A Simple Multi-Cycle Central Processing Unit (CPU) Design EE 203 Digital Systems Design

Teaching Assistant: Diren Erdem

Teaching Assistant: Aysenaz Ergin

Instructor: Dr. Şuayb S. Arslan

Institution:MEF University

04 January 2020

Group Members:

Mustafa Mert Burma 041701033

Burak Çevik 041702025

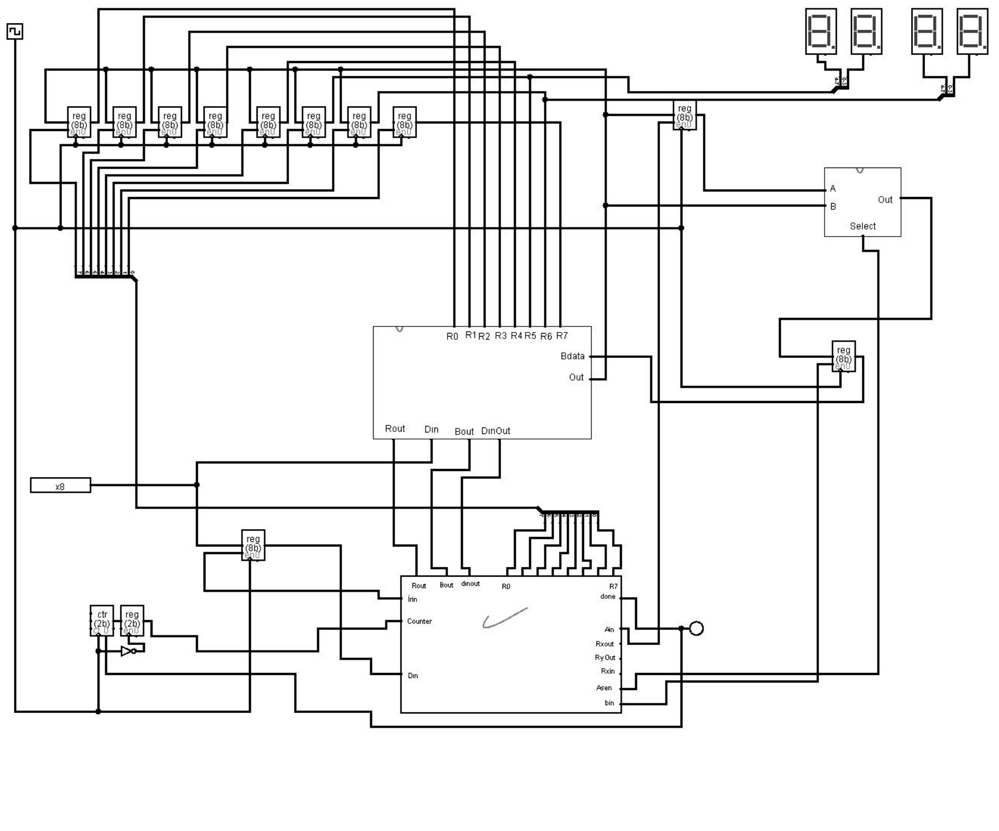
Introduction

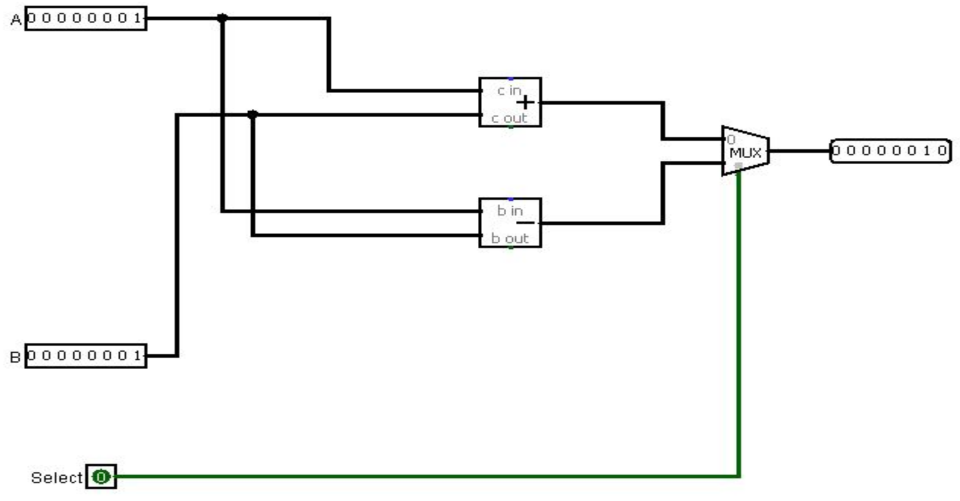
We have designed and implemented a simple multi-cycle central processing unit (CPU) with a bus unit and a wired controller that can process 8-bit data through various registry transfers and common Arithmetic Logic Unit (ALU) procedures.We realize this project in 3 steps
In the first stage, we tried to understand the given block schema. The second stage involves the design and implementation of the main components of a simple multi-loop CPU, such as the registry file, the arithmetic logic unit (ALU), and the counter in our default Multisim software.

The Design Work

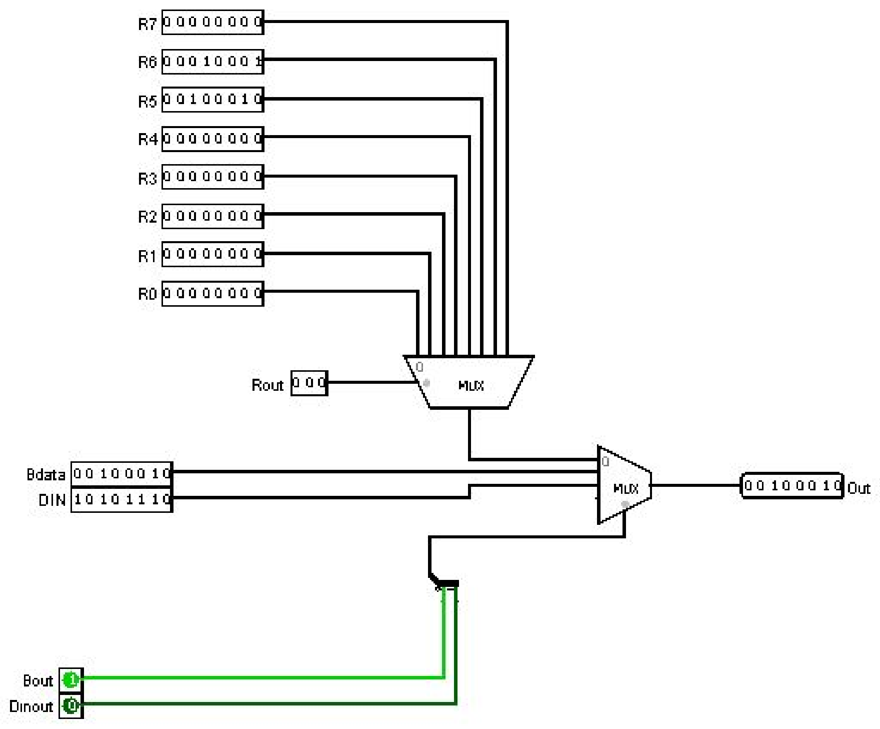
2.1

Datapath of our CPU



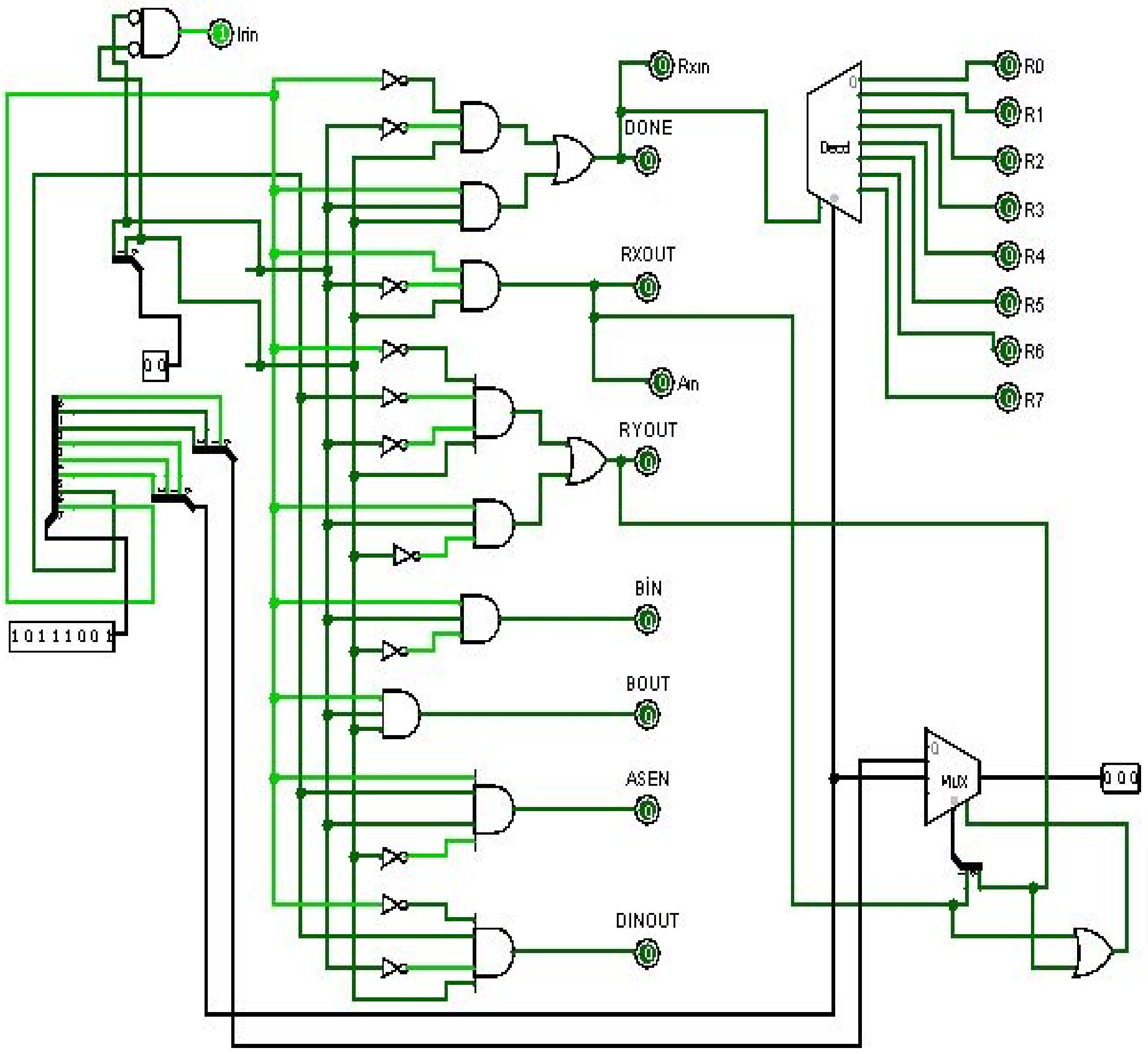


• ALU of our CPU



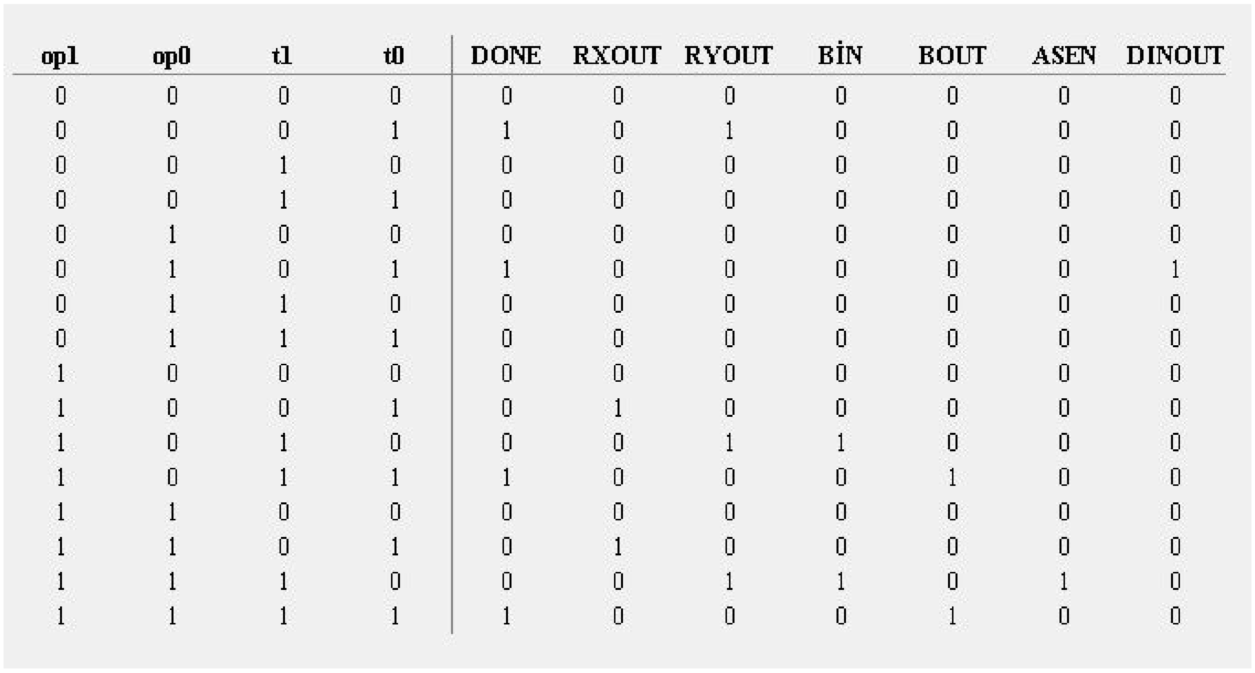
• Multiplexer of our CPU

2.2



• **Control Logic of our CPU**

- **Truth Table**



op1	op0	t1	t0	DONE	RXOUT	RYOUT	BİN	BOUT	ASEN	DINOUT
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0
1	0	1	1	1	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0
1	1	1	0	0	0	1	1	0	1	0
1	1	1	1	1	0	0	0	1	0	0

Expressions of Control Logic

Done/Rxin: A1'T0T1'T2' + A1T0'T1'T2

Rxout/Ain: A1T0T1'T2'

Ryout: A1T0'T1T2' + A1'A2'T0T1'T2'

Bin: A1T0'T1T2'

Bout: A1T0'T1'T2

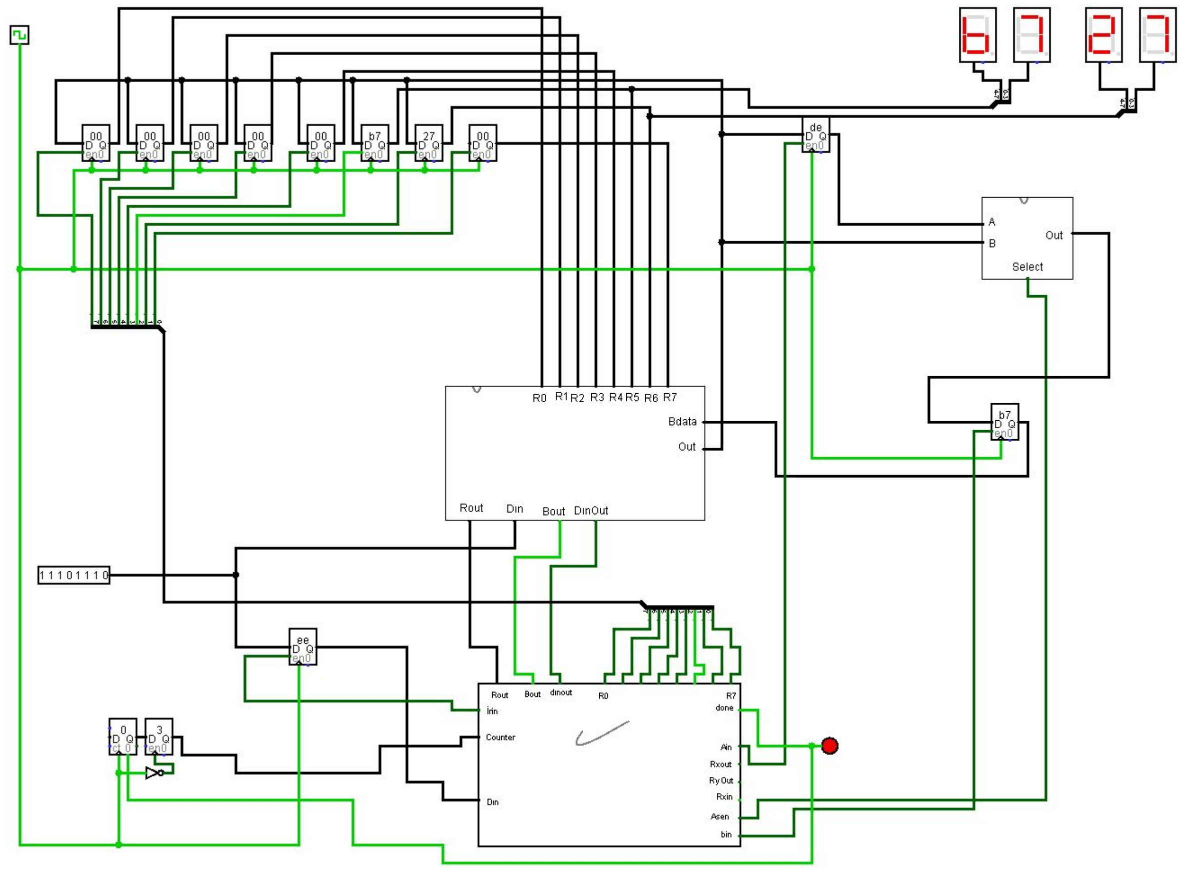
Asen: A1A2T0'T1T2'

DINout: A1'A2T0T1'T2'

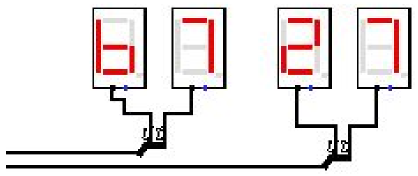
Demonstration

3.1

$$mvi\ R5, 0\text{xB8} \mid R5 \leftarrow 0\text{xB8}, \tag{1}$$
$$mv\ R6, R5 \mid R6 \leftarrow [R5], \tag{2}$$
$$mvi\ R5, 0\text{x6F} \mid R5 \leftarrow 0\text{x6F}, \tag{3}$$
$$add\ R6, R5 \mid R6 \leftarrow R6 + [R5], \tag{4}$$
$$add\ R5, R5 \mid R5 \leftarrow R5 + [R5], \tag{5}$$
$$sub\ R5, R6 \mid R5 \leftarrow R5 - [R6] \tag{6}$$



3.2



3.3

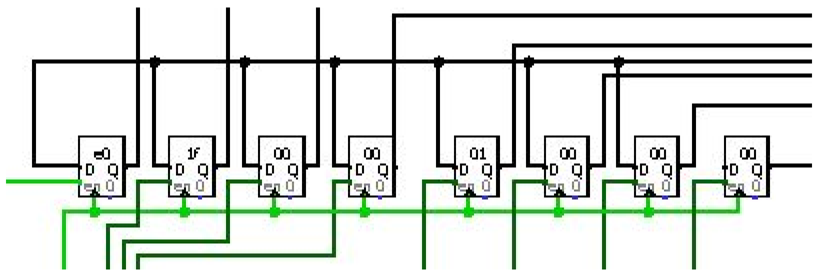
If we say: Ra => R0, Rb => R1, Rc => R2

I assigned a value of fourth register 1 to subtract 1

2 clock for mv and mvi

4 clock for add and sub

Total = 16



Work Distribution:

- **CPU, Control Logic,Multiplexer and Report has done by Mustafa Mert Burma**

- **ALU Unit has done by Burak Çevik**