CS2100: Computer Organisation

Tutorial #7: Combinational Circuits

(Week 9: 16 – 20 March 2026)

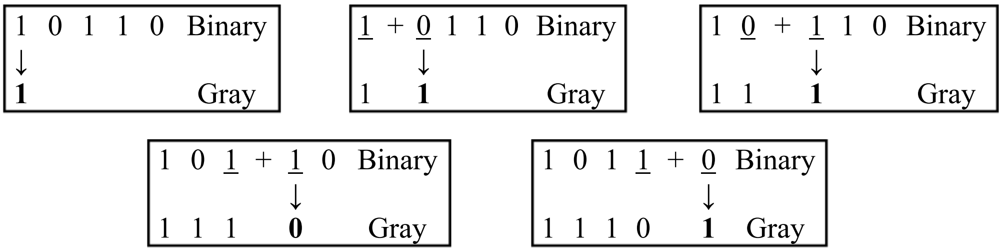
Self-Exploratory Questions

SE1. Design a circuit, without using any logic gate, that takes a 3-bit input *ABC* representing an unsigned integer *x*, and produces a 5-bit output *VWXYZ* which is equivalent to 4*x*+2. What are *V*, *W*, *X*, *Y* and *Z*?

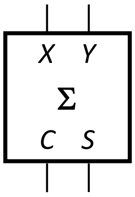
SE2. The following algorithm to convert binary to standard Gray code sequence is given in "Digital Logic Design" book, page 35:

1. Retain the MSB.
2. From left to right, add each adjacent pair of binary code bits to get the next Gray code bit, discarding the carry.

The following example shows the conversion of binary number $(10110)_2$ to its corresponding standard Gray code value, $(11101)_{Gray}$.



Given a half-adder as shown on the right where *X* and *Y* are its inputs and *C* (carry) and *S* (sum) its outputs, implement a 5-bit binary to Gray code converter to convert the binary value *ABCDE* to its equivalent Gray code *PQRST* by using the fewest number of half-adders without any additional logic gates.



SE3. [Past year's question]

A combinational circuit takes in a 5-bit input *ABCDE* and generates a 2-bit value *PQ* such that *PQ* represents the *distance* between the two closest 1s in the input. The distance is defined to be the number of 0s between the two closest 1s.

For example, if *ABCDE* is 01011, then the distance between the two closest 1s (the two rightmost 1s) is zero, therefore, *PQ*=00. If *ABCDE* is 10010, then the distance between the two closest 1s is 2, therefore, *PQ*=10.

You may assume that the distance is always determinable from the given input. Therefore, inputs such as 00000 and 01000 will not be supplied to this circuit.

Draw the K-maps for *P* and *Q* and write the simplified SOP expressions for *P* and *Q*.

Using the simplified SOP expressions for *P* and *Q* to implement the circuit, what is the output if the circuit is fed with the input *ABCDE*=00100?

(This question tests you on the application of don't-cares and the use of 5-variable K-maps.)

Tutorial questions

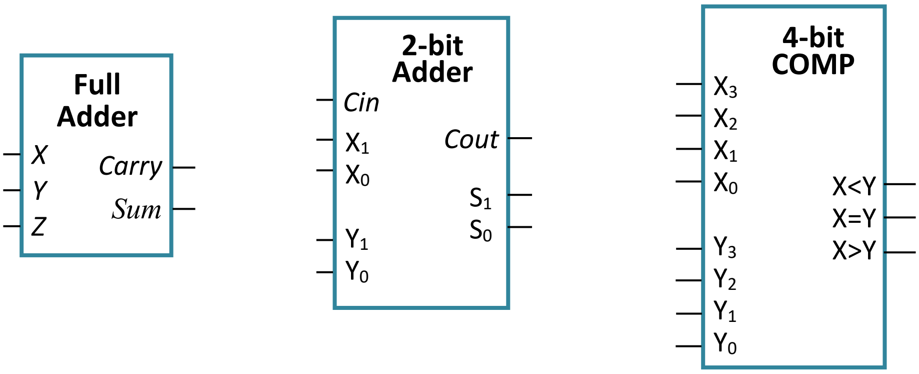
By default, we assume that complemented literals are NOT available, unless otherwise stated.

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/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 no other devices or logic gates. The block diagrams for these devices are shown below.



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).

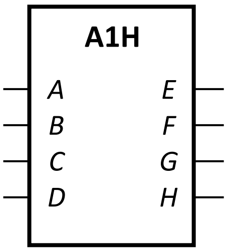
      Construct the above circuit using a single **4-bit parallel adder** and at most one logic gate with no restriction on its fan-in.

   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.



3. [Adapted from AY2024/25 Semester 1 exam question]
How do you implement a circuit that takes in a quaternary (base 4) digit *A* represented in 4-bit excess 3 code $A_3A_2A_1A_0$, and generates $F_3F_2F_1F_0$ which is *A* in excess 11 (eleven) code, without any logic gates?

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

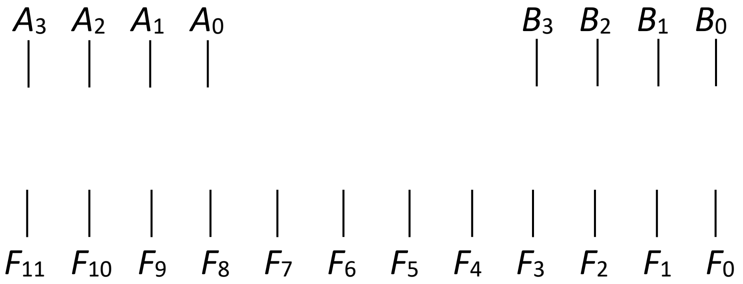
For example, the decimal value 396 is represented as 0011 1001 0110 in BCD code.

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 without using any logic gates 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	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								
1	0	0	0								
1	0	0	1								



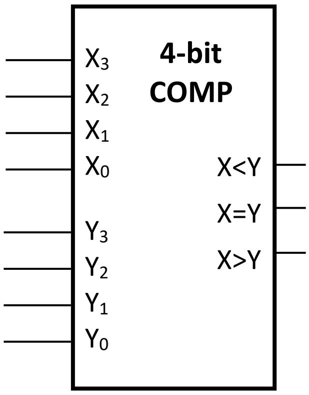
5. Given a 4-bit magnitude comparator as shown on the right, implement the following 4-variable Boolean functions using only this single magnitude comparator with no other logic gates. (Note that there could be multiple answers.)

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


4-bit
COMP
X3
X2
X1
X0
Y3
Y2
Y1
Y0
X<Y
X=Y
X>Y