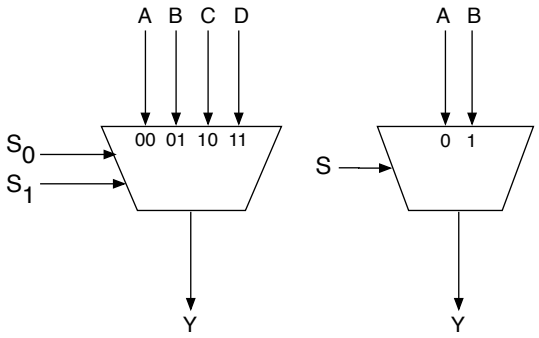
| **Intro. to Computer Architecture** | **Homework 3** |
|---|---|
| **CSE 240 Autumn 2005** | **DUE: Fri. 30 September 2005** |

Write your answers on these pages. Additional pages may be attached (with staple) if necessary. Please ensure that your answers are legible. Please show your work. Due at the ***beginning of class***. Total points: 37.

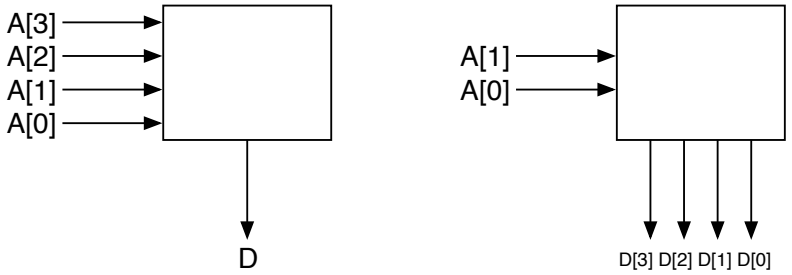
1. [8 Points] **Muxes and Memory.**

   (a) A 1-bit 4-to-1 mux (below, left) selects among four inputs using two selector lines. Specifically, the mux outputs $A$ when the selectors have the value 00 (with $S_1$ being the high-order bit), $B$ when they have the value 01, $C$ when they have the value 10, and $D$ when they have the value 11. Construct a 1-bit 4-to-1 mux (*i.e.*, build a circuit with the behavior of a 1-bit 4-to-1 mux) using three 1-bit 2-to-1 muxes (below, right). Be sure to label the inputs ($A$, $B$, $C$, $D$, $S_0$, and $S_1$) and output ($Y$).

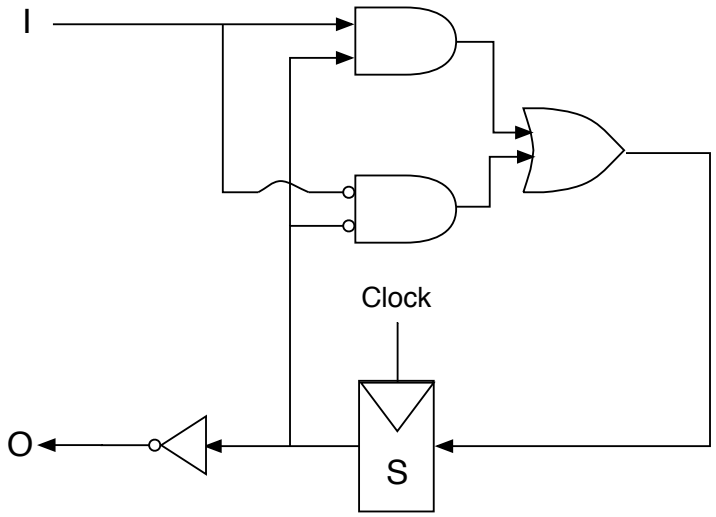



   (b) How many 1-bit 2-to-1 muxes are required to build a 1-bit $2^k$-to-1 mux (for $k > 0$)?

(c) Memories come in many shapes and sizes. If we need a memory of a particular size and addressability, we can often construct it out of other, differently-structured memories. Construct a $2^4$-by-1-bit memory (below, left) using a mux and a $2^2$-by-4-bit memory (below, right), as well as any gates you might need. Assume that the memories are read-only memories (ROMs), so you need only be concerned with reading. Be sure to label the inputs ($A[0]$ - $A[3]$) and output ($D$).

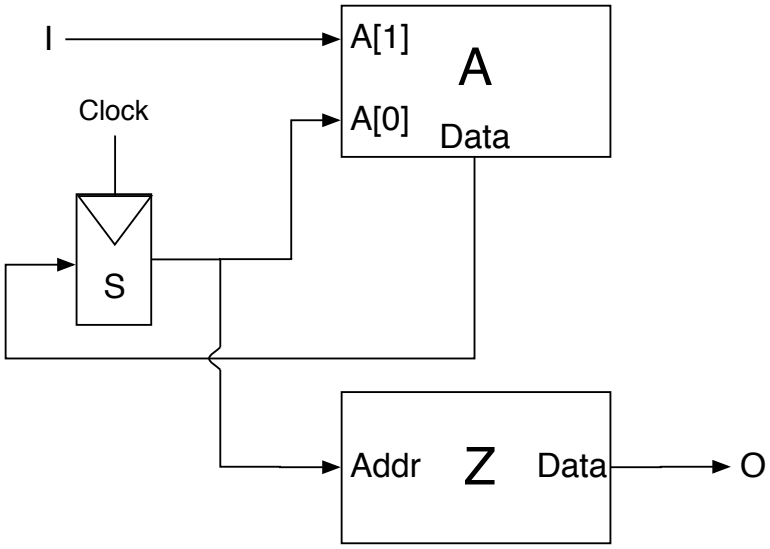


2. [8 Points] **Finite State Machines.** The circuit below implements a finite state machine $M$. The one bit register labeled $S$ stores the current state, $I$ is the input, and $O$ is the output.




(a) How many states does $M$ have? Draw a state diagram for $M$ (similar to that in the diagram on page 75 of the textbook).

(b) The circuit below is a programmable finite state machine. The components of this circuit are labeled according to the following key: (A) A $2^2$-by-1-bit memory. (Z) A $2^1$-by-1-bit memory. (S) A 1-bit register.

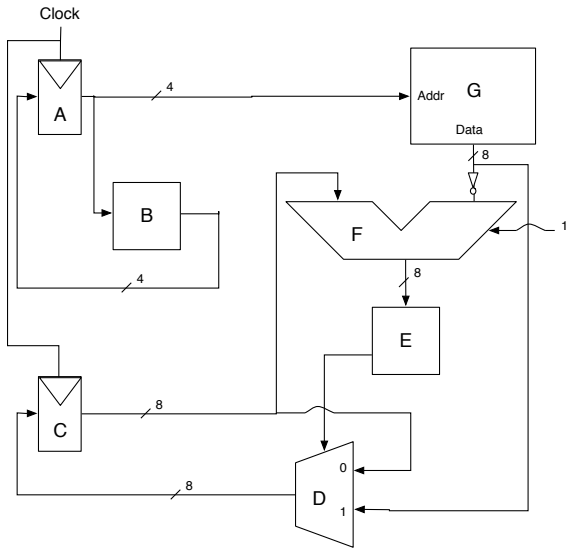



What should the contents of the memories be so that this circuit implements $M$ (from the previous problem)?

| Address | Contents of A |
|---|---|
| 0 0 | |
| 0 1 | |
| 1 0 | |
| 1 1 | |

| Address | Contents of Z |
|---|---|
| 0 | |
| 1 | |

3. [8 Points] **Sequential Logic Circuits.** Consider the following circuit.




The components of this circuit are labeled according to the following key: (A) A 4-bit register representing an unsigned integer (initially set to 0). (B) A 4-bit incrementer. (C) An 8-bit register storing a 2's complement integer (initially set to 127). (D) An 8-bit 2-to-1 mux. (E) A circuit whose output is true when the 8-bit 2's complement input is positive. (F) An 8-bit adder. (G) A $2^4$-by-8-bit memory. Assume the memory has the following contents.

| Address | Contents | Address | Contents |
|---|---|---|---|
| 0 | 31 | 8 | 27 |
| 1 | 47 | 9 | 95 |
| 2 | 5 | 10 | 88 |
| 3 | 16 | 11 | 67 |
| 4 | 112 | 12 | 63 |
| 5 | 3 | 13 | 54 |
| 6 | 59 | 14 | 80 |
| 7 | 8 | 15 | 110 |

(a) Determine the values stored in the A and C registers at the end of each clock cycle for 8 cycles. Complete the following table with this information.

| Clock Cycle | A | C |
|---|---|---|
| 0 | 0 | 127 |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |

(b) In a simple sentence, what does this circuit do?

4. [4 Points] **Reasoning About Numbers.**

(a) Draw a gate-level circuit that has a 1 output, $O$, only when the 5-bit 2's complement input, $S[0] - S[4]$, represents 0.

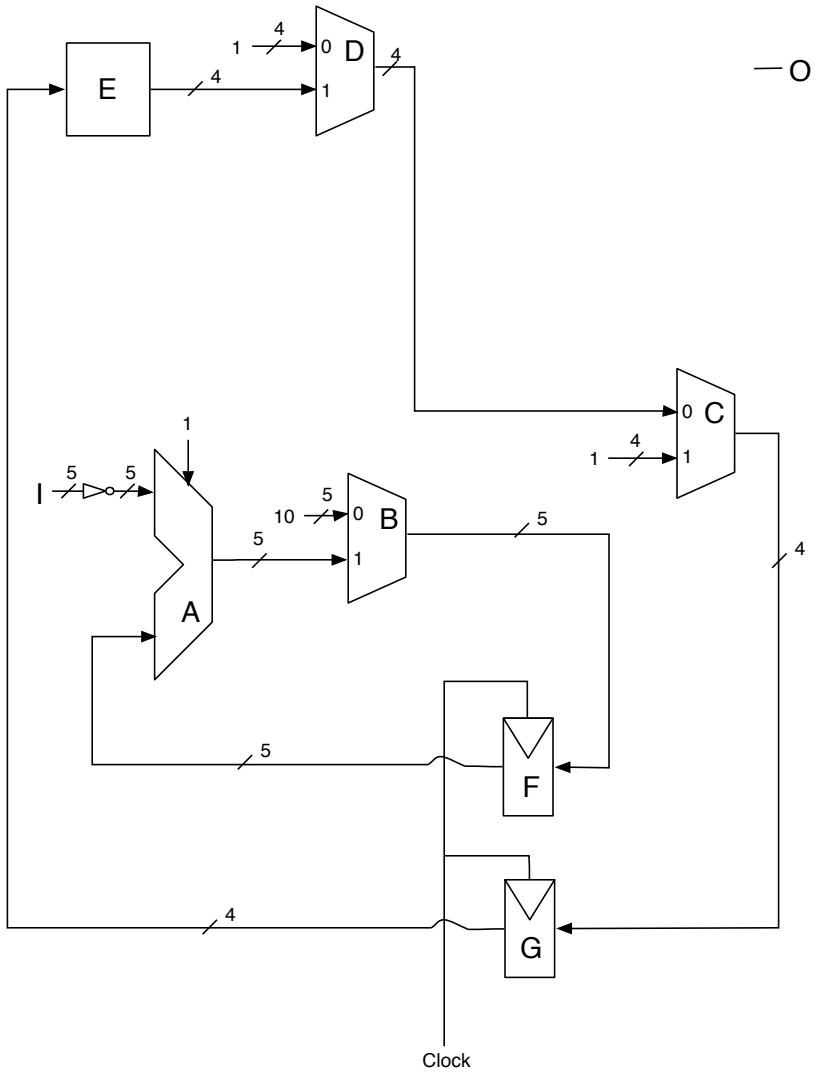
(b) Using your circuit from (a), draw a gate-level circuit that has a 1 output, $O$, only when the 5-bit 2's complement input, $S[0] - S[4]$, represents a positive integer. You don't need to redraw the circuit from (a); just draw it as a box labeled Z.

5. [9 Points] **More Sequential Circuits.**

Consider the incomplete circuit, below. The components of this circuit are labeled as follows: (A) A 5-bit adder. (B) A 5-bit 2-to-1 mux. (C) A 4-bit 2-to-1 mux. (D) A 4-bit 2-to-1 mux. (E) A 4-bit incrementer. (F) A 5-bit register. (G) A 4-bit register. $F$ stores a positive 2's complement integer less than or equal to 10, and $G$ stores a 2's complement integer between 1 and 4, inclusive. The input $I$ is a 5-bit 2's complement integer between 0 and 10, inclusive.

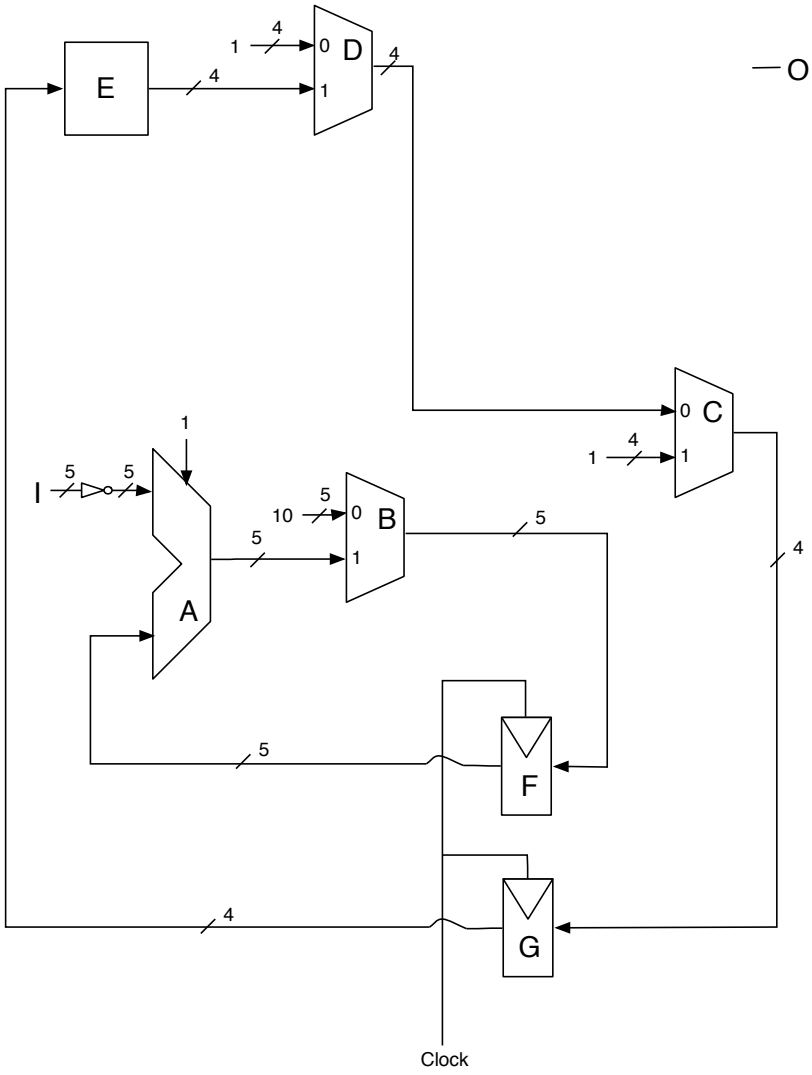
This incomplete circuit is intended to implement the following logic. In each cycle, except in the circumstances described below, (i) $F$ is updated to contain $F - I$, (ii) $G$ is updated to contain $G + 1$, and (iii) the output $O$ is 0. If $I$ is greater than or equal to $F$ or $G$ is equal to 5 (after the increment), 10 is written into $F$ (instead of $F - I$) and 1 is written into $G$ (instead of $G + 1$). If $G$ is equal to 5 (after the increment), the output $O$ is also set to 1 (instead of 0).

Unfortunately, the circuit is incomplete because the mux selector lines are missing as is the logic for computing the output $O$. It is your job to fill in the missing pieces of this circuit to achieve the intended functionality. (Note that while this circuit seems completely arbitrary, it is actually implementing a simplified version of the rules of American football. If it still seems completely arbitrary, well that's because the rules of American football are completely arbitrary!)

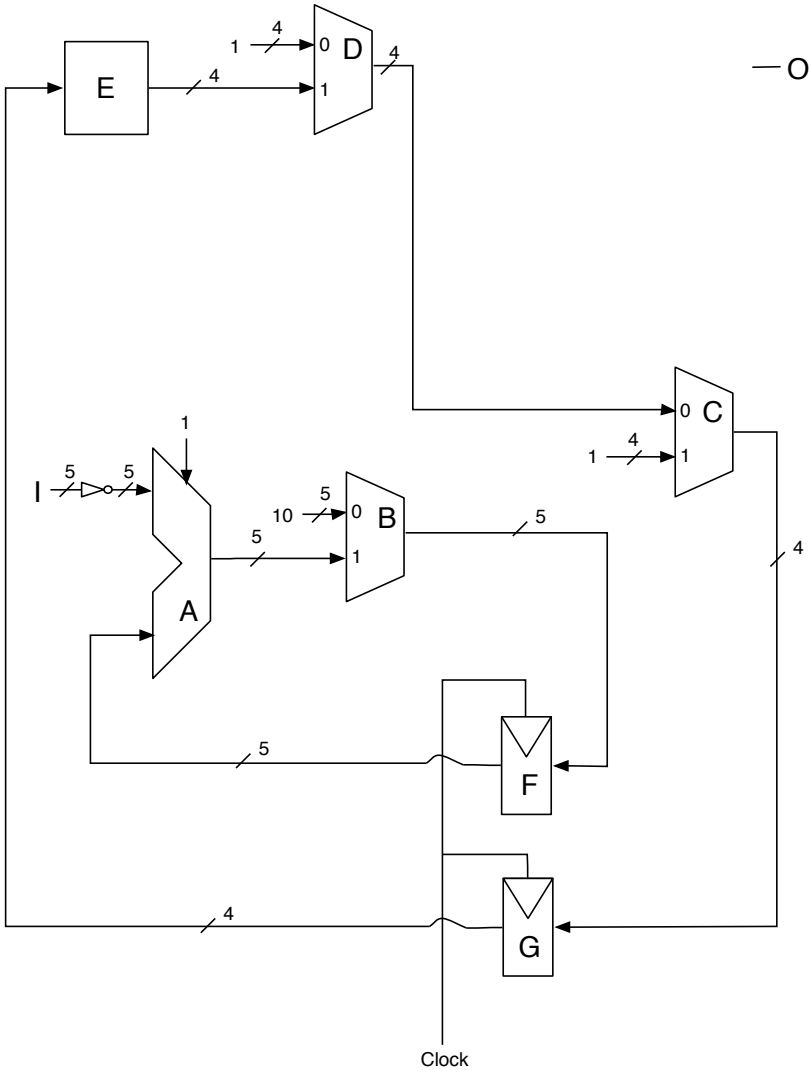



(a) Let's do this one step at a time. Using your positive-testing circuit from the previous problem (draw it as a box labeled $P$), augment the given circuit (above) with the logic for the $D$ mux selector. Hint: The $D$ mux determines whether to reset the value in $G$ to 1 due to the input $I$ being greater than or equal to the value stored in $F$.

(b) Using a 4-bit adder and the zero-testing circuit from the previous problem (draw it as a box labeled *Z*), augment the given circuit (below) with the logic for the *C* mux selector and the output *O*. Feel free to include the logic from the previous problem (but you don't have to). Make sure the output is connected to the line labeled *O*. Hint: The *C* mux determines whether to reset the value in *G* to 1 due to it being equal to 5 after incrementing.



(c) Augment the given circuit (below) with the logic for the *B* mux selector. You may find it helpful to redraw some or all of the logic from the previous steps.



6. [No Points] **Last and Most Important Question!** Please complete this question, and give us your feedback!

   (a) How many hours did you spend on this assignment?

   (b) On a scale of 1-5, how difficult did you find this assignment? (1-easiest, 5- most difficult)

   (c) Do you have any other comments on your experience completing this assignment? What are they?