CS2100: Computer Organisation

Tutorial #7: Combinational Circuits

(Week 9: 16 – 20 March 2026)

Answers to Selected Questions

Tutorial questions

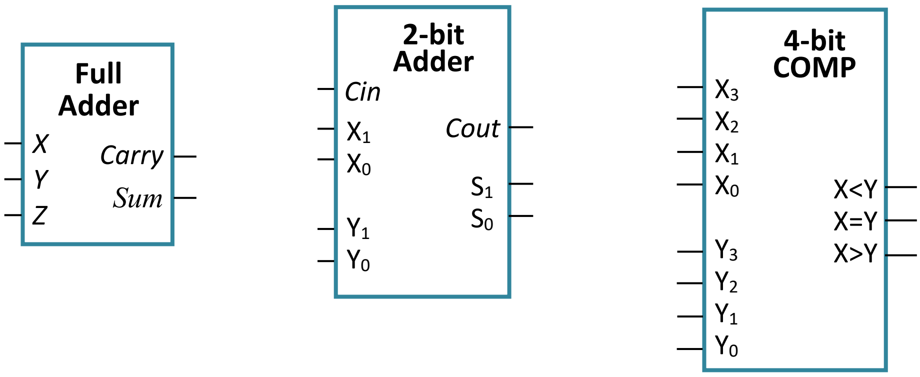
By default, we assume that complemented literals are <u>NOT</u> available, unless otherwise stated.

Logic constants (0 and 1) are always available, and they are considered (degenerate form of) SOP and POS expressions.

The above are to be assumed from now onwards and may not be repeated in future tutorials, assignments and final exam.

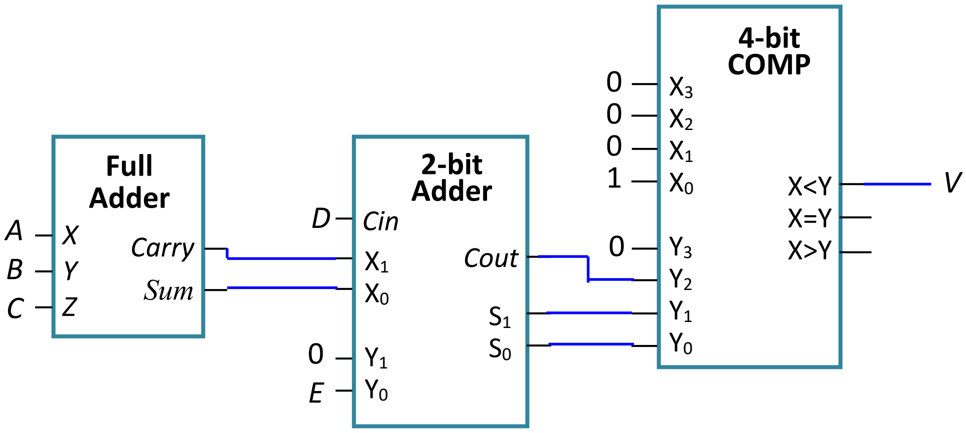
1. [Past-year's question]
   (You need to do SE3 before attempting this question.)

   You are to design a circuit to implement a function *V*(*A*,*B*,*C*,*D*,*E*) that takes in input *ABCDE* and generates output 1 if *ABCDE* is a valid input for the circuit in question SE3 above, or 0 if *ABCDE* is an invalid input. You are allowed to use only the following devices: full adder, 2-bit parallel adder, and 4-bit magnitude comparator. You should use the fewest number of these approved devices, and <u>no other devices or logic gates</u>. The block diagrams for these devices are shown below.



Answer:

Idea: Count the number of 1's in *ABCDE*. If count > 1, then it's a valid input.



2. [Past year's exam question]

   a. You want to construct a circuit that takes in a 4-bit unsigned binary number *ABCD* and outputs a 4-bit unsigned binary number *EFGH* where *EFGH* = (*ABCD* + 1) / 2. Note that the division is an integer division. For example, if *ABCD* = 0110 (or 6 in decimal), then *EFGH* = 0011 (or 3 in decimal); if *ABCD* = 1101 (or 13 in decimal), then *EFGH* = 0111 (or 7 in decimal).

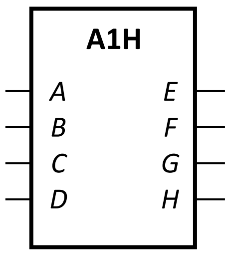
      Construct the above circuit using a single **4-bit parallel adder** and at most one logic gate with no restriction on its fan-in.

   b. The following table shows the 4221 code and 8421 code (also known as BCD code) for the ten decimal digits 0 through 9.

Digit	4221 code	8421 code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0110	0100
5	1001	0101
6	1100	0110
7	1101	0111
8	1110	1000
9	1111	1001

      You want to construct a 4221-to-8421 decimal code converter, which takes in a 4-bit 4221 decimal code *PQRS* and generates the corresponding 4-bit 8421 decimal code *WXYZ*.

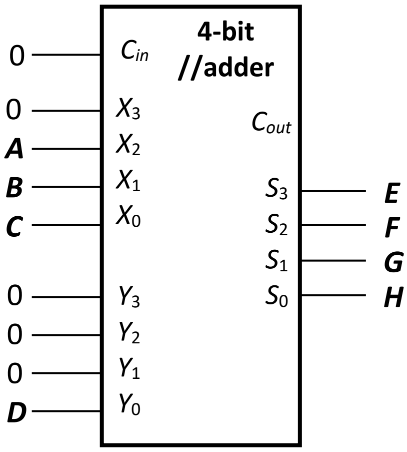
      Let's call the circuit you created in part (a) above the A1H (Add-1-then-Half) device, represented by the block diagram below. Implement your 4221-to-8421 decimal code converter using this A1H device with the fewest number of additional logic gates.



Answers:

2(a)

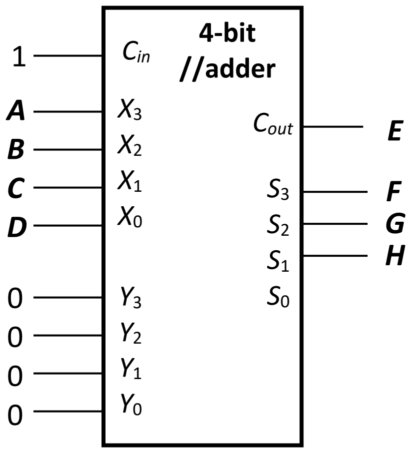
Solution 1:



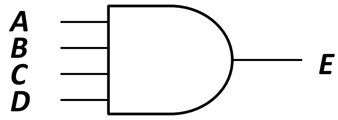
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	1
1	0	1	0	0	1	0	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0

Solution 2:

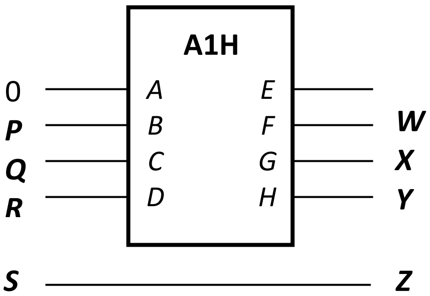
Or Y_0 = 1; C_{in} = 0



Alternative solution for *E*:



2(b)



P	*Q*	*R*	*S*	*W*	*X*	*Y*	*Z*
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	1	0	0	1	0	0
1	0	0	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

Notes on Decimal codes

Decimal codes are codes used to represent the ten decimal digits (0 – 9) of the decimal (base-10) system. Most of them are 4-bit long, but there are some that are longer (eg: biquinary code which is 6-bit long). Some of them are weighted codes, where the code name represents the weights of the respective positions (eg: in 84-2-1 code, the weights are 8, 4, -2 and -1 from left to right), while some are unweighted code (eg: excess 3). The table below (also available in the Digital Logic Design book) shows a few common decimal codes:

Decimal digit	BCD 8421	Excess 3	84-2-1	2421	Biquinary 5043210
0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

Note that as there are only ten valid code values, for codes that are 4-bit long, there will be 6 don't-care values in their 4-variable K-map of the function on such codes.

4. [Past year's exam question]
The BCD code (also known as 8421 code) values for the ten decimal digits are given below:

Digit:	0	1	2	3	4	5	6	7	8	9
Code:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

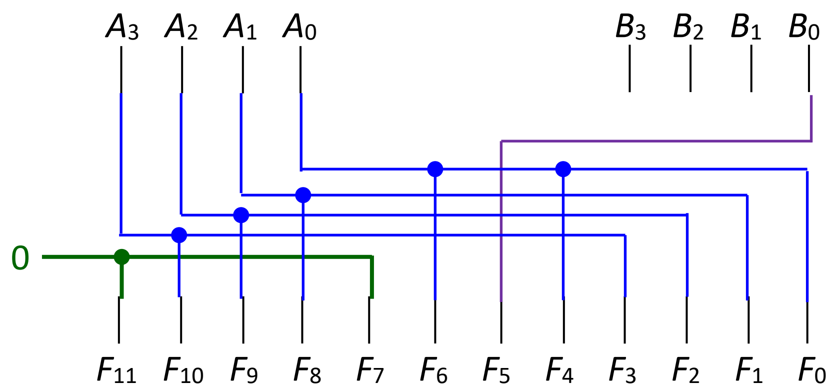
For example, the decimal value 396 is represented as 0011 1001 0110 in BCD code.

Given two decimal digits *A* and *B*, represented by their BCD codes $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ respectively, implement a circuit <u>without using any logic gates</u> to calculate the BCD code of the 3-digit output of $(51\times A) + (20\times(B\%2))$, where % is the modulo operator. Name the outputs $F_{11}F_{10}F_9F_8$ $F_7F_6F_5F_4$ $F_3F_2F_1F_0$.

For example, if *A*=2 (or 0010 in BCD) and *B*=7 (or 0111 in BCD), then $(51\times A) + (20\times(B\%2))$ = 122 or 0001 0010 0010 in BCD. Hence, the circuit is to produce the output 0001 0010 0010 for the inputs 0010 and 0111.

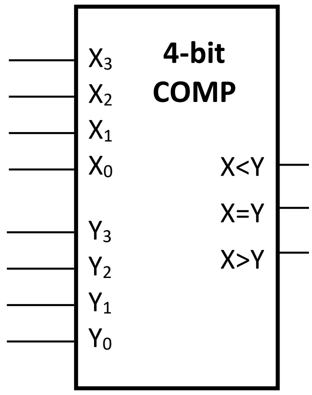
[*Hint:* Fill in the table below that computes 5×*A*.]

A				5×*A*							
A_3	A_2	A_1	A_0								
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1	0	1	0	1
0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	1	0	1
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0	1



See **tut7ans_Q4.pptx** on Canvas > Files and CS2100 website:
https://www.comp.nus.edu.sg/~cs2100/3_ca/tutorials.html

5. Given a 4-bit magnitude comparator as shown on the right, implement the following 4-variable Boolean functions using only this single magnitude comparator with <u>no</u> other logic gates. (Note that there could be multiple answers.)



(a) $F(A,B,C,D) = \Sigma m(12 – 15)$.

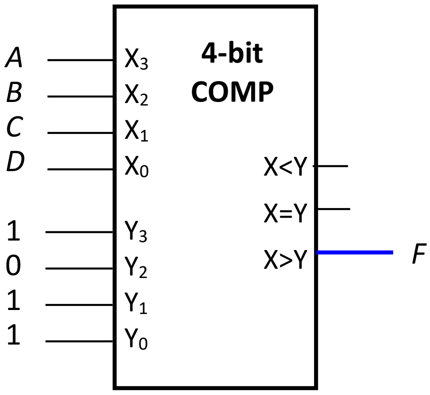
(b) $G(A,B,C,D) = \Sigma m(0, 6, 9, 15)$.

(c) $H(A,B,C,D) = \Sigma m(0, 1, 6, 7, 8, 9, 14, 15)$.

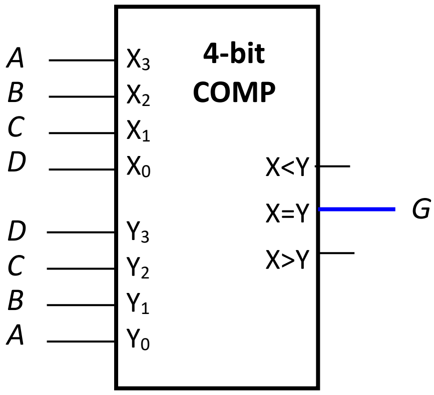
(d) $Z(A,B,C,D) = \Sigma m(1, 3, 5, 7, 9, 11, 13)$.

Answers:

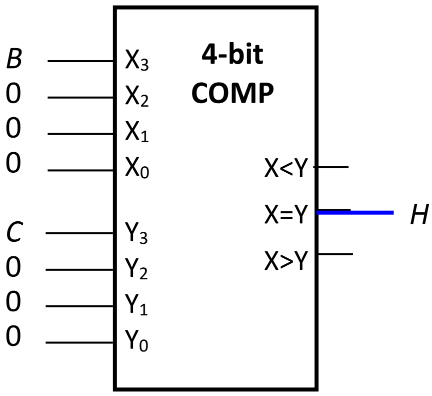
(a)



(b)

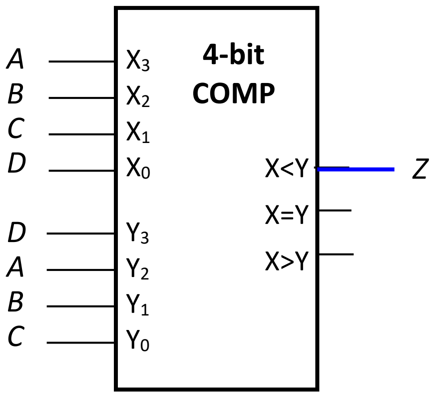


(c)

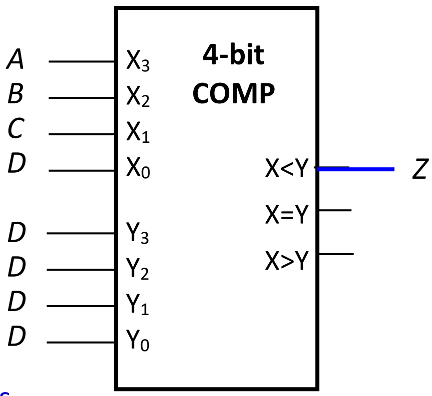


As long as $B = C$.

(d)



or (possibly other answers?)



For this type of question, there could be many possible answers.