CSE4358/CSE5358 Microprocessor Systems
Summer 2023
SDRAM Controller Design

1 Overview

The goal of this project is to design an SDRAM controller that allows SDRAM memory to be interfaced with a microprocessor having only asynchronous memory support. There is no requirement to build the hardware, but a complete written report containing schematics and theory of operation is required.

2 Detailed Requirements

The SDRAM controller should be designed to interface with one or more SDRAM memory devices. The minimum requirement is that support for a single MT48LC8M8A2 be provided. It is strongly suggested that you attempt to provide interfacing with the 80386DX processor, but you may interface with another processor with a 32-bit asynchronous memory data interface provided.

The detailed requirements follow:

1. Support the MT48LC8M8A2 memory device is required.
2. Support for a 80386DX or similar processor with 32-bit data bus not having SDRAM support is required.
3. The use of “northbridge/MCH/IMCH” chipsets, SDRAM controller (including reference design or commercial FPGA/CPLD/ASIC/hard copy solutions is not allowed).
4. A complete controller solution including state machine, row, column, and bank signal generation, data masking, data flow, ready logic, and refresh support must be provided.
5. Interfacing with ~ADS, W/~R, and M/~IO control signals (or equivalent) will be required.
6. Support the sending of AUTO REFRESH commands at a rate sufficient to ensure memory integrity.
7. Support the READY logic of the selected microprocessor system, especially when refresh is occurring, unless a bus locking protocol is used.
8. Support is only required for burst length 4 transfers, although you may add different burst length support for DMA operations if you wish for extra credit or support different bit-width memories with different burst lengths as an option.
9. Extra credit steps:
    a. Adding hardware external I/O control pins to support variable timing, with changes in both the FSM timing and the LMR values being loaded.
    b. Adding hardware external I/O to support 4- and 16-bit wide memories with burst lengths of 8 and 2 appropriately with all changes needed to the FSM and SDRAM interface
    c. Steps to minimize microprocessor waiting due to refresh will be worth more credit.

The report should include:

1. A detailed theory of operation for your design.
2. A top-level pinning of the SDRAM controller showing all inputs and outputs to the CPU, decoder logic, and SDRAM memory. This should be placed in a Verilog module with ports representing all the pins.
3. A detailed set of schematics for the complete design.
4. A complete FSM including initialization, refresh, reading, and writing.
5. A state transition table
6. Equations for every output as a function of the state and other parameters.
7. In lieu of steps 4-6, a Verilog sdram_top design that implements the FSM with inputs and outputs similar to that used in the DMA example in class.
8. Timing diagrams

A few hints to mention again:

1. Your design should not need any counters except for the refresh and initialization 100us period timers. The implicit state machine counter logic need not be specified in the design other than the state transition table.
2. Your design should only need a latch for reading the bursts from SDRAM. If pipelining is supported, then the address should also be latched as it changed from T1p to T2p. A latch may also be needed for the refresh request timer.
3. You should generate all SDRAM control signals directly as a function of the state.
4. Ready generation should be a function of the state.

3 Deadlines

The project is due at the time indicated in the syllabus. Oral defenses will be given during class time during the time indicated in the syllabus. Teams will consist of no more than 2 members (1 member for CSE 6351 students). All members of the team participate equally.

Have fun!