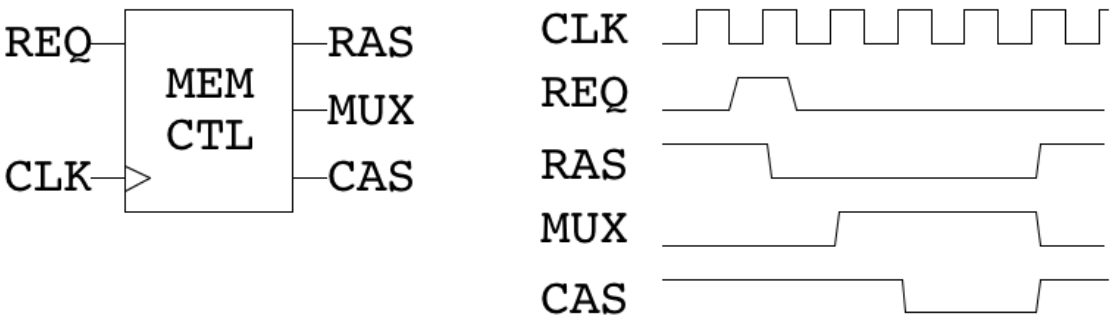
M A S S A C H U S E T T S I N S T I T U T E O F T E C H N O L O G Y
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.111 Introductory Digital Systems Laboratory
Fall 2019

Lecture PSet #5
Upload as PDF by 14:30 Thu, 09/26/19
Optional docx available for submission

Problem: To address 4GB on a memory chip (DRAMS), 32 address lines are required. Having 32 I/O pins for address lines requires a lot chip area. Fortunately, memory is arranged internally on the chip as a grid of columns and rows which allows for a simple solution: share column and row address on the same lines thus reducing the I/O pins by 2. As a result, memory chips require several control signals to be asserted in a particular sequence to perform a memory read.

Row address is presented on the address bus and RAS (**R**ow **A**ddress **S**trobe) is used to strobe in the row address. Some time later, row address is removed and column address is present and CAS (**C**olumn **A**ddress **S**trobe) is used to strobe in the column address. The following figure shows a control module and the waveforms it must generate in response to a read request: FSMs are often used to generate sequences of waveforms necessary to communicate with a component.



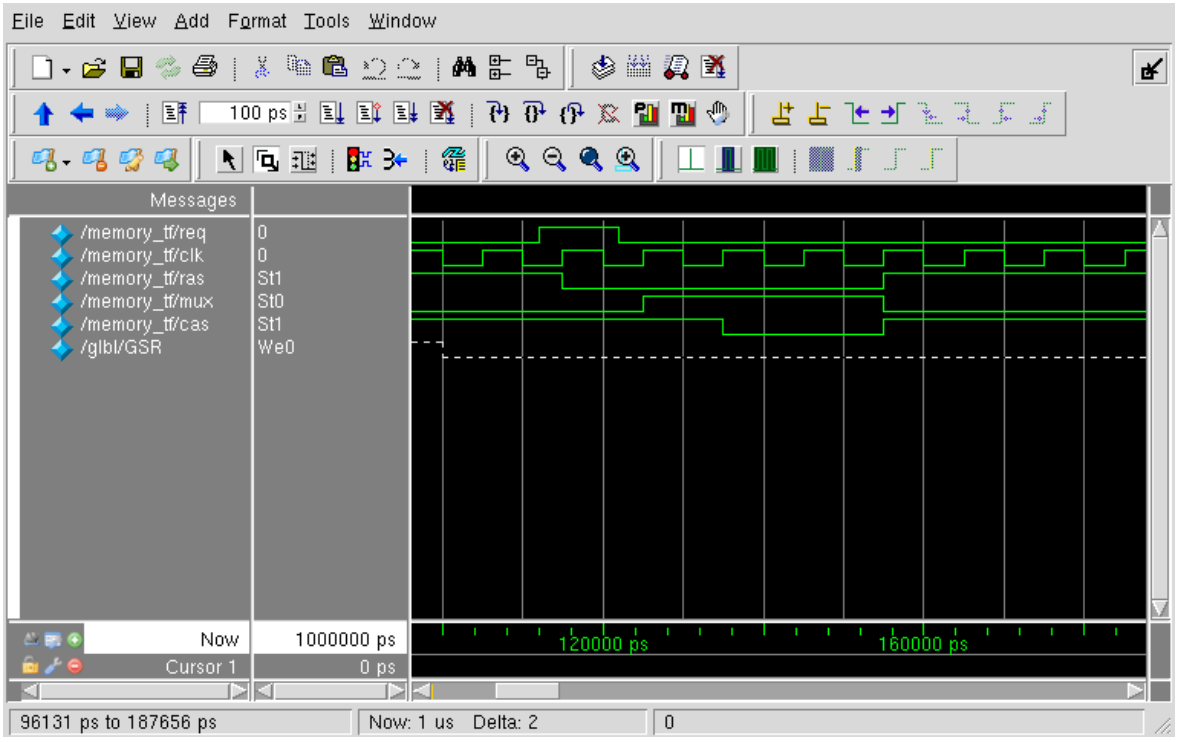
The module sits idle with RAS=1, MUX=0 and CAS=1 until it detects REQ=1 on the rising edge of CLK. In the first cycle of the request it should assert RAS=0, in the second cycle MUX=1 and in the third cycle CAS=0. These signals are held during the fourth cycle and return to their idle values in the fifth cycle. The module then waits for a new request; it ignores requests made while in the middle of processing the last request. [The timing and description is a gross simplification of an actual RAM timing and control. We didn't want to make the problem a 2 week design problem.]

(A) Draw a state transition diagram for a FSM that will generate the appropriate sequence of signals. (Use may draw the diagram using software or import a photo of a hand drawn diagram). [2 points]

(B) Write the Verilog for the module, choosing a state encoding and providing the appropriate combinational logic for generating the next state and output signals. But there's a hitch (this is the real world)! Using combinatorial logic to create RAS, MUX and CAS control signals could result in glitches . This is undesirable for memories. Select a state encoding that will generate glitch free RAS, MUX and CAS signals. There are at least two solutions to this problem. Attach your **Verilog and test bench.** [4 points]

(C) Explain how RAS, MUX and CAS are glitch free. [2 points]

(C) Verify your design by running a simulation. Note that **req** is asserted some time before the rising edge of **clk.** Attach your test bench and a screenshot of the simulation. Your screenshot should be identical this one with no extra cycles. [2 points]



For those with a thirst for knowledge: DRAM memory chips have performance specified as four numbers separated with dashes e.g. 7-8-8-24 (CL, tRCD, tRP, and tRAS). The parameters, expressed in clock cycles, are memory timings or RAM timings measure the performance of DRAM memory.

- CL: CAS latency, the number of cycles from column address to beginning of data,
- tRCD: Row address to column delay, the number of cycles between row and column address,
- tRP: Row precharge time, the number of cycles before recharge and opening the next row
- tRAS: Row active time, the number of cycles between row active and the next precharge.

Clock frequency can range from 400MHz (DDR2 400) to 3.6GHz (DDR4 3600) and higher. A typical DDR4 3600 has 18-20-20-40 timing parameters giving an absolute system performance of 18x(1/3.6Ghz) = 5ns.

To achieve higher performance, memory will have to be on the CPU die vs external memory DRAM.