**Technion**

*Electrical Engineering Department*

High Speed Digital System Lab

Memory Management

Documentation

**Version**: 1.0

**Last update**: 28.04.2011

**Written By**: Beeri Schreiber and Alon Yavich

Table of content

[Table of Changes 3](#_Toc295274758)

[Scope 3](#_Toc295274759)

[Abbreviations 3](#_Toc295274760)

[General Description 3](#_Toc295274761)

[Memory Management Pinout 4](#_Toc295274762)

[Memory Management Generic Parameters 5](#_Toc295274763)

[Mem\_Ctrl\_Wr Pinout 6](#_Toc295274764)

[Mem\_Ctrl\_Wr Generic Parameters 7](#_Toc295274765)

[Mem\_Ctrl\_Rd Pinout 7](#_Toc295274766)

[Mem\_Ctrl\_Rd Generic Parameters 8](#_Toc295274767)

[Arbiter Pinout 9](#_Toc295274768)

[Arbiter Generic Parameters 10](#_Toc295274769)

[Resources 10](#_Toc295274770)

[Memory Management Waves 11](#_Toc295274771)

[Mem\_Ctrl\_Wr Waves 11](#_Toc295274772)

[Burst 11](#_Toc295274773)

[Summary Chunk 12](#_Toc295274774)

[Debug Mode 13](#_Toc295274775)

[Cycle Split 14](#_Toc295274776)

[Mem\_Ctrl\_Wr Waves 15](#_Toc295274777)

[Burst 15](#_Toc295274778)

[Restart from initial address 16](#_Toc295274779)

[Cycle Split 17](#_Toc295274780)

[Arbiter Waves 18](#_Toc295274781)

[Performed Tests for Memory Management 19](#_Toc295274782)

[Performed Test for Mem\_Ctrl\_Wr 19](#_Toc295274783)

[Performed Test for Mem\_Ctrl\_Rd 19](#_Toc295274784)

Table of Tables

[Table 1 – Table of Changes 3](#_Toc295194250)

[Table 2 – Memory Management Pinout 5](#_Toc295194251)

[Table 3 – Memory Management Parameters 5](#_Toc295194252)

[Table 4 – mem\_ctrl\_wr Pinout 6](#_Toc295194253)

[Table 5 – mem\_ctrl\_wr Parameters 7](#_Toc295194254)

[Table 6 – mem\_ctrl\_rd Pinout 8](#_Toc295194255)

[Table 7 – Memory Management Parameters 8](#_Toc295194256)

[Table 8 – Arbiter Pinout 10](#_Toc295194257)

[Table 9 – Arbiter Generic Parameters 10](#_Toc295194258)

Tables of Figures

[Figure 1 - General Scheme 3](#_Toc295194259)

[Figure 2 – Memory Management Pinout 4](#_Toc295194260)

[Figure 3 – mem\_ctrl\_wr Pinout 6](#_Toc295194261)

[Figure 4 – mem\_ctrl\_rd Pinout 7](#_Toc295194262)

[Figure 5 – Arbiter Pinout 9](#_Toc295194263)

[Figure 6 – Mem\_Ctrl\_Wr Burst 11](#_Toc295194264)

[Figure 7 – Mem\_Ctrl\_Wr Summary Chunk 12](#_Toc295194265)

[Figure 8 – Mem\_Ctrl\_Wr Debug Mode 13](#_Toc295194266)

[Figure 9 – Mem\_Ctrl\_Wr Cycle Split 14](#_Toc295194267)

[Figure 10 – Mem\_Ctrl\_Rd Burst 15](#_Toc295194268)

[Figure 11 – Mem\_Ctrl\_Rd Burst 16](#_Toc295194269)

[Figure 12 – Mem\_Ctrl\_Rd Cycle Split 17](#_Toc295194270)

[Figure 13 – Arbiter 18](#_Toc295194271)

# Table of Changes

|  |  |  |
| --- | --- | --- |
| Version | Date of Change | Description |
| 1.0 | 28.04.2011 | Creation of documentation |

Table 1 – Table of Changes

# Scope

This document aims to describe the working method of the Memory Management Block.

# Abbreviations

1. SDRAM – Synchronous Dynamic Random Access Memory
2. RAS – Row Address
3. CAS – Column Address
4. WBM – Wishbone Master
5. WBS – Wishbone Slave
6. TB – Test bench

# General Description

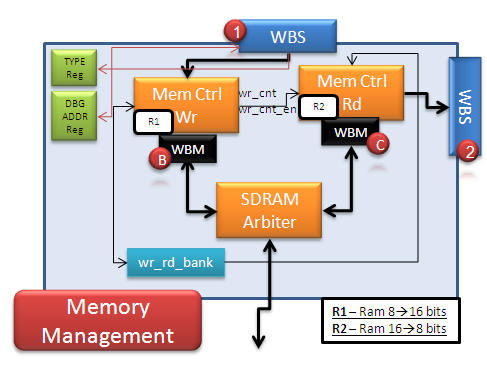


Figure 1 - General Scheme

The Memory Management block manages the write and read transaction to and from the SDRAM, through the SDRAM Controller.

The Memory Management block is consumed of the following components:

1. **Mem\_Ctrl\_Wr** – Responsible to receive data, through WBS, and store it, using SDRAM controller, to the SDRAM.  
   Data (up to 1KByte) is received from WBS (1) and stored into internal RAM (R1). As soon as half of the burst has been transmitted (Input RAM is 8 bits, while input SDRAM is 16 bits, therefore RAM transaction and SDRAM transaction will end at about the same time), SDRAM transaction is initialized, which stores the data from the internal RAM to the SDRAM, through the SDRAM Controller.
2. **Mem\_Ctrl\_Rd** – Responsible to transmit data, using SDRAM controller, to the data requester, through WBS.As soon as some words (about 10) are written to the internal RAM (R2) from the SDRAM, the WBS (2) negates STALL\_O and transmits data from the internal RAM.
3. **Arbiter** – Toggles between *Mem\_Ctrl\_Wr* WBM (B) and *Mem\_Ctrl\_Rd* WBM (C), where the *Mem\_Ctrl\_Wr* has higher priority, to prevent data loss.
4. **Wr\_rd\_bank** – Holds the MSB address bit of the SDRAM write / read address, which is actually the double bank. While mem*\_ctrl\_wr* writes to bank '**1**0', *mem\_ctrl\_rd* reads from bank '**0**0'.
5. **Registers** – Hold necessary data for above blocks:
   1. **Type Register** –Type of message (Address 1h).
   2. **Debug Address Register** – Write / Read SDRAM address for SDRAM in Debug mode (Address 2h🡪4h).

# Memory Management Pinout



Figure 2 – Memory Management Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Clk\_i | In | Wishbone Clock (133MHz) |
| Rst | In | Reset |
| Wr\_wbs\_adr\_i[9..0] | In | WBS Address (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_tga\_i[9..0] | In | WBS Tag Address – Burst length (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_dat\_i[7..0] | In | WBS Input Data (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_cyc\_i | In | WBS Cycle (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_stb\_i | In | WBS Strobe (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_we\_i | In | WBS Write Enable (to *mem\_ctrl\_wr*) |
| Wr\_wbs\_tgc\_i | In | WBS Tag Cycle – Write to components ('0') or registers ('1') (to *mem\_ctrl\_wr*) |
| Rd\_wbs\_adr\_i[9..0] | In | WBS Address (to *mem\_ctrl\_rd*) |
| Rd\_wbs\_tga\_i[9..0] | In | WBS Tag Address – Burst length (to *mem\_ctrl\_rd*) |
| Rd\_wbs\_cyc\_i | In | WBS Cycle (to *mem\_ctrl\_rd* |
| Rd\_wbs\_stb\_i | In | WBS Strobe (to *mem\_ctrl\_rd*) |
| Rd\_wbs\_tgc\_i | In | WBS Tag Cycle – Write to components ('0') or registers ('1') (to *mem\_ctrl\_rd*) |
| Wbm\_dat\_i[15..0] | In | WBM Input Data (From SDRAM Controller) |
| Wbm\_stall\_i | In | WBM Stall – Repeat last strobe (From SDRAM Controller) |
| Wbm\_err\_i | In | WBM Error (From SDRAM Controller) |
| Wbm\_ack\_i | In | WBM Acknowledged (From SDRAM Controller) |
| Wr\_wbs\_dat\_o[7..0] | Out | WBS Output Data (Registers' value) |
| Wr\_wbs\_stall\_o | Out | WBS Stall – Repeat last strobe (Payload / Registers) |
| Wr\_wbs\_ack\_o | Out | WBS Acknowledged (Payload / Registers) |
| Wr\_wbs\_err\_o | Out | WBS Error (From *mem\_ctrl\_wr*) |
| Rd\_wbs\_dat\_o[7..0] | Out | WBS Output Data, from SDRAM (From *mem\_ctrl\_rd*) |
| Rd\_wbs\_stall\_o | Out | WBS Stall (From *mem\_ctrl\_rd*) |
| Rd\_wbs\_ack\_o | Out | WBS Acknowledged (From *mem\_ctrl\_rd*) |
| Rd\_wbs\_err\_o | Out | WBS Error (From *mem\_ctrl\_rd*) |
| Wbm\_adr\_o[21..0] | Out | WBM Address to SDRAM Controller |
| Wbm\_dat\_o[15..0] | Out | WBM Data to SDRAM Controller |
| Wbm\_we\_o | Out | WBM Write Enable to SDRAM Controller |
| Wbm\_tga\_o[7..0] | Out | WBM Tag Address – Burst length to SDRAM Controller |
| wbm\_cyc\_o | Out | WBM Cycle to SDRAM Controller |
| Wbm\_stb\_o | Out | WBM Strobe to SDRAM Controller |

Table 2 – Memory Management Pinout

# Memory Management Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |
| mode\_g | 0 | Relevant bit in type register, which represent Normal ('0') or Debug ('1') mode |
| message\_g | 1 | Relevant bit in type register, which represent Image chunk ('0') or Summary chunk ('1') mode |
| img\_hor\_pixels\_g | 640 | Horizontal Pixels in image |
| img\_ver\_lines\_g | 480 | Vertical lines in image |

Table 3 – Memory Management Parameters

# Mem\_Ctrl\_Wr Pinout



Figure 3 – mem\_ctrl\_wr Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Clk\_i | In | Wishbone Clock (133MHz) |
| Rst | In | Reset |
| wbs\_adr\_i[9..0] | In | WBS Address |
| wbs\_tga\_i[9..0] | In | WBS Tag Address – Burst length |
| wbs\_dat\_i[7..0] | In | WBS Input Data |
| wbs\_cyc\_i | In | WBS Cycle |
| wbs\_stb\_i | In | WBS Strobe |
| Wbm\_stall\_i | In | WBM Stall – Repeat last strobe (From SDRAM Controller) |
| Wbm\_err\_i | In | WBM Error (From SDRAM Controller) |
| Wbm\_ack\_i | In | WBM Acknowledged (From SDRAM Controller) |
| Arbiter\_gnt | In | Grant on SDRAM Controller from Arbiter |
| Bank\_val | In | SDRAM's bank (1 / 0) |
| Type\_reg[7..0] | In | Type Register value |
| Wr\_addr\_reg | In | Debug address register value |
| wbs\_stall\_o | Out | WBS Stall – Repeat last strobe |
| wbs\_ack\_o | Out | WBS Acknowledged |
| wbs\_err\_o | Out | WBS Error |
| Wbm\_adr\_o[21..0] | Out | WBM Address to SDRAM Controller |
| Wbm\_dat\_o[15..0] | Out | WBM Data to SDRAM Controller |
| Wbm\_we\_o | Out | WBM Write Enable to SDRAM Controller |
| Wbm\_tga\_o[7..0] | Out | WBM Tag Address – Burst length to SDRAM Controller |
| wbm\_cyc\_o | Out | WBM Cycle to SDRAM Controller |
| Wbm\_stb\_o | Out | WBM Strobe to SDRAM Controller |
| Arbiter\_req | Out | Request for grant on SDRAM Controller from Arbiter |
| Bank\_switch | Out | '1' to switch between SDRAM banks (bank\_val) |
| Wr\_cnt\_val[18..0] | Out | Number of words, written so far to SDRAM |
| Wr\_cnt\_en | Out | All image has been transmitted to SDRAM, and wr\_cnt\_val value is valid |

Table 4 – mem\_ctrl\_wr Pinout

# Mem\_Ctrl\_Wr Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |
| mode\_g | 0 | Relevant bit in type register, which represent Normal ('0') or Debug ('1') mode |
| message\_g | 1 | Relevant bit in type register, which represent Image chunk ('0') or Summary chunk ('1') mode |
| img\_hor\_pixels\_g | 640 | Horizontal Pixels in image |
| img\_ver\_lines\_g | 480 | Vertical lines in image |

Table 5 – mem\_ctrl\_wr Parameters

# Mem\_Ctrl\_Rd Pinout



Figure 4 – mem\_ctrl\_rd Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Clk\_i | In | Wishbone Clock (133MHz) |
| Rst | In | Reset |
| wbs\_adr\_i[9..0] | In | WBS Address |
| wbs\_tga\_i[9..0] | In | WBS Tag Address – Burst length |
| wbs\_cyc\_i | In | WBS Cycle |
| wbs\_tgc\_i | In | WBS Tag Cycle. '1' to start reading from the first address in the chosen bank (according to bank\_val) |
| wbs\_stb\_i | In | WBS Strobe |
| Wbm\_dat\_i[15..0] | In | WBM Data from SDRAM Controller |
| Wbm\_stall\_i | In | WBM Stall – Repeat last strobe (From SDRAM Controller) |
| Wbm\_err\_i | In | WBM Error (From SDRAM Controller) |
| Wbm\_ack\_i | In | WBM Acknowledged (From SDRAM Controller) |
| Arbiter\_gnt | In | Grant on SDRAM Controller from Arbiter |
| Bank\_val | In | SDRAM's bank (1 / 0) |
| Type\_reg[7..0] | In | Type Register value |
| rd\_addr\_reg | In | Debug address register value |
| rd\_cnt\_val[18..0] | In | Number of words, written to SDRAM |
| rd\_cnt\_en | In | All image has been transmitted to SDRAM, and wr\_cnt\_val value is valid |
| wbs\_dat\_o[7..0] | In | WBS Output Data |
| wbs\_stall\_o | Out | WBS Stall – Repeat last strobe |
| wbs\_ack\_o | Out | WBS Acknowledged |
| wbs\_err\_o | Out | WBS Error |
| Wbm\_adr\_o[21..0] | Out | WBM Address to SDRAM Controller |
| Wbm\_we\_o | Out | WBM Write Enable to SDRAM Controller |
| Wbm\_tga\_o[7..0] | Out | WBM Tag Address – Burst length to SDRAM Controller |
| wbm\_cyc\_o | Out | WBM Cycle to SDRAM Controller |
| Wbm\_stb\_o | Out | WBM Strobe to SDRAM Controller |
| Arbiter\_req | Out | Request for grant on SDRAM Controller from Arbiter |

Table 6 – mem\_ctrl\_rd Pinout

# Mem\_Ctrl\_Rd Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |
| mode\_g | 0 | Relevant bit in type register, which represent Normal ('0') or Debug ('1') mode |
| img\_hor\_pixels\_g | 640 | Horizontal Pixels in image |
| img\_ver\_lines\_g | 480 | Vertical lines in image |

Table 7 – Memory Management Parameters

# Arbiter Pinout



Figure 5 – Arbiter Pinout

| Pin Name | Direction | Description |
| --- | --- | --- |
| Clk | In | Clock (133MHz) |
| Rst | In | Reset |
| Wr\_req | In | Request for grant on SDRAM controller from mem\_ctrl\_wr |
| Rd\_req | In | Request for grant on SDRAM controller from mem\_ctrl\_rd |
| Wr\_wbm\_adr\_o[21..0] | In | WBM Address to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Wr\_wbm\_dat\_o[15..0] | In | WBM Data to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Wr\_wbm\_we\_o | In | WBM Write Enable to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Wr\_wbm\_tga\_o[7..0] | In | WBM Tag Address – Burst length to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Wr\_wbm\_cyc\_o | In | WBM Cycle to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Wr\_wbm\_stb\_o | In | WBM Strobe to SDRAM Controller (From *mem\_ctrl\_wr*) |
| Rd\_wbm\_adr\_o[21..0] | In | WBM Address to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Rd\_wbm\_dat\_o[15..0] | In | WBM Data to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Rd\_wbm\_we\_o | In | WBM Write Enable to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Rd\_wbm\_tga\_o[7..0] | In | WBM Tag Address – Burst length to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Rd\_wbm\_cyc\_o | In | WBM Cycle to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Rd\_wbm\_stb\_o | In | WBM Strobe to SDRAM Controller (From *mem\_ctrl\_rd*) |
| Wbm\_dat\_i[15..0] | In | WBM Input data from SDRAM Controller |
| Wbm\_stall\_i | In | WBM Stall from SDRAM Controller – Repeat last transaction |
| Wr\_gnt | Out | Grant on SDRAM Controller to mem*\_ctrl\_wr* |
| Rd\_gnt | Out | Grant on SDRAM Controller to mem*\_ctrl\_rd* |
| Wr\_wbm\_stall\_i | Out | WBM Stall from SDRAM Controller to *mem\_ctrl\_wr* |
| Wr\_wbm\_err\_i | Out | WBM error from SDRAM Controller to *mem\_ctrl\_wr* |
| Wr\_wbm\_ack\_i | Out | WBM acknowledged from SDRAM Controller to *mem\_ctrl\_wr* |
| Rd\_wbm\_dat\_i[15..0] | Out | Output data from SDRAM Controller to *mem\_ctrl\_rd* |
| Rd\_wbm\_stall\_i | Out | WBM Stall from SDRAM Controller to *mem\_ctrl\_rd* |
| Rd\_wbm\_err\_i | Out | WBM error from SDRAM Controller to *mem\_ctrl\_rd* |
| Rd\_wbm\_ack\_i | Out | WBM acknowledged from SDRAM Controller to *mem\_ctrl\_rd* |
| Wbm\_adr\_o[21..0] | Out | WBM Write address to SDRAM Controller |
| Wbm\_we\_o | Out | WBM Write Enable to SDRAM Controller |
| Wbm\_tga\_o[7..0] | Out | WBM Tag Address – Burst length to SDRAM Controller |
| wbm\_cyc\_o | Out | WBM Cycle to SDRAM Controller |
| Wbm\_stb\_o | Out | WBM Strobe to SDRAM Controller |
| Wbm\_dat\_o[15..0] | Out | WBM Output data to SDRAM Controller |

Table 8 – Arbiter Pinout

# Arbiter Generic Parameters

| Generic Parameter | Default Value | Description |
| --- | --- | --- |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |

Table 9 – Arbiter Generic Parameters

# Resources

Required resources, when synthesizing, using Synplify, for Altera's Cyclon II FPGA:

* 1162 Combinational Logic
* 32 EC (Embedded Cell: M4K, which is the RAM)
* 549 FF

**Maximum frequency**:

148.4 MHz

# Memory Management Waves

## Mem\_Ctrl\_Wr Waves

### Burst

1. 1030 bytes burst to SDRAM is split, by Data Provider, to 1024 bytes burst, and then 6 bytes burst.
2. Internal logic writes the data to the SDRAM in 256 words bursts.
3. Summary chunk, with total transmitted bytes (1030) is compared with total stored words \* 2 to SDRAM (1030 = 515 \* 2)
4. SDRAM Banks are switched ('0' to '1')

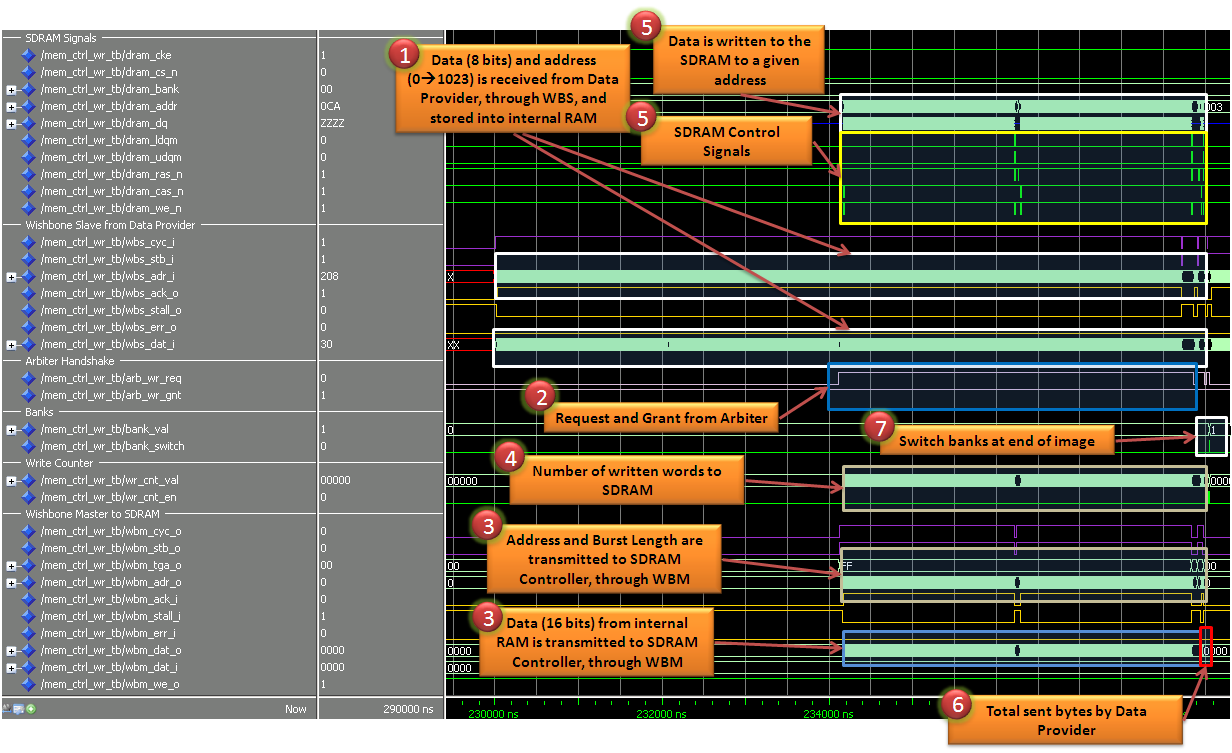


Figure 6 – Mem\_Ctrl\_Wr Burst

### Summary Chunk

1. Type register, bit number 1, is changed to '1', to indicate that summary chunk is being transmitted.
2. Received data is calculated, and compared with 2 \* total sent words to SDRAM.
3. The two values are equal 🡪 Bank switch is executed.

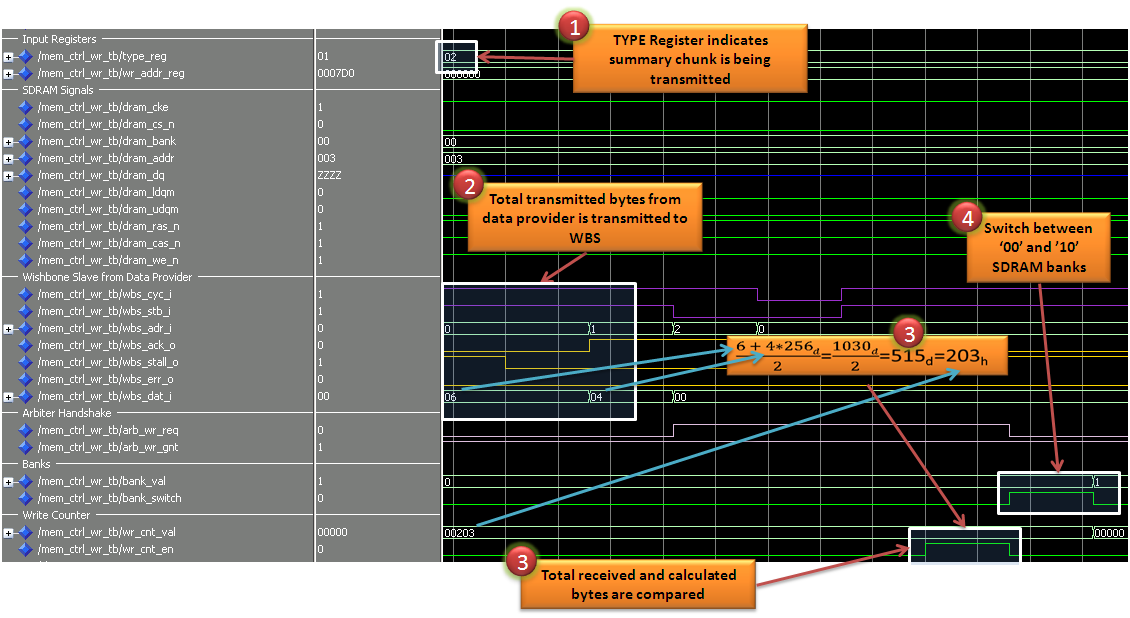


Figure 7 – Mem\_Ctrl\_Wr Summary Chunk

### Debug Mode

1. Type register, bit number 0 is set to '1', to indicate on Debug Mode.
2. Write Address register is set to the required start address, to write to the SDRAM from.
3. WBS received data to internal RAM.
4. Transaction is being initialized, to store the data from the internal RAM to the SDRAM.

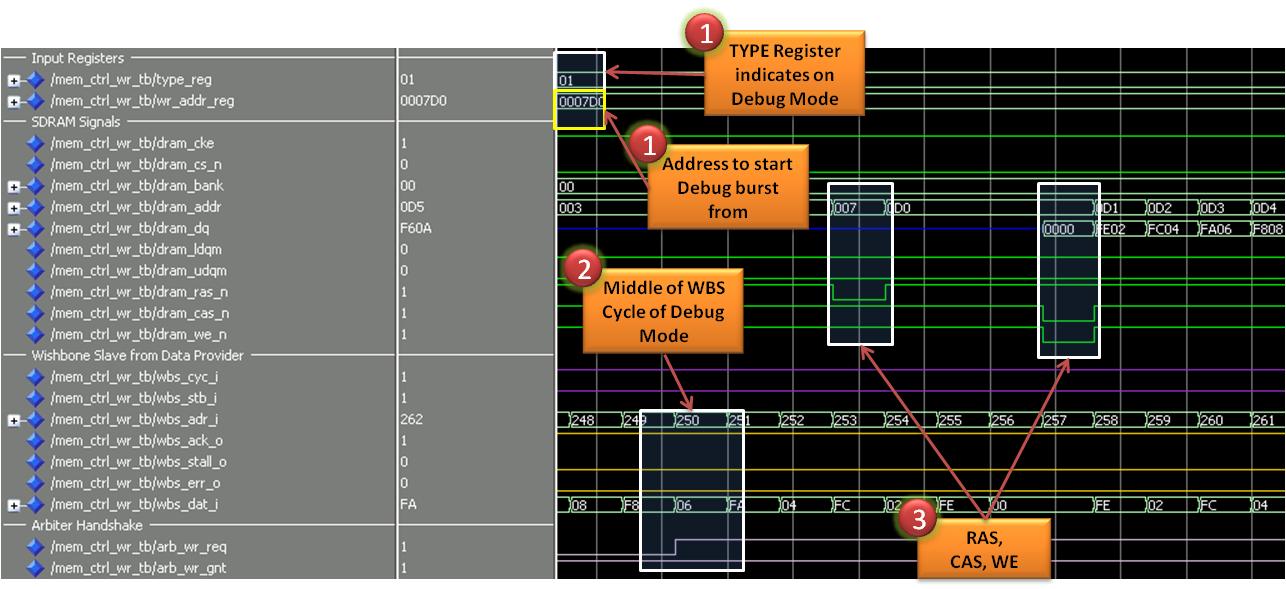


Figure 8 – Mem\_Ctrl\_Wr Debug Mode

### Cycle Split

1. End of SDRAM Page has been detected (End of burst to SDRAM by SDRAM controller).
2. WBM closes the cycle, to split burst, and re-initialize it from where it stops.
3. New SDRAM burst is being executed, from the new page.

### 

Figure 9 – Mem\_Ctrl\_Wr Cycle Split

## Mem\_Ctrl\_Wr Waves

### Burst

1. Data Requester initializes Wishbone Read Cycle
2. End of Write Cycle of *mem\_ctrl\_rd* causes to banks switch
3. Request for grant from Arbiter on SDRAM, and Arbiter grants
4. Data from SDRAM is read, and stored into internal 1KB RAM
5. Since the burst is 1KB, and SDRAM page is 512Byte, the SDRAM burst is split to two burst
6. All data has been acknowledged by data requester, which closes the burst

Figure 10 – Mem\_Ctrl\_Rd Burst

### Restart from initial address

When an image should be presented on the screen, the data should be read from the beginning of the SDRAM bank, and not from the current point. Therefore, the Data Requester can assert WBM\_TGC\_O, no indicate that SDRAM should provide data from the first bank's address.

1. Cycle Tag is received, therefore current address in SDRAM's read bank is changed to 0h.
2. 1KB of data is read.

Figure 11 – Mem\_Ctrl\_Rd Burst

### Cycle Split

Since SDRAM page is 256 words (=512Bytes), 1KB burst must be split to two burst.

1. Precharge, to end read transaction
2. End and Start of SDRAM transaction
3. RAS, CAS and WE, to start new SDRAM transaction

Figure 12 – Mem\_Ctrl\_Rd Cycle Split

## Arbiter Waves

1. When no instantiation request for grant on SDRAM, *Mem\_Ctrl\_Wr* grants the control on the SDRAM.
2. *Mem\_Ctrl\_Rd* grants the control on the SDRAM, when it request for the grant, and as soon as the request from *Mem\_Ctrl\_Wr* is being negated.

## 

Figure 13 – Arbiter

# Performed Tests for Memory Management

## Performed Test for Mem\_Ctrl\_Wr

1. 2 bytes burst
2. 4 bytes burst
3. 6 bytes burst
4. 1024 bytes burst
5. 1026 bytes burst
6. 1030 bytes burst
7. Two 1030 bytes bursts
8. Two burst, with 1 clock cycle separates between them.
9. 500 bytes burst, in Debug mode, from address 2000decimal = 7D0hex
10. Summary chunk after 1030 burst, to switch banks
11. Summary chunk after two 1030 bursts, to switch banks
12. Error injections:
    1. Incorrect summary chunk after 1030 burst. Bank has not been switched
    2. Error from SDRAM controller (WBM\_ERR\_I) caused end of cycle
    3. Incorrect address to WBS (expecting sequential addressing, starting from 0) causes WBS\_ERR\_O signal to rise

## Performed Test for Mem\_Ctrl\_Rd

1. 2 bytes burst
2. 4 bytes burst
3. 6 bytes burst
4. 1024 bytes burst
5. 1026 bytes burst
6. 1030 bytes burst
7. Two 1030 bytes bursts
8. Two burst, with 1 clock cycle separates between them.
9. 500 bytes burst, in Debug mode, from address 2000decimal = 7D0hex
10. Error injections:
    1. Error from SDRAM controller (WBM\_ERR\_I) caused end of cycle
    2. Incorrect address to WBS (expecting sequential addressing, starting from 0) – Waits for the next correct address.