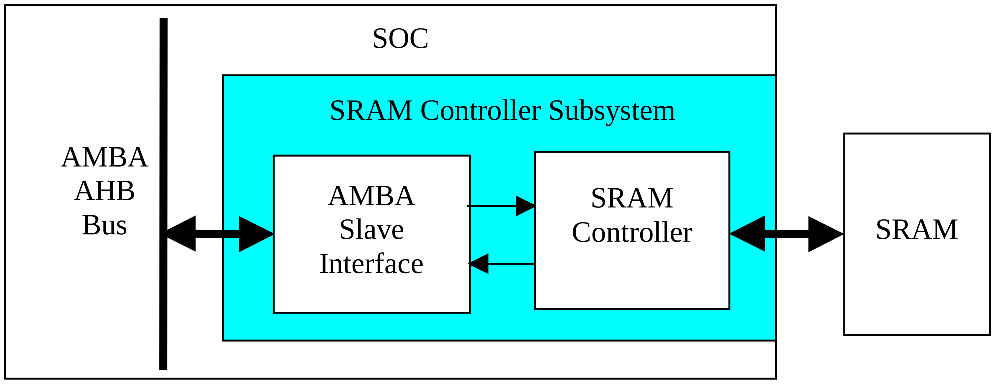
Aurora VLSI

Unleashing the Power of the Internet

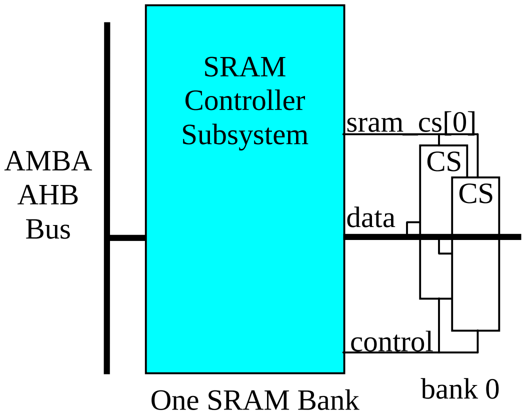
AU-MB1000: SRAM Controller AMBA Subsystem Core

AMBA AHB Bus SRAM Controller

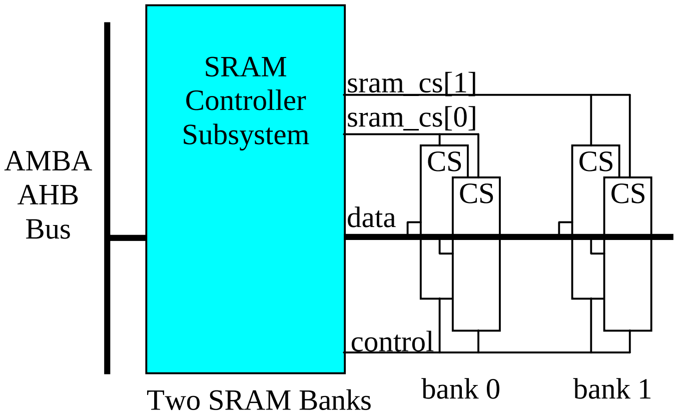
The AU-MB1000 SRAM Controller AMBA Subsystem provides an SRAM Controller peripheral subsystem for AMBA based SOCs. It contains an SRAM Controller that connects seamlessly to the AMBA AHB Bus as an AMBA Bus slave. The SRAM Controller AMBA Subsystem Core is available as a synthesizable Verilog model from Aurora VLSI, Inc. Contact CustomerService@auroravlsi.com.



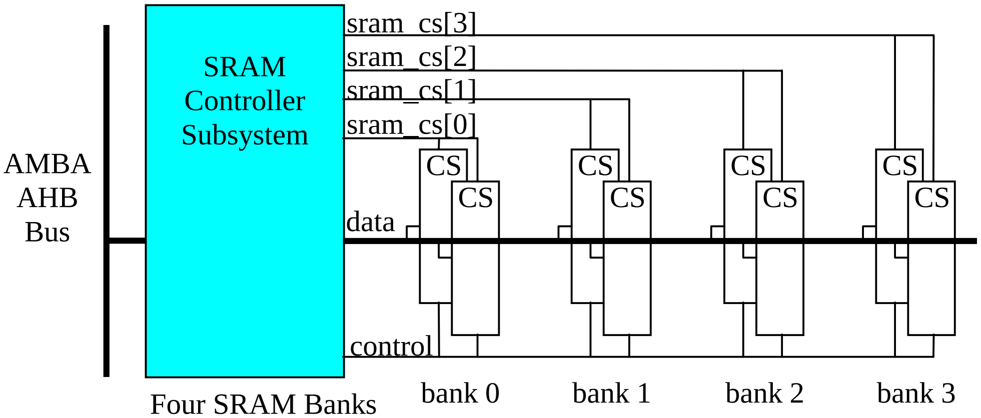
The SRAM Controller Subsystem supports one, two, or four SRAM banks in SRAM memory systems through its SRAM chip select outputs- sram_cs[3:0]. The SRAM data bus, control lines, and clock are common to all banks of SRAM.



One SRAM Bank



Two SRAM Banks



Four SRAM Banks

AMBA is a trademark of ARM Limited

The SRAM Controller Subsystem includes a versatile, pipelined, high performance SRAM controller. Several SRAM types are supported including flow through ZBT synchronous SRAMs, pipelined ZBT synchronous SRAMs, flow through syncburst synchronous SRAMs, SCD syncburst synchronous SRAMs, DCD syncburst synchronous SRAMs, and asynchronous SRAMs. The SRAM Controller supports SRAM memory systems from 512 Kbytes to 512 Mbytes. The SRAM data bus width is user configurable to 32 or 64 bits. To conserve power the SRAMs can be put in low power mode.

Internal to an SOC, the SRAM Controller Subsystem is a bus slave peripheral on the AMBA AHB Bus. The SRAM Controller Subsystem can interface to either a 32 bit or 64 bit AMBA AHB Bus. A Verilog parameter indicates the AMBA Bus width. AMBA Bus transactions that target the SRAMs, recognized by the AMBA Slave Interface of the SRAM Controller Subsystem. The AMBA Slave Interface then initiates an SRAM request at the requester interface of the SRAM Controller block. To complete the transaction, the AMBA Slave Interface drives the appropriate AMBA response onto the AMBA Bus.

SRAM Controller AMBA Subsystem features are summarized:

SRAM Controller

- Supports several SRAM types- flow through ZBT synchronous, pipelined ZBT synchronous, flow through syncburst synchronous, SCD syncburst synchronous, DCD syncburst synchronous, and asynchronous SRAMs
- 32 bit or 64 bit SRAM data bus
- 512 Kbyte to 512 Mbyte SRAM memory system
- Pipelined accesses for highest performance
- 0, 1, or 2 cycle read latency
- 0, 1, or 2 cycle write latency
- 1, 2, or 4 banks of SRAM
- 17 to 24 address bits
- SRAM powerdown supported

AMBA Slave Interface

- AMBA AHB Bus slave
- 32 bit or 64 bit AMBA AHB Bus- user configurable
- Supports all required AMBA AHB Bus features
- Implements AMBA Bus timeout and RETRY response
- Read data prefetching
- Write data packing
- Same cycle device request/response is supported for highest throughput
- Handles all data packing/unpacking and data alignment for data transfer sizes that do not match the AMBA Bus width and/or SRAM data bus width
- User configurable for big or little endian AMBA Bus and memory
- AMBA Bus and SRAM interface can be asynchronous to each other

The core is delivered as a synthesizeable RTL Verilog model. Deliverables include:

- RTL Verilog source code model of the core
- Verilog testbench and test cases
- Synthesis scripts examples
- Complete detailed documentation and training class notes

AMBA is a trademark of ARM Limited

SRAM Controller

The SRAM Controller Subsystem includes the AU-M1000 SRAM Controller Core. Additional logic at the requester interface of the SRAM Controller provides an AMBA Bus slave interface, read prefetching logic, and write data packing.

The SRAM Controller accepts SRAM requests and converts them into pipelined SRAM accesses within the SRAM Controller. This results in a peak bandwidth of 4 bytes/cycle with a 32 bit SRAM data bus, and 8 bytes/cycle with a 64 bit SRAM data bus. The SRAM Controller ensures that correct latencies and bus turn around times are met.

The number of SRAM address bits and SRAM banks is software configurable. Seventeen to twenty four address bits are supported. One, two, or four banks of SRAM can be built into the system. The SRAM chip select pins are used to identify the accessed bank of SRAM. This flexibility permits SRAM memory system of 512 Kbytes to 512 Mbytes.

SRAM read and write latencies are software programmable. This allows the SRAM Controller to be used with a wide variety of SRAMs. Asynchronous SRAMs and many types of synchronous SRAMs are supported.

Upon reset, SRAM accesses are optionally enabled or disabled according to a user configurable Verilog parameter. Software may also enable and disable SRAM accesses. Read and write latencies, the number of address bits, and number of SRAM banks are also user configurable upon reset with Verilog parameters. After reset de-assertion, software may overwrite the read and write latencies, number of address bits, and number of SRAM bank reset values.

The user may put the SRAMs into low power mode through software. Low power mode is exited when a read/write request occurs or when software exits low power mode.

AMBA Slave Interface

The AMBA Slave Interface of the SRAM Controller Subsystem, accepts SRAM requests from the AMBA Bus. The AMBA Slave Interface supports all required AMBA AHB Bus features including all AMBA burst and wrapping types, AMBA sizes up to the AMBA Bus width, and all AMBA Bus responses. When the AMBA Bus data transfer size does not match the AMBA Bus width or SRAM data bus width the AMBA Slave Interface packs and/or unpacks the data and aligns the data, for the most efficient transfer of data to/from the SRAMs and to/from the AMBA Bus.

The AMBA Slave Interface can respond in the cycle after it receives an AMBA Bus request. This ensures a slave response on the AMBA Bus with no wait states, and therefore highest system performance. To support this fast response time, read data is prefetched whenever possible.

Typically, the SRAMs and the AMBA Bus do not run at the same clock rate. The AMBA Bus and SRAM interface can be completely asynchronous to each other due to the two independent clock domains of the SRAM Controller Subsystem. One clock domain includes the AMBA Bus interface logic. The SRAM interface logic is in the second clock domain. These two clock domains come together in the AMBA Slave Interface block. Each clock domain has its own SRAM Controller Subsystem clock input.