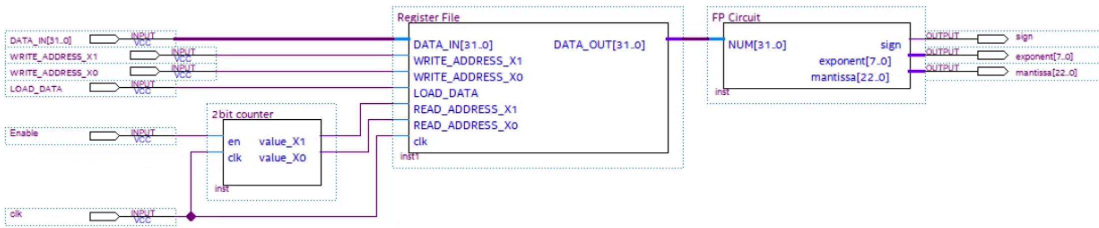
**Cpr E 281 HW09**
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

# State Assignment, Counters, and Register Files

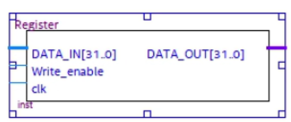
## Due Date: Oct. 26, 2020

**P1 (30 points):** Floating Point Register File.
Consider the circuit below. This circuits iterates through a list of four floating point numbers stored in a register file and finds the sign, exponent, and mantissa portion of each number.




A) (10 points) The following four floating point numbers are currently stored in the register file. For each number find the values for **sign**, **exponent**, and **mantissa**. (Ex: 3.0, sign = 0, exponent = $10000000_2$ , mantissa = $10000000000000000000000_2$.)
   a. 23
   b. -127
   c. -53
   d. 1.6

B) (10 points) Use any number of 32-bit registers, 2-to-4 decoders, 32-bit 4-to-1 multiplexers, and any necessary gates to construct a register file that can store four floating point numbers. Use the following design for your 32-bit registers.




C) (10 points) Implement a 2-bit counter to iterate through all four values stored in the register file.
   a. Use TFF to design the 2-bit counter
   b. Use DFF to implement the 2-bit counter

# State Assignment, Counters, and Register Files
## Due Date: Oct. 26, 2020

**P2 (20 points):** We want to design a circuit with input W and an output Z, where Z will equal 1 if, for the last three clock cycles, W has been 1.
A: Draw a state diagram for a Moore Finite State Machine (FSM) that implements this circuit in four states, specified as follows:

RESET

**S-0 : 0** **S-I : 0** **S-II : 0** **S-III : 1**

B: Complete a state table for the above state diagram with the state assignments as shown below for state variables $D_1$ and $D_0$.

| | W=0 | W=1 | Z |
|---|---|---|---|
| S-0 : 00 | | | |
| S-I : 01 | | | |
| S-II : 10 | | | |
| S-III : 11 | | | |

C: Use K-maps to show that the output and next-state variables can be expressed as:

$$D_1^{new} = (w)(D_1 + D_0)$$
$$D_0^{new} = (w)(D_1 + \overline{D}_0)$$
$$z = D_1 D_0$$

D: Let's consider if the states were encoded as:

S-0 = 00, S-I = 01, S-II = 11, S-III = 10

Use K-maps to show that the output and next-state variables with this new encoding can be expressed as:

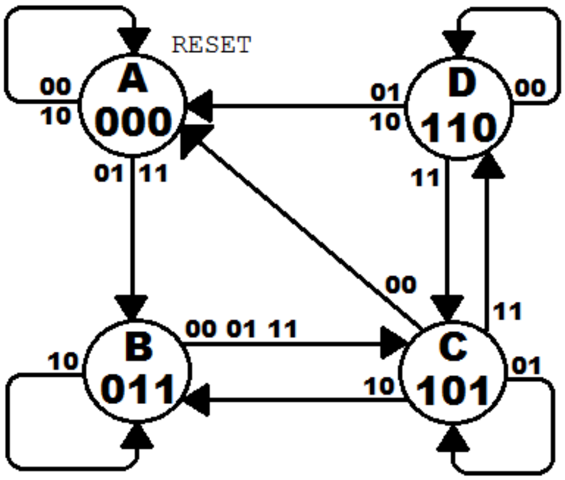
$$D_1^{new} = (w)(D_1 + D_0)$$
$$D_0^{new} = w\overline{D}_1$$
$$z = D_1 \overline{D}_0$$

E: Draw the circuit for this FSM (developed in part D) using only DFFs, AND gates, and one OR gate (Do not use any NOT gates).

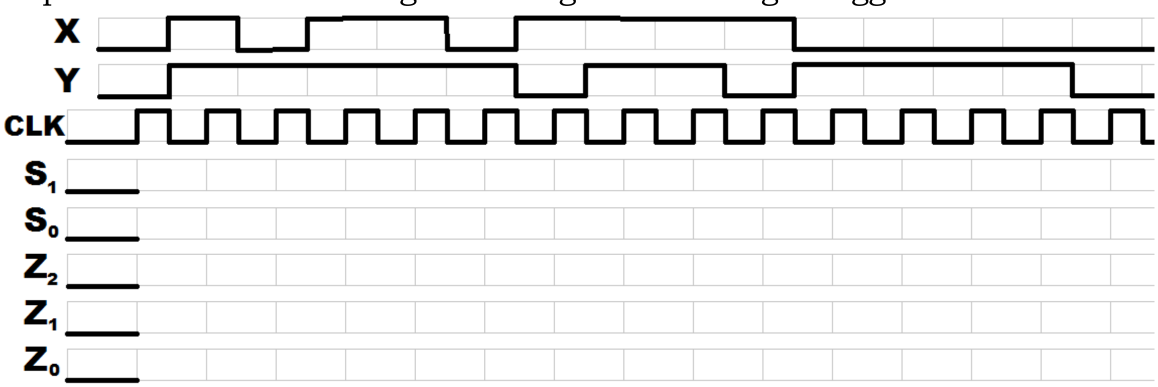
**P3 (15 points):** Design a circuit that with one bit input N and 3-bit output P that operates as follows:
When N=0, the new value of P will be $P_{new} = 2 * P_{old}$. When N=1, the new value of P will be $P_{new} = 2 * P_{old} + 1$. Since P is only three bits, each operation must be modulo 8 (i.e., if P=5, then 2*P=10 is too large to fit into the 3-bit value for P, so we subtract 8 from 10 to get 2, which should be the new value for P).
A: Draw the state table for the Moore FSM that implements this circuit. Your circuit should have a state for each possible output.
B: Show the next state Boolean expressions for each state variable.
C: The circuit that you end up with for this circuit is equivalent in functionality to a circuit that we have already discussed. What circuit is this?

**P4 (15 points):** Draw the state diagram for a Moore FSM that has a 1-bit input P and a 1-bit output Q. P will be either 1 or 0 on any particular clock cycle. Q=0 if P has been 1 for an even number of clock cycles; Q=1 if P has been 1 for an odd number of clock cycles.
A: Draw the state diagram for this Moore FSM.
B: Draw the state table for this FSM.
C: Draw a state assigned table for this FSM. The state should be the same as the output: Q.
D: Draw the truth table for this FSM's next-state variable.
E: Derive the expression for the next state variable and the output Q.
F: Draw the circuit for this FSM. If done properly, the circuit you create will implement a component that you have seen before. What component have you implemented?

**P5 (20 points):** Look at the state diagram below. The input variables are X and Y. The state variables are $S_1$ and $S_0$. The state encodings are as follows: A=00, B=01, C=10, and D=11. The output variables are $Z_2$, $Z_1$, and $Z_0$.




I: Fill in the timing diagram below given the state diagram for a circuit that implements this state diagram using Positive-Edge-Triggered DFFs.




II: Fill in the state table with state assignments

| | X=0, Y=0 | X=0, Y=1 | X=1, Y=0 | X=1, Y=1 | $Z_2 Z_1 Z_0$ |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |

III: Draw the truth table and show that the next-state expressions can be expressed as follows:

$$S_0^{new} = \bar{S}_1\bar{S}_0(Y) + \bar{S}_1 S_0(X\bar{Y}) + S_1\bar{S}_0(X) + S_1 S_0(\bar{X}\bar{Y})$$
$$S_1^{new} = \bar{S}_1\bar{S}_0(0) + \bar{S}_1 S_0(\bar{X} + Y) + S_1\bar{S}_0(Y) + S_1 S_0(XY + \bar{X}\bar{Y})$$

IV: Derive expressions for the output variables $Z_2$, $Z_1$, and $Z_0$ in terms of $S_1$ and $S_0$.