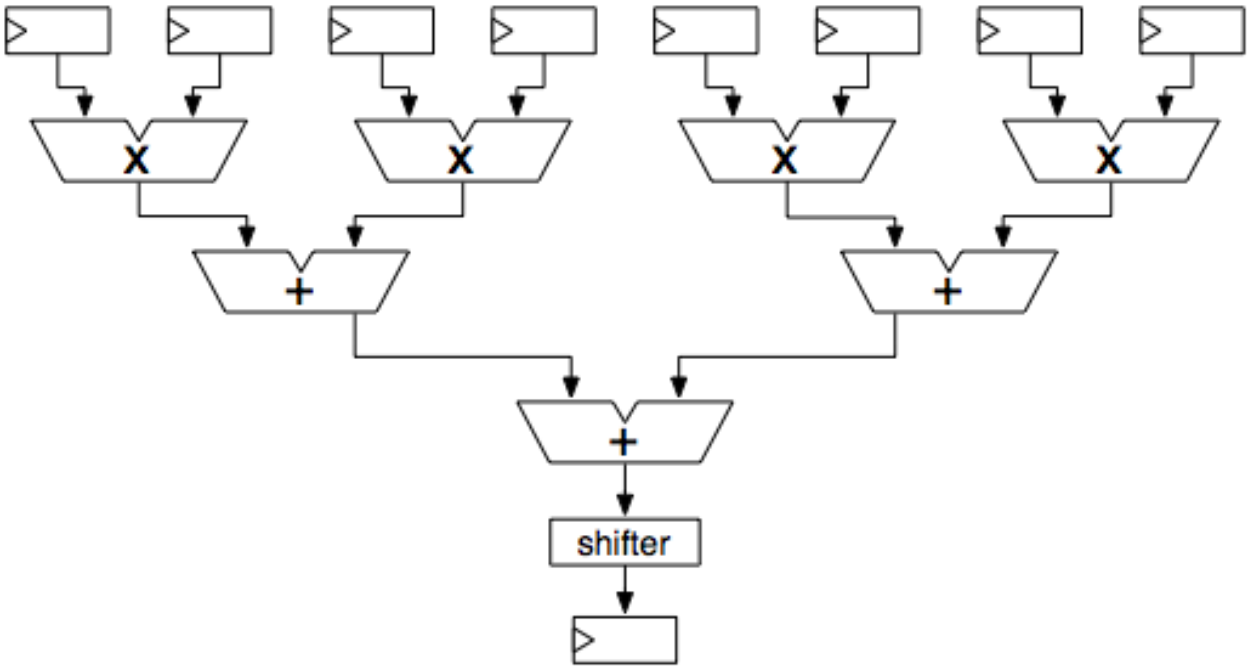
Clocking Methodology

- The input signal to each state element must stabilize before each rising edge.
- Critical path: Longest delay path between state elements in the circuit.
- Min clock period = $t_{clk\text{-}to\text{-}q} + t_{CL} + t_{setup}$, where t_{CL} is the Combinational Logic delay in the critical path.
- If we place registers in the critical path, we can shorten the period by reducing the amount of logic between registers

Clocking Problem

- The circuit below computes the weighted average of 4 values
- Logic Delays - t_{mult} = 55ns, t_{add} = 19ns, t_{shift} = 2ns
- Register Parameters - t_{setup} = 2ns, t_{hold} = 1ns, $t_{clk\text{-}to\text{-}q}$ = 3ns

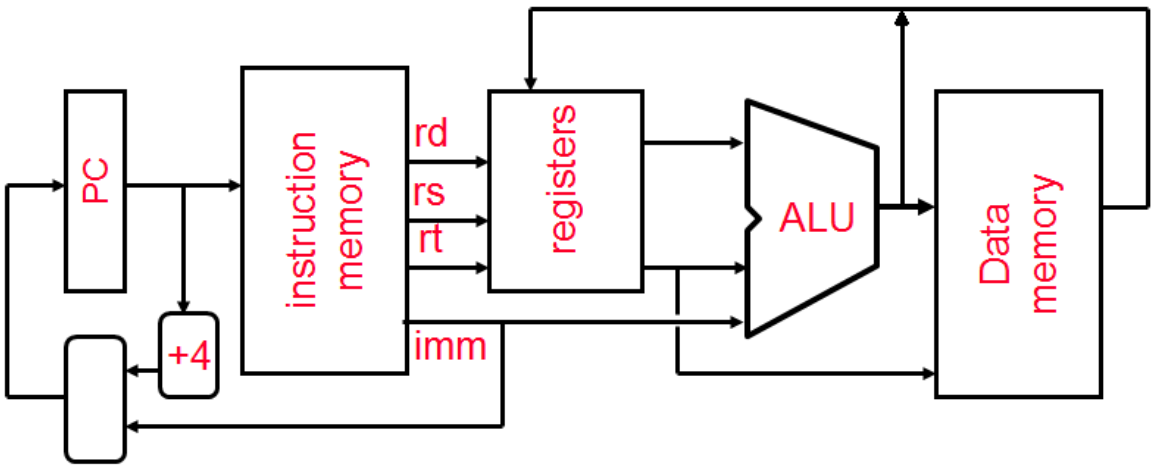


1. What is the critical path delay and the maximum clock rate this circuit can operate at?

2. If you add one stage of registers (pipelining), what is the highest clock rate you can get?

CPU Design

Here is the basic datapath as discussed in lecture, shown in simplified (i.e. incomplete) format:



`rd`, `rs`, and `rt` are 5-bit wires, `imm` is a 16-bit wire. All other wires are 32 bits wide. Assume that the ALU can output an Equals signal, which is on when its two inputs are equal.

1. Add control signals and missing elements (such as multiplexers) to the diagram so that the datapath can execute the following instructions: `add`, `lui`, `sw`, `bne`, `j`.

2. Fill out the values for the control signals from part 2 (write the names of your control signals in the second row):

Instr	Control Signals							
add								
lui								
sw								
bne								
j								

3. Suppose you wanted to add a new instruction, `beqr`, which will be used like this: `beqr $x, $y, $z` will branch to the address in `$z` if `$x` and `$y` are equal, otherwise continue to the next instruction. Show any changes that would need to be made to the datapath above to make this instruction work.