DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Dr. Anjad Badran

Experiment No. 2 - Comparators, Adders and Subtractors

Section 11

Islam Jihad Joma 1191375

T. Enas Jawabrah

3-4-2021

Abstract

In this experiment we have to learn how to create memory using flip flops and chips to create bigger and bigger storage in ROM or RAM or as a normal storage, we have to link them together to create longer as vertical or horizontal, then simulate them by simulation button and try some inputs to check and fill tables.

Table of content:

Introduction (Theory)

We have to create those 2 figures

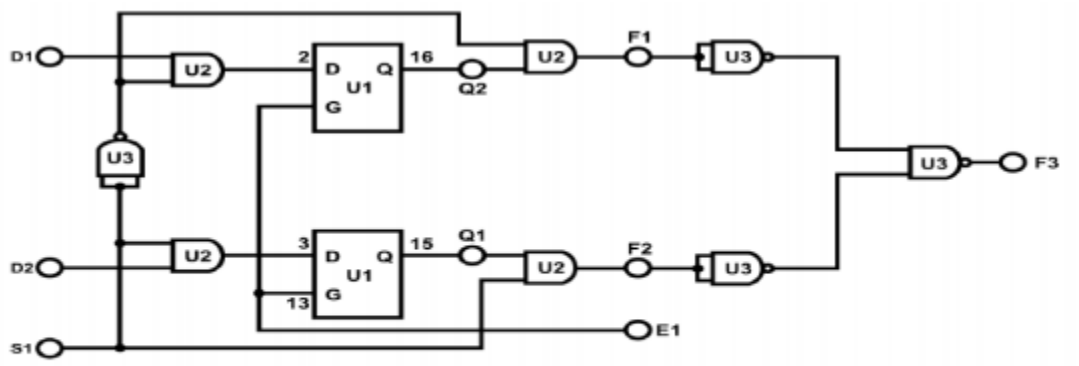


Figure 7.5

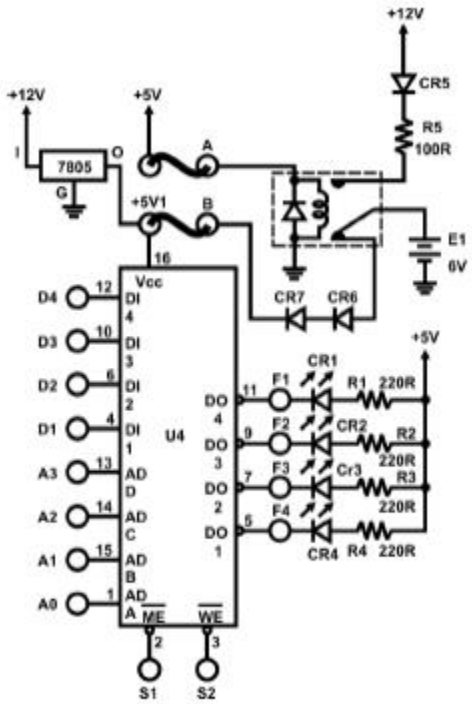


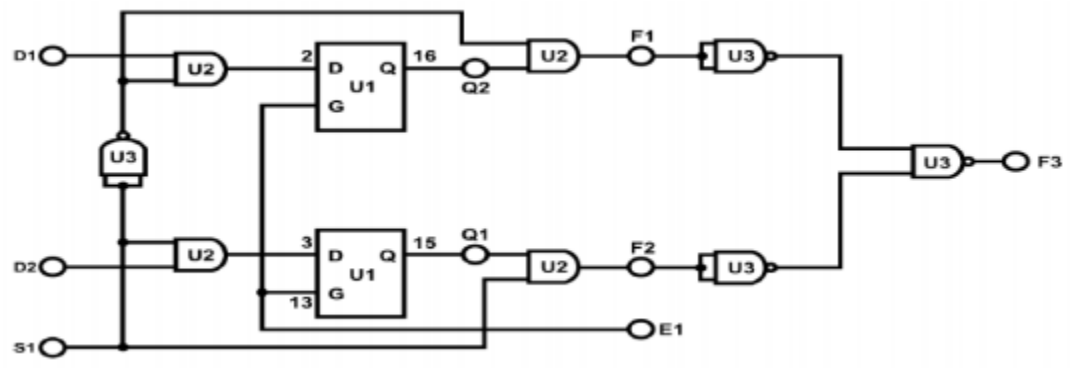
Figure 7.5

We need to create RAM block with D Flip-Flop so it simulates the RAM work in only 2 bits as an example. We need to create Ram IC circuit that save 64 bits.

We need to simulate them too, and other connecting stuff.

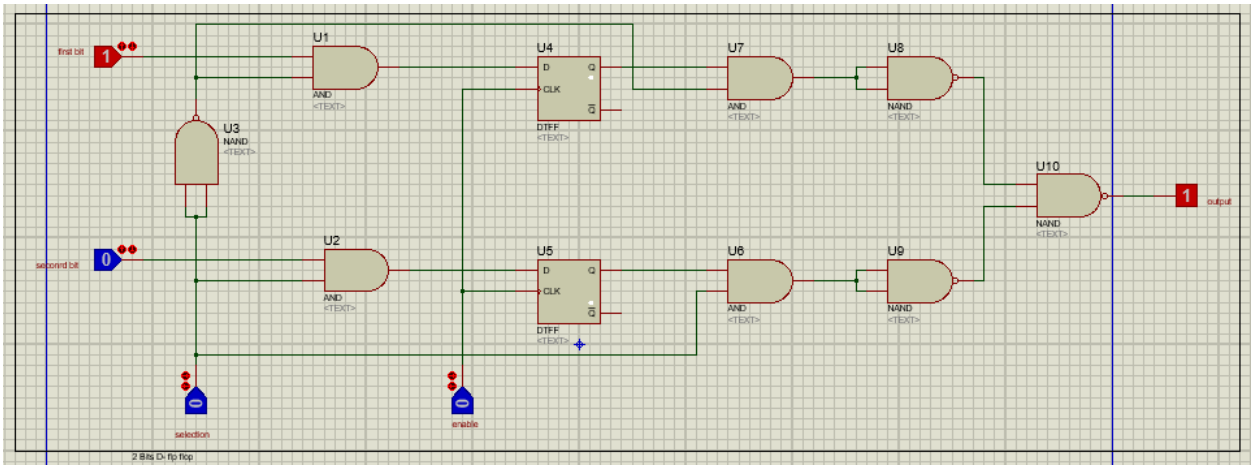
Procedure (Discussion & Results)

RAM block with D Flip-Flop



We used Portus of course to create the circuit, we needed the input and output buttons and the NAND and AND gates and two D flip flops and a selection button and enable button that's work as clock.

I'll share a picture of the circuit



Then I simulated it and got results like this:

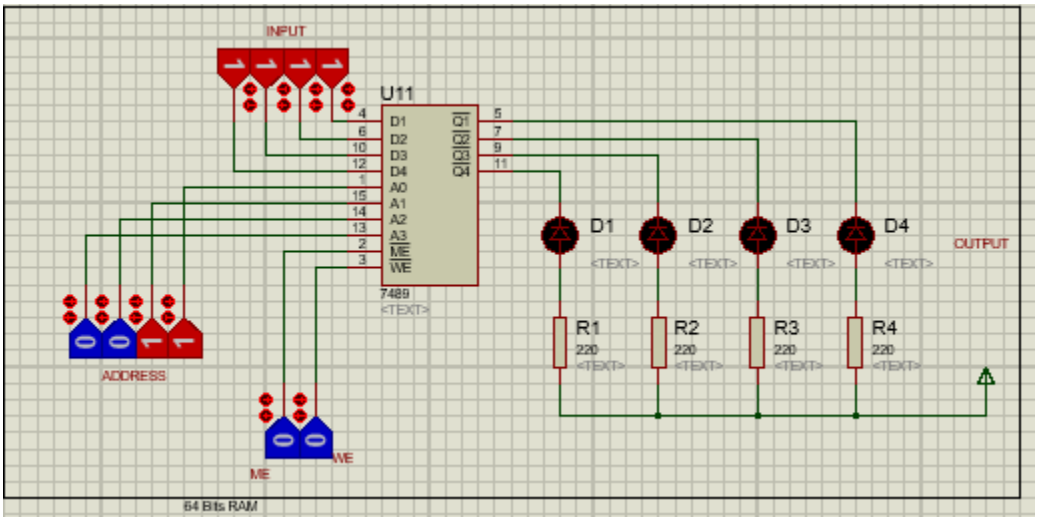
Input				Output		
E1	S1	D2	D1	F3	F2	F1
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Table 7.1

RAM IC circuit that save 64 bits

We used Portus of course to create the circuit, we needed the input buttons and the 64 bits IC and resistors to reduce the power and a 5V supply to connect the output which are LEDs instead of logicp buttons .

I'll share a picture of the circuit



Then I simulated it and got results like this:

Address				Write						Read					
A3	A2	A1	A0	$\overline{ME}$	$\overline{WE}$	D4	D3	D2	D1	$\overline{ME}$	$\overline{WE}$	F4	F3	F2	F1
0	0	0	0	↓	0	0	0	0	0	↑	1	0	0	0	0
0	0	0	1	↓	0	0	0	0	1	↑	1	0	0	0	1
0	0	1	0	↓	0	0	0	1	0	↑	1	0	0	1	0
0	0	1	1	↓	0	0	0	1	1	↑	1	0	0	1	1
0	1	0	0	↓	0	0	1	0	0	↑	1	0	1	0	0
0	1	0	1	↓	0	0	1	0	1	↑	1	0	1	0	1
0	1	1	0	↓	0	0	1	1	0	↑	1	0	1	1	0
0	1	1	1	↓	0	0	1	1	1	↑	1	0	1	1	1
1	0	0	0	↓	0	1	0	0	0	↑	1	1	0	0	0
1	0	0	1	↓	0	1	0	0	1	↑	1	1	0	0	1
1	0	1	0	↓	0	1	0	1	0	↑	1	1	0	1	0
1	0	1	1	↓	0	1	0	1	1	↑	1	1	0	1	1
1	1	0	0	↓	0	1	1	0	0	↑	1	1	1	0	0
1	1	0	1	↓	0	1	1	0	1	↑	1	1	1	0	1
1	1	1	0	↓	0	1	1	1	0	↑	1	1	1	1	0
1	1	1	1	↓	0	1	1	1	1	↑	1	1	1	1	1

Table 7.2

That’s every register give the same value stored in it

Implement 256 bit RAM

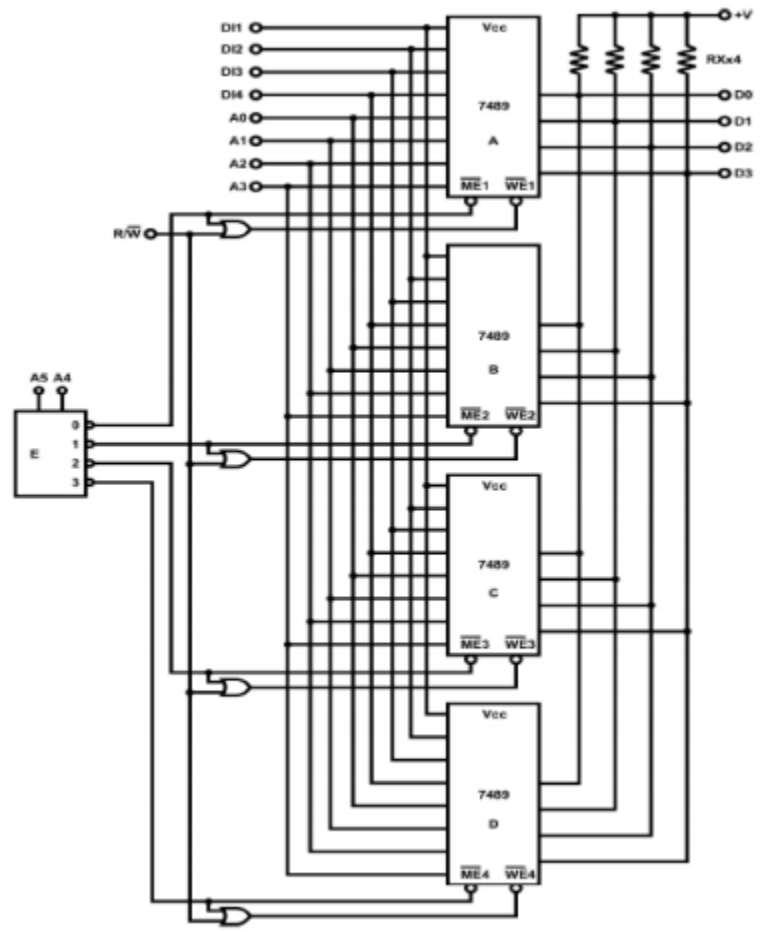
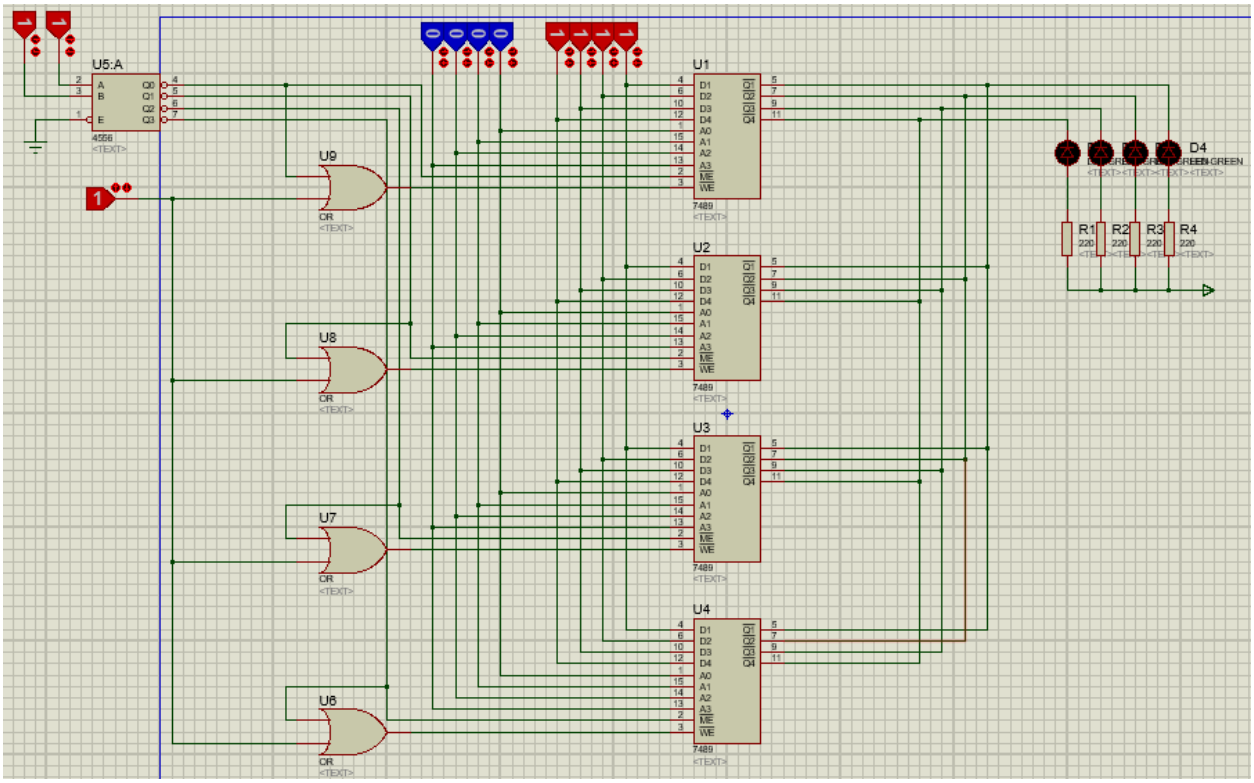


Figure 7.4

We need 4 ICs and decoder and the OR gate to get only 00 & 01 bits inputs for WM & WE

We used the same for one IC with small edits

here is a screenshot of the circuit



Here we connected them vertically that's means the memory got bigger space with the same extend for the one IC (4 bits) but now we have 256 bits instead of 64 bits.

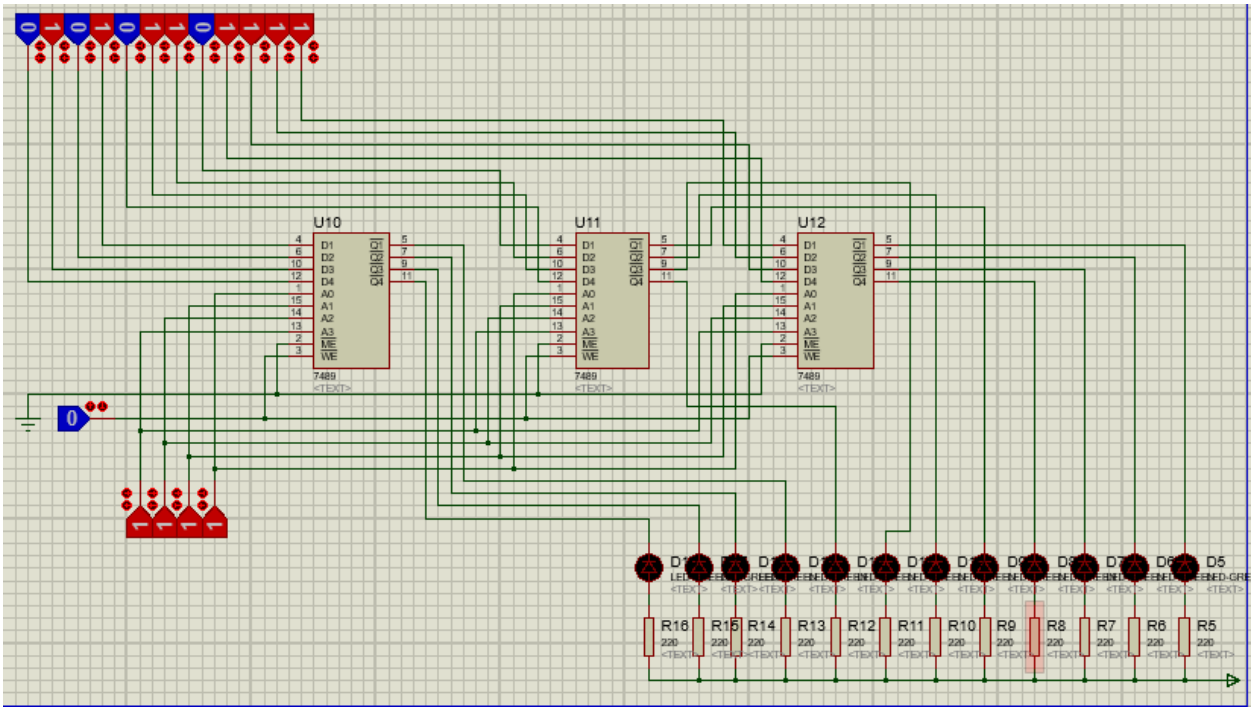
The outputs are the same as inputs for every address as we have 4 bits address and 2 bits for selection and one bit to chose to read or right data.

Implement 192 bit RAM, as 12*16 bit RAM

We need 3 ICs, the ground is to get only 00 & 01 bits inputs for WM & WE

We used the same for one IC with small edits

here is a screenshot of the circuit



Here we connected them horizontally that's means the memory got the same space with extended long (12 bits) but we still have the same storage of 64 bits.

The outputs are the same as inputs for every address as we have 12 bits address b and one bit to choose to read or right data.

Conclusion

In this experiment I became familiar with Storages and how they save data in it and how to make the number I want of bits to store even vertically or horizontally and how to connect them together. And I have verified the correctness of my work by using the simulation button for every circuit.

References

In this expirement I didn’t get help from outside the ALL Experimets PDF, only my own work.

Appendix

I didn’t get help from outside the ALL Experimets PDF, only my own work.