CS61c Summer 2014 Discussion 11 – Clocking and finite state machines

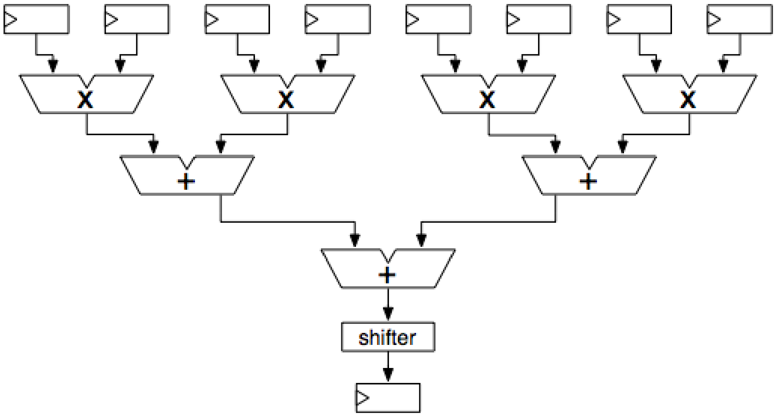
1 Clocking

1.1 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 = $t_{clk-to-q} + t_{CL} + t_{setup}$, where t_{CL} is the critical path in the comnibational logic.
- If we place registers in the critical path, we can shorten the period by reducing the amount of logic between registers.

1.2 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, $t_{clk-to-q}$ = 3ns

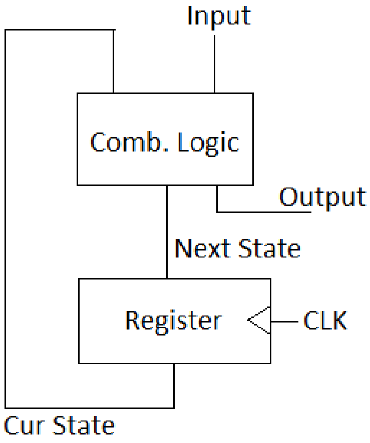


Exercise 1. What is the critical path delay and the maximum clock rate this circuit can operate at?

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

2 Finite State Machines

FSMs can be an incredibly useful computational tool. They have a straightforward implementation in hardware:



The register holds the current state (encoded as a particular combination of bits), and the combinational logic block maps from {current state, input} to {next state, output}.

Exercise 3. Draw a transition diagram for an FSM that can take in an input sequence one bit at a time, and after each input is received, output whether the number of 1s is divisible by 3.

Exercise 4. Write out the truth table that the combinational logic block must implement (remember to assign each state a binary encoding).

Exercise 5. Finally, write the Boolean algebra expressions that implement the FSM's truth table.