Project Title: A Simple Multi-Cycle Central Processing Unit (CPU) Design

EE 203 Digital Systems Design

Date Assigned: Nov. 27, 2019

Teaching Assistant: Diren Erdem

Teaching Assistant: Ayşenaz Ergin

Instructor: Dr. Şuayb Ş. Arslan

Institution: MEF University

Ertuğrul Yılmaz

Student ID: 041701030

Zeynep Yılmaz

Student ID: 041701019

OP[0]	OP[1]	T_0	T_1	T_2	Done	Rxin	Rxout	Ryout	Ain	Bin	Bout	Asen	DINout
0	0	1	0	0	1	1	0	1	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	1	0	1	0	0	0	0
1	0	0	1	0	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	0	1	0	1	0	0	0	0
1	1	0	1	0	0	0	0	1	0	1	0	1	0
1	1	0	0	1	1	1	0	0	0	0	1	0	0

Table3: Truth Table for control circuit

Operations for inputs and outputs;

- DINout=(OP[0])'*OP[1]
- Asen=OP[1]*T_1
- Bout=T_2
- Bin=T_1
- Ain=OP[0]*T_0
- Ryout=(OP[0])'*(OP[1])'+T_1
- Rxout=OP[0]*T_0
- Rxin=(OP[0])'+T_2
- Done=T_2+(OP[0])'

In the control logic, We split the instruction bit into 8-bits, which came to control logic, by using the splitter. We take the 2nd, 3rd and 4th bit and turn them into XXX by using the splitter. After, we used splitter as the demux selection. Outputs of demux entered XXX register's enable. As we

mentioned on the truth table, we took MSB and LSB from the counter and connected them with the decoder's selection, by doing this we get the T0, T1, and T2 as the output of the decoder. We implemented the outputs of the control logic by using necessary AOI gates and circuit elements. We get the control logic equations, like; done, clear, Ain, etc.

	T_0	T_1	T_2
00	0	0	0
01	1	0	0
10	0	1	0
11	0	0	1

Current State		Next State	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

A(t+1)=A'B+AB'=A⊕B

B(t+1)=B'

A'B=T_0

AB'=T_1

AB=T_2

While implementing the register file of our circuit, we connected the circuit element called “destination” to a decoder. We also connected the decoder with the register’s enable because according to income destination from the circuit, data will be given to register. Arriving data, which every time the clock runs, will be written to the destination register. Even if the destination shows another register, data remains its position. If the destination shows the same register after a few operations, data will be overwritten on the register, the register’s old data will be gone.

Usually, when we implement a circuit there is an enable. The reason why we connected our decoder to register’s enable is, we will write our operations to register.

In our datapath, we take X and Y data into our circuit. X and Y go to ALU after muxes. Data will be processed depending on the chosen(given) operation. The output will continue its operation after this process. After all of this, the register will be chosen and the data will be written on the register. This will repeat for a few times.

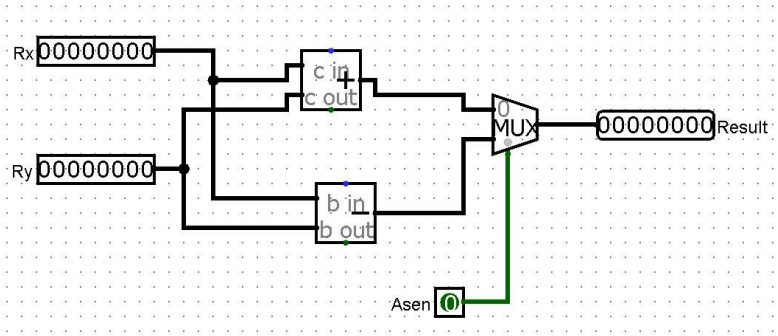
On the control unit, if the variable “done” is equal to 1, we can write our data to register; if it is equal to 0, we will have a reserve situation. “Asen” provides addition and subtraction operations. If the value of Asen is equal to zero we make addition; if the value of Asen is equal to 1 we make the subtraction operation.

On the CPU, where we run the main circuit of the central processing unit (CPU), we have 8 registers, 4 muxes, the control unit, a ram, the counter, the arithmetic logic unit (ALU), 3 registers, and an AND gate.

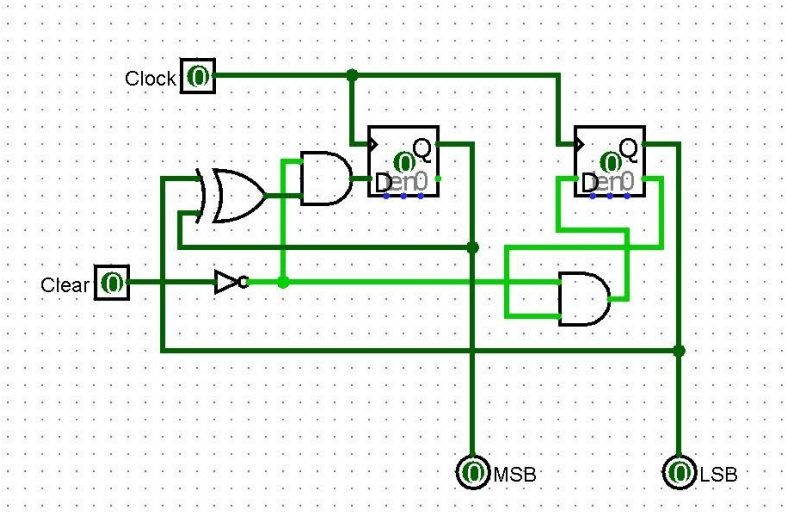
In counter, We used digital circuit design techniques to create the state diagram and state table. Then we implemented the logic diagram of the 2-bit counter. We used 2 D flip-flops and AOI gates. We took the MSB and LSB as the output of the counter. And we had an input which named as clear to start over the counter.

In the arithmetic logic unit (ALU), We used 3 inputs. One of them is Rx and the other one is Ry and Asen. We also used two arithmetic logic elements; an 8-bit adder and an 8-bit subtractor. We connected these 2 outputs to MUX's input. We also connected Asen to the MUX which we get from the logic unit. It shows that the arriving instruction will do subtraction or addition. We get the result as output.

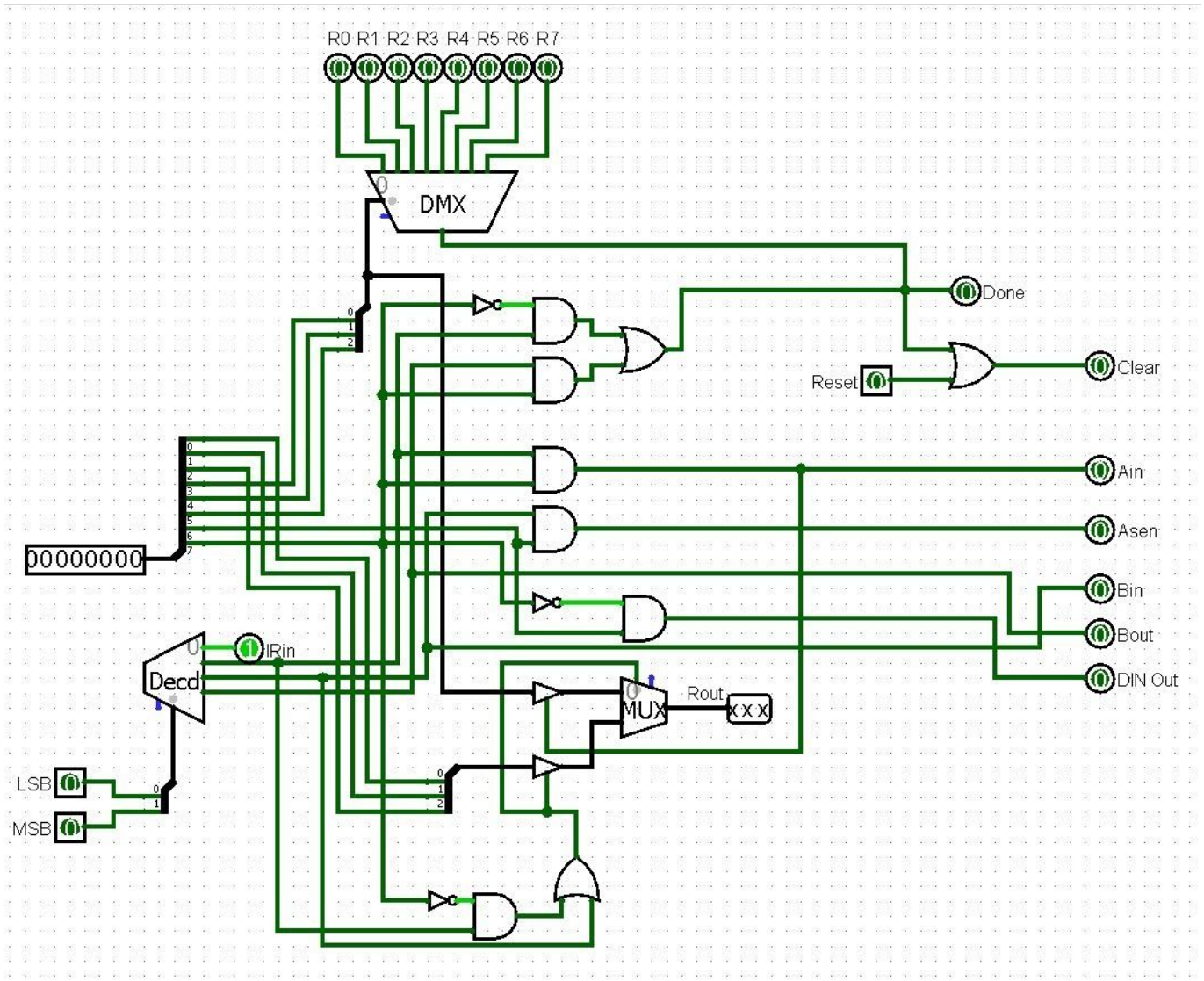
In the central processing unit (CPU), we have buttons to reset RAM and CPU.



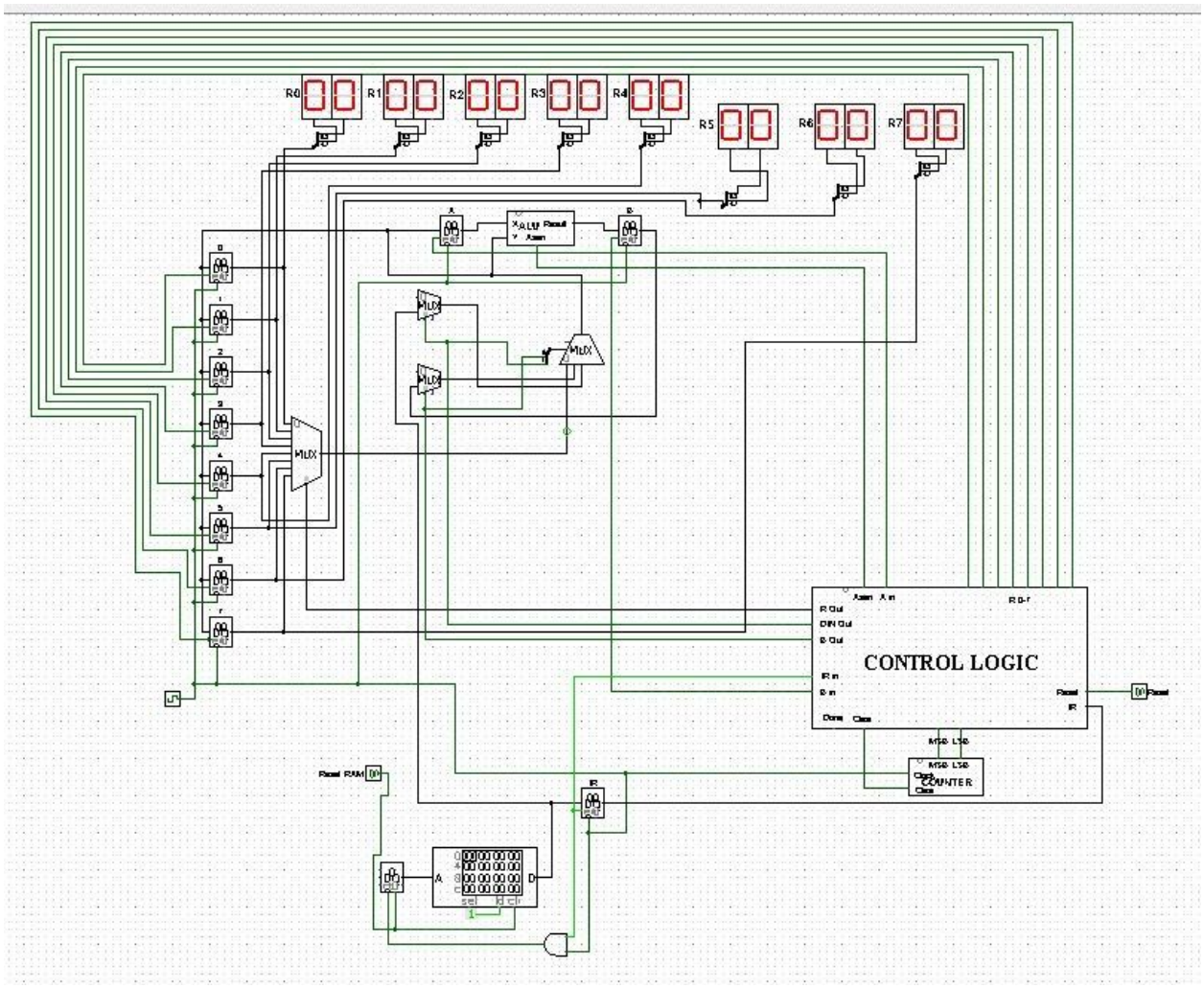
Circuit 1: Arithmetic Logic Unit (ALU)



Circuit 2: Counter



Circuit 3: The Control Unit



Final Circuit: Central Processing Unit (CPU)