



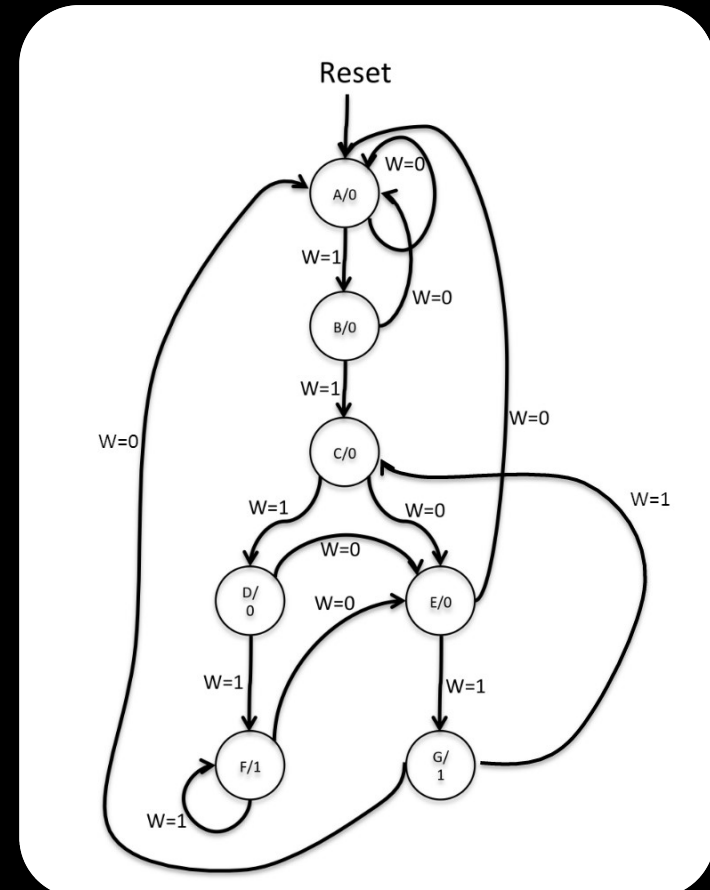
Lab 6 Preparation

Lab 6 Components

- **Part I:** Create a Finite State Machine
 - Make a clocked sequence recognizer.
- **Part II:** Control a datapath
 - Combine datapath + FSM to perform ALU functions.
- **Part III:** Divider (bonus)
 - Dividing number using a simple adder/subtractor
 - **This is a bonus part**, for those who are looking for a little extra challenge 😊
 - You can still get full marks for leaving this out, and 10 marks out of 8 for completing it perfectly.

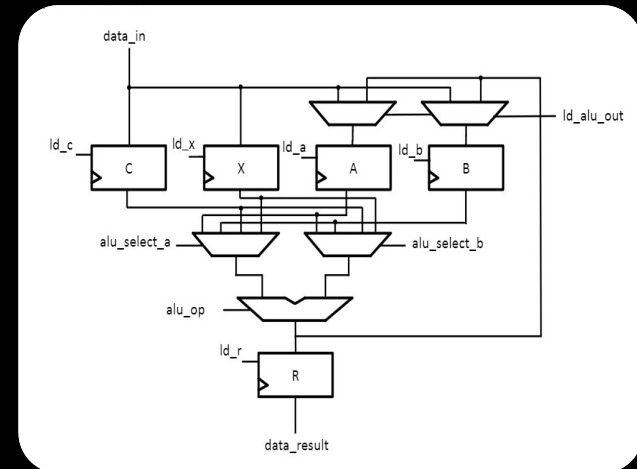
Part I: Finite State Machine

- **Sequence recognizer:**
 - Make output high if the sequence 1111 or 1101 was seen on the input.
- Starter circuit provided!
 - Assign flip-flop values to each state in diagram
 - Create **state logic** to assign new flip-flop values based on previous values.
 - Many ways to do this!



Part II: Datapath Control

- Recall the ALU datapath example we did in class.
 - This is the same thing 😊
- We provide the datapath circuit, you provide the FSM for the controller.
 - Controller turns the datapath signals on or off to:
 - Move data from registers to the ALU
 - Perform an ALU operation
 - Store the result back into a destination register.



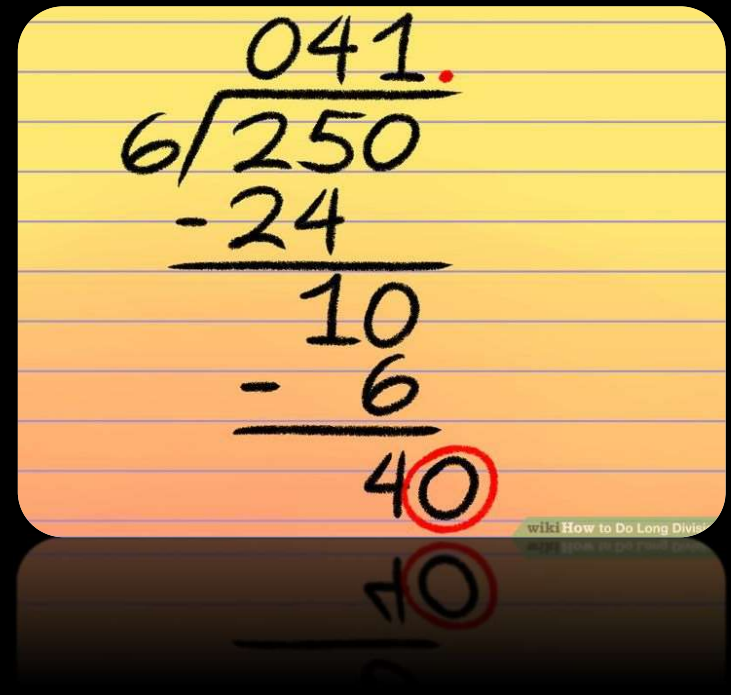
1100

- Steps to follow:

-

Part III: Divider Circuit

- Note: This part is optional, but can be done for bonus marks in the course.
- Basic idea from decimal long division:
 - From left to right, find where the divisor can be subtracted from the dividend.
 - Doing this in binary is simpler, except that we keep the divisor static, and move everything else!



A handwritten long division problem on lined paper. The divisor is 6, and the dividend is 250. The quotient is 41 with a decimal point. The steps shown are: 6 goes into 25 four times (24), leaving a remainder of 10. Then, 6 goes into 10 one time (6), leaving a remainder of 40. The final remainder 40 is circled in red. A small watermark 'wiki How to Do Long Division' is visible at the bottom right of the paper.

$$\begin{array}{r} 041. \\ 6 \overline{) 250} \\ \underline{-24} \\ 10 \\ \underline{-6} \\ 40 \end{array}$$

Thoughts for Lab 6

- You're all grown up now.
- The only restriction for Lab 6 is that you must use the datapath that we provide and not change it.
- Beyond that, there are no limitations to the implementation approach that you use.
 - You can use the files that we provide, or not.
 - You can implement your FSM the way we did in class, or not.

Thoughts for Lab 6

- The one thing we will check this week is the **readability** and **modularity** of your design.
 - If your *.circ file is getting full and/or complicated, break it up into smaller modules (like helper functions).
 - The collection of modules we provide for each datapath signal is a guide to this (but not a required approach).
 - Your modules should contain circuits that are simple and compact, either through module creation or an elegant design approach.
 - Consider using the Tunnel (under Wiring) to make connections across long distances:
 - <http://www.cburch.com/logisim/docs/2.6.o/en/libs/base/tunnel.html>
- **Only submit the modules that you use.**