

General Description

SDRAM is commonly used in cost-sensitive applications requiring large amounts of memory. Though SDRAM is inexpensive, the implementation of refresh operations, open row management, and various delays and command sequences requires logic. The Nios SDRAM Controller with an Avalon interface, transparently handles SDRAM initialization, refresh, and all other SDRAM requirements and appears to software as a simple linear memory interface (flat address space) with a wait signal. The SDRAM Controller supports standard SDRAM as described in PC100. With the SDRAM Controller, users can access SDRAM subsystems with data widths of 8, 16, 32, or 64-bits, various memory sizes, and multiple chip selects.

Users may also choose to share the tristate data and address buses with other tristate devices. This feature is valuable in systems which have multiple types of external memory devices, but limited I/O pins. The Avalon tristate interface is required to share pins.



See the *Avalon Interface to Off-Chip Devices* chapter of the *Avalon Bus Specification Reference Manual* for more information about the Avalon tristate interface.

Performance Considerations

SDRAM chips are arranged as multiple banks of memory with each bank capable of independent open row address management. This controller takes advantage of open row management for a single bank. Systems or applications that frequently change the destination bank for operations will require extra management cycles (row closings and openings) to access data. Continuous reads or writes within the same row and bank will operate at rates approaching one word per clock.

When the controller shares data pins, it will automatically maintain control of the associated tristate bus as long as back-to-back read or write transactions within the same row and bank persist. A break in back-to-back transactions, or a required refresh transaction, will force a row to be closed. This both prevents the controller from permanently blocking access to other tristate devices and guarantees that the controller does not violate the SDRAM's row open time limit. When the controller shares data pins, it requires more row open and close overhead cycles.

PTF Assignments

A particular instantiation of the SDRAM Controller is generated according to parameters configured in the system PTF file. The parameters take into account:

- system clock frequency
- SDRAM timing assignments
- size of SDRAM row and column addresses
- number of SDRAM chipselect signals
- choice of options governing the performance vs. complexity trade-off

Designers use the SDRAM Controller configuration wizard in SOPC Builder to assign timing parameters for a specific SDRAM device. The timing parameters (including the requested system frequency) are used to determine all of the controller's timing values (in integer multiples of system clock cycles). Regardless of the input value, the timing will be based on the system clock frequency and timing will be the number of clock ticks that provide a value greater than or equal to the input value.

The SDRAM Controller is built for a particular application according to the assignments in the system PTF file. Generating a controller for a new SDRAM configuration requires configuring the SDRAM Controller's size and timing. This can be done in one of two ways:

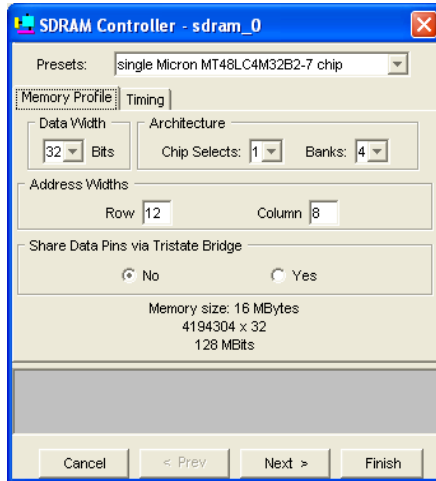
1. Change the settings in the SDRAM Controller's configuration wizard.
2. Edit the SDRAM Controller PTF file's size and timing assignments.



See the *SOPC Builder Data Sheet* for more information about the SOPC Builder configuration wizards and see the *SOPC Builder PTF File Reference Manual* for PTF file assignment details.

[Table 1 on page 3](#) – [Table 3 on page 7](#) list the SDRAM Controller PTF file parameters. All of these parameters are located in the system PTF files. Most of these settings may be set in the SDRAM Controller configuration wizards (see [Figure 1](#) and [Figure 2](#)). The configuration wizard settings associated with the PTF file parameters are listed in the “Wizard Field Name/Control” column in [Table 1](#) and the “Wizard Field” column in [Table 2](#). SDRAM Controller PTF settings that cannot be set in the configuration wizards, are noted in [Table 1 - Table 3](#) as “not available via wizard.”

The default PTF values match the settings for the preset value “single Micron MT48LC4M32B2-7 chip” shown in [Figure 1](#). Several other presets are available to work with other SDRAM chips.

Figure 1. SDRAM Controller Memory Profile Wizard**Table 1. SDRAM Controller Memory Profile Parameters (Part 1 of 2)**

Parameter	Wizard Field Name/Control	Type	Allowed Value	Default	Units	Description
sdram_data_width	Data Width/Bits	Integer	8, 16, 32, 64	32	bits	SDRAM data bus width.
sdram_addr_width	(Not available via wizard: defaults to sdram_row_width)	Integer	(*)	12	bits	Number of SDRAM address pins. Normally the same as sdram_row_width.
sdram_num_chipselects	Architecture/Chip Selects	Integer	1, 2, 4, 8	1	bits	Number of independent chip selects in the SDRAM system.
sdram_num_banks	Architecture/Banks	Integer	2, 4	4	banks	Number of SDRAM banks. The number of bank address pins (sdram_bank_width) is determined for this value.

Table 1. SDRAM Controller Memory Profile Parameters (Part 2 of 2)

Parameter	Wizard Field Name/Control	Type	Allowed Value	Default	Units	Description
s dram_row_width	Address Widths/ Rows	Integer	≥ 11 and ≤ 14	12	bits	Number of row address bits. This value depends on SDRAM geometry. For example, an SDRAM organized as 4096 rows by 512 columns has a s dram_row_width value of 12.
s dram_col_width	Address Widths/ Columns	Integer	≥ 8 and \leq s dram_row_ width	8	bits	Number of column address bits. For example, the SDRAM organized as 4096 rows by 512 columns has a s dram_col_width value of 9.
shared_data	Share Data Pins via Tristate Bridge	Boolean	0, 1	0	-	Requires tristate-bus bridge

Figure 2. SDRAM Controller Timing Wizard

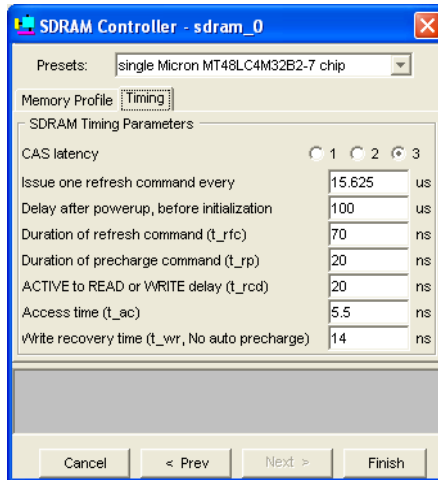


Table 2. SDRAM Controller Timing Parameters (Part 1 of 2)

Parameter	Wizard Field	Type	Allowed Values	Default	Units	Description
cas_latency	CAS latency	Integer	1, 2, 3	3	clock cycles	Latency in clocks from a READ command to data out. Typical SDRAM chips support CAS latency of 2 or 3; rare examples also support CAS latency = 1. For best performance, use the lowest CAS latency possible, but beware of increased t _{ac} at lower CAS latencies.
refresh_period	Issue one refresh command every	Floating Point	(*)	15.625	μs	One refresh command is executed each refresh_period. A typical SDRAM requires 4096 refresh commands every 64ms, which can be met by issuing one refresh command every 64ms / 4096 = 15.625μs.

Table 2. SDRAM Controller Timing Parameters (Part 2 of 2)

Parameter	Wizard Field	Type	Allowed Values	Default	Units	Description
powerup_delay	Delay after power up before initialization	Floating Point	(*)	100	μs	The delay from stable clock and power to SDRAM initialization.
t_rfc	Duration of refresh command (t_rfc)	Floating Point	(*)	70	ns	Auto Refresh period.
t_rp	Duration of precharge command (t_rp)	Floating Point	(*)	20	ns	PRECHARGE command period.
t_rcd	Active to READ or WRITE delay (t_rcd)	Floating Point	(*)	20	ns	ACTIVE to READ or WRITE delay.
t_ac	Access time (t_ac)	Floating Point	(*)	17	ns	Access time from clock edge. This value may depend on CAS latency.
t_wr	Write recovery time. (t_wr, No auto precharge)	Floating Point	(*)	14	ns	Write recovery if explicit PRECHARGE commands are issued. This SDRAM Controller always issues explicit precharge commands.

Table 3. SDRAM Controller Advanced Parameters (not available via wizard)

Parameter	Type	Allowed Values	Default	Units	Description
init_refresh_commands	Integer	1, 2, 3, 4, 5, 6, 7, 8	2	Auto Refresh commands	The number of refresh commands required during initialization. The SDRAM controller logic is minimized by smaller values of this parameter.
t_mrd	Floating Point	(*)	2	clocks	LOAD MODE REGISTER command to ACTIVE or REFRESH command. JEDEC and PC100 specify 3 clocks; other parts accept 2 clocks.
register_data_in	Boolean	0, 1	1	–	Uses fast input registers for data returned by SDRAM READs. Otherwise, this parameter lets data flow directly from RAM back through the Avalon bus unregistered.
starvation_indicator	Boolean	0, 1	0	–	Requires a tristate-bus bridge. This parameter generates an extra output port from the controller that pulses logic high when the controller is starved off the tristate bridge for too long to issue Auto Refresh commands.

(*) Correct values depend upon the information provided in the SDRAM manufacturer's data sheet.

Examples

The following examples show how to connect the SDRAM controller outputs to an SDRAM chip or chips. The bus labeled `ctl` is an aggregate of the signals `cas_n`, `ras_n`, `cke` and `we_n`.

Figure 3 shows a single 128 Mbit SDRAM chip with 32-bit data. Address, data and control signals are wired directly from the controller to the chip. The result is a 128Mbit (16Mbyte) memory space.

Figure 3. Single 128 Mbit SDRAM Chip with 32-Bit data

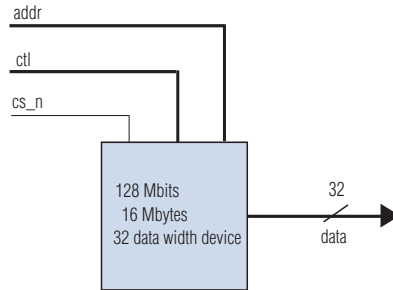


Figure 4 shows two 64Mbit SDRAM chips, each with 16-bit data. Address and control signals wire in parallel to both chips. Note that chipselect (`cs_n`) is shared by the chips. Each chip provides half of the 32-bit data bus. The result is a logical 128Mbit (16Mbyte) 32-bit data memory.

Figure 4. Two 64Mbit SDRAM Chips each with 16-Bit Data

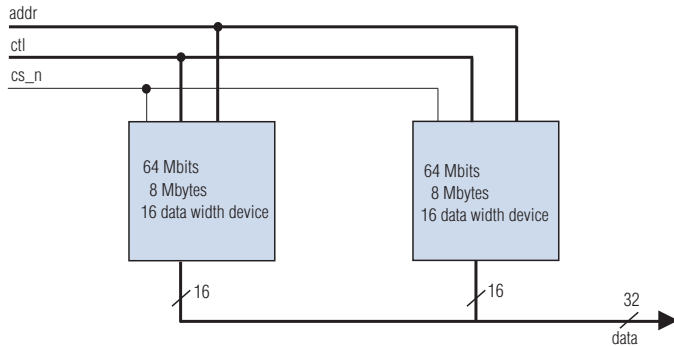
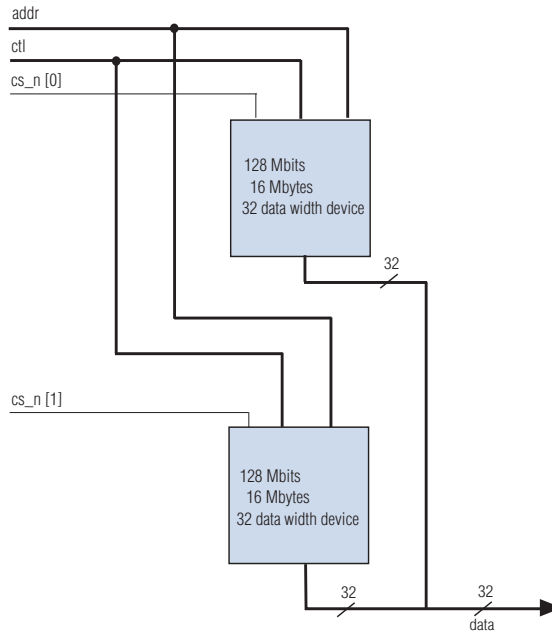


Figure 5 shows two 128Mbit SDRAM chips, each with 32-bit data. Control, address and data signals wire in parallel to the two chips. The chipselect bus ($cs_n[1:0]$) determines which chip is selected. The result is a logical 256Mbit 32-bit wide memory.

Figure 5. Two 128Mbit SDRAM Chips each with 32-Bit Data





101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com

Copyright © 2003 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, mask work rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

