CS-2110 A/B/C Quiz 1 (C)

Aaryan Vinay Potdar

TOTAL POINTS

96 / 100

QUESTION 1

1 1A **4 / 4**

✓ **+ 4 pts** *Correct (1010 0001, no overflow)*

+ 2 pts ANS or Overflow incorrect

+ 0 pts incorrect

QUESTION 2

2 1B **4 / 4**

✓ **+ 4 pts** *Correct (1100 1011, yes overflow)*

+ 2 pts ANS or Overflow incorrect

+ 0 pts incorrect

QUESTION 3

3 1C **6 / 6**

✓ **+ 6 pts** *Correct (255)*

+ 0 pts Incorrect

+ 2 pts Partial - Binary (11111111)

QUESTION 4

4 1D **6 / 6**

✓ **+ 6 pts** *Correct (6, 6)*

+ 3 pts Partially correct - one answer correct

+ 0 pts Incorrect

QUESTION 5

5 1E **6 / 6**

✓ **+ 6 pts** *Correct (0xFFB7, 0x003F)*

+ 3 pts Partially correct - one answer correct

+ 2 pts Partial credit - both correct but in binary

+ 0 pts Incorrect

QUESTION 6

1F 24 pts

6.1 i **4 / 4**

✓ **+ 4 pts** *Correct (True)*

+ 2 pts Partially Correct (True, wrong explanation)

+ 0 pts incorrect

6.2 ii **4 / 4**

✓ **+ 4 pts** *Correct (True)*

+ 2 pts Partially Correct (True, wrong explanation or false, right explanation)

+ 0 pts Incorrect

6.3 iii **4 / 4**

✓ **+ 4 pts** *Correct (True)*

+ 2 pts Partially Correct (True, wrong explanation)

+ 0 pts incorrect

6.4 iv **0 / 4**

+ 4 pts Correct (True)

+ 2 pts Partially correct (True + Wrong explanation)

✓ **+ 0 pts** *Incorrect*

6.5 v **4 / 4**

✓ **+ 4 pts** *Correct (False)*

+ 2 pts Partially correct (False + Wrong explanation)

+ 0 pts Incorrect

6.6 vi **4 / 4**

✓ **+ 4 pts** *Correct (False)*

+ 2 pts Partially correct (False + Wrong explanation)

+ 0 pts incorrect

QUESTION 7

7 2A **8 / 8**

✓ **+ 8 pts** *Fully Correct (0,0,0,0,0,0,0,0)*

+ 6 pts 1 row incorrect

+ 4 pts 2 rows incorrect

+ 2 pts 3 rows incorrect

+ 4 pts Inverse

+ 0 pts 4 or more row incorrect

QUESTION 8

8 2B **8 / 8**

✓ **+ 8 pts** *Correct (NOT)*

+ 4 pts incorrect(NAND)

+ 0 pts incorrect

QUESTION 9

9 2C **9 / 9**

✓ **+ 9 pts** *Correct*

![Screenshot_2023-09-13_at_8.10.16_PM.png](/files/da1c10f8-403c-48f6-8fcb-cb0282643545)

+ 0 pts incorrect

+ 6 pts Mostly correct (e.g. no output wire, switched n/p types). Drawing a NAND doesn't count (wrong)

QUESTION 10

10 3A **8 / 8**

✓ **+ 8 pts** *Fully Correct (1,0,0,1,1,1,0,1)*

+ 4 pts Inverse

+ 6 pts 1 row incorrect

+ 4 pts 2 rows incorrect

+ 2 pts 3 rows incorrect

+ 0 pts 4 or more row incorrect

QUESTION 11

11 3B **8 / 8**

✓ **+ 8 pts**

![Screenshot_2023-09-13_at_8.17.40_PM.png](/files/c5a79ac4-70ba-4939-b59a-f9cea4b0aaee)Correct (Example or equivalent)

+ 0 pts incorrect

QUESTION 12

12 3C **9 / 9**

✓ **+ 9 pts** *Correct (Example or equivalent)*

![Image_9-13-23_at_8.24_PM.png](/files/55b0e243-6e66-4a38-8282-782edc576e79)

+ 5 pts Structure is correct (e.g 2 muxes feeding into a 3rd mux), but inputs/positioning

+ 0 pts incorrect

Name [PRINT CLEARLY]: AARYAN POTDAR

GT username (e.g. gburdell3): apotdar 31

CS 2110: Computer Organization and Programming
Gupta/Conte/Adams Fall 2023

QUIZ 1
VERSION C

This exam is given under the Georgia Tech Honor Code System. Anyone found to have submitted copied work instead of original work will be dealt with in full accordance with Institute policies.

Georgia Tech Honor Pledge: *"I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community."*

[MUST sign:] AnP

- THIS IS A CLOSED BOOK, CLOSED NOTES EXAM
- NO CALCULATORS
- This examination handout has **5** pages.
- Do all your work in this examination handout.
- Use the back of the exam sheets if necessary.
- WHERE NEEDED, SHOW ALL YOUR INTERMEDIATE RESULTS TO RECEIVE FULL CREDIT

In case you forgot, here are some good facts to know:

Hex	Dec
0x1	1
0x2	2
0x3	3
0x4	4
0x5	5
0x6	6
0x7	7
0x8	8
0x9	9
0xA	10
0xB	11
0xC	12
0xD	13
0xE	14
0xF	15

x	2^x
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16,384
15	32,768
16	65,536

Problem	Points	Score
1	50	
2	25	
3	25	
TOTAL	100	

GOOD LUCK!

More good facts to know:

1K = 2^{10}
1M = 2^{20}
1G = 2^{30}
1T = 2^{40}
1P = 2^{50}
1E = 2^{60}

1. [50 pts] Answer the following short questions. Show your work (where needed) to receive full credit.

(a) Compute the following operation on two **2's complement 8-bit binary numbers**. Write your answer as a **2's complement 8-bit binary number**. Then circle "Yes" if the calculation results in overflow, otherwise circle "No." The word size is **8 bits**.

```
 1 1111
  1010 1001
+ 1111 1000
-----------
1|1010 0001
```

ANS: 1010 0001

Overflow (circle): Yes / [✓ No]

(b) Compute the following operation on two **2's complement 8-bit binary numbers**. Write your answer as a **2's complement 8-bit binary number**. Then circle "Yes" if the calculation results in overflow, otherwise circle "No." The word size is **8 bits**.

```
 01 11
  0101 1010
+ 0111 0001
-----------
  1100 1011
```

ANS: 1100 1011

Overflow (circle): [✓ Yes] / No

(c) Compute the following operation that uses bitwise operators on **unsigned 8-bit hex numbers**. Write your answer as an **unsigned decimal number**.

(0xA9 AND 0x8D) OR 0x7F

ANS: 255

A9 = 10101001

8D 10001101

1000 1001 OR 0111 1111

10001001 1111 1111 → $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 =

(d) In CMOS design, what is the **minimum** number of transistors needed to build:

(1) NAND3: 6 (3p, 3n)

(2) OR2: 6 (3p 3n)

(e) Sign-extend the following **2's complement 8-bit hexadecimal numbers** to **16** bits. Write your answer as a **2's complement 16-bit hexadecimal number**. We have already written the conventional hexadecimal prefix (0x) for you.

0xB7

1111 1111 10110111 → FF B7

ANS: **0x** FFB7

0x3F

ANS: **0x** 003F

3F

0000 0000 0011

(f) Answer the following true/false questions by circling "true" or "false," and then give a reason for each answer:

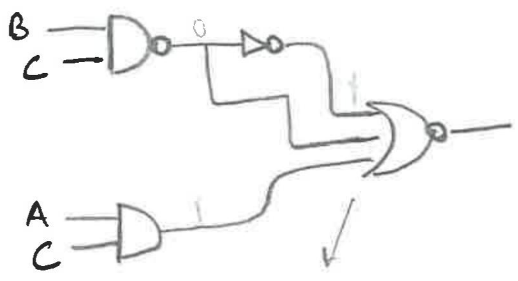
[✓ TRUE] or FALSE	A and B are numbers encoded in IEEE floating point. Because of the ordering of the sign bit, encoded-exponent and fraction fields in IEEE floating point, testing "is A is greater than B?" can be done by assuming A and B are already encoded as sign-magnitude integers. Why or why not? the first bit corresponds to the sign bit. A '0' indicates a +ve # and a 1 indicates a -ve number. Hence, sign-magnitude integer comparisons can be made.
[✓ TRUE] or FALSE	You cannot buy a "Turing Machine" (as Alan Turing conceived of it) at your local electronics store because it does ***not*** refer to a physical contraption that Turing built. Why or why not? A turing machine is run by another turing machine. A turing machine refers to a simple algo. Most electronics are composed multiple connected turing machines
[✓ TRUE] or FALSE	A 5-to-32 decoder has 5 selector bit lines. Why or why not? $n=5$ $2^5 = 32$ hence 5 selector bits. Each selector bit can have a selector bit line.
TRUE or [✓ FALSE]	Signed magnitude defines an encoding for both +0 and -0. Why or why not? Signed magnitude only encodes (+)0. There's no eg ⇒ 0000 → 0 → toggle bits 1111 add 1 +1 = 1 0000 → 0000. This is the representation for +ve 0.
TRUE or [✓ FALSE]	An 8-to-1 mux has 4 selector bit lines. Why or why not? $8 = 2^n \Rightarrow n = 3$. There will be 3 selector bits, hence 3 selector bit lines.
TRUE or [✓ FALSE]	I can represent 64 with two's complement and a word size of 7 bits. Why or why not? max # → $2^{7-1} - 1 = 2^6 - 1 = 64 - 1 = 63$ 64 is greater than 63, so it cannot be represented with 7 bit word size.

2. [25 pts] Answer the following questions about transistors. **Show your work.**

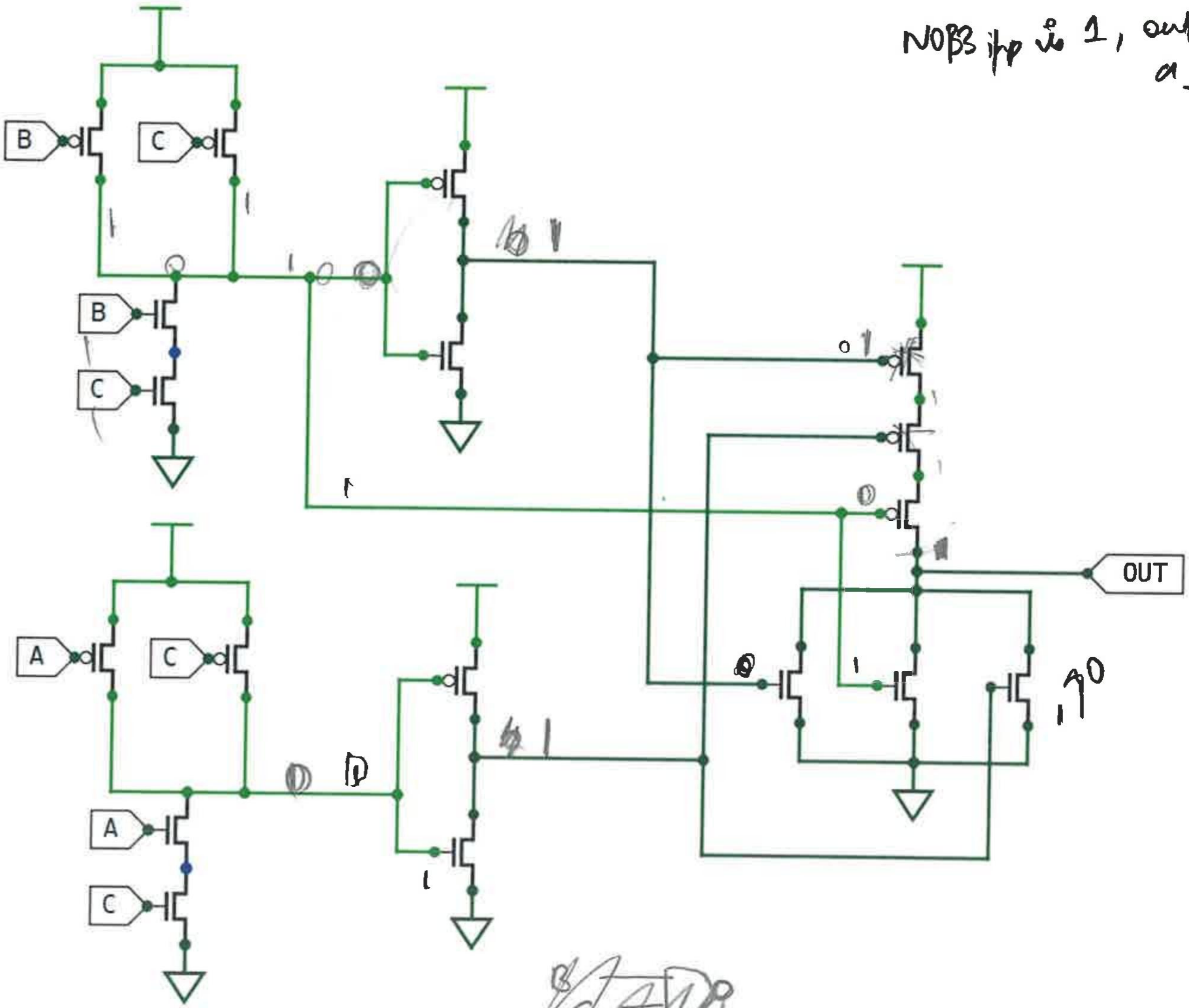
(a) Complete the truth table for the following CMOS transistor diagram:

A	B	C	OUT
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

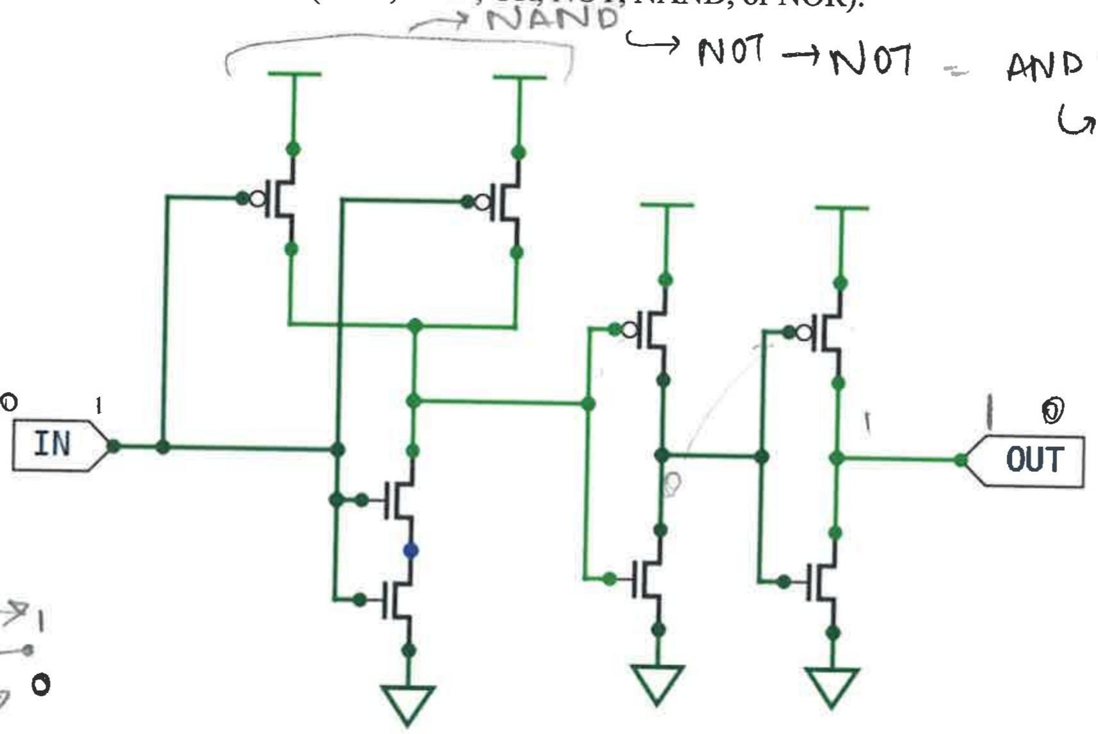
Gate diagram:



in all cases one of the NOR3 i/p is 1, outputting a 0.

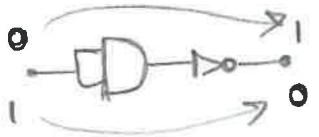


(b) Which logic gate does the following circuit correspond to? Your answer should be the name of a gate we covered in class (XOR, AND, OR, NOT, NAND, or NOR).



↳ NOT → NOT = AND → NOT

↳ NAND gate



Name of the gate this circuit performs: NOT

0→1

1→0

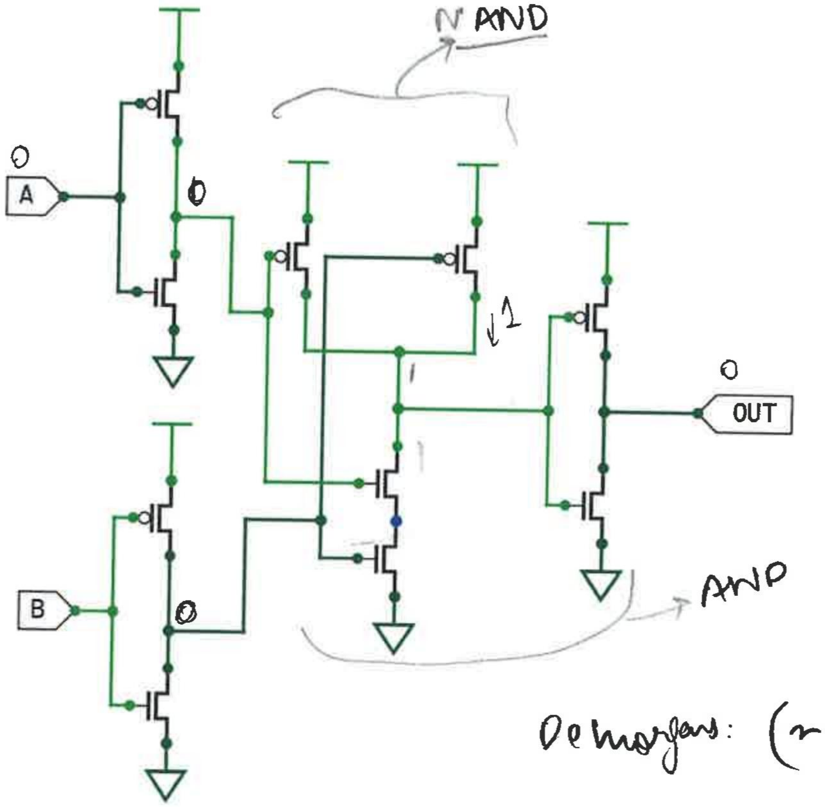
Explanation → the diagram represents a NAND gate followed by 2 NOT gates. This acts as a NAND gate. However, we only have 1 input. the input gets flipped in the NAND gate, behaving as a NOT Gate. Hence, it acts as a NOT gate.

A	B	O → NAND
0	0	1
0	1	1
1	0	1
1	1	0

(c) Re-draw the following circuit using only 4 transistors by applying DeMorgan's Laws.

NOR gate

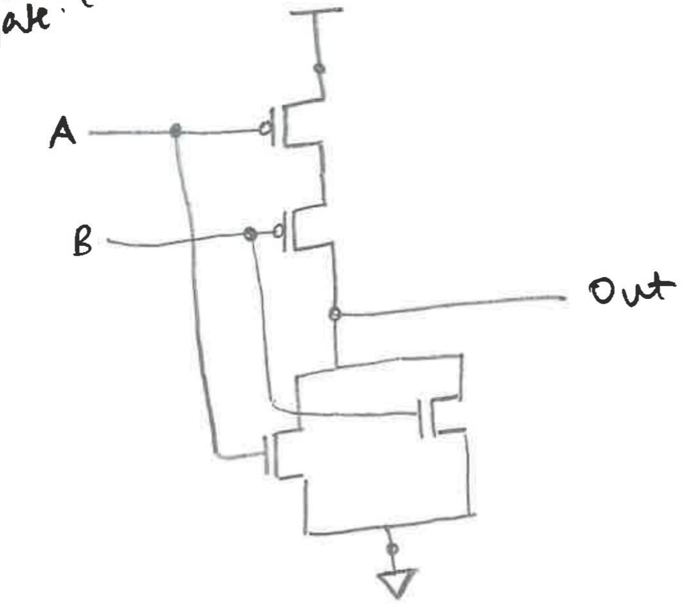
A	B	O
0	0	1
0	1	0
1	0	0
1	1	0



Demorgans: $(\sim A \cdot \sim B)$

$= \sim(A+B)$ → NOR gate.

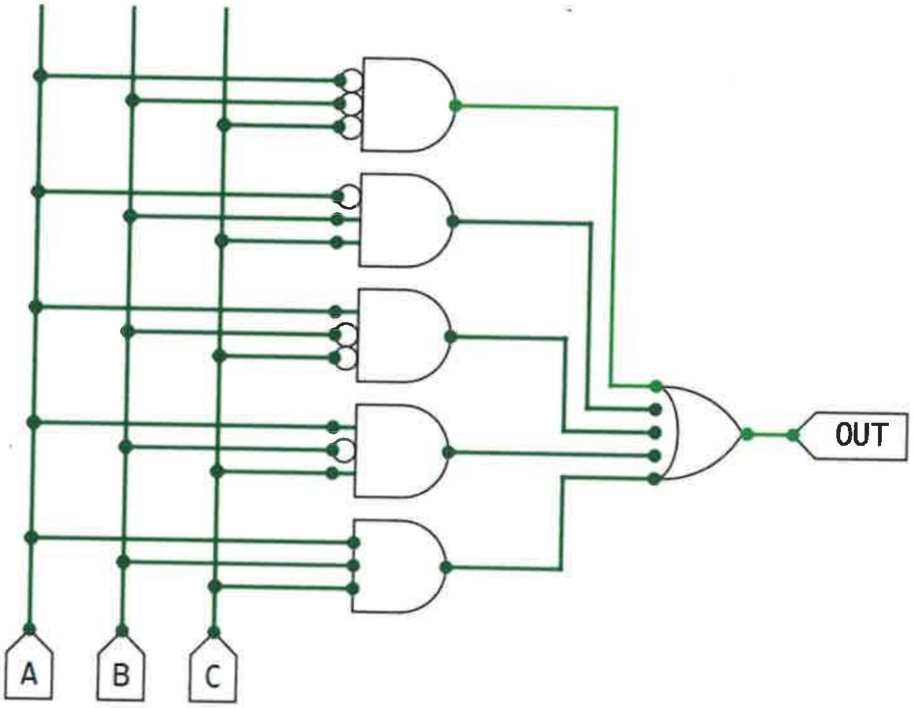
NOR gate: (4 transistors)



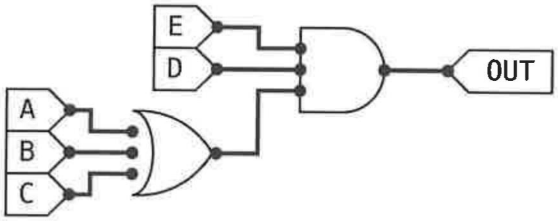
3. [25 pts] Answer the following questions about gates. **Show your work.**

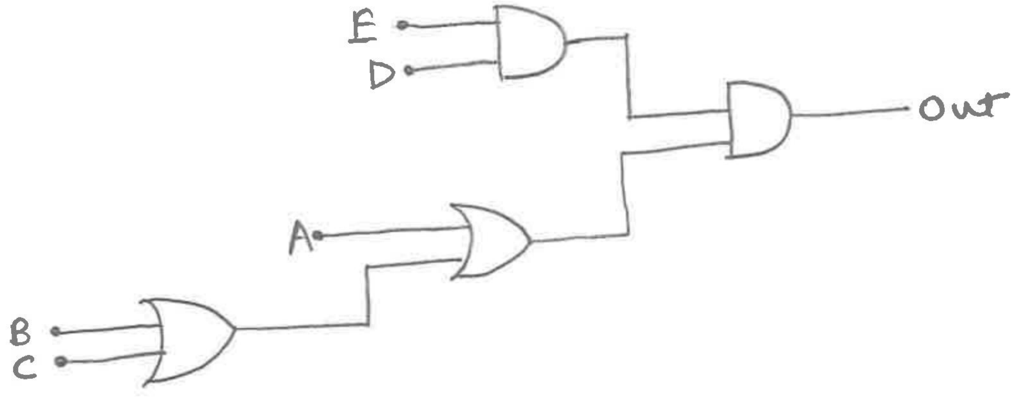
(a) Complete the truth table for the logic diagram as shown.

A	B	C	OUT
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



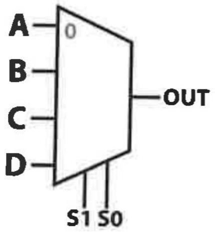
(b) Re-draw the following circuit using **only** AND and OR gates with two inputs (AND2 and OR2 gates). Do ***not*** attempt to minimize the logic or simplify it.





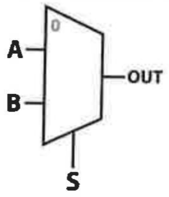
(c) Consider a 4-to-1 mux with 4 input signals for A, B, C, and D, and 2 selector signals S0 and S1. The symbol and a table that describes the behavior of a 4-to-1 mux follow:

S1	S0	Output
0	0	A
0	1	B
1	0	C
1	1	D

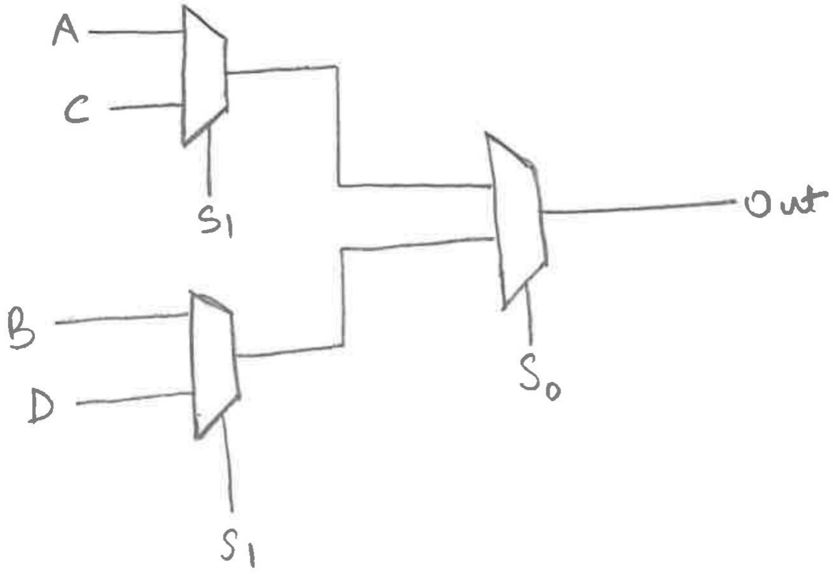


However, you only have 2-to-1 muxes and wires that you can use to build this 4-to-1 mux. The symbol and a table that describes the behavior of a 2-to-1 mux follow:

S	Output
0	A
1	B



Construct a 4-to-1 mux using only 2-to-1 muxes and wires:



check:

S1	S0		
0	0	→ A	✓
0	1	→ B	✓
1	0	→ C	✓
1	1	→ D	✓