University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

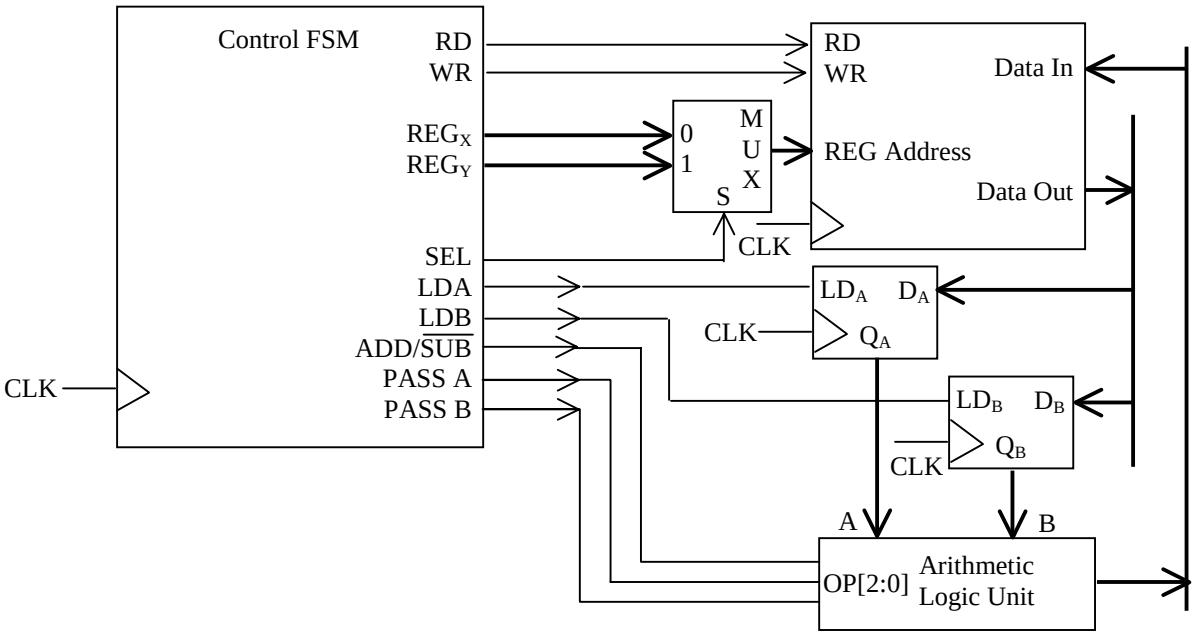
EECS 150 R. H. Katz
Spring 2004

Problem Set # 7 (Assigned 11 March, Due 19 March)

1. You are to implement a MOORE MACHINE *controller* that does the following. It "swaps" the values in two registers within a register file. That is, the high level operation SWAP (REG_X, REG_Y) places the contents of REG_X into REG_Y and REG_Y into REG_X, where REG_X and REG_Y are two registers within a register file. The contents of a register can be swapped with itself. The register file can read one register or write one register during any clock period, but not both (this is a single port read/single port write register file). The signals RD and WR control reading from and writing to the register file respectively. REG ADDRESS indicates the register number within the register file. The RD signal is *asynchronous*. Reading begins as soon as the signal is asserted. You may assume that data will be available at Data Out well before the rising edge of the clock after RD is asserted and REG ADDRESS becomes stable (i.e., propagation delay through the register file is short compared with the clock period). The WR signal is *synchronous*, and writing takes place on the rising edge of the clock whenever WR is true. You may assume that the set-up time for REG Address and Data In is much shorter than the clock period and that the hold time is 0.

   There are two buffer registers, A and B, with synchronous LD control inputs outside the register file. They provide the A-side and B-side inputs to an arithmetic/logic unit (ALU). The ALU can perform the four operations A PLUS B (OP=100) (sum A and B inputs), A MINUS B (OP=000) (subtract B from A), PASS A (OP=010) (pass through the A input), or PASS B (OP=001) (pass through the B input). You may assume that any of the ALU operations take place with a propagation delay much less than the clock period.

   The datapath fragment is the following. Thick lines are multi-bit busses; thin lines are single bit:



(i) Draw a state diagram fragment that implements the SWAP(REG_A, REG_B) sequence. Indicate which control signals are asserted in each state (assume a global RESET signal puts the controller into an initial state named S0).

(ii) Write a Verilog fragment that implements this state machine behavior.

(iii) Complete a timing diagram such as the figure below, indicating in detail when control signals are asserted and unasserted with respect to the clock.