SDRAM CONTROLLER
Specification

Author: Dinesh Annayya

dinesha@opencores.org

Rev. 0.0
January 18, 2012

This page has been intentionally left blank

Revision History

Rev.	Date	Author	Description
0.0	01/17/2012	Dinesh Annayya	First draft release

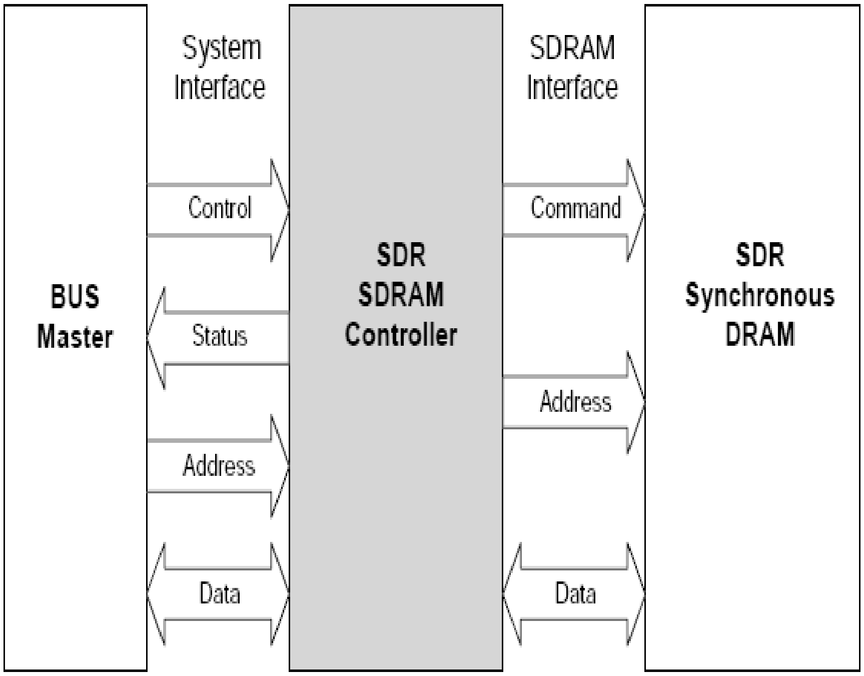
1

Introduction

Synchronous DRAM (SDRAM) has become a mainstream memory of choice in embedded system memory design. For high-end applications using processors the interface to the SDRAM is supported by the processor's built-in peripheral module. However, for other applications, the system designer must design a controller to provide proper commands for SDRAM initialization, read/write accesses and memory refresh.

This SDRAM controller reference design, located between the SDRAM and the bus master, reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the bus master. Figure 1 shows the relationship of the controller between the bus master and SDRAM. The bus master can be either a microprocessor or a user's proprietary module interface.

Figure 1. SDR SDRAM Controller System



FEATURES

- Support for industry-standard SDRAM devices and modules
- Supports all standard SDRAM functions
- Fully Synchronous; All signals registered on positive edge of system clock
- One chip-select signals
- Support SDRAM with four bank
- Programmable CAS latency
- Data mask signals for partial write operations
- Bank management architecture, which minimizes latency
- 16/32 Configurable data width
- Automatic controlled refresh
- Static synchronous design
- Fully synthesizable

2

IO ports

2.1 Core Parameters

Parameter	Type	Default	Description
SDR_DW	Bit	16	SDRAM DATA Width Selection: 16 – 16 Bit SDRAM Mode 32 – 32 Bit SDRAM Mode
SDR_BW	Bit	2	SDRAM BYTE Width Selection 2 – 16 Bit SDRAM Mode 4 – 32 Bit SDRAM Mode

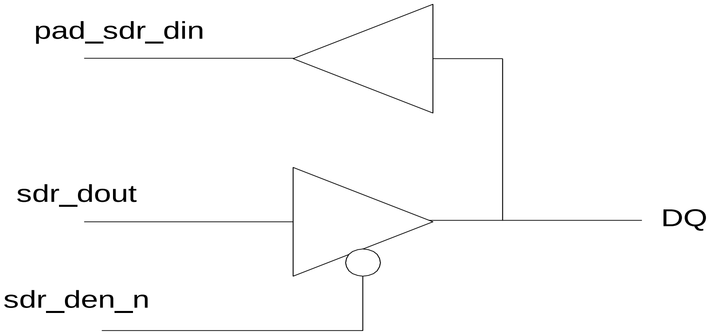
2.2 Application interface signals

Port	Width	Direction	Description
app_req	1	Input	Application Request
app_req_addr	30	Input	Address
app_req_addr_mask	29	Input	Address Mask
app_req_wr_n	1	Input	0 - Write, 1 - Read
app_req_wrap	1	Input	Address Wrap
app_req_ack	1	Output	Application Request Ack
sdr_core_busy_n	1	Output	SDRAM Controller Busy Indication, 0 - busy, 1 - free
app_wr_data	32	Input	Write Data
app_wr_next_req	1	Input	Next Write Data Request
app_wr_en_n	4	Output	Byte wise write Enable, Active low
app_rd_data	32	Input	Read Data
app_rd_valid	1	Output	Read Valid

2.3 SDRAM External connections

Port	Width	Direction	Description
sdr_cke	1	Output	SDRAM clock enable
sdr_cs_n	1	Output	SDRAM command inputs CS#
sdr_ras_n	1	Output	SDRAM command inputs RAS#
sdr_cas_n	1	Output	SDRAM command inputs CAS#
sdr_we_n	1	Output	SDRAM command inputs WE#
sdr_dqm	2/4	Output	SDRAM data bus mask
sdr_ba	2	Output	SDRAM bank address
sdr_addr	12	Output	SDRAM address bus
pad_sdr_din	16/32	Input	SDRAM Data Inut
sdr_dout	16/32	Output	SDRAM Data Output
sdr_den_n	16/32	Output	SDRAM Data Output enable

The tri-state buffers for the DQ lines must be added at a higher hierarchical level. Connections should be made according to the following figure:



Verilog code for 32 BIT SDRAM:

```
assign Dq[7:0]    = (sdr_den_n[0] == 1'b0) ? sdr_dout[7:0]   : 8'hZZ;
assign Dq[15:8]   = (sdr_den_n[1] == 1'b0) ? sdr_dout[15:8]  : 8'hZZ;
assign Dq[23:16]  = (sdr_den_n[2] == 1'b0) ? sdr_dout[23:16] : 8'hZZ;
assign Dq[31:24]  = (sdr_den_n[3] == 1'b0) ? sdr_dout[31:24] : 8'hZZ;
```

Verilog code for 16 BIT SDRAM:

```
assign Dq[7:0]    = (sdr_den_n[0] == 1'b0) ? sdr_dout[7:0]   : 8'hZZ;
assign Dq[15:8]   = (sdr_den_n[1] == 1'b0) ? sdr_dout[15:8]  : 8'hZZ;
```

Simulation

Run Directorty: **sdr_ctrl/trunk/verif/run**

1. Compiling and Simulating in 16 BIT SDR Mode
   ./run_modelsim SDR_16BIT
2. Compiling and Simulating in 32 BIT SDR Mode
   ./run_modelsim SDR_32BIT

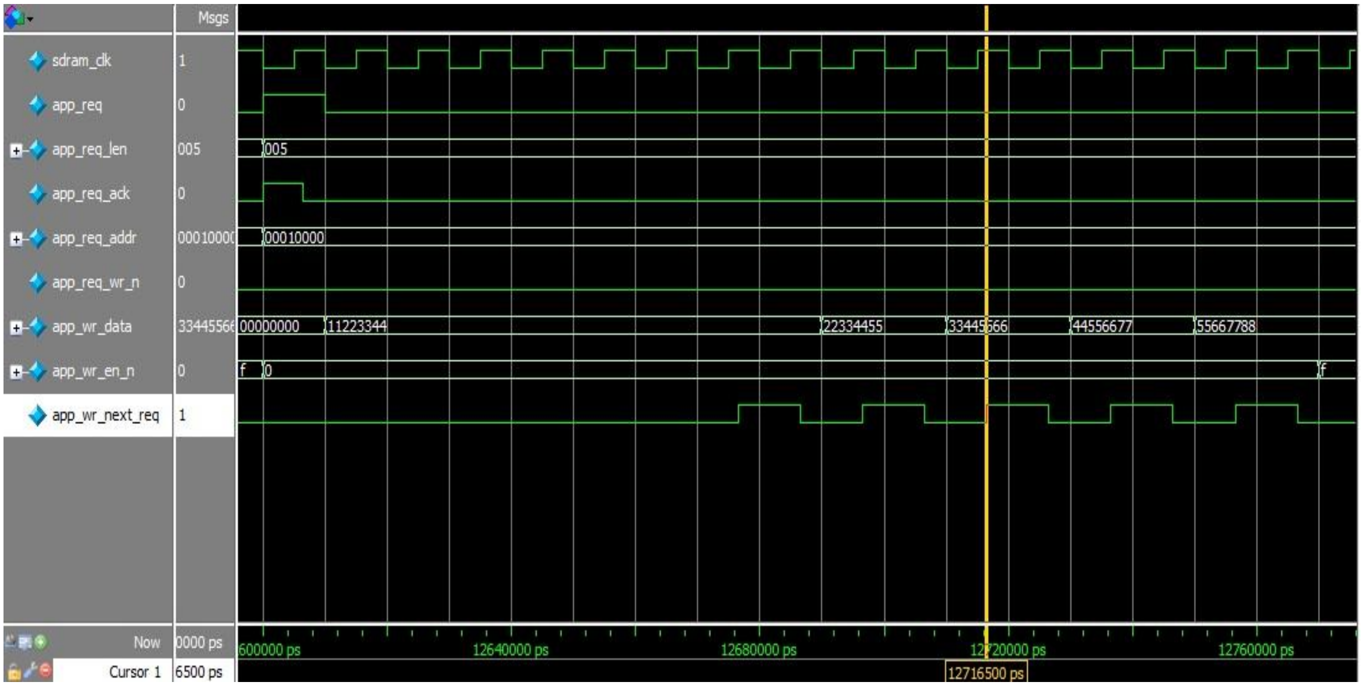
Golden Log file are available under: **sdr_ctrl/trunk/verif/log**

- **SDR_16BIT_complie.log** - SDR-16BIT Compile log
- **sdr16_sim.log** – SDR-16BIT Simulation log
- **SDR_16BIT_basic_test1.log** -- SDR-16BIT Basic Test simulation log
- **SDR_32BIT_complie.log** - SDR-32BIT Compile log
- **sdr32_sim.log** -- SDR-32BIT Simulation log
- **SDR_32BIT_basic_test1.log** – SDR-32BIT Basic Test Simulation log

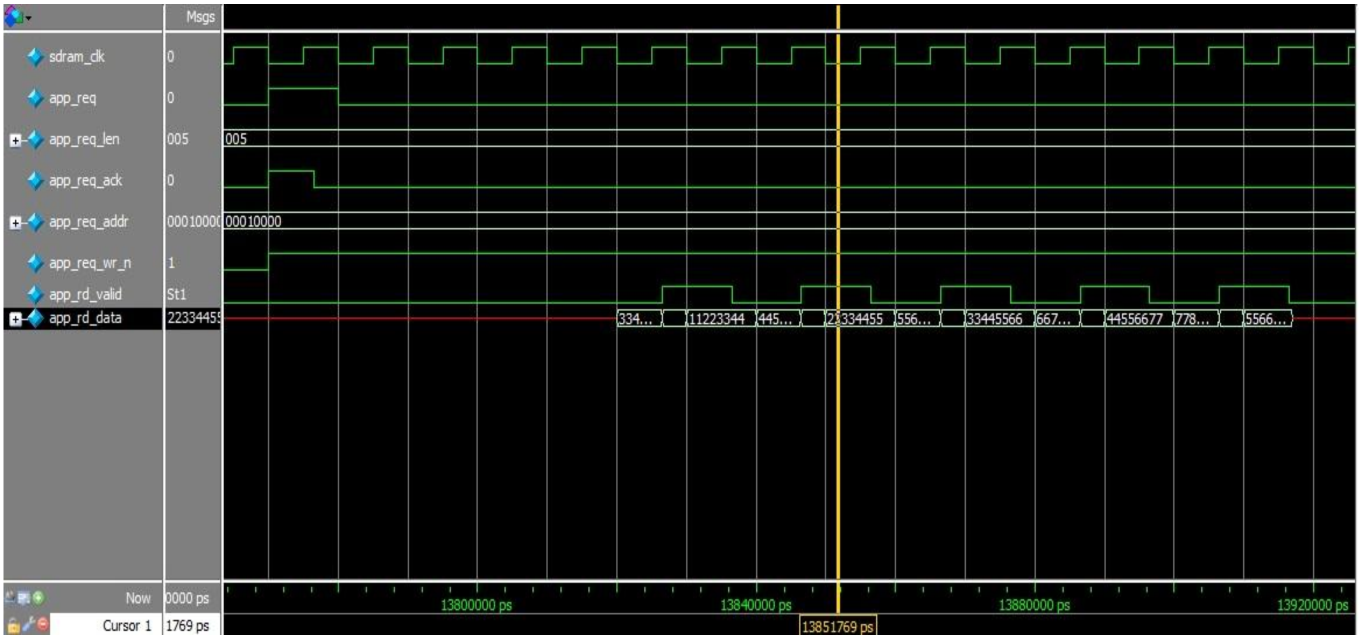
WAVEFORM

Image files are available under: **sdr_ctrl\trunk\verif\dump**

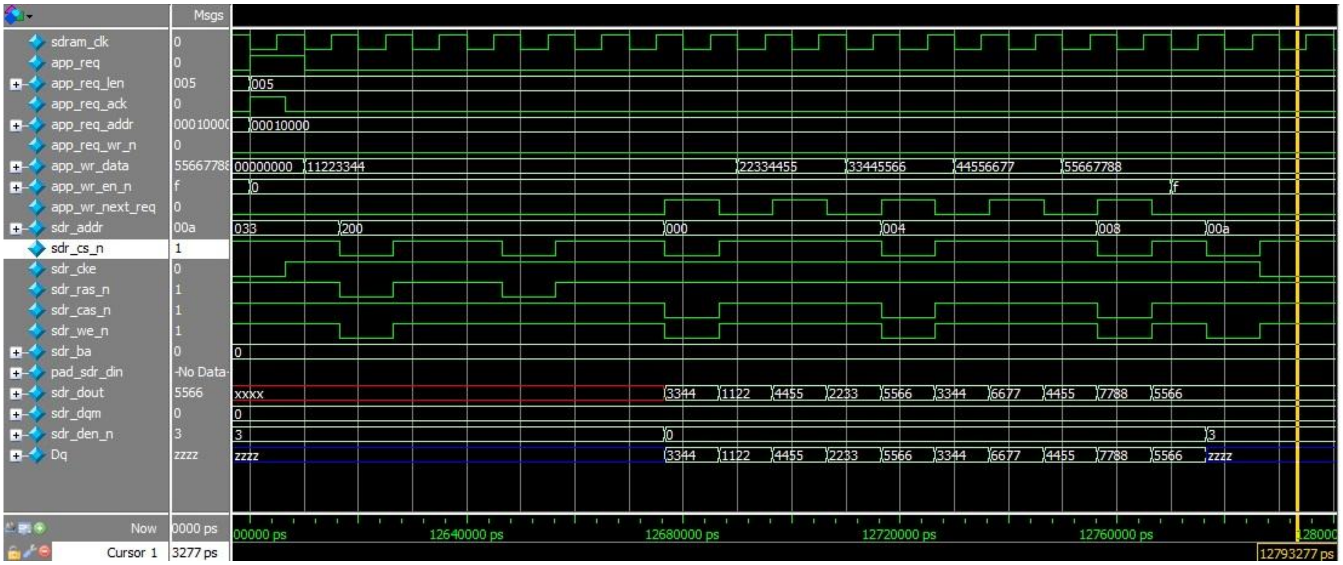
1. Application Write Request:



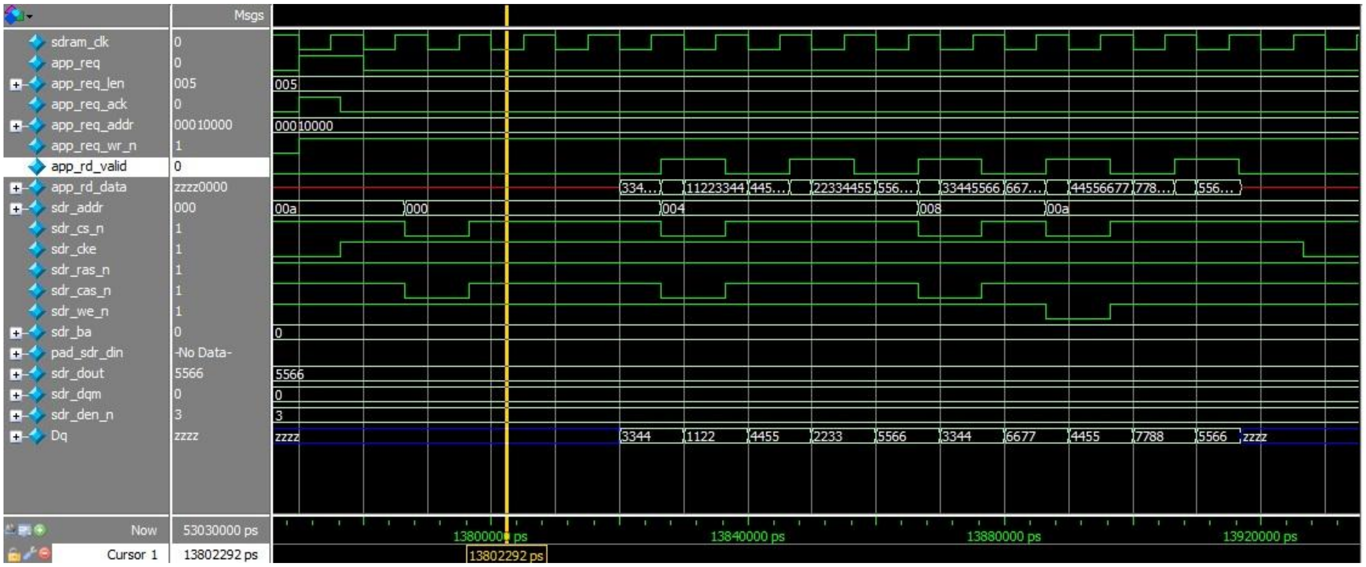
2. Application Read Request



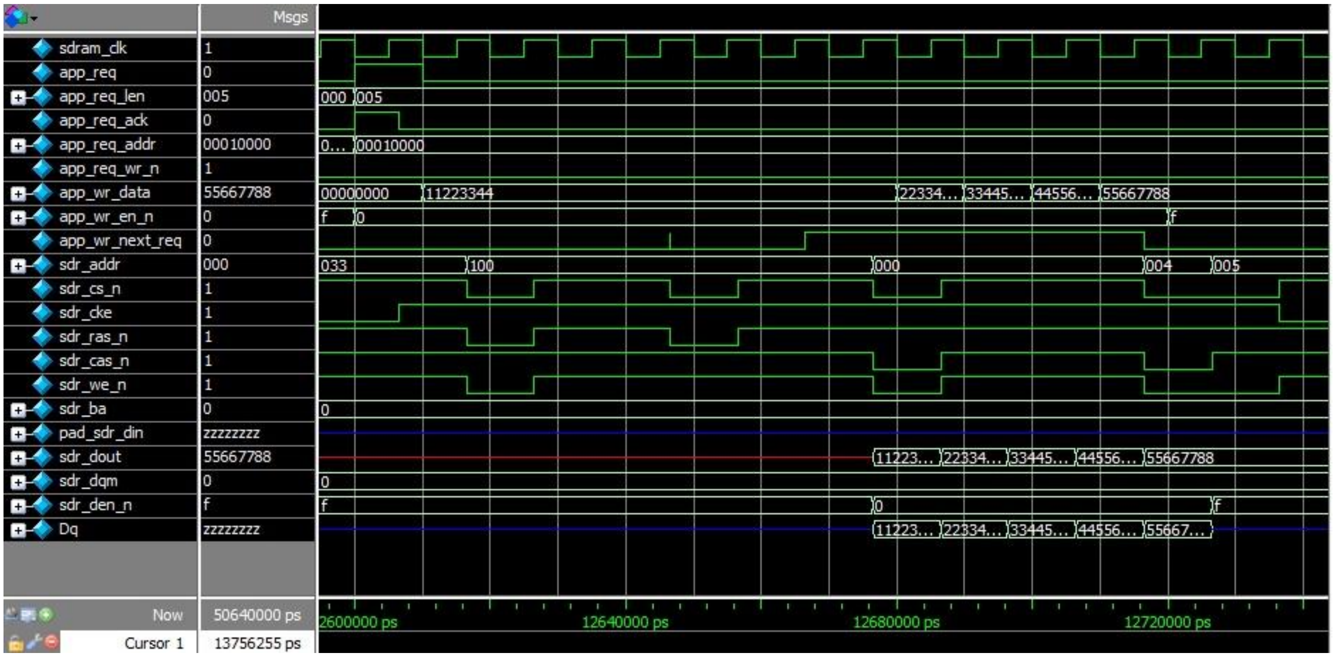
3. SDRAM 16 Bit Write Transaction



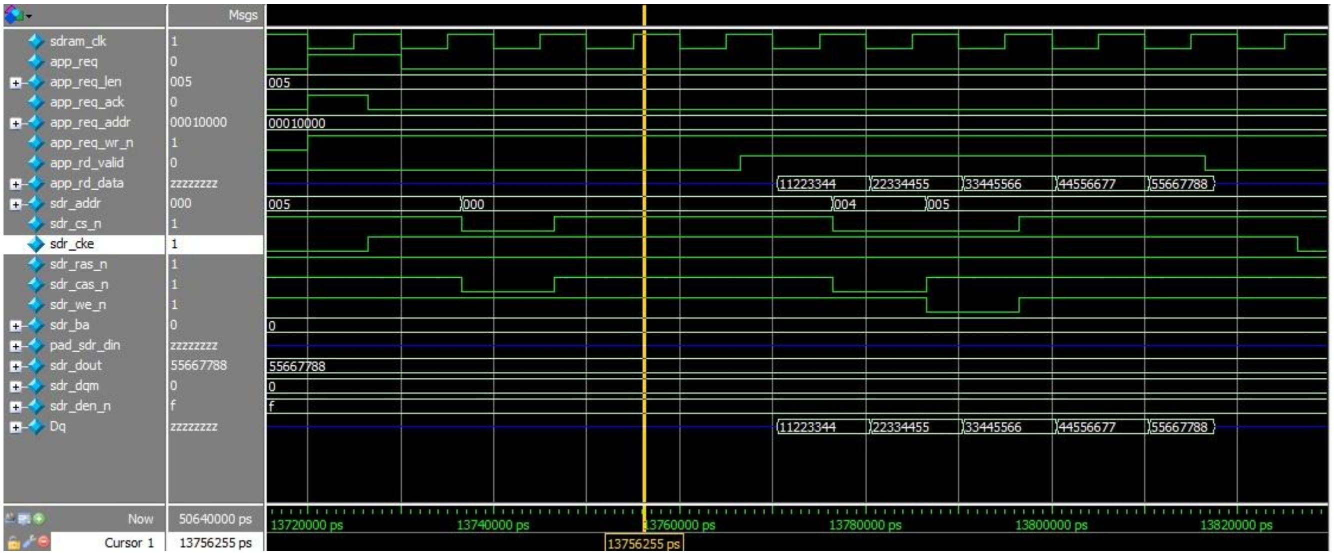
4. SDRAM 16 Bit Read Transaction



5. SDRAM 32 Bit Write Transaction



6. SDRAM 32 bit Read Transaction


Msgs
sdram_clk 1
app_req 0
app_req_len 005
app_req_ack 0
app_req_addr 00010000
app_req_wr_n 1
app_rd_valid 0
app_rd_data zzzzzzzz
sdr_addr 000
sdr_cs_n 1
sdr_cke 1
sdr_ras_n 1
sdr_cas_n 1
sdr_we_n 1
sdr_ba 0
pad_sdr_din zzzzzzzz
sdr_dout 55667788
sdr_dqm 0
sdr_den_n f
Dq zzzzzzzz
005
00010000
11223344
22334455
33445566
44556677
55667788
005
000
004
005
0
55667788
0
f
11223344
22334455
33445566
44556677
55667788
Now 50640000 ps
Cursor 1 13756255 ps
13720000 ps
13740000 ps
13760000 ps
13780000 ps
13800000 ps
13820000 ps
13756255 ps

Appendix A

Synthesis results