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

EECS150/EE141/EE241A, Fall 2014

Homework 6: High Level Design, Processors, and Register Circuits

Due: Friday Oct 24, 5pm

Please include your name, SID and specify either CS150, EE141 or EE241A at the top of your homework handin. Homeworks must be submitted electronically.

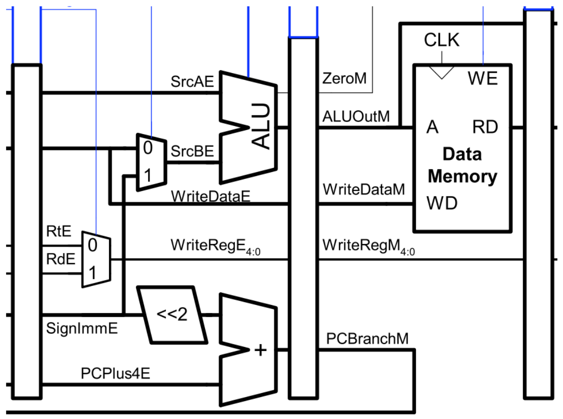
1. You are given the assembly code shown below, and it is run on a MIPS processor with a 5 stage pipeline.

```
0x7C        add $t1, $t2, $t3
0x80        add $t5, $t4, $t1
```

(a) Explain what kind of hazard is present in the code above, and how many extra cycles are introduced if the MIPS processor has no data-forwarding. Assuming all state elements in the processor are posedge triggered.

(b) Assuming the register file is written on the posedge of the clock, how many cycles would the two instructions take?

(c) To improve the performance, forwarding is implemented such that the inputs to ALU can be from the Memory stage. Now how many extra cycles are caused by the hazard? How would you modify the part of the datapath shown below for this to be possible? Name any new control signals you have added.

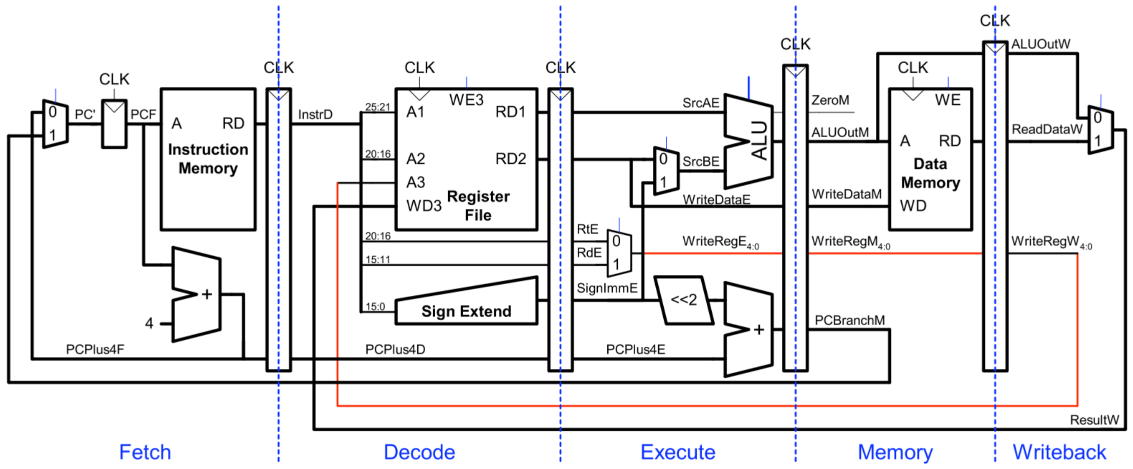


(d) In Verilog, design the control circuit coordinating the forwarding. Use the name of the signals from the lecture notes, and from part (c).

2. You are to add a *store with postincrement* instruction to a 5-stage MIPS processor pipeline. The instruction **swinc** updates the index register to point to the next memory word after completing the store. **swinc $rt, imm($rs)** is equivalent to the following two instructions:

```
sw         $rt, imm($rs)
addi       $rs, $rs, 4
```

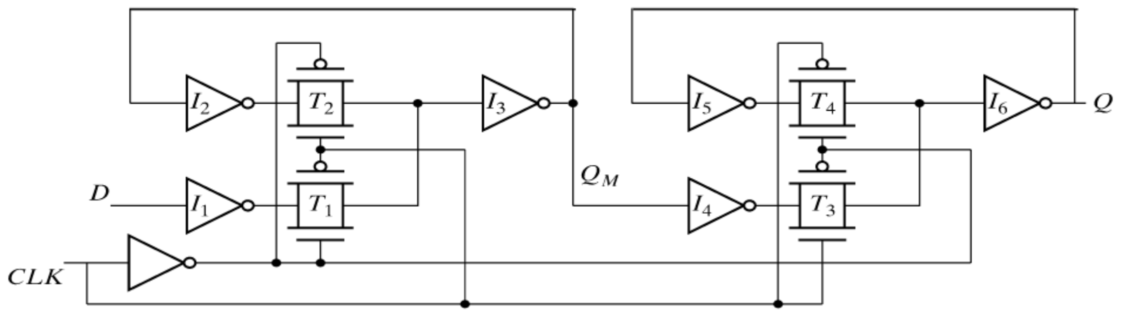
   (a) How would you modify the following datapath to accommodate this instruction? Try to add as little hardware as possible.



   (b) The following assembly program uses the new instruction, what hazard do you see? Modify the datapath you created in part (a) to eliminate the stalls introduced.

```
swinc      $t3, 0x10($s2)
add        $t3, $s2, $s3
```

3. Pictured here is the master-slave register circuit presented in lecture:



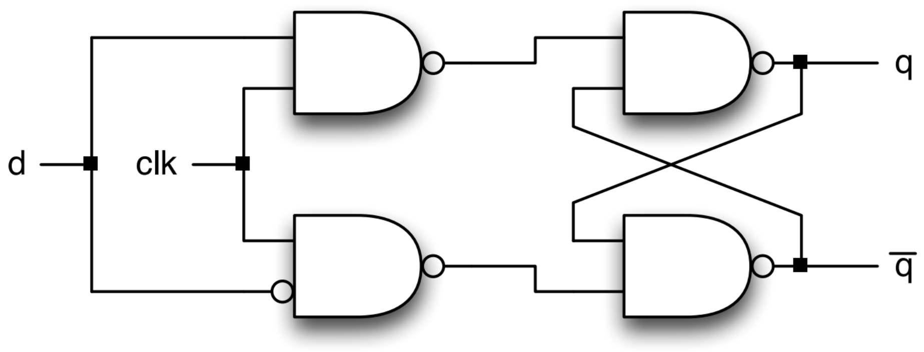
How would you modify the circuit to add the following features? Show the modifications you would make for each feature. Try to minimize the total number of transistors. You may use any simple 2 input gates and NMOS or PMOS transistors, and you may replace the inverters in the flip flop circuit, as long as you maintain the desired functionality.

(a) Asynchrounous reset

(b) Synchronous set

(c) Clock enable

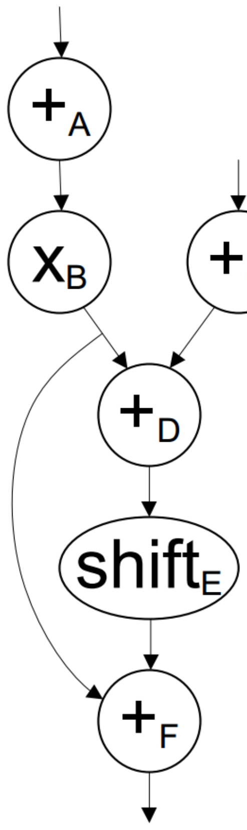
4. A positive level sensitive latch design is shown below



(a) Draw a circuit diagram showing how you would use the design principle of this latch as a building block to implement a NEGATIVE edge triggered D-type flip-flop.

(b) Assume a gate delay of 1ns for inverters and 2ns for NAND gates, what are the setup and clock-to-q times for your flip-flop.

5. You are given a datapath that has four computation units; two adders, a multiplier, and a shifter. Each unit requires an entire clock cycle (minus flip-flop overheads) to complete its operation and is followed by a register to hold its output. You can think of all the computation units as being in parallel, where you can use them all simultaneously on any given cycle. The graph below represents an iterative operation to be completed on the datapath. Each node is labeled with the name of the computation unit that it requires plus a unique letter identifying the node. Note that there is no feedback (or loop carry dependence) in this computation.

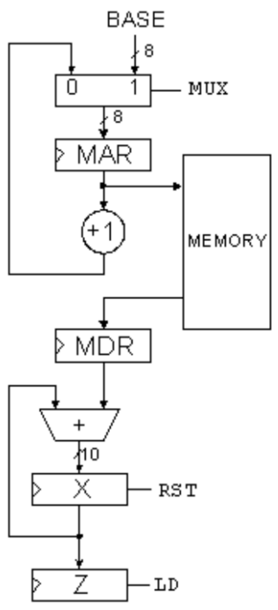


Fill in the following resource utilization chart to show how to complete four iterations of the loop in the minimum number of cycles during steady state operation. Show your work. Then, fill in the chart shown below the unique integer node numbers from the graph. Use subscripts (1, 2, 3, and 4) to indicate the iteration number. For instance, "C2" indicates node C of iteration 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
adder1																			
adder2																			
mult																			
shift																			

clock cycle

6. **EE241A only:** A simple processor is designed to add the contents of blocks of 4 bytes in consecutive memory locations. The datapath is shown below



The processor has one data input (8-bit wide) named **BASE**, an input control signal named **ENABLE**, and 3 internal control signals - **MUX, LD,** and **RST**. The datapath contains three data registers - **MAR, MDR,** and **X**. After the processor performs its operation, the **Z** register is left with the sum of memory locations **BASE, BASE+1, BASE+2,** and **BASE+3**. We assume that a controller (not shown) will take as input the **ENABLE** signals and generate **MUX, RST,** and **LD**. To begin the addition operation, an external circuit asserts **ENABLE** for 1 clock cycle then lowers it for a minimum of 12 cycles.

Write the register transfer language level description (following the syntax presented in lecture) for the sequence of transfers that must occur after the **ENABLE** signal is asserted. Try to minimize the total number of cycles.