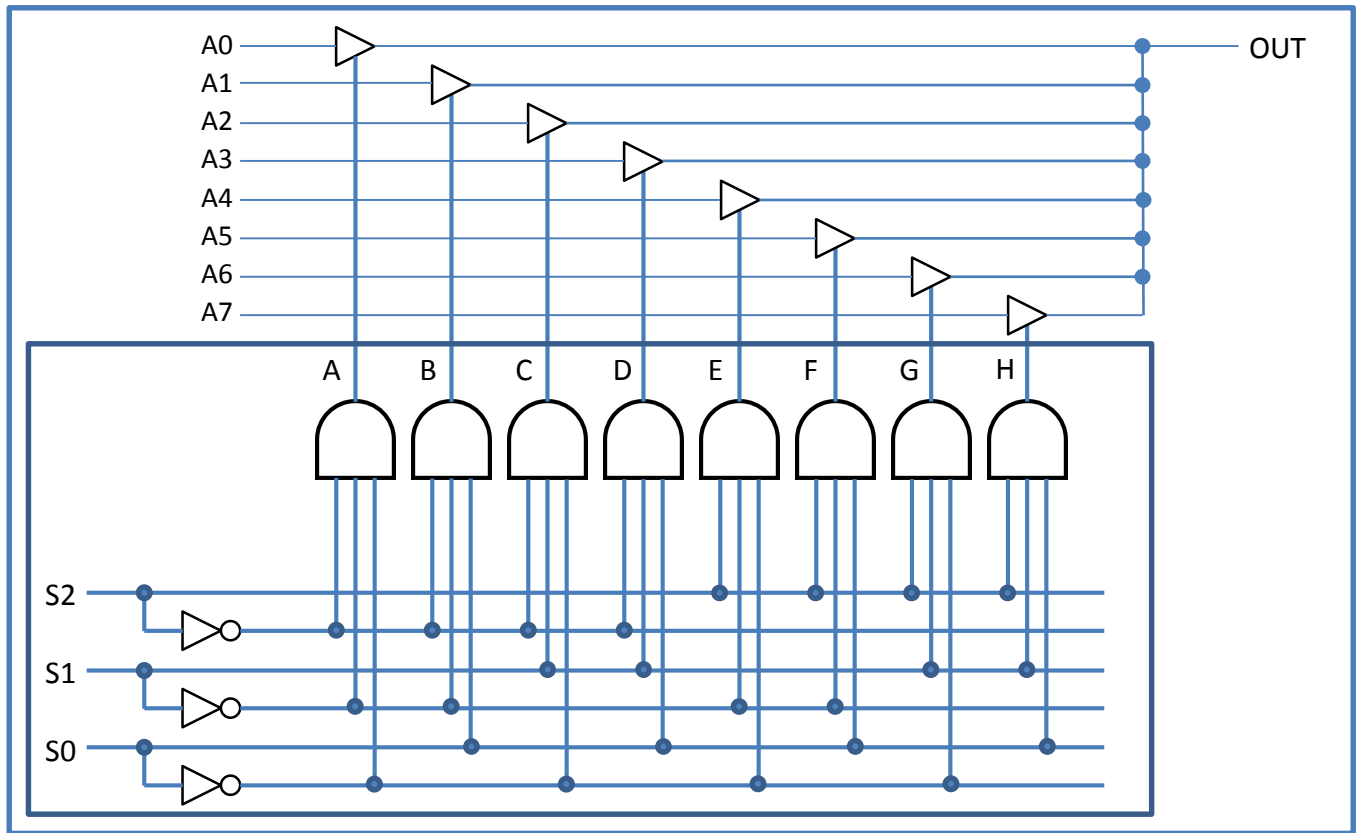


ECE 2504: Introduction to Computer Engineering  
Homework Assignment 4 (80 points)

Bowei Zhao

Problem 1 (10 points)

Consider the circuit shown below.



Problem 1 (continued)

a. Analyze the circuit in the smaller box by completing the truth table shown below.

S2	S1	S0	A	B	C	D	E	F	G	H
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

b. Based on its behavior, what kind of device does the circuit in the smaller box implement?

A decoder

c. Use the result of part (a) to analyze the circuit in the larger box; complete the truth table shown below.  
(The devices in the upper portion of the larger box are tri-state buffers.)

S2	S1	S0	A	B	C	D	E	F	G	H	OUT = ?
0	0	0	1	0	0	0	0	0	0	0	A0
0	0	1	0	1	0	0	0	0	0	0	A1
0	1	0	0	0	1	0	0	0	0	0	A2
0	1	1	0	0	0	1	0	0	0	0	A3
1	0	0	0	0	0	0	1	0	0	0	A4
1	0	1	0	0	0	0	0	1	0	0	A5
1	1	0	0	0	0	0	0	0	1	0	A6
1	1	1	0	0	0	0	0	0	0	1	A7

d. Based on its behavior, what kind of device does the circuit in the larger box implement?

A multiplexer that takes  $2^3(8)$  inputs with one output

Problem 2 (10 points)

A **composite number** is an integer that factors other than one and itself. For example:

- 8 is a composite number; it has factors 1, 2, 4, and 8.
- 11 is not a composite number; its only factors are 1 and 11.
- 0 is not a composite number; one of its factors is always itself.
- 1 is not a composite number; its only factor is 1.

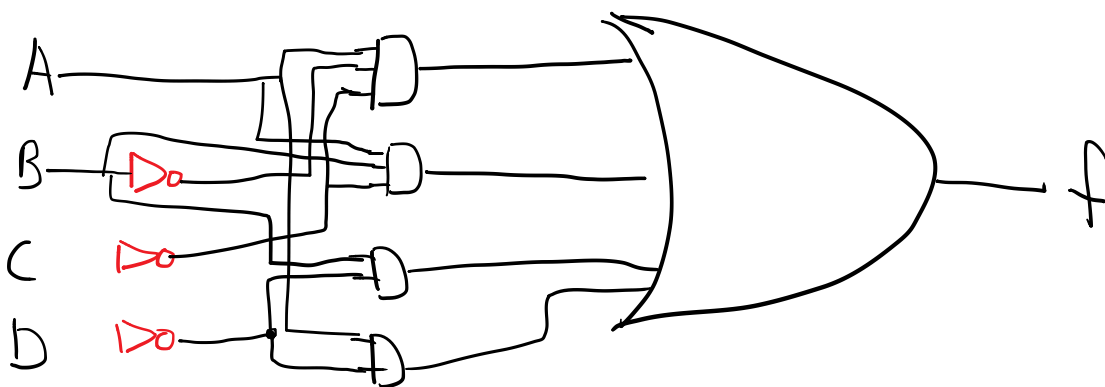
(11 is a prime number, while 0 and 1 are neither prime nor composite. For the purpose of this problem, it is sufficient to say that all three numbers are “not composite.”)

Design and implement a composite number detector circuit. The logic circuit should have four inputs and one output. When a four-bit input is applied to the circuit, the output should be one if the decimal equivalent of the binary number is a composite number, and zero if is not composite.

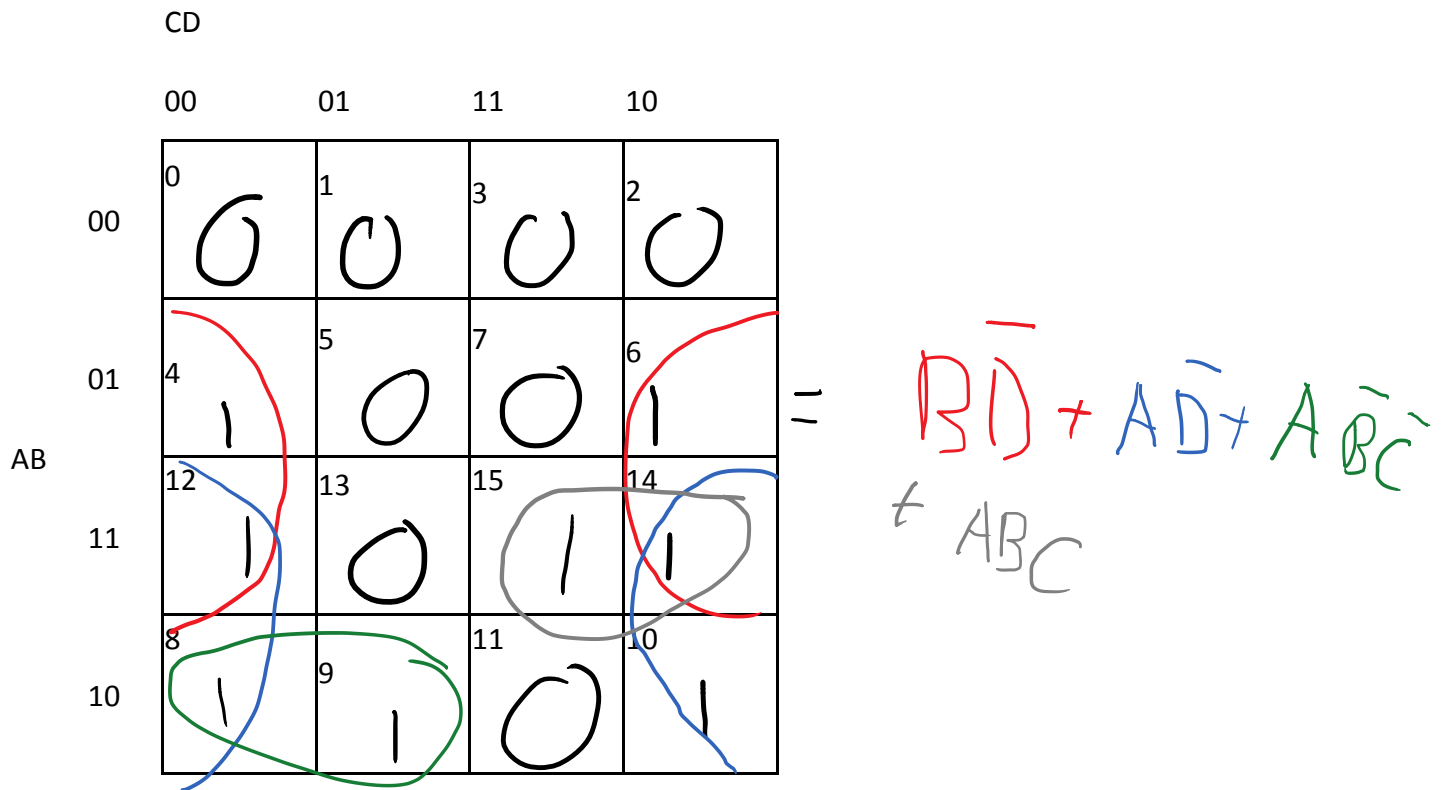
Show the truth table for the composite number detector.

INPUT A	INPUT B	INPUT C	INPUT D	OUTPUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

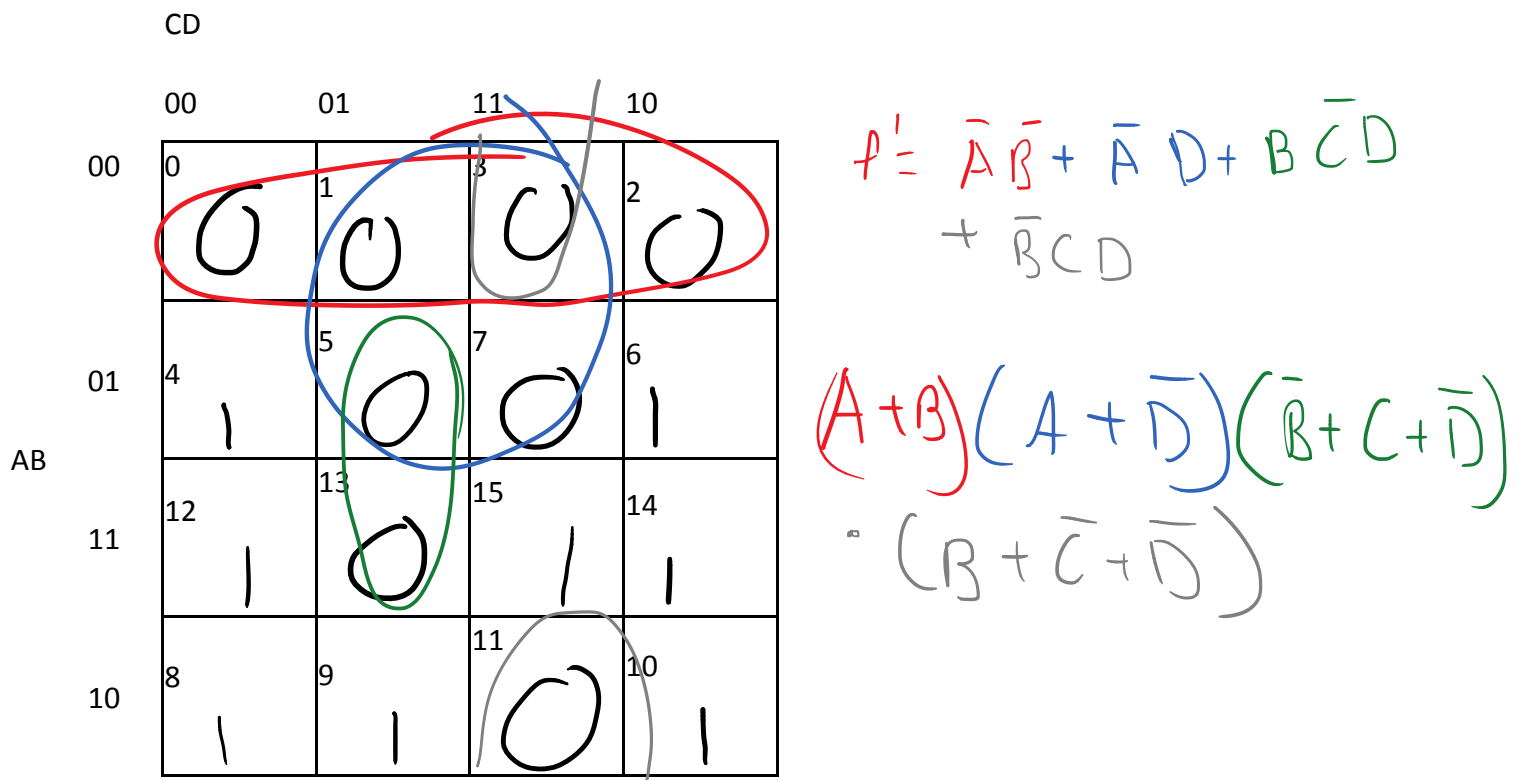
Implemented)  $B\bar{D} + A\bar{D} + A\bar{B}\bar{C} + ABC$



- a. Obtain the minimal sum-of-products implementation for the composite number detector circuit.



- b. Obtain the minimal product-of-sums implementation for the composite number detector circuit.



Problem 3 (10 points)

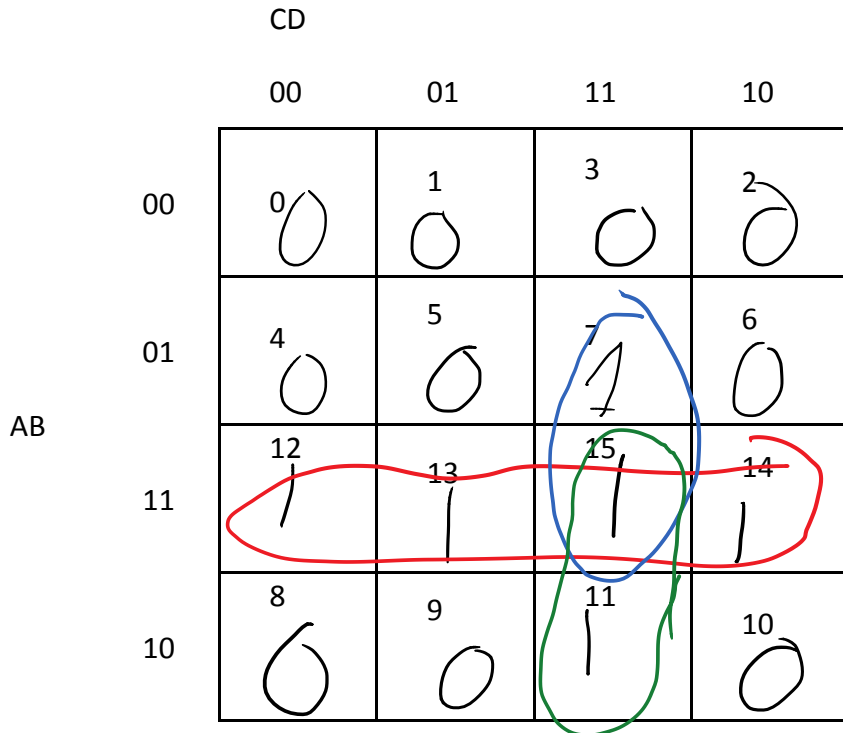
Computer engineers Adam and Barbara have two children, son Charles and daughter Diane. When the family wants to go out to dinner, someone first proposes a restaurant. Then, they vote on whether or not to go to that restaurant. If a majority of the family votes in favor of going to that restaurant, then they go. If a majority of the family does not vote in favor of going to that restaurant, then they don't go. Of course, Mom and Dad are paying for dinner, so the only exception to the aforementioned situation is when **both parents** vote in favor of going to the proposed restaurant, in which case the family also goes to that restaurant.

(Tonight's proposed restaurant happens to be "Professor McNasty's Rib Shack and Boolean Algebra Emporium," but that is actually immaterial to the story.)

- a. Create a truth table that shows all possible combinations of votes, and the outcome of each vote. For each family member, take "0" to indicate that he or she voted **no**, and take "1" to indicate that he or she voted **yes**. For the outcome, use a "0" to indicate that the family **is not going** to the proposed restaurant, and use a "1" to indicate that the family **is going** to the proposed restaurant.

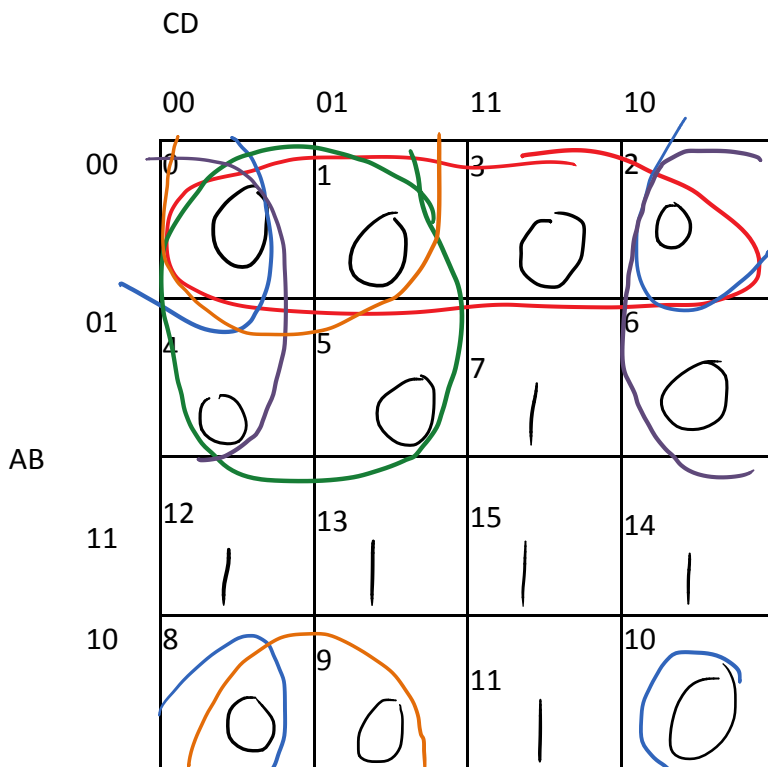
INPUT A = Adam	INPUT B = Barbara	INPUT C =Charles	INPUT D =Diane	OUTPUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- b. Using your truth table and a Karnaugh map, obtain the minimal sum-of-products function that expresses the conditions necessary for the family to go to Professor McNasty



$$= AB + BCD + AC$$

- c. Using your truth table and a Karnaugh map, obtain the minimal product-of-sums function that expresses the conditions necessary for the family to go to Professor McNasty's.



$$F = \overline{A}\overline{B} + \overline{B}\overline{D} + \overline{A}\overline{C} + \overline{A}\overline{D} + \overline{B}\overline{C}$$

$$POS = (A+B)(B+D)(A+C)(A+D)(B+C)$$

#### Problem 4 (20 points)

Standard binary-coded decimal is not the only way that we can encode decimal digits in binary. For example, we've seen the format called **excess-3 binary-coded decimal**. Another such format is called 2421 code. The name refers to the place value weighting of each place. (In this regard, we could call BCD "8421 code.")

Here is a table that shows the correspondence between decimal digits and their 2421 counterparts:

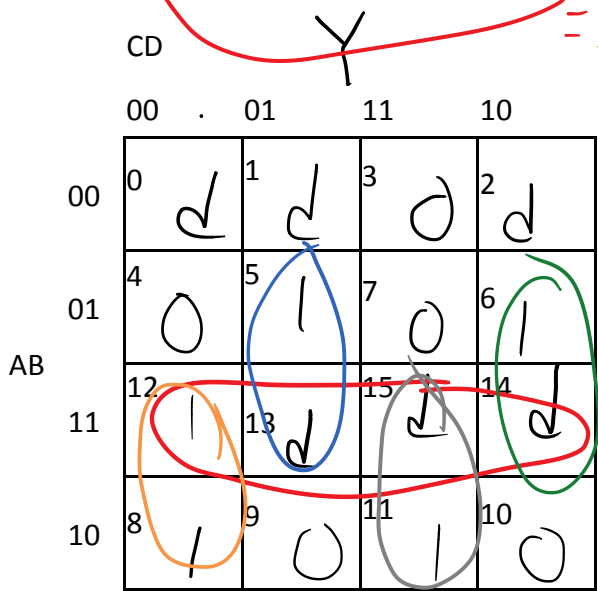
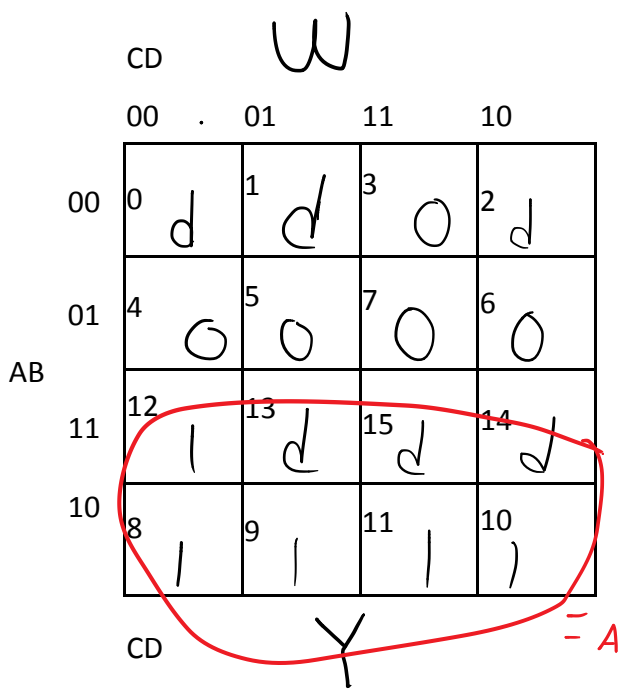
Decimal Digit	0	1	2	3	4	5	6	7	8	9
2421 code	0000	0001	0010	0011	0100	1011	1100	1101	1110	1111

There are actually two ways that we could encode certain digits, but the table shows the standard 2421 code. Notice that if two digits add up to 9, then their codes are complements of each other. (Excess-3 BCD actually has the same quality.)

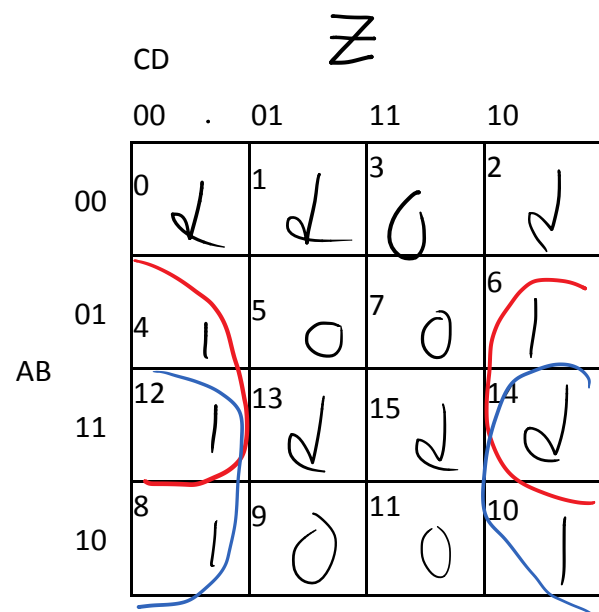
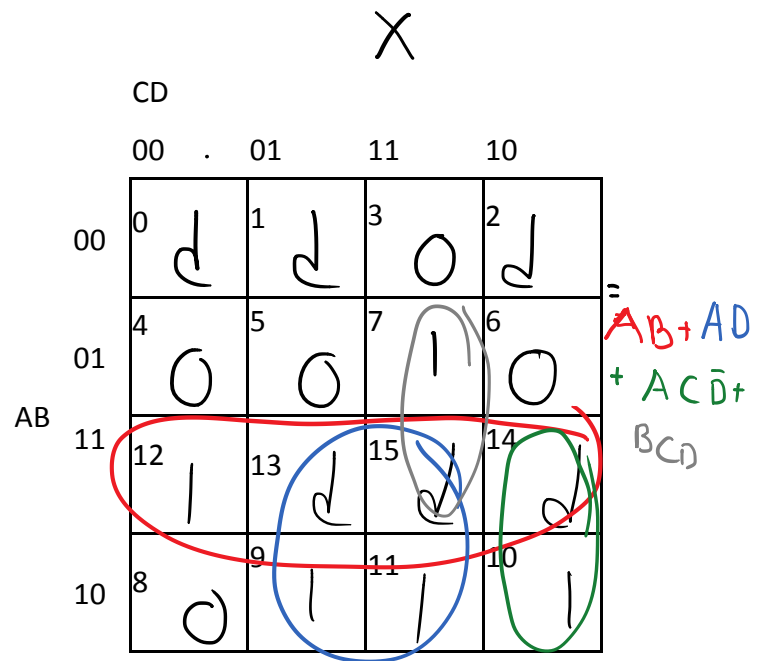
A **code converter** has four inputs (A, B, C, D) and four outputs (W, X, Y, Z). The inputs (taken together) represent the code for a decimal digit in **excess-3 binary-coded decimal** format. The outputs (taken together) represent the code for a decimal digit in **2421 code** format. When the user applies the *valid* excess-3 code for some decimal digit as the input, the circuit outputs the binary-coded decimal for the same decimal digit. The converter will only operate upon codes that represent valid decimal digits.

- Show the truth table for the logic circuit described above.
- Obtain the minimal sum-of-products implementation for each output of the code converter circuit.

Regular Binary Value and MinTerms	Excess 3 BCD Value	INPUT A (8)	INPUT B (4)	INPUT C (2)	INPUT D (1)	OUTPUT W (2)	OUTPUT X (4)	OUTPUT Y (2)	OUTPUT Z (1)
0		0	0	0	0	d	d	d	D
1		0	0	0	1	d	d	d	D
2		0	0	1	0	d	d	d	d
3	0	0	0	1	1	0	0	0	0
4	1	0	1	0	0	0	0	0	1
5	2	0	1	0	1	0	0	1	0
6	3	0	1	1	0	0	0	1	1
7	4	0	1	1	1	0	1	0	0
8	5	1	0	0	0	1	0	1	1
9	6	1	0	0	1	1	1	0	0
10	7	1	0	1	0	1	1	0	1
11	8	1	0	1	1	1	1	1	0
12	9	1	1	0	0	1	1	1	1
13		1	1	0	1	d	d	d	D
14		1	1	1	0	d	d	d	D
15		1	1	1	1	d	d	d	d



$$= \bar{A}B + B\bar{C}D + BC\bar{D} + ACD + A\bar{C}\bar{D}$$



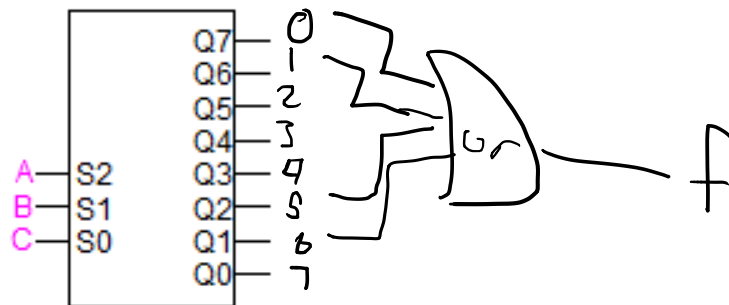
$$= B\bar{D} + A\bar{D}$$



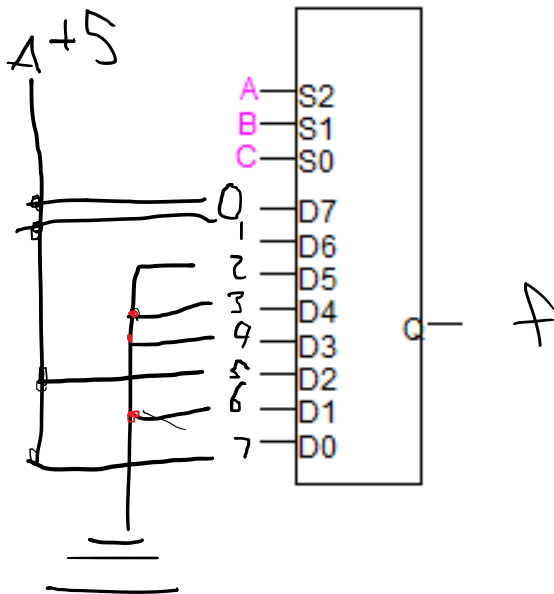
Problem 5 (12 points)

Implement the function  $f(A,B,C) = \sum m(0,1,5,6)$  in each of the following ways:

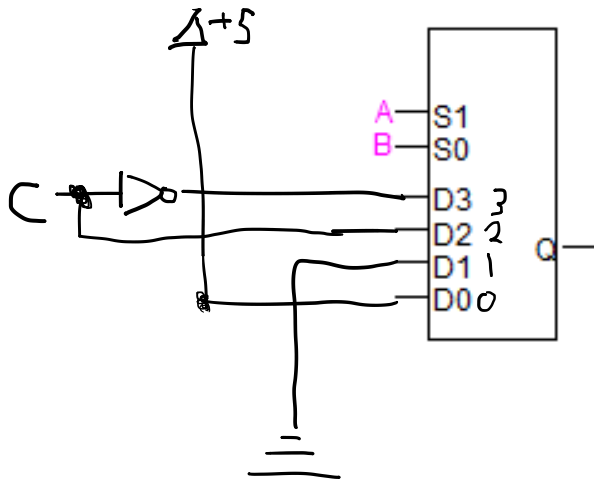
- a. Using one 3-to-8 decoder and one OR gate. (Draw an OR gate having an appropriate number of inputs.)



- b. Using one 8-to-1 multiplexer.



- c. Using one 4-to-1 multiplexer and (at most) one inverter. (Place the inverter appropriately and connect it to the appropriate multiplexer inputs.)

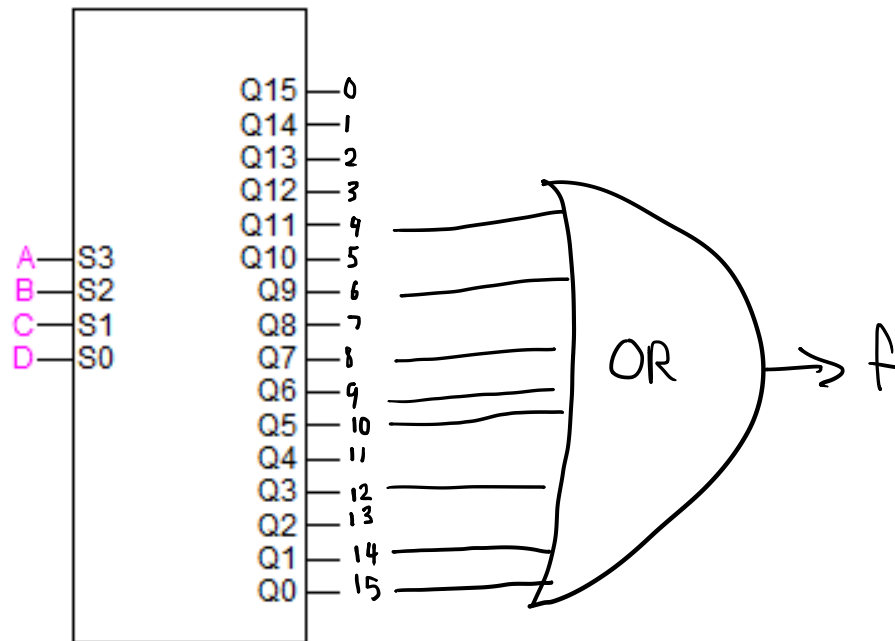


A	B	C	What happens between C and f	f	Min Value	Term
0	0	0	$F = 1$ but this c is 0	1	0	0
0	0	1	$F = 1$	1	1	1
0	1	0	$F = 0$	0	2	2
0	1	1	$F = 0$ but this c is 1	0	3	3
1	0	0	$F = c$	0	4	4
1	0	1	$F = c$	1	5	5
1	1	0	$F = C'$	1	6	6
1	1	1	$F = C'$	0	7	7

**Problem 6 (12 points)**

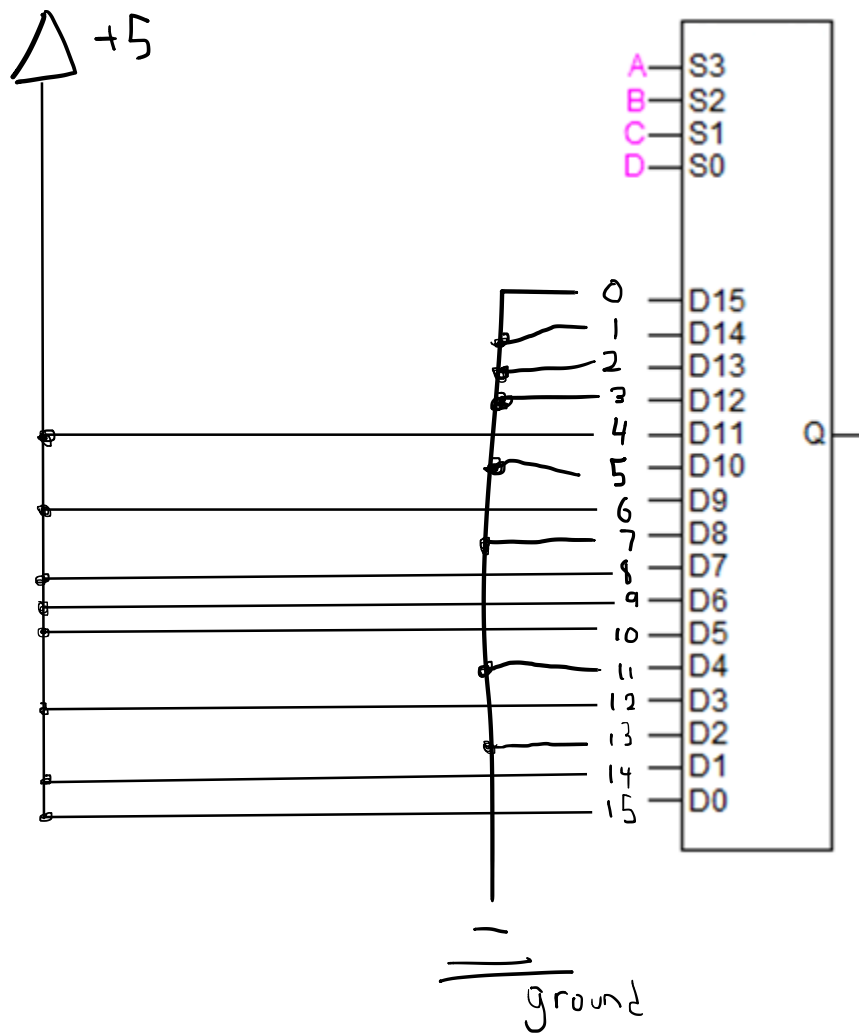
In Problem 2, you had to implement a composite number detector. Implement the same composite number detector in each of the following ways:

a. Using one 4-to-16 decoder and one OR gate.



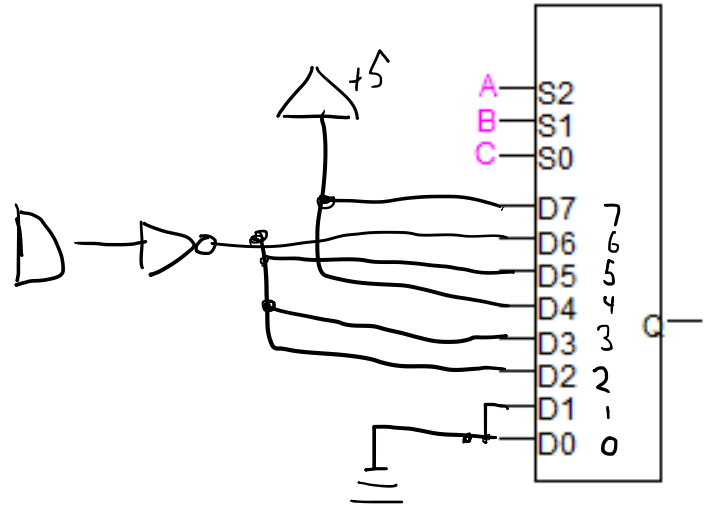
INPUT A	INPUT B	INPUT C	INPUT D	OUTPUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

b. Using one 16-to-1 multiplexer.



### Problem 6 (continued)

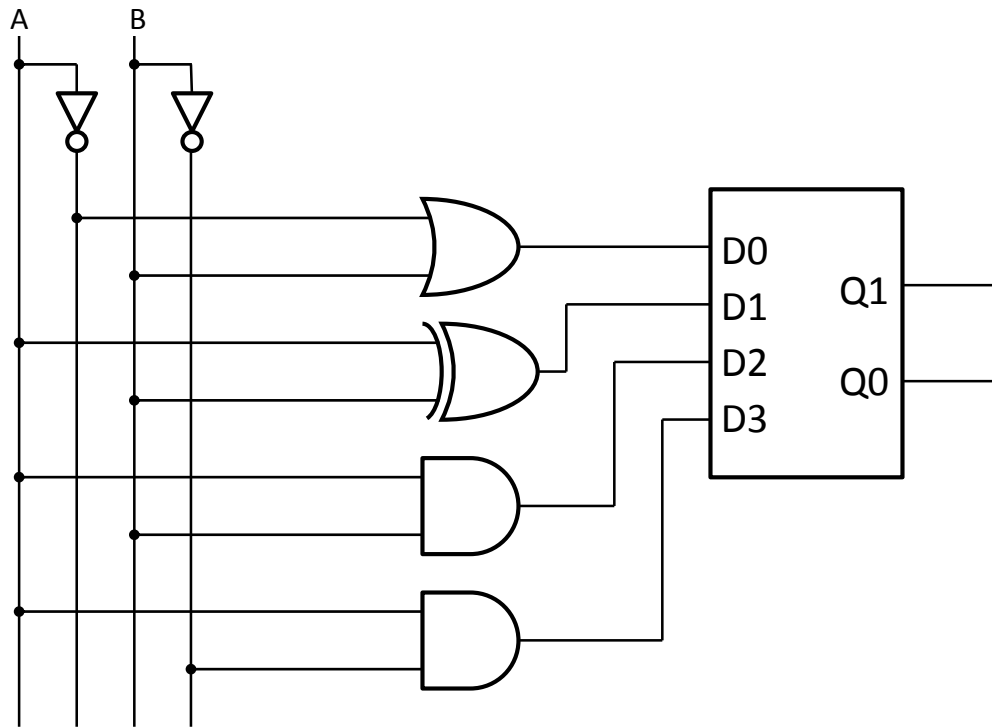
- c. Using one 8-to-1 multiplexer and (at most) one inverter. (If necessary, place the inverter appropriately and connect it to the appropriate multiplexer inputs.)



INPUT A	INPUT B	INPUT C	INPUT D	OUTPUT	
0	0	0	0	0	d0 ground
0	0	0	1	0	
0	0	1	0	0	d1 ground
0	0	1	1	0	
0	1	0	0	1	d2 Inv
0	1	0	1	0	
0	1	1	0	1	d3 Inv
0	1	1	1	0	
1	0	0	0	1	d4 pow
1	0	0	1	1	
1	0	1	0	1	d5 Inv
1	0	1	1	0	
1	1	0	0	1	d6 Inv
1	1	0	1	0	
1	1	1	0	1	d7 pow
1	1	1	1	1	

Problem 7 (6 points)

Consider the circuit shown below, which contains a priority encoder. For this particular priority encoder, *higher-numbered* inputs have priority over lower-numbered inputs.



Complete the truth table below.

A	B	D0	D1	D2	D3	Q1	Q0
0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	1	0	1	1	1
1	1	1	0	1	0	1	0