XSA SDRAM Controller

September 5, 2002 (Version 1.1) Application Note by D. Vanden Bout

Summary

This application note describes the interface and circuits for the XSA Board SDRAM controller.

XSA SDRAM Controller Interface

The XSA synchronous DRAM (SDRAM) controller module accepts simple read and write requests on the host side and generates the timed waveforms required to perform these operations on the SDRAM while keeping the SDRAM refreshed.

The interface for the module is shown in Figure 1. The functions of the I/O signals are as follows:

clkin: This is the master clock input. The clock from the DS1075 programmable oscillator enters the FPGA through a global clock input pin and drives this module input. The clock frequency must be 25 MHz or greater or else the DLLs in the FPGA will not lock and the SDRAM controller will operate incorrectly.

sclk: This output is derived from the master clock and it drives the clock input of the external SDRAM.

sclkfb: This input is a copy of the SDRAM clock signal with delays added by its passage out of the FPGA to the SDRAM and back into the FPGA through a global clock input. The SDRAM controller uses a DLL to compensate for these delays.

clk0: Miscellaneous logic circuitry in the FPGA is clocked with this signal that is derived from the master clock. The DLLs are used to synchronize sclk and clk0.

clk2x: This is a clock-doubled version of the master clock and it is also synchronized to sclk and clk0.

lock: This signal goes high when the DLLs in the SDRAM controller are locked which indicates the sclk, clk0 and clk2x clock signals are valid.

bufclk: This is just a globally buffered version of the master clock. It is available even when the DLLs have not locked.

rst: This is an active-high, synchronous reset for the internal logic of the SDRAM controller. The reset also causes the controller to initialize the SDRAM for use.

rd: This active-high input initiates a read of a single word from the SDRAM. It is sampled on the rising clock edge and must be held high throughout the read operation. The read control must be lowered after the done signal goes high and before the next rising clock edge or else another read operation will start.

wr: This active-high input initiates a write of a single word from the SDRAM. It is sampled on the rising clock edge and must be held high throughout the write operation. The write control must be lowered after the done signal goes high and before the next rising clock edge or else another write operation will start.

done: This synchronous output signal goes high to indicate the completion of the currently active read or write operation.

hAddr: The address of the SDRAM word that is to be read or written is passed through this input bus. The address value must be held stable throughout the read or write operation.

hDIn: The data to be written to the SDRAM enters through this input bus. The data value must be held stable throughout the write operation.

hDOut: The data read from the SDRAM comes out on this bus. This data must be latched by the host side logic on the rising clock edge after the done signal goes high.

cke: This output drives the clock-enable input of the SDRAM.

cs_n: This output drives the chip-select of the SDRAM.

ras_n: This output drives the RAS input of the SDRAM.

cas_n: This output drives the CAS input of the SDRAM.

we_n: This output drives the write-enable input of the SDRAM.

ba: This two-bit bus activates one of the four banks of memory in the SDRAM.

sAddr: The row and column address fields for the SDRAM memory location are output on this bus.

sData: The data word to be written to SDRAM exits the FPGA on this bus during write operations, and data from the SDRAM enters the FPGA on this bus during read operations.

dqmh: This output drives the SDRAM input that controls the drivers for the upper half of the data bus during read operations.

dqml: This output drives the SDRAM input that controls the drivers for the lower half of the data bus during read operations.

sdramCntl_state: The current state of the SDRAM controller is made available on this four-bit bus. This is used only for diagnostic purposes.

SDRAM Controller Application

A simple memory tester application is used to demonstrate the use of the SDRAM controller module. The memory tester performs the following functions:

- It initializes a pseudo-random number generator (RNG) with a known seed value.
- It writes a sequence of random numbers throughout a range of memory addresses.
- It re-initializes the RNG with the seed value.
- It reads back the contents of memory and compares it to the sequence from the RNG. Any mismatch indicates an error reading or writing the memory.

The source files that describe this application are:

xs_pckg.vhd: Some functions and definitions useful in many applications are provided in this file.

memtest.vhd: A generic memory tester is described in this file.

randgen.vhd: This file contains the RNG used by the generic memory tester.

sdramcnt.vhd: This file describes the SDRAM controller.

tst50.vhd, tst100.vhd: These are the top-level files that combine the generic memory tester and the SDRAM controller modules to make the complete SDRAM tester for the XSA-50 or XSA-100 Board.

xsa.ucf: The pin assignments for mapping the SDRAM tester to the XSA Board are listed in this file.

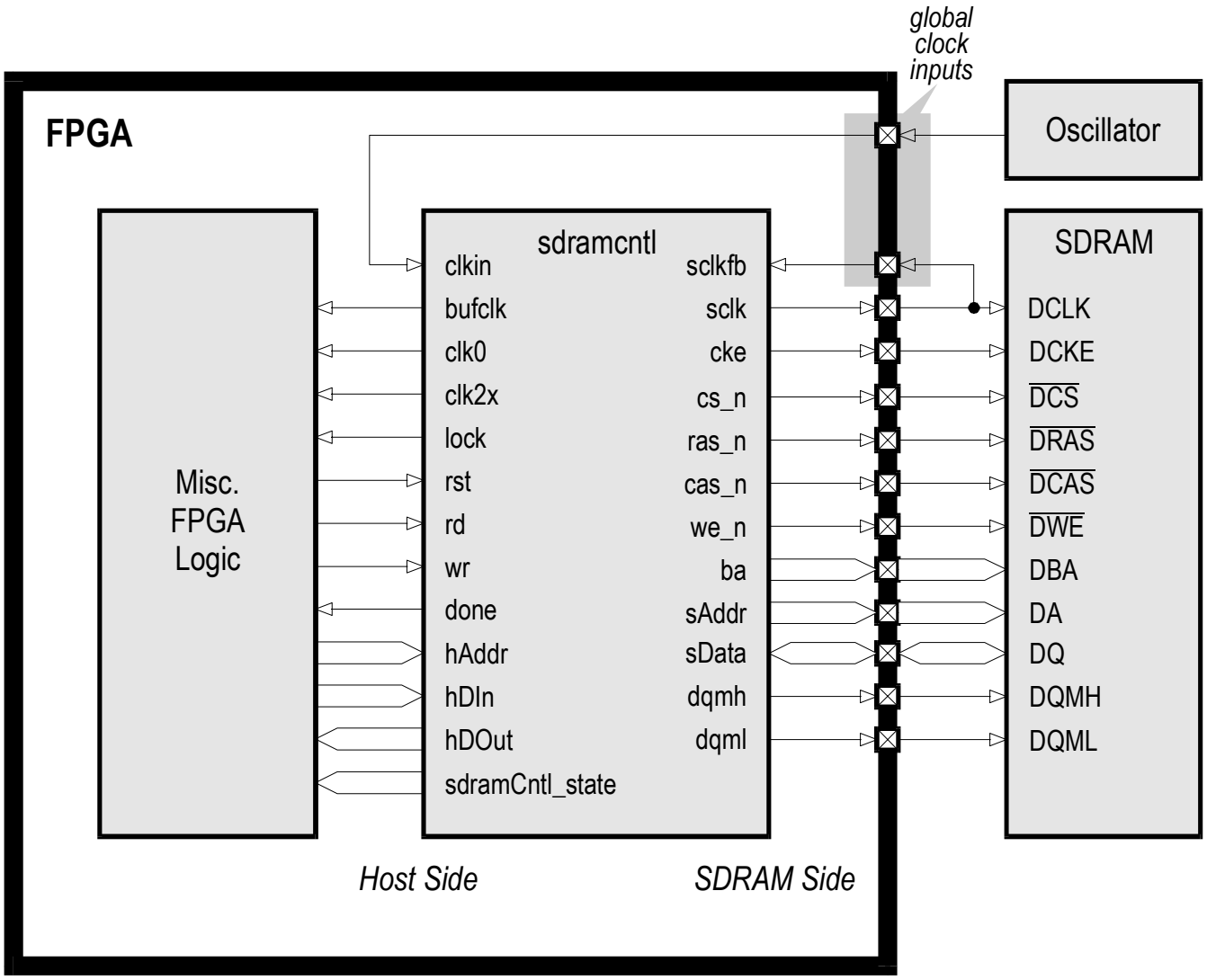

Figure 1: XSA Board SDRAM controller interfaces.