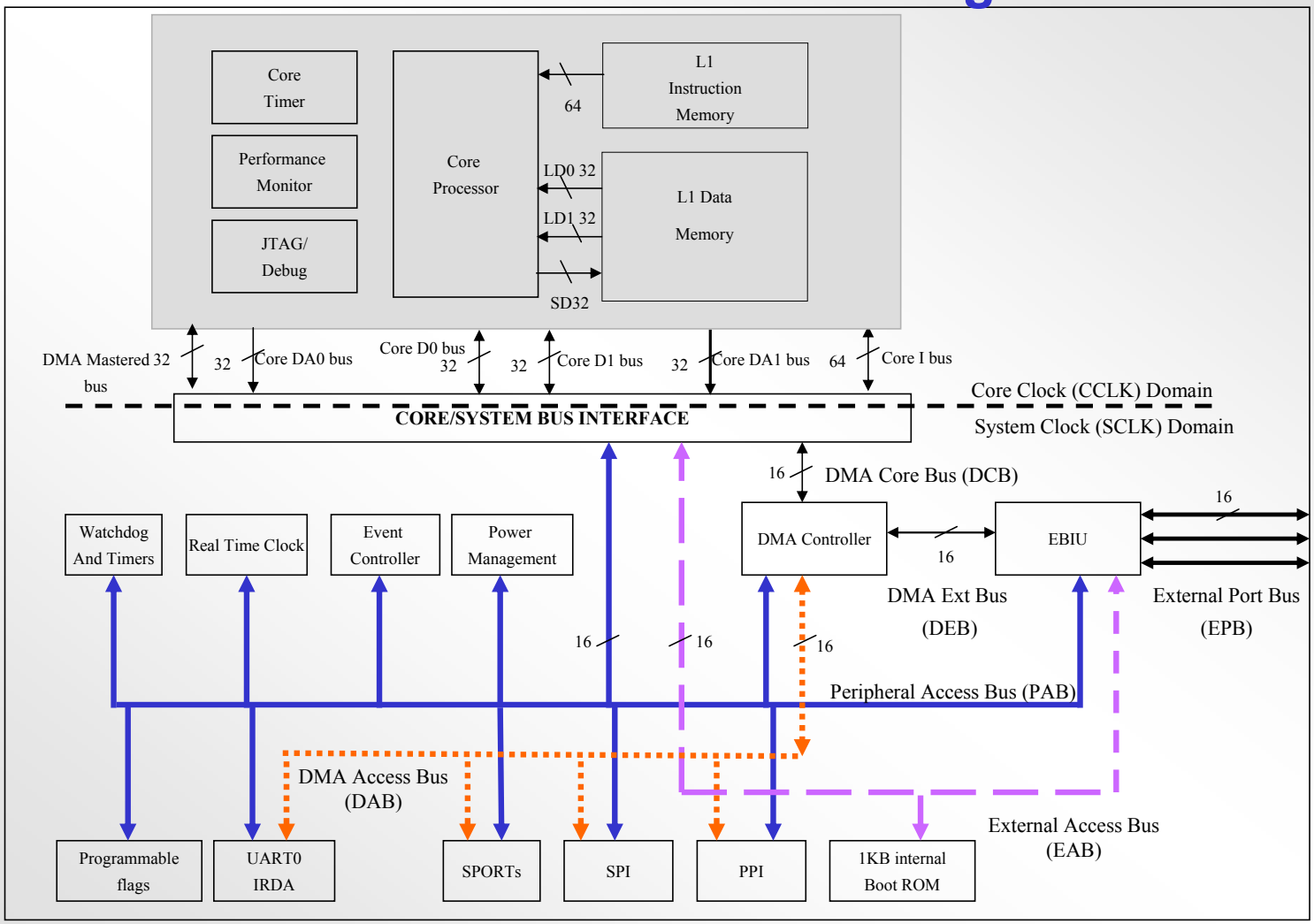
Section 12

External Bus Interface Unit (EBIU)

ADSP-BF533 Block Diagram



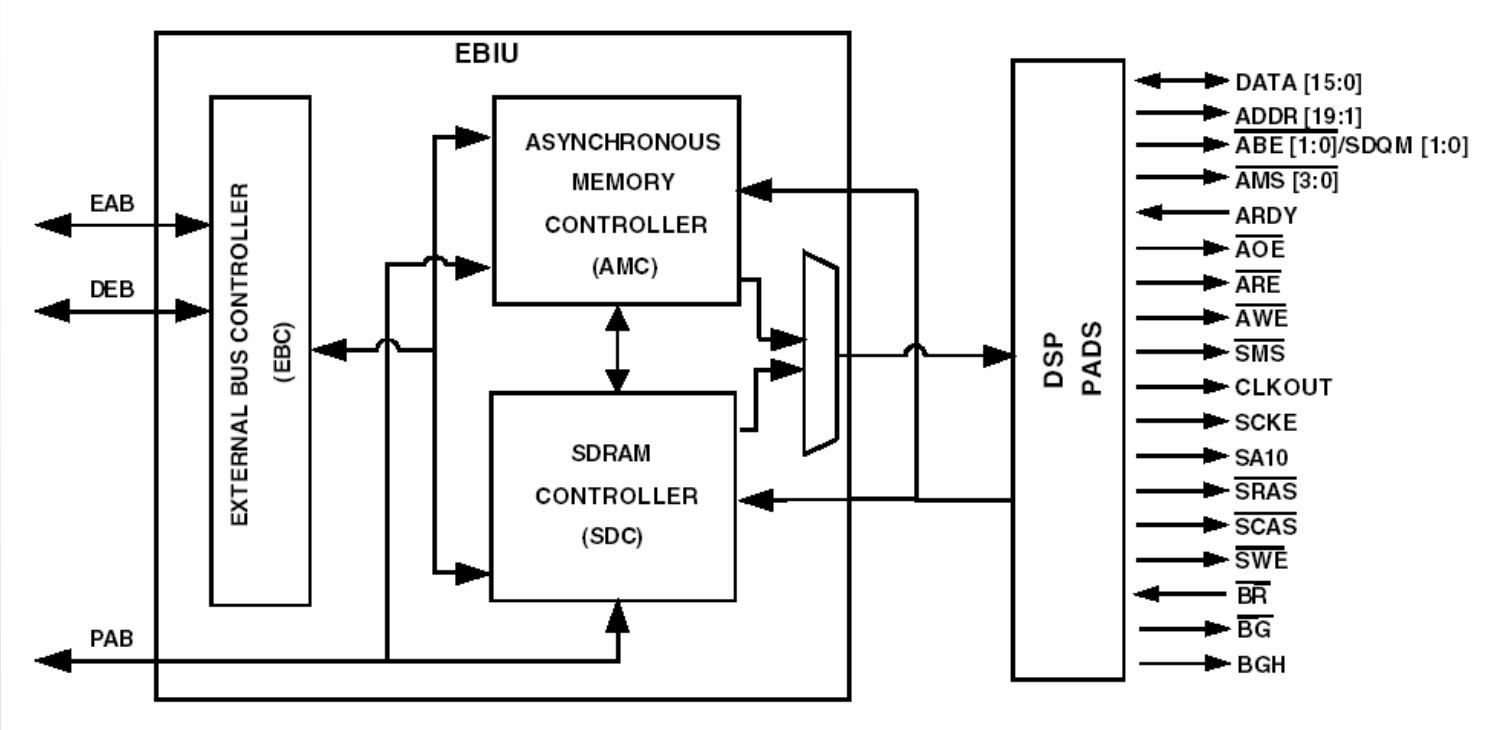
BF533 EBIU Overview

- Provides a glueless interface to external synchronous and asynchronous memories.
  - On-chip SDRAM controller (SDC) supports PC-100 and PC-133 SDRAM
- Asynchronous Memory Controller (AMC) and Synchronous DRAM Controller (SDC) arbitrate for internal bus resources and shared external pin resources.
- EBIU runs at the system clock rate (SCLK).
  - All synchronous memories interfaced to the BF533 operate at this frequency.
- SDC and AMC both support 16-bit accesses
  - True 8-bit read accesses are supported on SDC only

Bus Interfaces to EBIU

- Three internal 16-bit busses are connected to the EBIU:
  - External Access Bus (EAB): Mastered by the core memory management unit to access external memory
  - Peripheral Access Bus (PAB): Used to provide access to EBIU MMRs
  - DMA External Bus (DEB): Mastered by the DMA controller to access external memory
- External Port Bus (EPB) connects the output of EBIU to external devices
- Transactions from the core have priority over DMA accesses unless the DMA detects an urgent condition (e.g. peripheral FIFO filling up)
- Packing modes are available for DMA transfers
  - 16-bit transfers make the most efficient use of the DMA buses

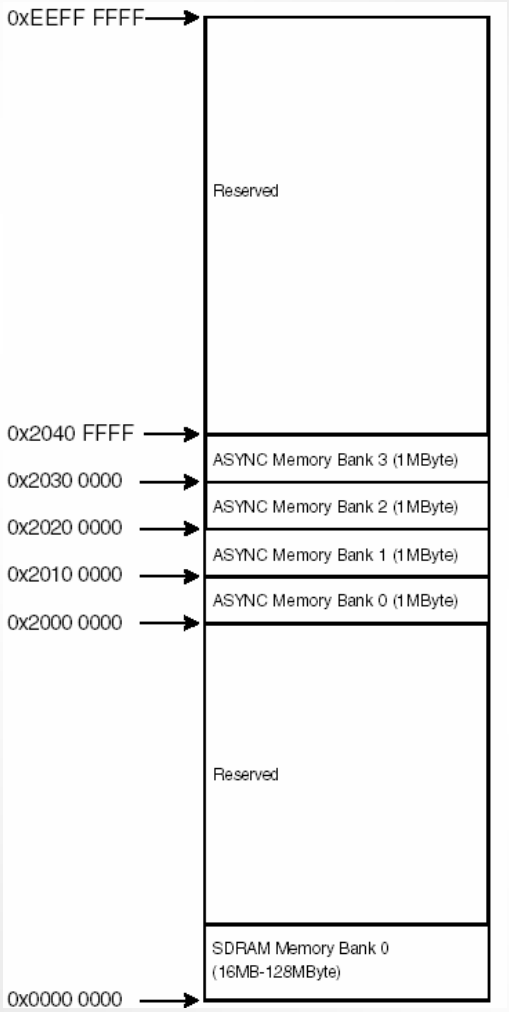
BF533 EBIU Block Diagram



Shared Memory Interface Pins

- **The AMC and SDC share the following external pins:**
  - DATA[15:0], data bus
  - ADDR[19:1], address bus
  - /ABE [1:0]/SDQM[1:0], AMC byte enables/SDC data masks
  - /BR, /BG, /BGH

ADSP-BF533 External Memory Map



Asynchronous Memory Controller

Asynchronous Memory Controller Features

- Supports up to 4 MB of addressable memory comprised of four 1MB banks.
- Each 1MB asynchronous memory bank has its own memory select signal
  - /AMS0, /AMS1, /AMS2, /AMS3
- EBIU supports 16-bit accesses
  - True byte accesses not supported because EBIU always fetches 16-bits
    - Core will return upper or lower byte as needed using instruction of the form R0 = B[P0];
    - Booting option does exist from 8-bit flash
      - Performed by making 16-bit access and using least significant 8 bits
- Memory bank(s) must be enabled in the EBIU_AMGCTL register if a device is present
- Glueless interface to SRAM, flash
  - Can also be used to map peripherals (A/D's, Video decoders, etc)

Interface Signals Unique to Asynchronous Memory

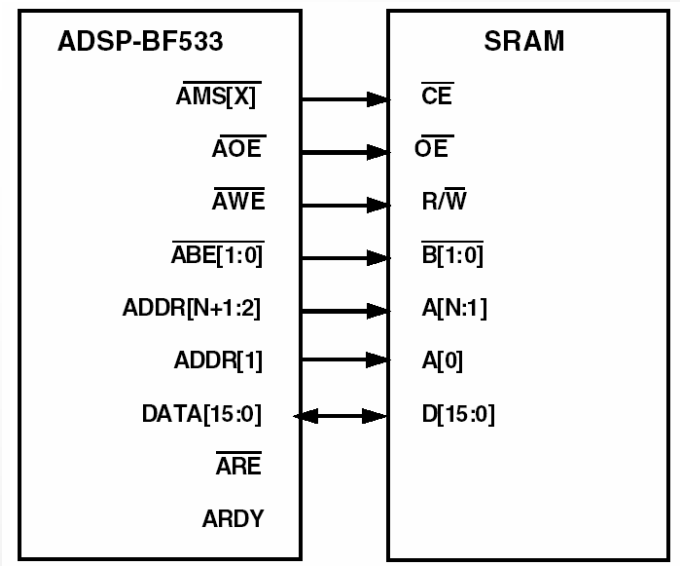
EBIU Pin Name	Pin Type	Description
ADDR[19:1]	O	External Address Bus
DATA[15:0]	I/O	External Data Bus
$\overline{\text{AMS}}$[3:0]	O	Asynchronous Memory Selects
$\overline{\text{AWE}}$	O	Asynchronous Memory Write Enable
$\overline{\text{ARE}}$	O	Asynchronous Memory Read Enable
$\overline{\text{AOE}}$	O	Asynchronous Memory Read Enable
ARDY	I	Asynchronous Memory Ready Response
$\overline{\text{ABE}}$[1:0]	O	Byte Enables

Pin Types: I = Input, O = Output, I/O = Input/Output

ABE Signals

- /ABE0 and /ABE1 are byte enable signals that allow byte writes to 16-bit memory
  - They do not function as an address 0 pin
  - If an 8-bit flash is used, the max bank size is 512KB
- /ABE0 and /ABE1 are both low during read operations
- /ABE0 is low and /ABE1 is high during a write to the lower byte of 16-bit ASYNC memory
- /ABE0 is high and /ABE1 is low during a write to the upper byte of 16-bit ASYNC memory

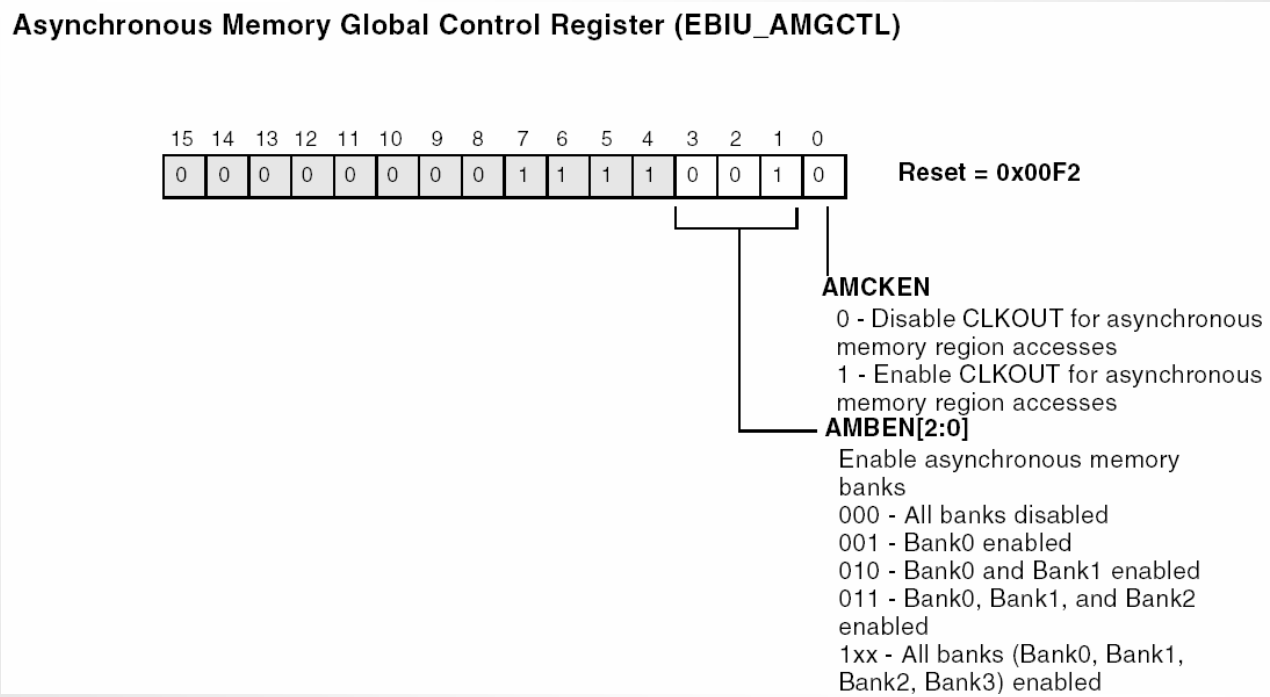
16-bit SRAM Interface Example



AMC Control Registers

- Asynchronous Memory Global Control Register (EBIU_AMGCTL)
  - Enable/disable CLKOUT signal (SCLK)
  - All banks enabled/disabled, bank 0 enabled only, bank 0 and 1 enabled only, banks 0 & 1, 2 enabled only.
- Asynchronous Memory Bank Control Registers (EBIU_AMBCTL0, EBIU_AMBCTL1)
  - Define wait states, ARDY enable/disable, and setup and hold times for each asynchronous memory bank.

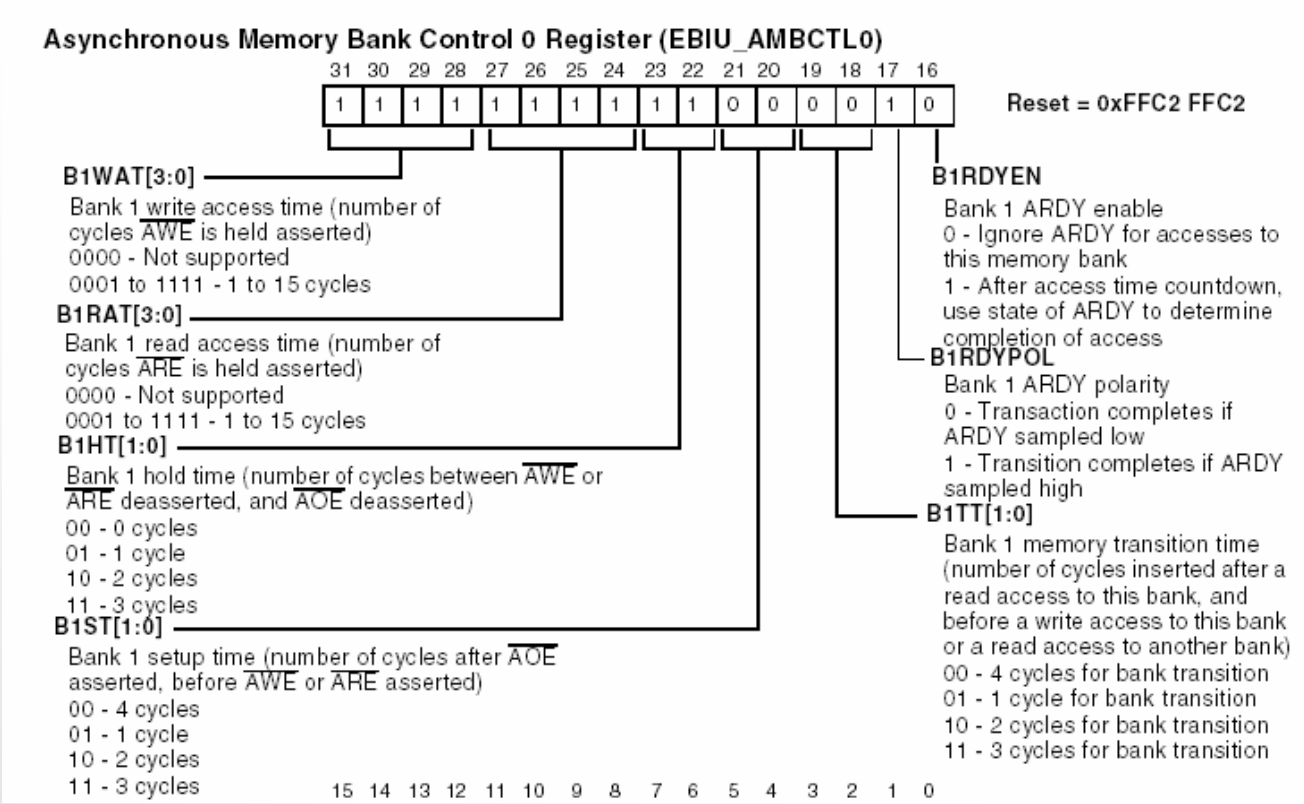
Asynchronous Memory Global Control Register (EBIU_AMGCTL)



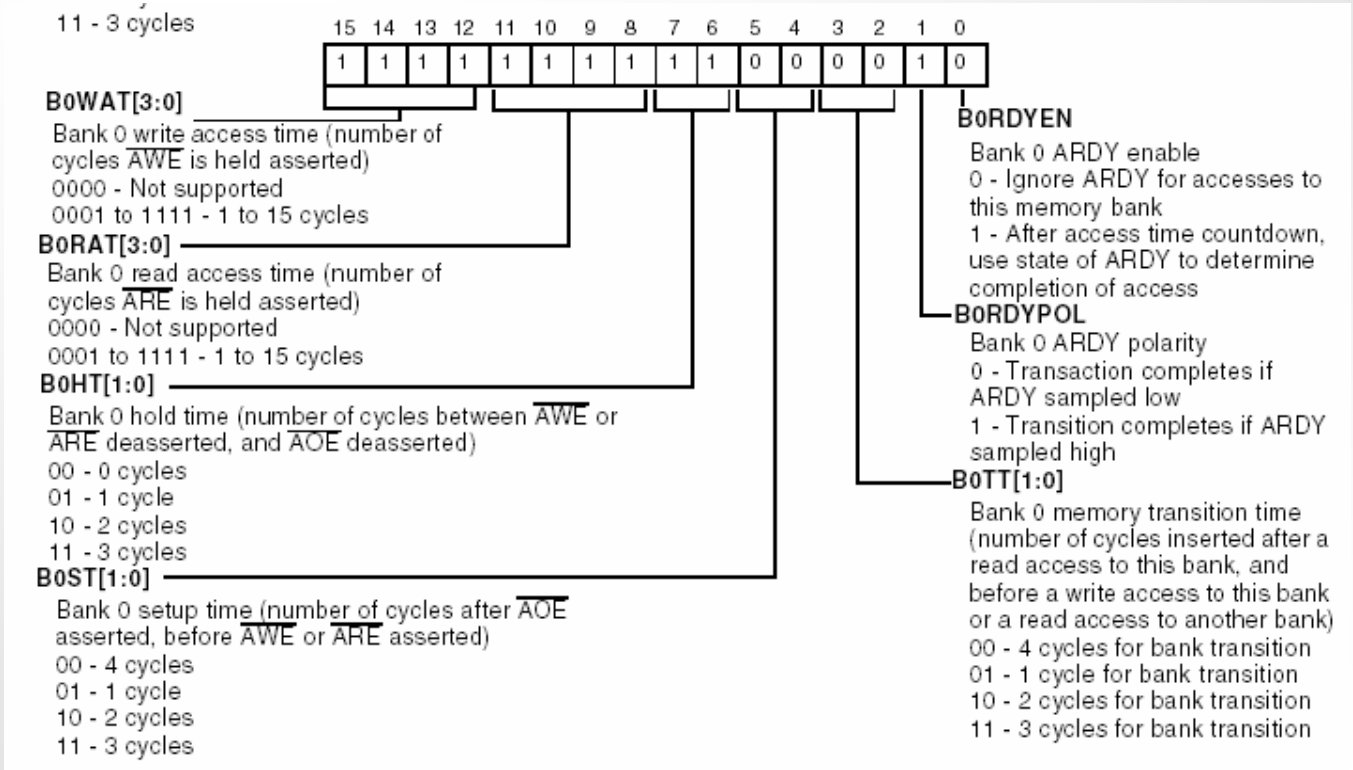
Asynchronous Memory Bank Control Register 1 of 3

- The Asynchronous Memory Controller has two memory bank control registers that control the following parameters:
  - Setup:
    - The time between the beginning of a memory cycle and the assertion of the read or write enable.
  - Read Access:
    - The time between the read enable assertion and negation.
  - Write Access:
    - The time between write enable assertion and negation.
  - Hold:
    - The time between read or write enable negation and the end of the memory cycle.
- Each of these parameters can be programmed in terms of duration of EBIU clock cycles (SCLK).
- Additional wait states can be added via the ARDY pin.
  - ARDY sampled on the clock cycle before the end of the programmed strobe period.

Asynchronous Memory Bank Control Register 2 of 3

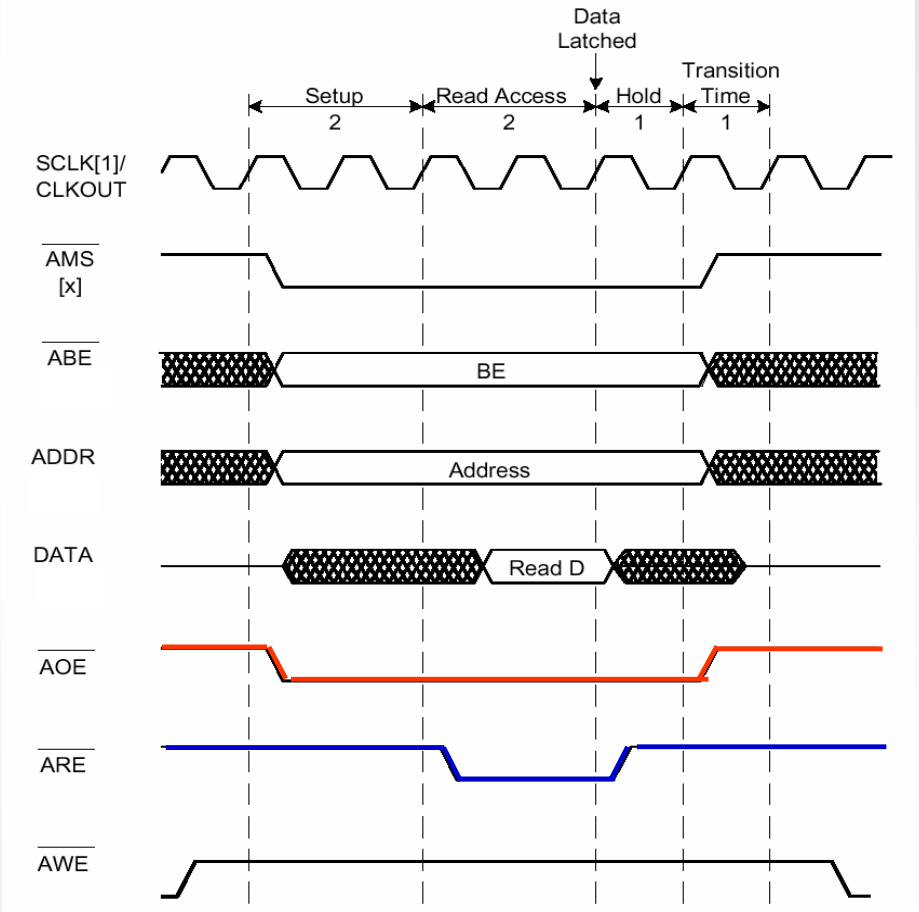


Asynchronous Memory Bank Control Register 3 of 3



$\overline{\text{AOE}}$ or $\overline{\text{ARE}}$?

- Both the $\overline{\text{AOE}}$ and $\overline{\text{ARE}}$ signals can be used to interface with the $\overline{\text{OE}}$ pin of an asynchronous device.
- Two pins provided to increase flexibility



Asynchronous Memory Read Timing Example

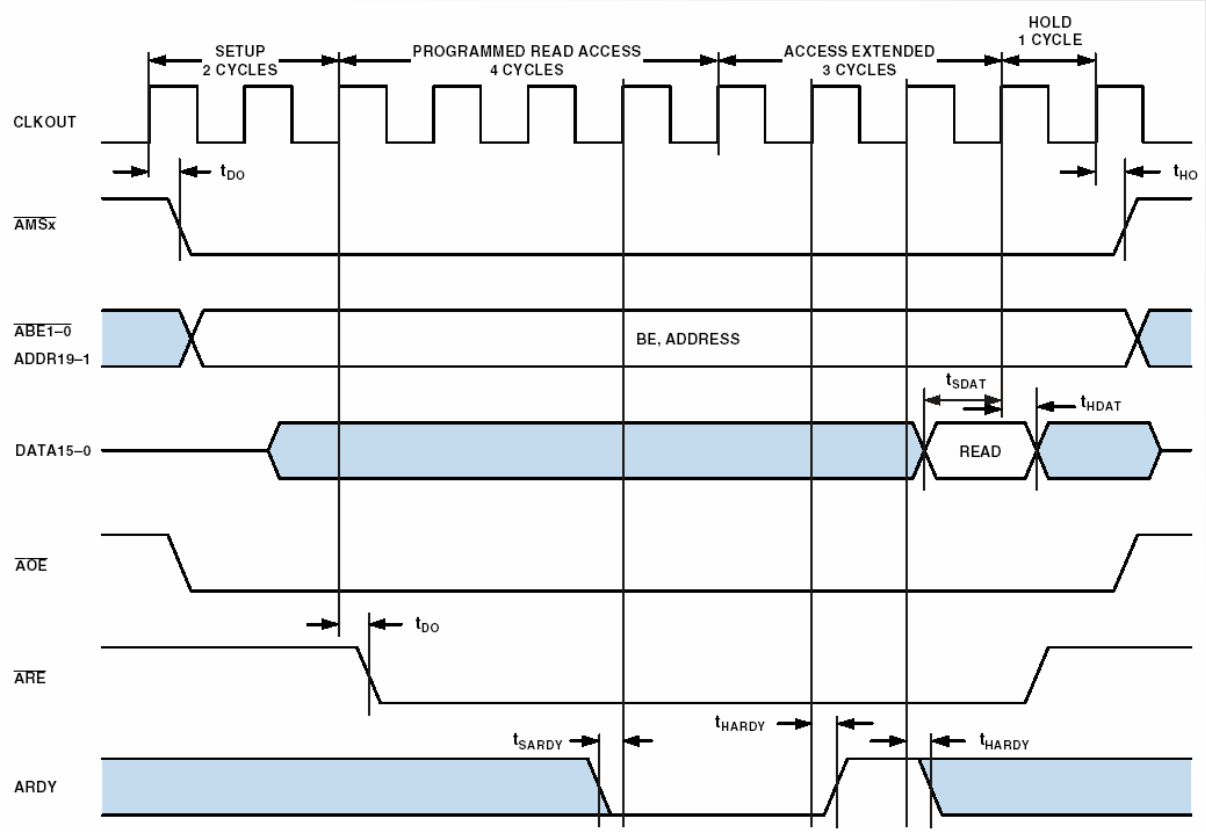


Figure 12. Asynchronous Memory Read Cycle Timing

Asynchronous Memory Write Timing Example

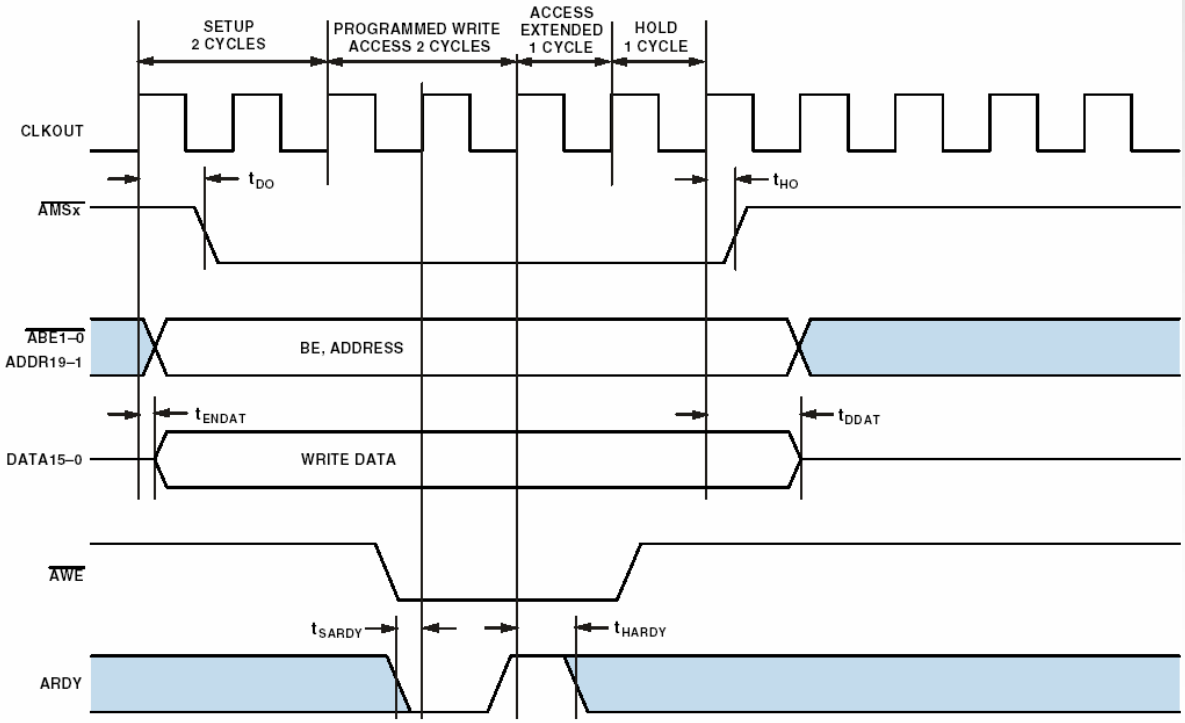


Figure 13. Asynchronous Memory Write Cycle Timing

Synchronous DRAM Controller

SDC Features

- SDC supports one bank of standard SDRAMs of 64Mbit, 128Mbit, 256Mbit, and 512Mbit with configurations x4, x8, and x16.
  - Access up to 128MB of memory.
- Provides a programmable refresh counter.
- Supports self refresh mode.
- Provides two SDRAM power-up options.
- Allows up to 4 SDRAM pages to be open at any one time
  - Open SDRAM internal pages reduce the number of page refreshes when multiple accesses are ongoing
  - Greatly improves performance

Interface Signals Unique to Synchronous Memory

EBIU Pin Name	Pin Type	Description
ADDR[19:18], ADDR[16:1]	O	External Address Bus. (Bank address is output on ADDR[19:18]; connect to SDRAM's BA[1:0] pins.)
DATA[15:0]	I/O	External Data Bus.
SRAS	O	SDRAM Row Address Strobe. (Connect to SDRAM's RAS pin.)
SCAS	O	SDRAM Column Address Strobe. (Connect to SDRAM's CAS pin.)
SWE	O	SDRAM Write Enable pin. (Connect to SDRAM's WE pin.)
SDQM[1:0]	O	SDRAM Data Mask pins. (Connect to SDRAM's DQM pins.)
SMS	O	Memory select pin of external memory bank configured for SDRAM.
SA10	O	SDRAM A10 pin. (Used for SDRAM refreshes; connect to SDRAM's A[10] pin.)
SCKE	O	SDRAM Clock Enable pin. (Connect to SDRAM's CKE pin.)
CLKOUT	O	SDRAM Clock pin. (Connect to SDRAM's CLK pin. Operates at SCLK frequency.)

16MB SDRAM System Example

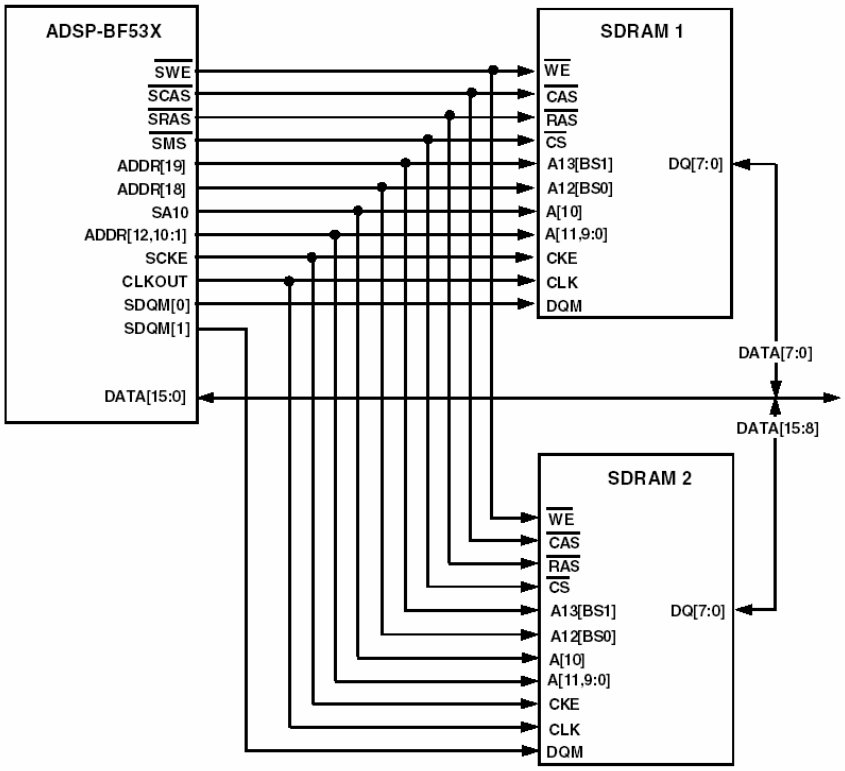


Figure 17-14. 16 MB SDRAM System Example

Internal Address Mapping (16 Bit Config)

Bank Width (bits)	Bank Size (Mbyte)	Col. Addr. Width (CAW)	Page Size (kbyte)	Bank Address	Row Address	Page	
						Column Address	Byte Address
16	128	11	4	IA26—25	IA24—12	A11—1	IA0
16	128	10	2	IA26—25	IA24—11	IA10—1	IA0
16	128	9	1	1A26—25	IA24—10	IA9—1	IA0
16	128	8	.5	IA26—25	IA24—9	IA8—1	IA0
16	64	11	4	IA25—24	IA23—12	IA11—1	IA0
16	64	10	2	IA25—24	IA23—11	IA10—1	IA0
16	64	9	1	IA25—24	IA23—10	IA9—1	IA0
16	64	8	.5	IA25—24	IA23—9	IA8—1	IA0
16	32	11	4	IA24—23	IA22—12	IA11—1	IA0
16	32	10	2	IA24—23	IA22—11	IA10—1	IA0
16	32	9	1	IA24—23	IA22—10	IA9—1	IA0
16	32	8	.5	IA24—23	IA22—9	IA8—1	IA0
16	16	11	4	IA23—22	IA21—12	IA11—1	IA0
16	16	10	2	IA23—22	IA21—11	IA10—1	IA0
16	16	9	1	IA23—22	IA21—10	IA9—1	IA0
16	16	8	.5	IA23—22	IA21—9	IA8—1	IA0

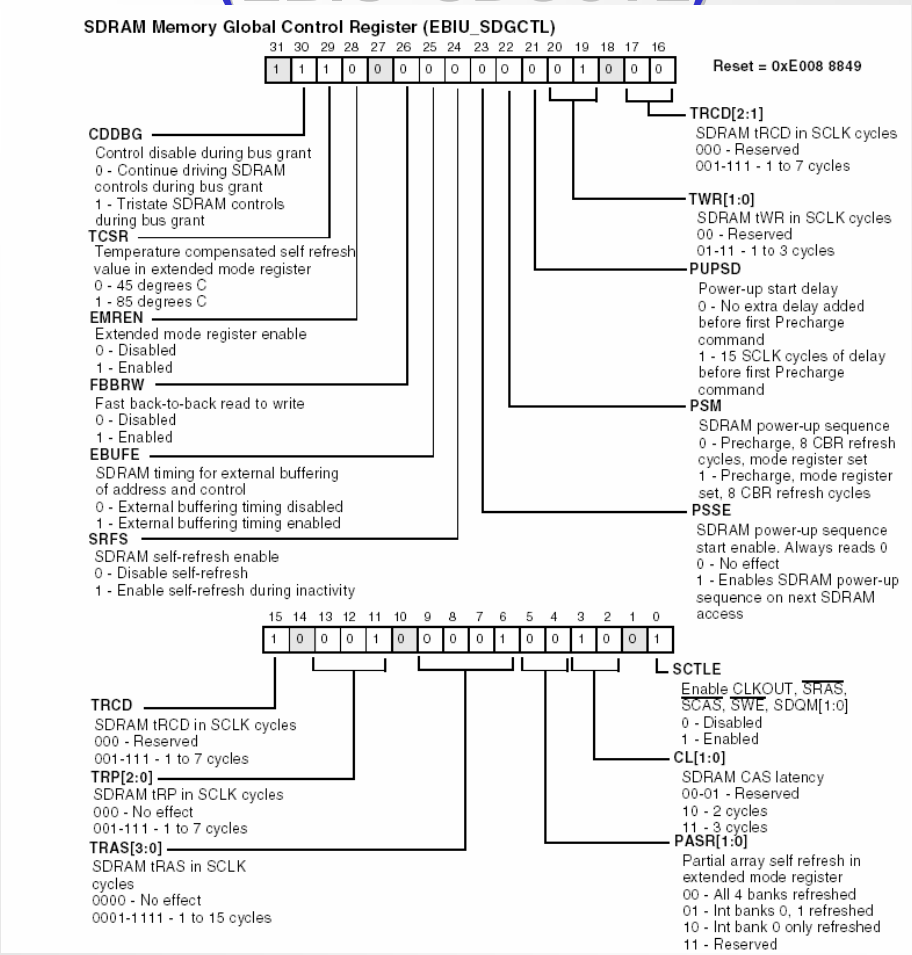
SDRAM Data Mask (SDQM[1:0]) Encoding

- During write transfers to SDRAM the SDQM[1:0] pins are used to mask writes to bytes that are not accessed.
- During read transfers to SDRAM banks, reads are always done for all bytes in the bank regardless of the transfer size.
  - For 16-bit SDRAM accesses SDQM[1:0] = 0,

Internal Address IA[0]	Internal Transfer Size		
	byte	halfword	word
0	SDQM[1]=1 SDQM[0]=0	SDQM[1]=0 SDQM[0]=0	SDQM[1]=0 SDQM[0]=0
1	SDQM[1]=0 SDQM[0]=1		

16-bit SDRAM

SDRAM Memory Global Control Register (EBIU SDGCTL)



SDRAM Memory Bank Control Register (EBIU_SDBCTL)

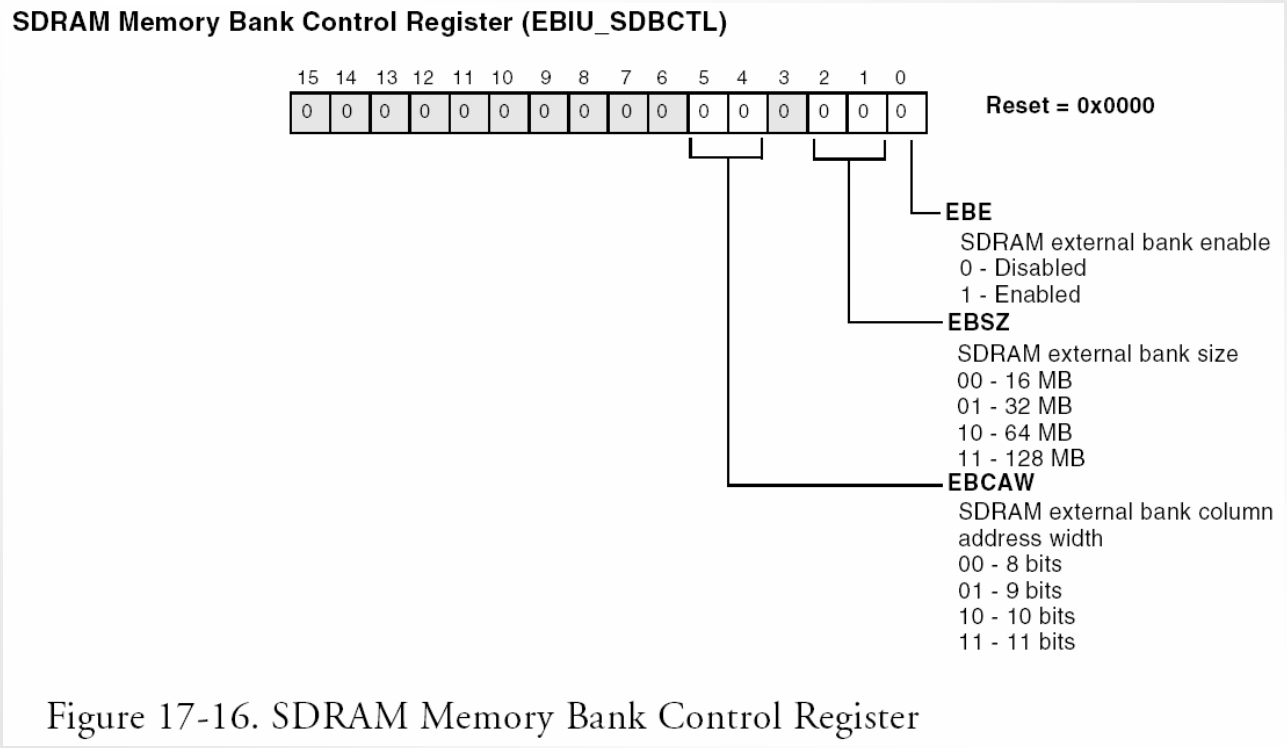
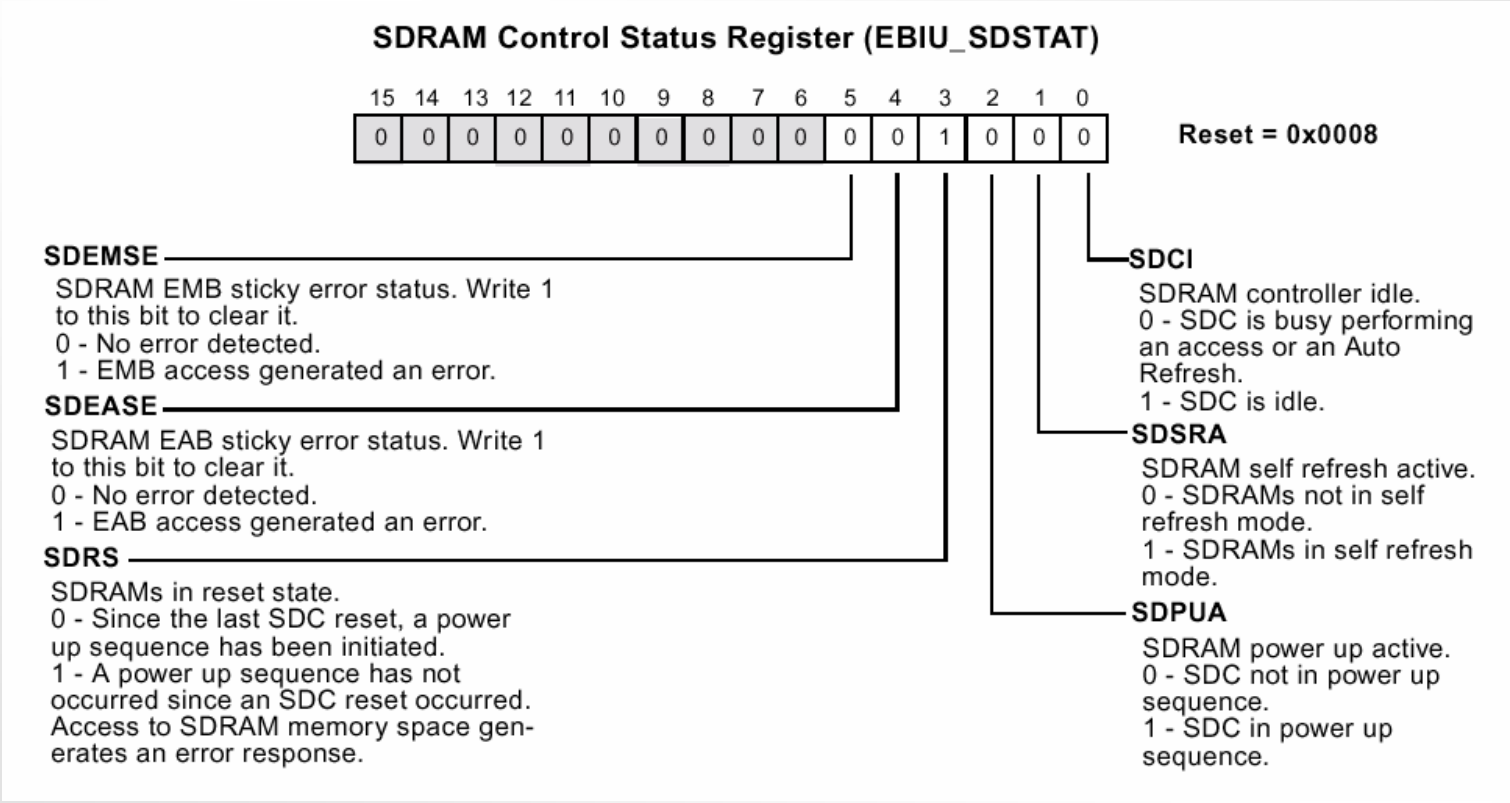
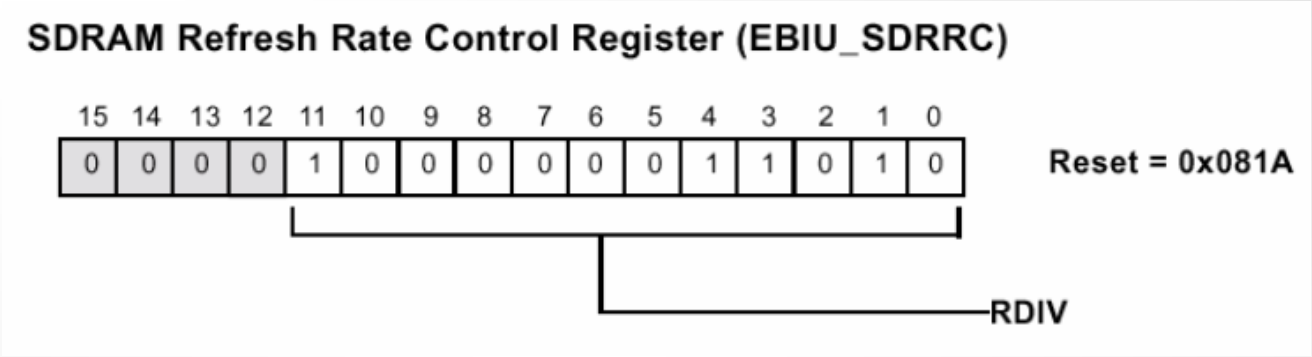


Figure 17-16. SDRAM Memory Bank Control Register

SDRAM Control Status Register (EBIU_SDSTAT)



SDRAM Refresh Rate Control Register (EBIU_SDRRC)



To calculate the value that should be written to the EBIU_SDRRC register, use the following equation:

$$RDIV = ((f_{SCLK} \times t_{REF})/NRA) - (t_{RAS} + t_{RP})$$

Where:

- f_{SCLK} = SCLK frequency (system clock frequency).
- t_{REF} = SDRAM refresh period.
- NRA = Number of row addresses in SDRAM (refresh cycles to refresh whole SDRAM).
- t_{RAS} = Active to Precharge time (TRAS in the SDRAM Memory Global Control register) in number of clock cycles.
- t_{RP} = RAS to Precharge time (TRP in the SDRAM Memory Global Control register) in number of clock cycles.

Bus Grant/ Bus Request

Bus Request and Grant

- Processor tri-states the memory interface to allow an external controller access to ASYNC or synch memory banks
  - The sequence starts when the external device asserts /BR
  - If no internal request is pending, the processor tri-states the data, control and address lines of the async memory
    - The synchronous interface is optionally tri-stated
  - At this point, /BG is asserted by the core
  - Once the bus has been granted, the processor asserts /BGH when it is ready to access external memory but it is being held off
  - When the external device releases /BR, the processor de-asserts /BG
- Note: the processor will stall if an internal core access is required to the external bus when the bus has been granted

EBIU Performance

Ideal memDMA EBIU System Throughput *

Source	Destination	Approximate SCLKS for N Words
16-bit SDRAM	L1 Data Memory	N+14
L1 Data Memory	16-bit SDRAM	N+11
16-bit ASYNC Memory	L1 Data Memory	XN+12
L1 Data Memory	16-bit ASYNC Memory	XN+9
16-bit SDRAM	16-bit SDRAM	10+(17*N/7)
16-bit ASYNC Memory	16-bit ASYNC Memory	10+2XN

2 < X = # of (wait states + setup + hold time)

* Measured numbers may be slightly higher on hardware

Core SDRAM Accesses

- Even though the EBIU bus width is 16-bits,
  - A 32-bit core access of the form:

    R0 = [P0]; // load from SDRAM memory takes 10 SCLKs

    Will be more efficient than two 16-bit core accesses of the form:

    R0 = w[P0++] ; // load from SDRAM memory takes 9 SCLKS

    R0 = w[P0++] ; // load from SDRAM memory takes 9 SCLKS

    This is due to the fact that it is more efficient to have the core make one EBIU request vs. two EBIU requests