPR is cleared during hardware reset

NOTE 1 Refresh requests are not accumulated and therefore in a fast refresh request rate not all the refresh requests will be served (e.g. the combination BRF(7:0) = $00 and BRP = 0 will generate refresh request every clock cycle, but a refresh access takes at least 5 clock cycles).

NOTE 2 When programming the periodic refresh rate the user must consider the $\overline{\text{RAS}}$ time-out period. There is no hardware support for the $\overline{\text{RAS}}$ time-out restriction.