Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Digital Lab (ENCS2110)

Experiment No.4 Pre-Lab

Title: Digital Circuits Implementation using Breadboard

Prepared by:

Name: Aya Dahbour	**Number:** 1201738
Instructor: Dr. Bilal karaki	**TA:** Eng. Ali Hamodeh
Section: 2	**Date:** April 14 2022

Q3:

Design and Implement a Full Adder using the gates on the chips. Your final circuit must include the IC's, their pin numbers, and the connections between the pins.

In the previous experiments, we derived the Boolean expression of the outputs of Full adder circuit using both truth table and K-map. Meanwhile, the Full adder circuit has three-inputs and two-outputs.

Inputs: A, B, Cin (previous carry).

Outputs: F (summation), Cout (carry).

$F = A \oplus B \oplus Cin$

$Cout = (A \cdot B) + \big(Cin \cdot (A \oplus B)\big)$

It's noticeable for F we need two XOR gates, the first for $(A \oplus B)$ and the second, the result of $(A \oplus B)$ and C which is F.

Cout is a bit harder, we need AND gate (AB), XOR gate $(A \oplus B)$, and another AND gate $\big(Cin \cdot (A \oplus B)\big)$, then we need or gate.

These steps are shown clearly in Figure 1.

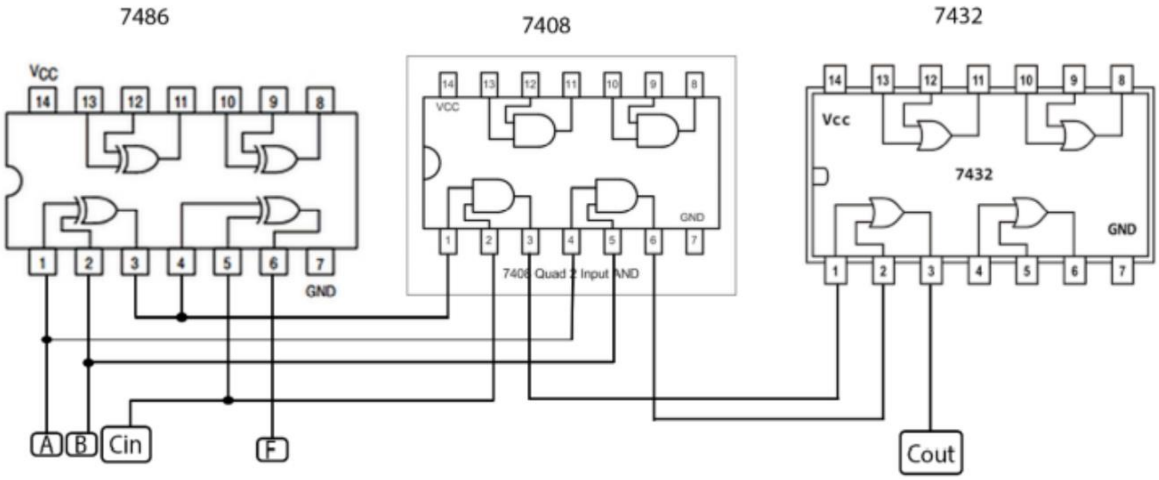


Figure 1: Full adder circuit using breadboard

Q4:

Design and Implement a 4x1 multiplexer using the gates on the chips. Your final circuit must include the IC's, their pin numbers, and the connections between the pins.

Previously shown that the MUX has multiple inputs and one output. Also, it has selection lines.

The truth table of 4-to-1 MUX is:

Table 1: 4-to-1 MUX truth table

Input		Output
S1	S0	F
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Somehow, I0 is the first input of the MUX, I1 is the second, the third is I2 and I3 is the fourth.

From the truth table we deliver that the Boolean expression of the MUX is as follows.

$F = (S1' \cdot S0' \cdot I0) + (S1' \cdot S0 \cdot I1) + (S1 \cdot S0' \cdot I2) + (S1 \cdot S0 \cdot I3)$.

To design the 4-to-1 MUX using breadboard, we need AND IC, OR IC, and NOT IC.

By implementing the Boolean expression F using breadboard ICs, the final circuit is as shown in Figure 2.

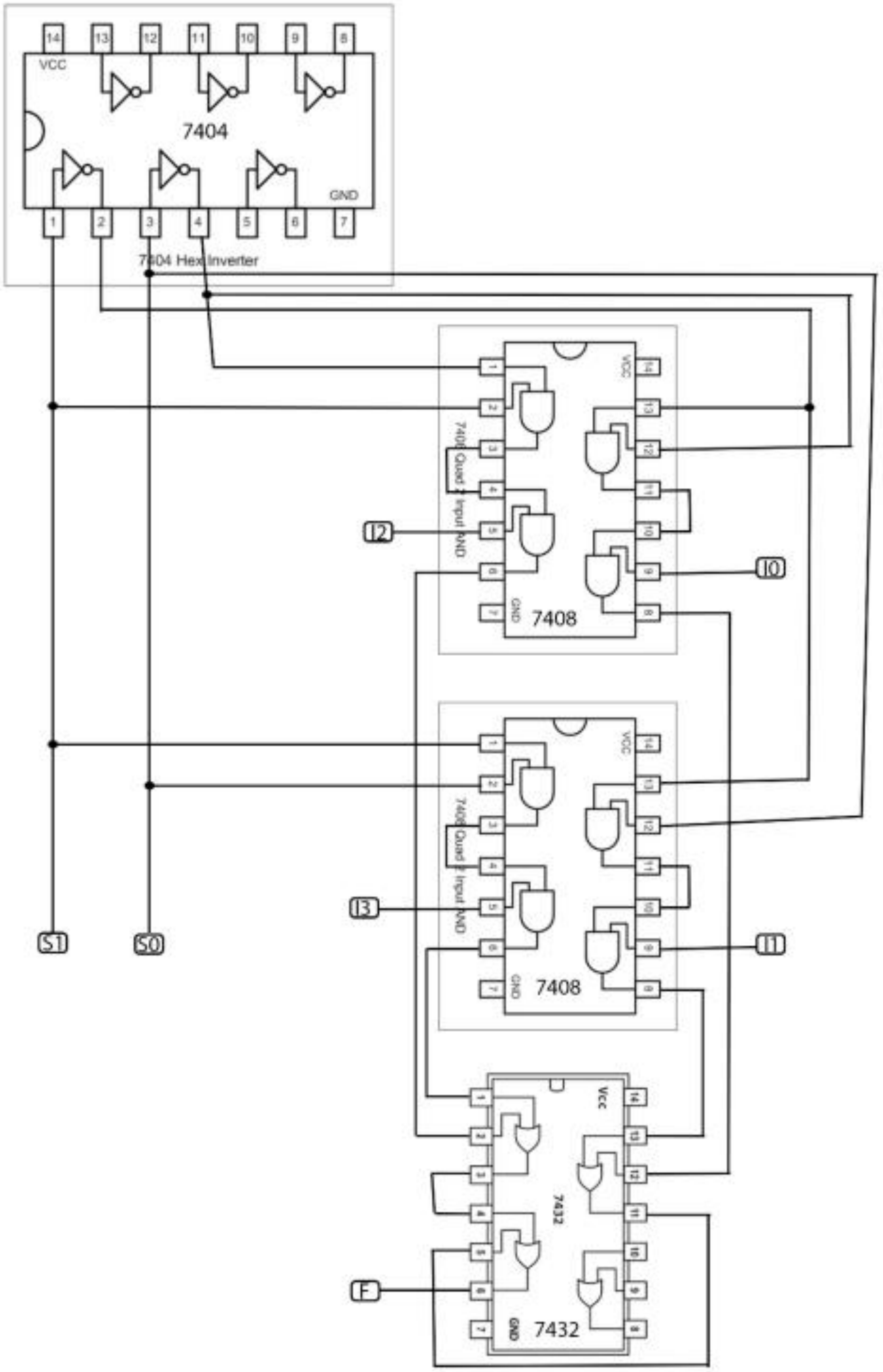


Figure 2: 4-to-1 MUX circuit using breadboard

Q5:

Design and Implement a 2-4 “active-low” decoder using the gates on the chips. Your final circuit must include the IC’s, their pin numbers, and the connections between the pins.

As mentioned in past experiments, the 2-4 decoder, has 2 inputs and 2^2 outputs. Using the truth table and K-map, we can find the Boolean expressions for the decoder outputs.

Table 2: 2-to “active-low” decoder truth table

Input		Output			
I1	I0	D3	D2	D1	D0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$D3 = (I1 \cdot I0)'$

$D2 = (I1' \cdot I0)'$

$D1 = (I1 \cdot I0')'$

$D0 = (I1' \cdot I0')'$

Basically, the Boolean expressions were derived using K-map, or by simplifying it manually.

So, to construct this circuit we need NOT IC and NAND IC. And, the circuit is as in Figure 3.

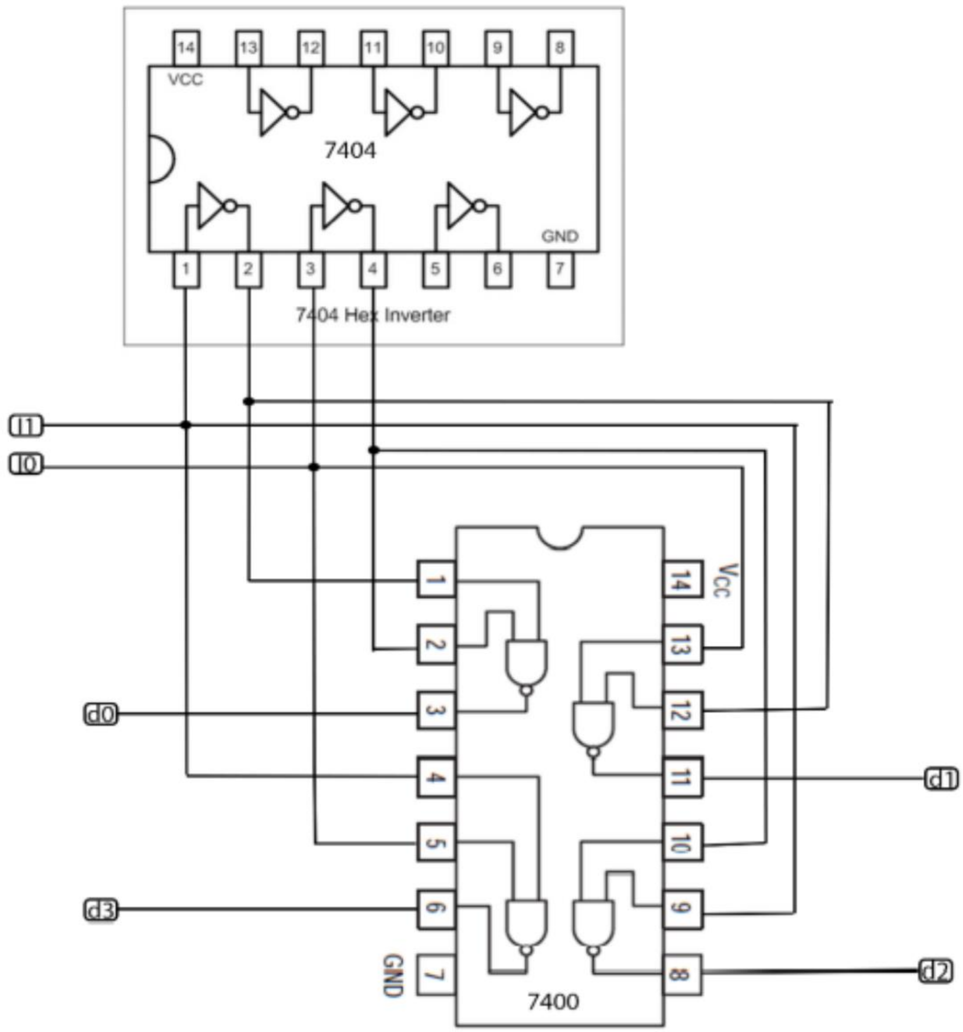


Figure 3: 2-4 "active-low" decoder using breadboard

Q6:

Use the just constructed 4x1 multiplexer to design a three-input network that gives 1 if the majority of its inputs are 1 and outputs a zero otherwise.

Table 3: 4-to-1 MUX with Boolean function truth table

Input			Output	
A	B	C	F	
0	0	0	0	0
0	0	1	0	
0	1	0	0	C
0	1	1	1	
1	0	0	0	C
1	0	1	1	
1	1	0	1	1
1	1	1	1	

The MUX circuit is already designed. We only change the inputs.

I0 = 0, I1 = I2 = C, I3 = 1.

0 means GND, 1 means Vcc. But, in the design I placed it 0 and 1.

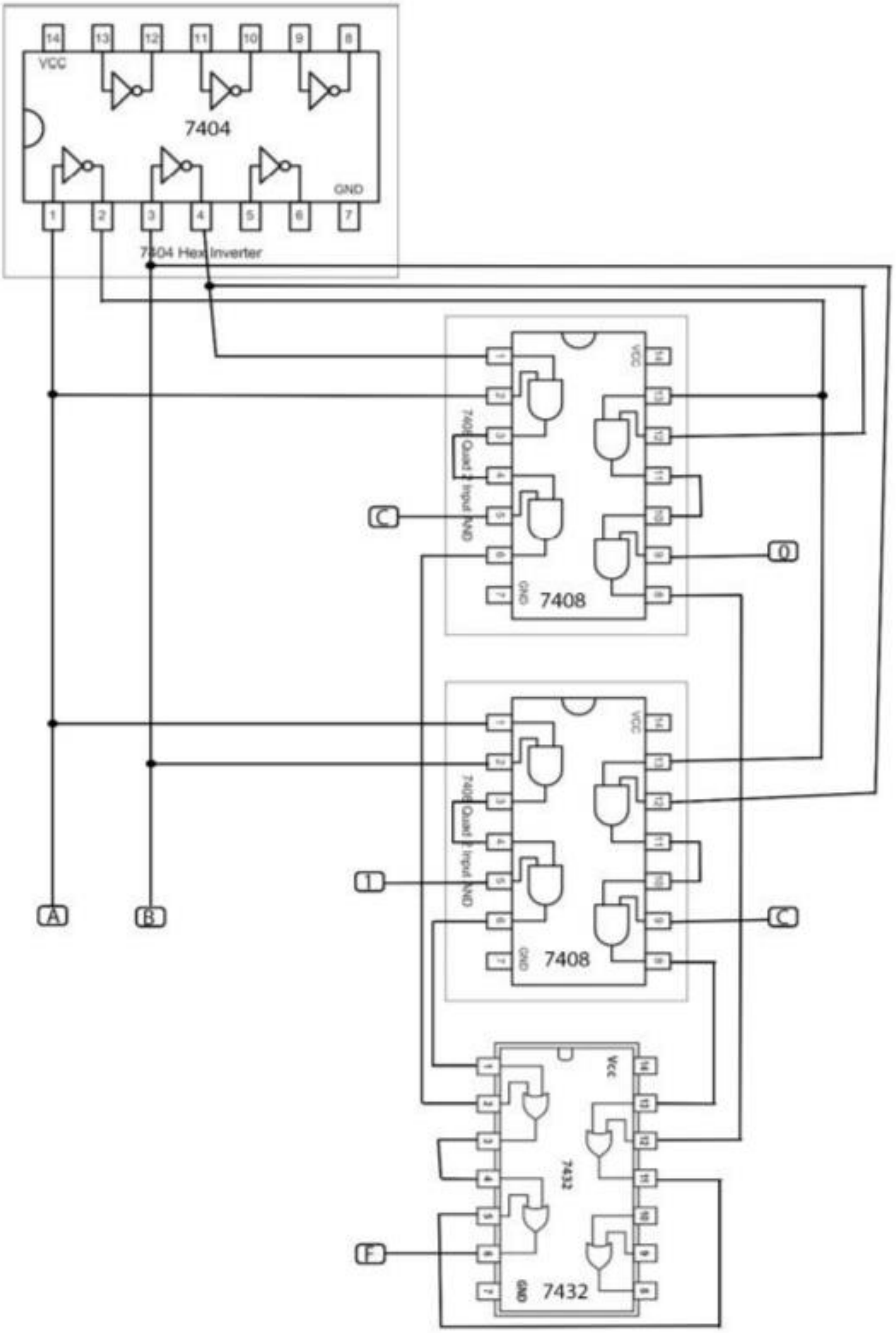


Figure 4: Boolean function circuit using 4-to-1 MUX by breadboard

Q7:

(Optional??) Use the 2-4 decoder to implement a 2-input function that acts like an equivalence gate: gives 1 on the output if both inputs are equal.

Table 4: 2-4 decoder truth table

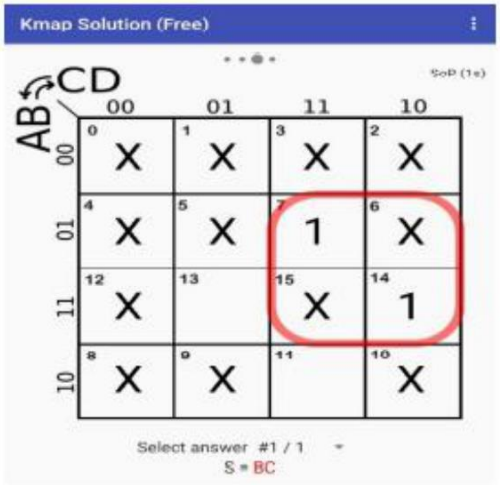
Input		Output			
I1	I0	D3	D2	D1	D0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Now, deriving the truth table of the function using the output of the decoder:

Table 5: 2-4 decoder with Boolean function truth table

Decoder outputs				F Input		Output
D3	D2	D1	D0	A	B	F
1	1	1	0	0	0	1
1	1	0	1	0	1	0
1	0	1	1	1	0	0
0	1	1	1	1	1	1

The decoder input in table 5, are 16 bits, only 4 values are used, others are unused. So, using the K-map for the following function, we get:



Kmap Solution (Free)

SoP (1s)

AB \ CD	00	01	11	10
00	0 X	1 X	3 X	2 X
01	4 X	5 X	7 1	6 X
11	12 X	13	15 X	14 1
10	8 X	9 X	11	10 X

Select answer #1 / 1

S = BC

Figure 5: Boolean function K-map with decoder outputs

F in terms of Decoder output: F = D2 · D1. So, by adding AND IC to the decoder implemented in Q6 we get:

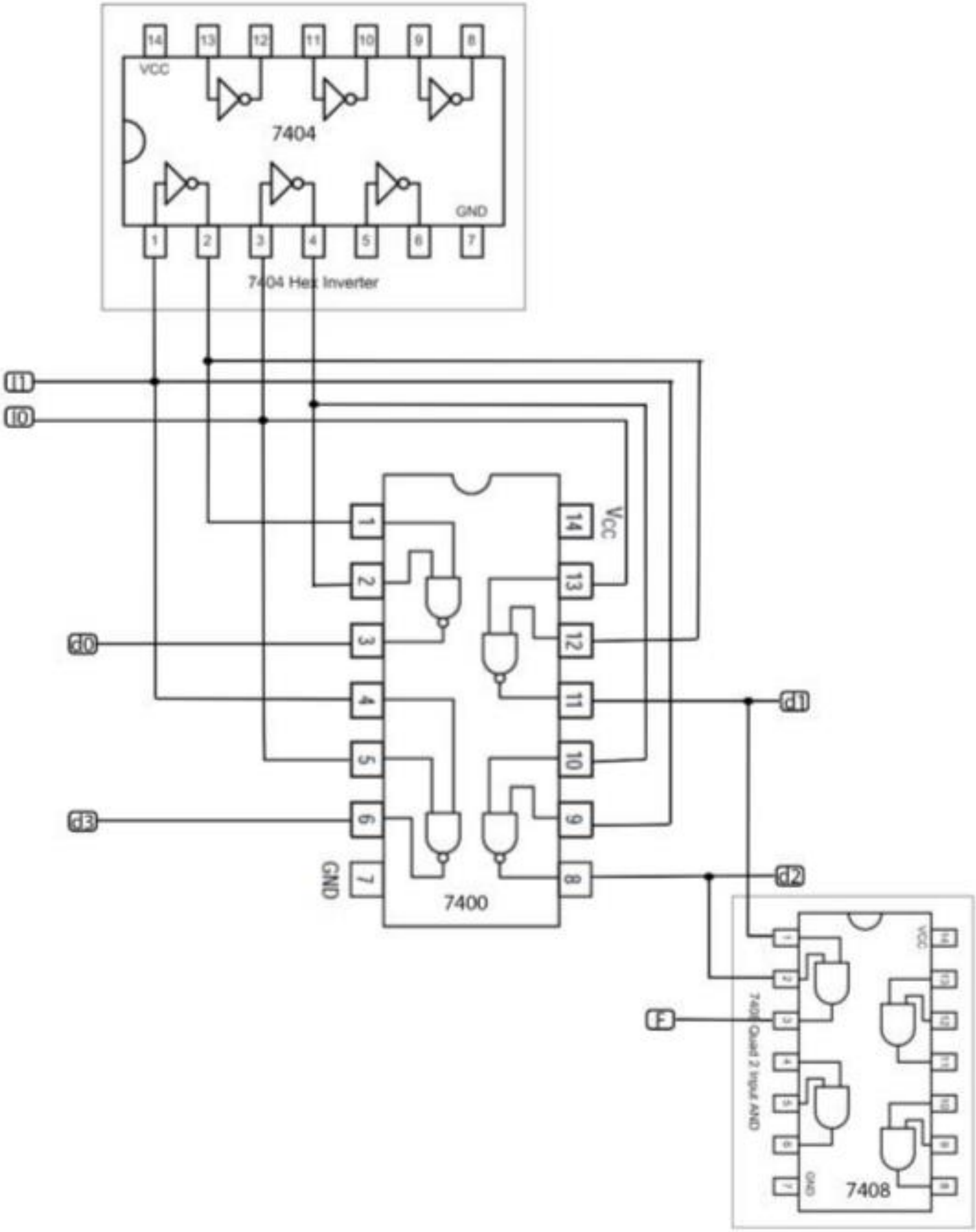


Figure 6: Boolean function circuit using breadboard