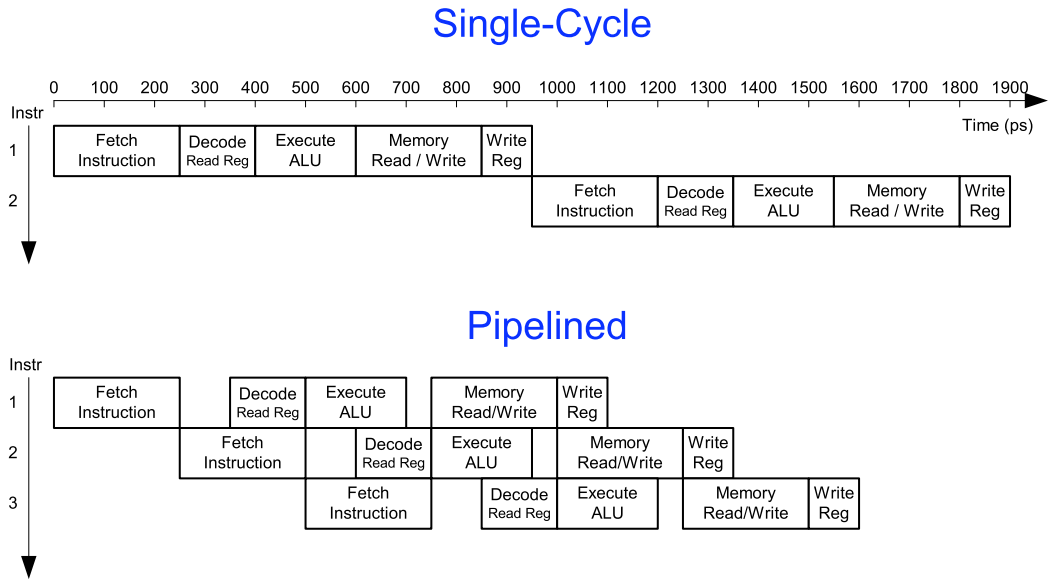
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2013

Homework Assignment 4: High-level Design, Multiplier, Pipelining and Parallelism
Due Feburary 26^{th}, 2pm

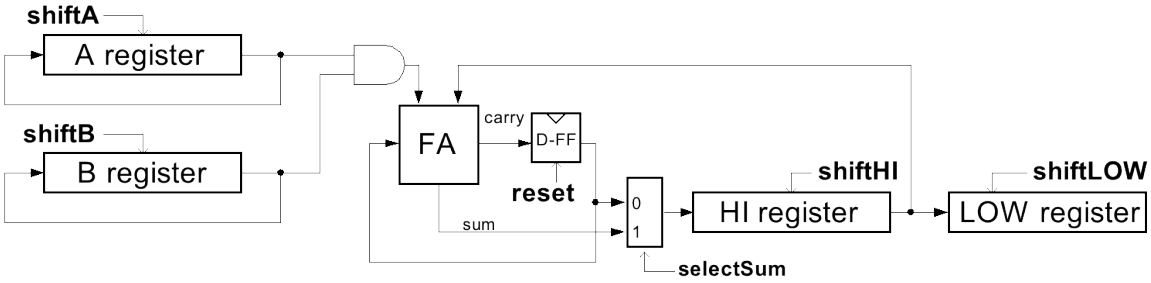
1. The diagram below shows a single cycle MIPS and 5 stage MIPS pipeline. Assume $FetchInstruction$ takes 250ps, $Decode/ReadReg$ takes 150ps, $Execute/ALU$ takes 200ps, $MemoryRead/Write$ takes 250ps and $WriteReg$ takes 100ps, and also assume no performance overhead is introduced when the pipelining registers are inserted.



   (a) We have a program of 2000 instructions running on the pipelined MIPS processor. The MIPS processor is designed such that it runs without stalls. After the pipeline is filled, one instruction is completed each cycle. What is the total time required to run this program in the pipelined processor? How long would it take if you run the program on the single-cycle processor?

   (b) A student redesigned the ALU, now $Execute/ALU$ only takes 150ps. Repeat part 1a.

2. Shown below is a bit-serial multiplier.



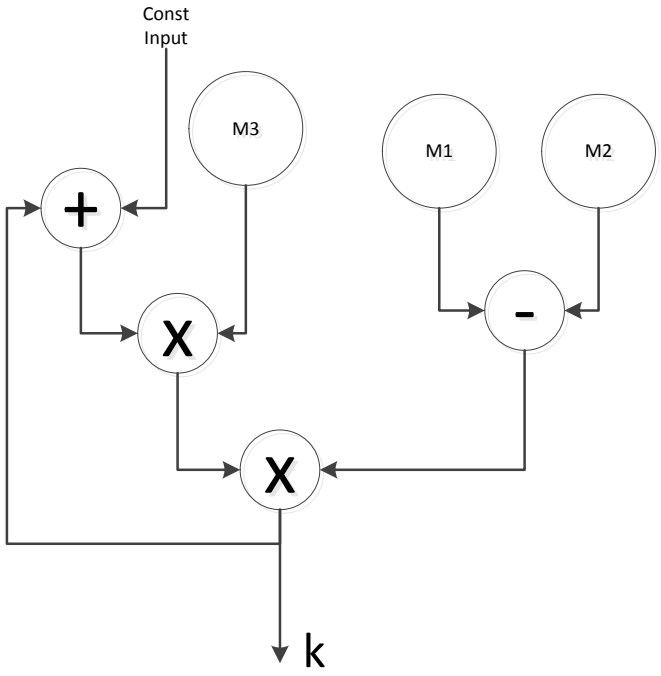
(a) Tabulate the change of value in each register when 4'd6 is multiplied by 4'd10.

(b) How should the multiplier be changed to handle signed 2's complement multiplication. Describe the change in the datapath, and rewrite the control algorithm you have seen in class.

hint:The two's complement of an N-bit number can be computed as

$$X = \sum_{i=0}^{n-2} x_i 2^i - x_{n-1} 2^{n-1}$$

3. The dataflow graph below shows the computation needed to generate one output K. M1, M2 and M3 are memory reads each of which can produce one new output 1ns after every clock edge. Addition and subtraction both take 5ns and multiplication takes 10ns. Register setup time and clock to q time are both 0.5ns.



(a) Without rearranging operators in the graph, we want to have a circuit which keeps producing new K from the data in the memory. How would you insert registers to maximize the throughput? How fast is your clock? How many new K would be produced every seconds?

(b) Is the implementation in the previous part the best we can do? If yes, explain why. If not, draw a better implementation, and how many new K can this implementation produce every second?

4. You are given 9 bit adders and 9 bit multipliers.

   (a) Implement a 18-bit multiplier. You can use multiple 9 bit adders and 9 bit multipliers, try to reduce the delay of the 18-bit multiplier. What is the critical path in your design?

   (b) Given one 18-bit multiplier and one 9-bit adder, implement a 36 bit multiplier. You can add registers and muxes when necessary. Show the datapath and design the control FSM, state clearly what happens in each state, in terms of the control signals in your datapath.

5. Consider the design of a special processor connected to a dual-ported memory, shown below. In the memory, an array of 8-bit integers is stored, starting at address 0. When started, the processor begins at location 0 and moves through memory forming the sum of all the integers up to that point, storing the sum in each memory location as it goes. The process continues for the entire array. An input signal call **START** is used to start the process, and an input called **ENDADDR** is used to specify the address of the final element in the array.

   Write the register transfer language description of the processor operation and draw the design of the processor datapath.

   • Use only the following circuit elements:

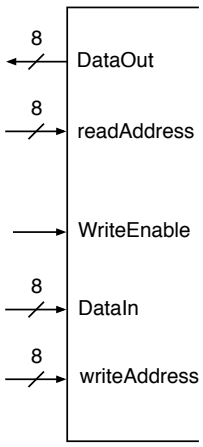
   (a) binary adder(s) of any width

   (b) register(s) with reset and load-enable,

   (c) equal-comparator(s), and the memory show below. The memory has asynchronous read and synchronous write operations.

   • In your design, minimize the processor cycle time and the number of cycles in the innerloop.

   • Remember to use a comma, ",", to separate RTL operations that occur on the same clock cycle, and a ";" to seperate operations on different cycles.


8
DataOut
8
readAddress
WriteEnable
8
DataIn
8
writeAddress