1.0 INTRODUCTION

The MicroBlaze DDR SDRAM Test Module from Avnet Design Services provides the user with a complete MicroBlaze DDR SDRAM Test Platform.

2.0 Block Diagram

Figure 2-1 shows the four main components of the test platform: MicroBlaze stub, Read/Write Buffers and Registers, Command Control unit, and the DDR Controller.

2.1.1 MicroBlaze

The MicroBlaze is a 32-bit soft processor core from Xilinx which is used for control by the test platform. MicroBlaze is a RISC processor with separate instruction and data busses that can execute programs from both internal block RAM and external memory. MicroBlaze can run up to 75MHz in the FPGA. Included in this document is an instruction set on how to use the MicroBlaze to execute the tests.

2.1.2 Buffers and Registers

The test platform contains two 8KB buffers and a set of registers which are used by the MicroBlaze to perform burst write/read accesses to/from the DDR SDRAM. The buffers and registers reside in MicroBlaze address space. The functions of the registers are defined in the Memory Map section.

2.1.3 Command Control Unit

The Command Control Unit is a state machine which interfaces the MicroBlaze buffers to the DDR Controller. The Command Control Unit communicates with the MicroBlaze through two read/write communication register ports. The Unit communicates with the DDR Controller through a set of DDR SDRAM commands defined by the controller. When the MicroBlaze wants a read or write test to be performed, it initializes the appropriate registers and sets the respective register port bit. The command control unit, which constantly monitors the ports, issues the appropriate SDRAM commands to the controller, and the controller then issues the memory commands (rasb, casb, web, csb) to the DDR SDRAM. The Command Control Unit clears the communication port at the end of the test to inform the MicroBlaze that the test was successfully completed.

2.1.4 DDR Controller

The DDR Controller is a modified version of Xilinx's DDR Controller [XAPP200, PC:64-bit] which can be downloaded from Xilinx website: http://www.xilinx.com/apps/xapp.htm. The website also contains an application note that explains the details of the controller. To achieve consistent place and routing timing, the controller clocks had to be modified to use one clock domain for all modules. Other modifications were made to the Data and Address Path units.

The DDR Controller communicates with the Command Control Unit through a set 7-bit DDR SDRAM commands:

- NOP : u_cmd = 0000001
- Extended Mode Register Set (EMRS): u_cmd = 0000010 with BA1 = 1, BA0 = 1
- Mode Register Set (MRS) : u_cmd = 0000010 with BA1 = 0, BA0 = 0
- Read with autoprecharge : u_cmd = 0000100
- Write with autoprecharge : u_cmd = 0001000
- Precharge all banks: u_cmd = 0010000
- Auto Refresh: u_cmd = 0100000 (Note 1)
- Burst stop: u_cmd = 1000000

Note 1:
The controller does not include a refresh counter, hence the Command Control Unit has a refresh counter which periodically issues a refresh command to the controller.

3.0 MicroBlaze Address Map

32-Bit Address Range (hex)	**Register**	**Type**	**Notes**
30FF0000	Revision	R	Revision Register
30FF0004	TestReg1	R/W	Test Register_1
30FF0008	TestReg2	R/W	Test Register_2
30FF000C - 30FF00FC	***Unused***		***Available***
30FF0100	wr_mem_startaddr	R/W	Write Memory Start Address
30FF0104	wr_mem_length	R/W	Write Memory Length (specified in bytes)
30FF0108	wr_comm._port	R/W	Write Communication Port
30FF010C	wr_buff_startaddr	R/W	Write Buffer Start Address
30FF0110 - 30FF01FC	***Unused***		***Available***
30FF0200	rd_mem_startaddr	R/W	Read Memory Start Address
30FF0204	rd_mem_length	R/W	Read Memory Length (specified in bytes)
30FF0208	rd_comm._port	R/W	Read Communication Port
30FF020C	rd_buff_startaddr	R/W	Read Buffer Start Address
30FF0210 - 30FF1FFC	***Unused***		***Available***
30FF2000 - 30FF3FFC	wr_buffer_memory	R/W	Write Buffer Memory
30FF4000 - 30FF5FFC	Rd_buffer_memory	R/W	Read Buffer Memory
30FF6000 - 30FFFFFC	***Unused***		***Available***

3.1 Revision Register (0x30FF0000) R

The Revision register contains the current revision number of the source code.

Updated: 09/17/2002
Revision:

3.2 Test Registers (0x30FF0004 and 0x30FF0008) R/W

Test Registers for device sanity check and debug.

3.3 Write Memory Start Address(wr_mem_startaddr) (0x30FF0100) R/W

The wr_mem_startaddr register is a 32-bit register used to indicate the DDR SDRAM start address during write burst tests. Values range from 0x00000000 to 0x07FFFFF0. The user always provides the memory start address for write memory test.

3.4 Write Memory Length (wr_mem_length) (0x30FF0104) R/W

The write length register contains the length of data transfer in bytes during write burst tests. Values range from 0x10 to 0x08000000. The user always provides the length for a write memory test.

3.5 Write Communication Port(wr_comm_port) (0x30FF0108) R/W

The write communication port is used by the MicroBlaze and the Command Control Unit to indicate the start and end of a write test. During a write test, the MicroBlaze writes the test data into the write buffer memory. It initializes the wr_mem_startaddr and wr_mem_length registers with the user provided start address and length data and sets the write communication port. The Command Control Unit, which constantly monitors the port, recognizes the write test command from the MicroBlaze and generates the appropriate commands to the DDR Controller. At the end of the test, the Command Control Unit clears the write communication port to indicate the successful completion of the test.

3.6 Write Buffer Start Address (wr_buffer_startaddr) (0x30FF010C) R/W

The write buffer start address register is initialized by the MicroBlaze during a write test. It contains the starting address of the write buffer memory from where the Command Control Unit starts retrieving data which is written into the DDR SDRAM memory.

3.7 Read Memory Start Address(rd_mem_startaddr) (0x30FF0200) R/W

Perfroms identical functions as the wr_mem_startaddr during read burst tests.

3.8 Read Memory Length (rd_mem_length) (0x30FF0204) R/W

Performs identical functions as the wr_mem_length register during read tests.

3.9 Read Communication Port(rd_comm_port) (0x30FF0208) R/W

Performs identical functions as the wr_comm._port register during read tests.

3.10 Read Buffer Start Address (rd_buffer_startaddr) (0x30FF020C) R/W

Performs identical functions as the wr_buffer_startaddr register during read tests.

4.0 MicroBlaze System

The MicroBlaze System is configured in the executable mode, hence the programmed FPGA should come up and ready for testing. A HyperTerminal is needed to run the tests. Before programming the FPGA, bring up the HyperTerminal and configure as follows:

HyperTerminal Settings

Port settings in File ⇒ Properties ⇒ configure:

Bits per second	=	19200
Data Bits	=	8
Parity	=	None
Stop Bits	=	1
Flow Control	=	Hardware

ASCII Setup in File ⇒ Properties ⇒ Settings :

√ - append line feeds
√ - wrap lines that exceed

When the device is programmed, the AVNET LOGO will appear on the HyperTerminal screen followed by [RLDRAM TEST >] prompt. Any of the commands listed below can now be executed.

5.0 Test Commands

There are two sets of commands which are used to access MicroBlaze address space and DDR SDRAM address space.

5.1 MicroBlaze Test Commands

These commands – `read`, `write`, `flush_read_buffer`, `cls` and `quit` – do not use RLDRAM addresses.

5.1.1 Read and write commands

The `read` and `write` commands directly access any portion of the MicroBlaze 32-bit address space. This includes FPGA block RAM, RLDRAM I/O buffers, RLDRAM registers, etc. Reads and writes are 32 bits wide.

All numbers are assumed to be in hex, always, whether or not prefixed with "0x".

Usage:

```
read {addr} [number of bytes]
```

If omitted, the number of bytes defaults to 4, that is, a single 32-bit read.

```
write {addr}{one or more values}
```

Examples:

- `read 0`
- `read 0 100 (this reads 256 bytes since 100 hex is 256 decimal)`
- `write 0 aa bbb ccccc 01234567`

5.1.2 flush_read_buffer

Usage:

`flush_read_buffer [fill value]`
Flushes the FPGA's 8K read buffer. If fill value is omitted, zero is used.

Examples:

```
flush_read_buffer
flush_read_buffer eeeeeeee
```

5.1.3 cls

Usage:

`cls`
Clears the screen.

5.1.4 quit

Usage:

`quit`
Returns control to XMD, if it is running. Otherwise, causes the program to stop functioning.

5.2 DDR SDRAM Test Commands

Unlike the `read` and `write` commands, these command take RLDRAM addresses and implicitly manage the RLDRAM I/O buffers for you.

Notes:

- All numbers are assumed to be in hex, always, whether or not prefixed with `0x`.
- The `v` argument to any command increases its verbosity. Or, use `f` for FPGA-only verbosity, or `q` for extra quiet.
- Commands may be aborted at any time by pressing the escape key.
- Since the RLDRAM is 128MB, addresses range from 0x00000000 to 0x07ffffff; number of bytes ranges from 0x00000004 to 0x08000000. Any start address plus length should not exceed 0x08000000. (The software does not check this for you; it allows too-large address ranges to wrap around to 0.)

5.2.1 peek

Usage:

```
peek   {RLDRAM addr} {#bytes} [v]
```

Note: The number of bytes will always be rounded up to a multiple of 16.

Examples:

```
peek 0x100000 0x200
peek 0x100000 0x200 v
peek 0x100000 0x200 f
```

5.2.2 fill

Usage:

```
fill   {RLDRAM addr} {#bytes} val     [v]
fill   {RLDRAM addr} {#bytes} rand    [v]
fill   {RLDRAM addr} {#bytes} addr    [v]
fill   {RLDRAM addr} {#bytes} revaddr [v]
```

The `val`, `rand`,`addr` and `revaddr` options are the same as for the `test` command, described below.

Examples:

```
fill   0x100000 0x200  0xabababab
fill   0x100000 0x200  rand
```

5.2.3 verify

Usage:

```
verify {RLDRAM addr} {#bytes} val     [v]
verify {RLDRAM addr} {#bytes} rand    [v]
verify {RLDRAM addr} {#bytes} addr    [v]
verify {RLDRAM addr} {#bytes} revaddr [v]
```

The `val`, `rand`,`addr` and `revaddr` options are the same as for the `test` command, described below.

Example:

```
verify 0x100000 0x200  0xabababab
verify 0x100000 0x200  0xabababab f
verify 0x100000 0x200  0xabababab q
```

5.2.4 test

Usage:

```
test   {RLDRAM addr} {#bytes} [tests or v]
```

Tests may be `00`, `ff`, `aa`, `55`, `addr`, `revaddr` or `rand`. If no tests are specified, all of them are done. Type the escape to abandon all tests; type `s` to skip the current one.

Details about the tests:

Updated: 09/17/2002
Revision:

- The `00` test fills each 32-bit word in the specified address range with 0x00000000, then verifies this.
- The `ff` test fills each 32-bit word in the specified address range with 0xffffffff, then verifies this.
- The `aa` test fills each 32-bit word in the specified address range with 0xaaaaaaaa, then verifies this.
- The `55` test fills each 32-bit word in the specified address range with 0x55555555, then verifies this.
- The `addr` test fills each 32-bit word in the specified address range with its own address, then verifies this.
- The `revaddr` test fills each 32-bit word in the specified address range with its own byte-swapped address, then verifies this. E.g. address 0x01234560 would contain the value 0x60452312.
- The `rand` test fills each 32-bit word in the specified address range with a pseudo-random value, then reseeds the generator with the same starting value and verifies the sequence. The pseudo-random number generator is a simple linear congruence.

Examples:

```
test 0x100000 0x200 q
test 0x100000 0x200 rand
test 0x100000 0x200 rand addr v
```

Figure 2-1

DDR SDRAM Test Block Diagram