Pre-lab [30 marks]

1. 64M = 64 \* 220 bytes

Convert to hex: 1048576 = 64 \* 165 = 0x0400 0000  
So the highest byte address of the SRDRAM chip is **0x04000 000**

1. Since the word size used in this lab is **16-bit, or 2-byte**. Thus, the addresses seen by the chip will only be multiples of 2-byte word type, or even multiples of types. For example, when requesting data from the lowest byte in the memory, the address seen by the chip should be 0x0000 0000, as shown below, then 0x0000 0002, 0x0000 0004

|  |  |
| --- | --- |
| 0x0000 0000 | 🡨 First Word |
| 0x0000 0001 |  |
| 0x0000 0002 | 🡨 Second Word |
| 0x0000 0003 |  |
| …… | …… |
| 0x03FF FFFE |  |
| 0x03FF FFFF | 🡨 Last Word |
| 0x0400 0000 |  |

and so on. When requesting data from the highest byte in the memory, the address seen by the chip should be the original address of the second last byte in the memory, which is **0x03ffffff**.

1. The address appearing on the SDRAM address line when the second byte is accessed will be the combined address of the first word, or the first and second byte, in the form of

**FIRST-WORD [15:0] 🡸 SECOND-BYTE [7:0]:FIRST-BYTE [7:0]**

which makes the address **0x0000 0000,** pointing to the first word in the SDRAM.

1. schematic of the circuit that is needed to implement SDRAM Controller.v.  
   Diagram

   Description automatically generated

Diagram, schematic

Description automatically generated

Lab Report (Part 2)[15 marks]

1. the screenshots in the “Develop Software” part are provided below.

Graphical user interface, text, application

Description automatically generated

1. Writing into memory takes less clock cycles than reading because the writing process is conducted by the registers into main memory, where register operated with a larger bandwidth, while reading takes data from the main memory with a smaller bandwidth, which makes it slower.
2. Because it’s writing multiple values at the same time into the memory.
3. Because the data width for the SDRAM is 16-bit, whereas the C compiler uses data of 8-bit width, thus the Avalon interconnect takes two data location at the same time and combine it into one 16-bit location.
4. The steps the Avalon interconnect takes to write a 32-bit integer into the 16-bit SDRAM memory are:
   1. The compiler takes the 32 bit integer and splits it up into an upper and lower half word
   2. The compiler then writes each word separately
   3. Once it has been sent, the words are joined on the memory itself
5. On the compilation report,
6. Total number of logic elements used: 1847/32070(6%)
7. Total number of memory bits used: 1126376/4065280(28%)
8. Total number of pins used: 57/457(12%)
9. The maximum number of logic elements that can fit in the FGPA: 32070
10. Considering just the maximum number of logic elements on the FPGA, approximately

32070/1847 ≅ 17.36 ≅ 17 SOPC systems like this could fit on the FPGA.

1. Considering just the maximum amount of memory on the FPGA, approximately

4065280/1126376 ≅ 3.57 ≅ 3 SOPC systems like this could fit on the FPGA.

1. Considering just the maximum number of pins on the FPGA, approximately

457/57 ≅ 8..02 ≅ 8 SOPC systems like this could fit on the FPGA.

Compilation Result

Graphical user interface, text, application, email

Description automatically generated