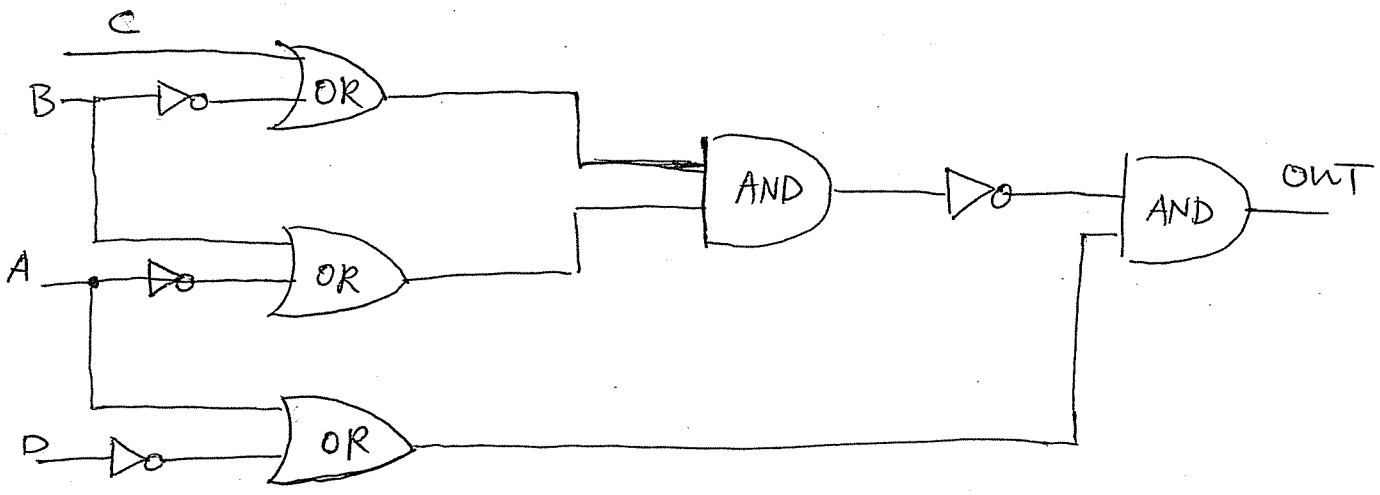
1. Draw a logic circuit corresponding to the following logic expression.
   NOT((NOT(A) OR B) AND (C OR NOT(B))) AND (NOT(D) OR A)



2. For the transistor-level circuit in Figure 1, fill in the truth table. What is Z in terms of A and B?

A	B	C	D	Z
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

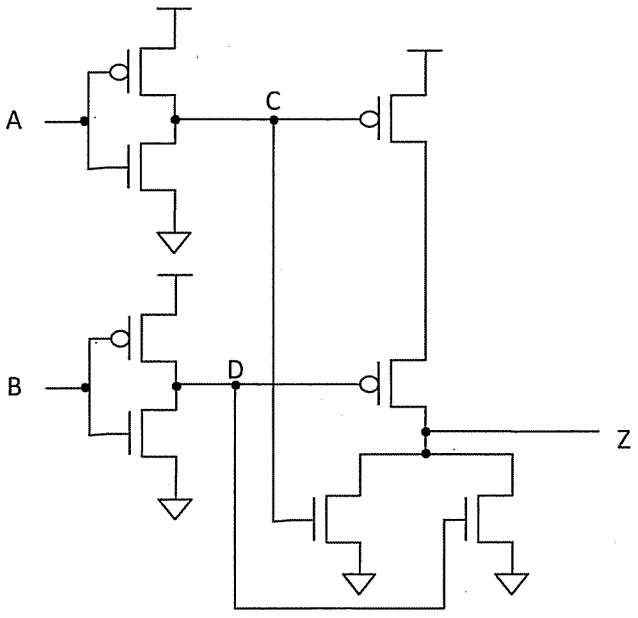


Figure 1

3. Fill in the truth table for the logical expression NOT(A) AND NOT(A OR B).

A	B	NOT(A) AND NOT(A OR B)
0	0	1
0	1	0
1	0	0
1	1	0

4. For this question, refer to the figure below

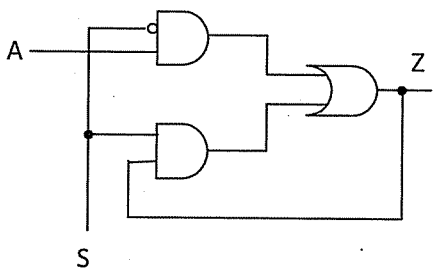


Figure 2

a. Describe the output of this logic circuit when the select line S is a logical 0. That is , what is the output Z for each value of A?
Answer: Z equals to A

b. If the select line S is switched from a logical 0 to 1, what will the output be?
Answer: Z equals to the previous value of A before S is switched.

c. Is this logic circuit a storage element?
Answer: Yes. When S is 0, Z will be the same value as A; whenever S is switched from 0 to1, the value of A will be latched in Z.

5. How many different memory locations can we have with 32-bit address? If the memory is byte-addressable, how many bits does the memory hold?
2^{32} different locations and $2^{32} \times 8 = 2^{35}$ bits

6. Given the logic circuit in Figure 3, fill in the truth table for the output value Z.

A	B	C	Z
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	0

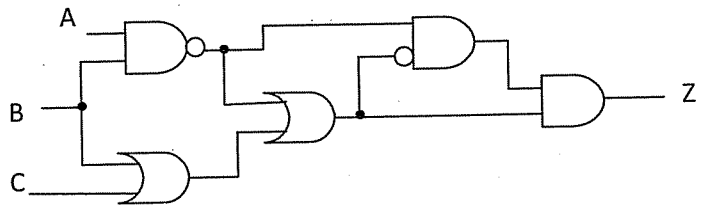


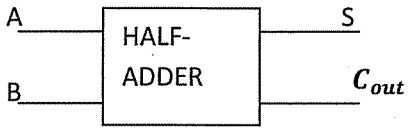
Figure 3

7. If A and B are four-bit unsigned binary numbers, 0111 and 1011, complete the table obtained when using a two-bit full adder(as in Page62), calculate each bit of the sum, S, of A and B. Check your answer by adding the decimal value of A and B and comparing the sum with S. Are the answers the same? Why or Why not?

C_{in}	**1**	**1**	**1**	**0**
A	0	1	1	1
B	1	0	1	1
S	0	0	1	0
C_{out}	1	1	1	1

In decimal format: $7 + 18 \neq 2$. The results are not the same because an overflow happened.

8. Half-adder is another kind of logic circuits that can perform binary addition. The difference between a full-adder and a half-adder is that a half-adder does not take the carry as input. A typical half-adder is illustrated in Figure 4. In fact, one way to build a full-adder is to use half-adders as basic building blocks. Now try to build a full-adder by drawing the necessary connections in Figure 5. (Note that you need one more 2-inputs gate ('AND' or 'OR') to finish the job. Feel free to choose one and add it in the box.)



A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure 4

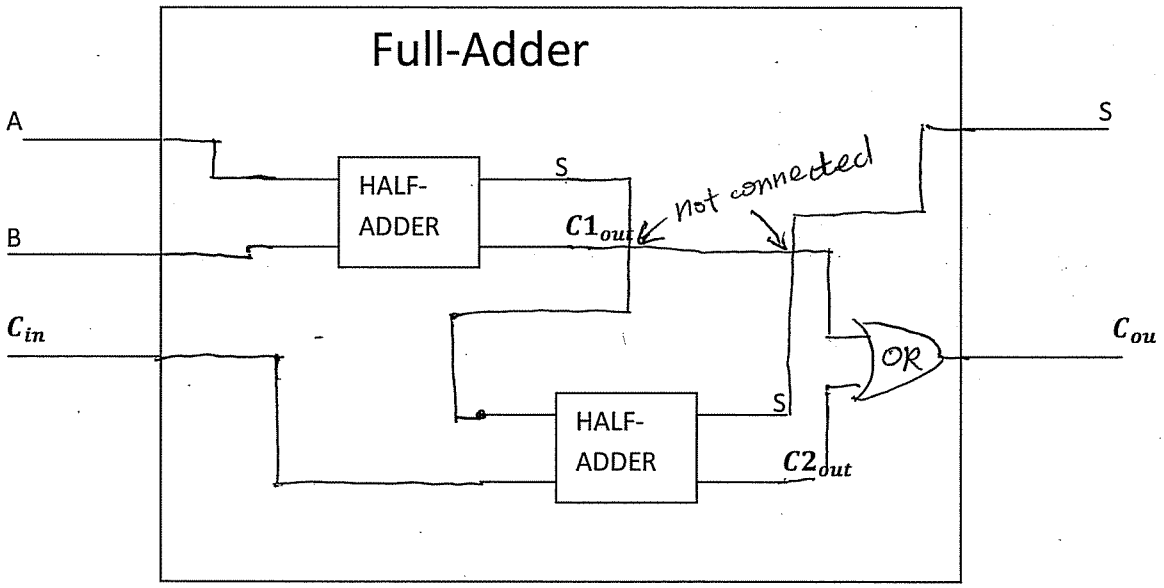


Figure 5