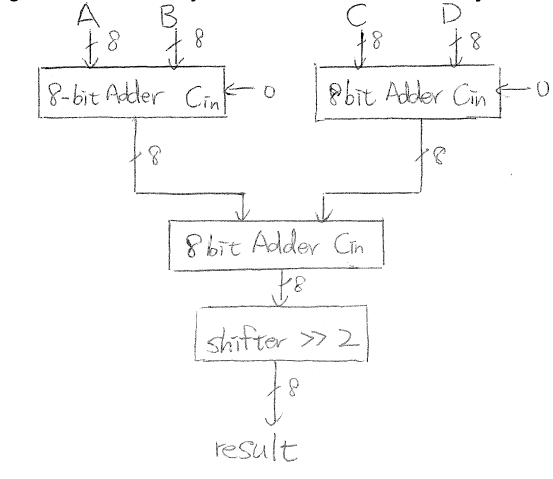
1. **.Design a circuit that outputs the average of 4 8-bit inputs representing positive binary numbers. You can use only the following components:**
   1. **Three 8-bit adders**
   2. **One 2-bit shifter**

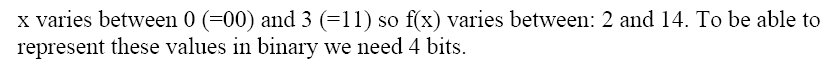
**HINT: a division by 2 is equivalent to a right shift by one place. Assume the numbers are small enough such that carry out of all adders are always zero.**

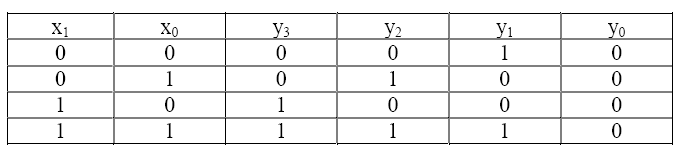


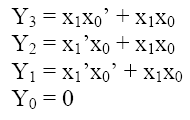
1. **We want to design a combinational circuit that computes the function f(x) = x2+x+2 for a 2-bit x:**

**How many bits do we need for the output?**

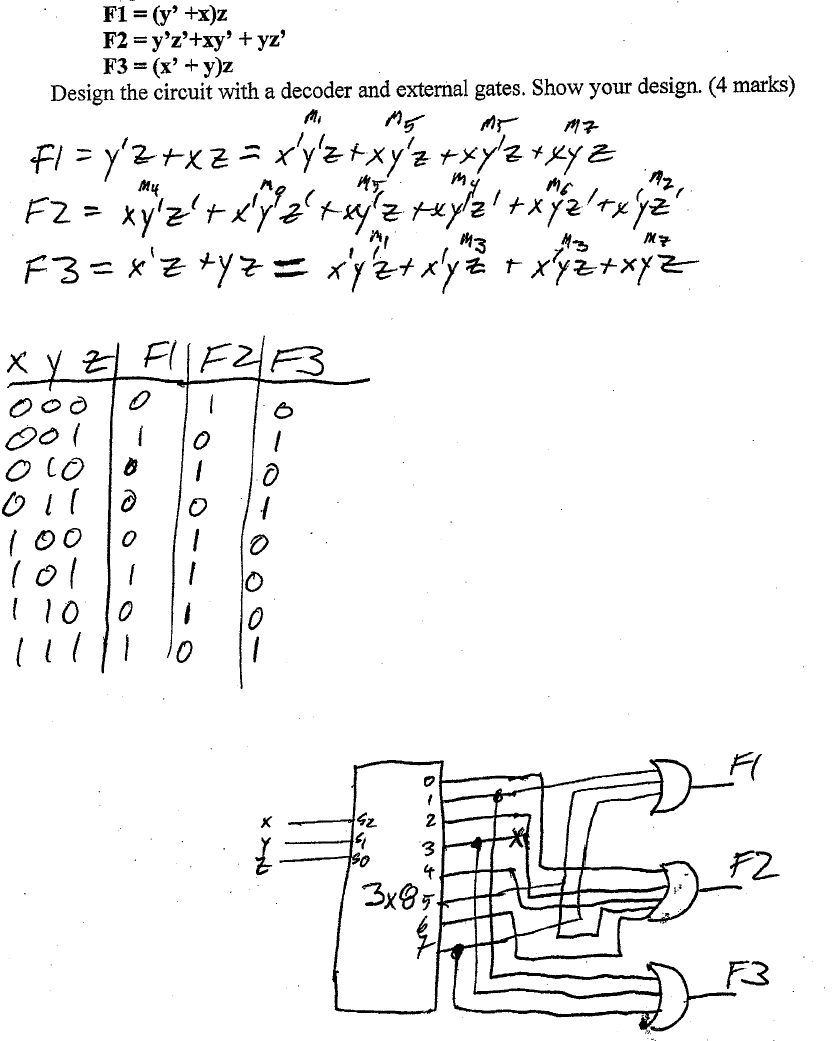
**Derive the truth table and draw the logic circuit using NAND and NOR**







1. **A 3 input combinational circuit is defined by the following three Boolean functions:**

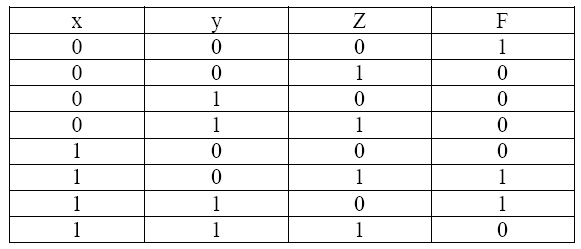
Y

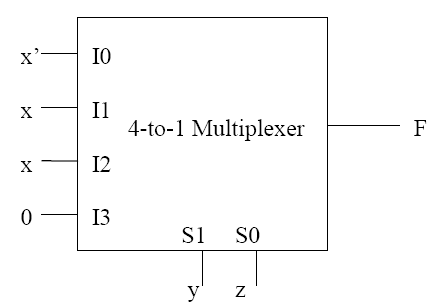
1. **For function F(x,y,z) = xy’z + xyz’ + x’y’z’ :**
   1. **Create the truth table**
   2. **Implement F by means of a 4-to-1 multiplexer (5 points)**

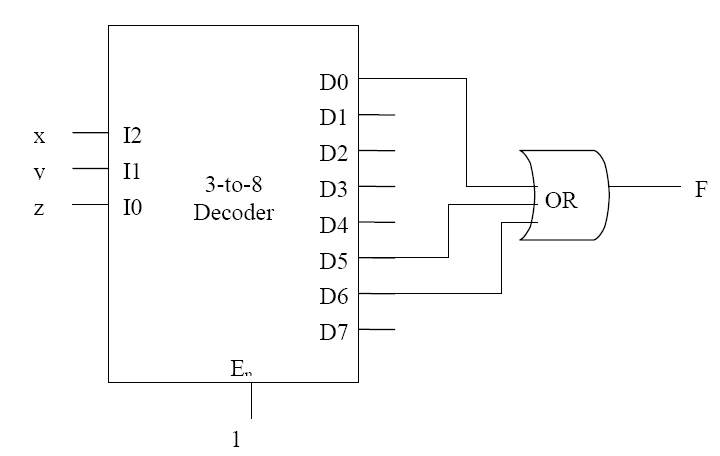
**(HINT: You can use NOT gates to inverse the input to the MUX)**

* 1. **Implement F by means of a decoder**
  2. **Compare the two design in mux and decoder based on the number of equivalent 2-input gates**

**(You don’t have to count the number of NOT gates used in the input of MUX)**







For 4-to-1 MUX the output equation is:

F = S1’S0’.I0 + S1’S0.I1 + S1S0’.I2 + S1S0.I3

To implement this function we totally need 8 AND and 3 OR gates which becomes 11 2- input gates.

For 3-to-8 Decoder the output equation is:

D0 = I2’I1’I0’ , D1 = I2’I1’I0 , D2 = I2’I1I0’ , D3 = I2’I1I0 , D4 = I2I1’I0’ ,

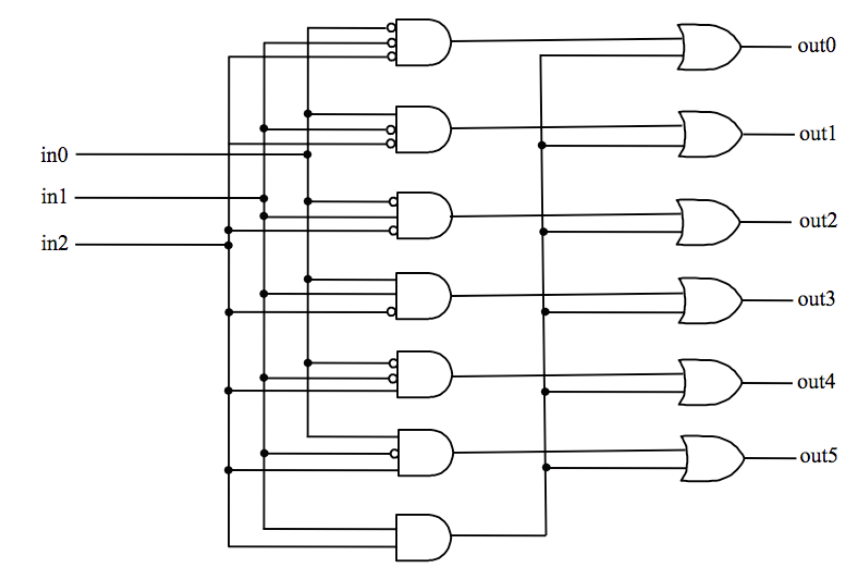
D5 = I2I1’I0 , D6 = I2I1I0’ , D7 = I2I1I0

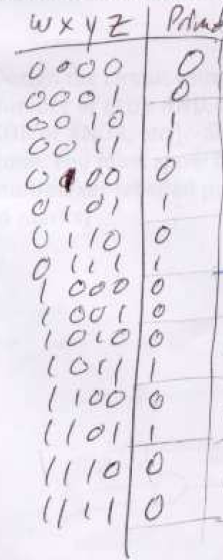
To implement the decoder we need 16 AND gates and for the output we need two additional OR gates. So we totally need 18, 2-input gates.

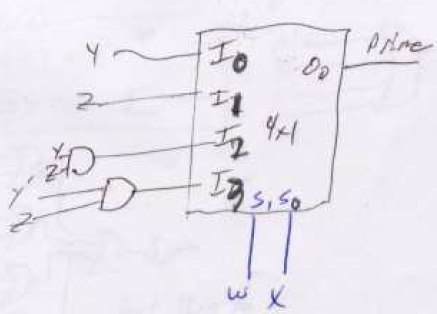
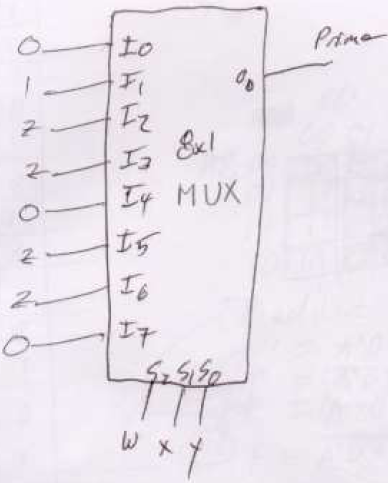
1. **Build a decoder with three input lines but with only six output lines. If the value of the input corresponds to 6 or 7, then all output lines should be asserted to signal an error. Show all 1’s as an output for error indication.**

**This is just like an ordinary 3-input, 8-output decoder, except that if the input1 and input2 are both 1 (resulting in an input value of 6 or 7), all the outputs should be 1. This is accomplished by ANDing input1 and input2 and then sending that result to be OR’d with the normal value for each output. See below.**

1. **You are required by your manager to design a 4-bit prime number checker – you want to output 1 if the binary number is primer else 0 (note 0 and 1 are not prime numbers)**
2. **Show the truth table for the primer number checker**
3. **Design the circuit using single 8x1 mux and a minimal number of extra AND, OR or NOT gates if needed (i.e., no NAND, NOR, XOR or XNOR, etc.). show your work including any simplifications done. You must show the logic diagram with all the pins of the mux labeled properly. State any assumptions that you make.**
4. **Design the circuit using single 4x1 mux and a minimal number of extra AND, OR or NOT gates if needed (i.e., no NAND, NOR, XOR or XNOR, etc.). show your work including any simplifications done. You must show the logic diagram with all the pins of the mux labeled properly. State any assumptions that you make.**



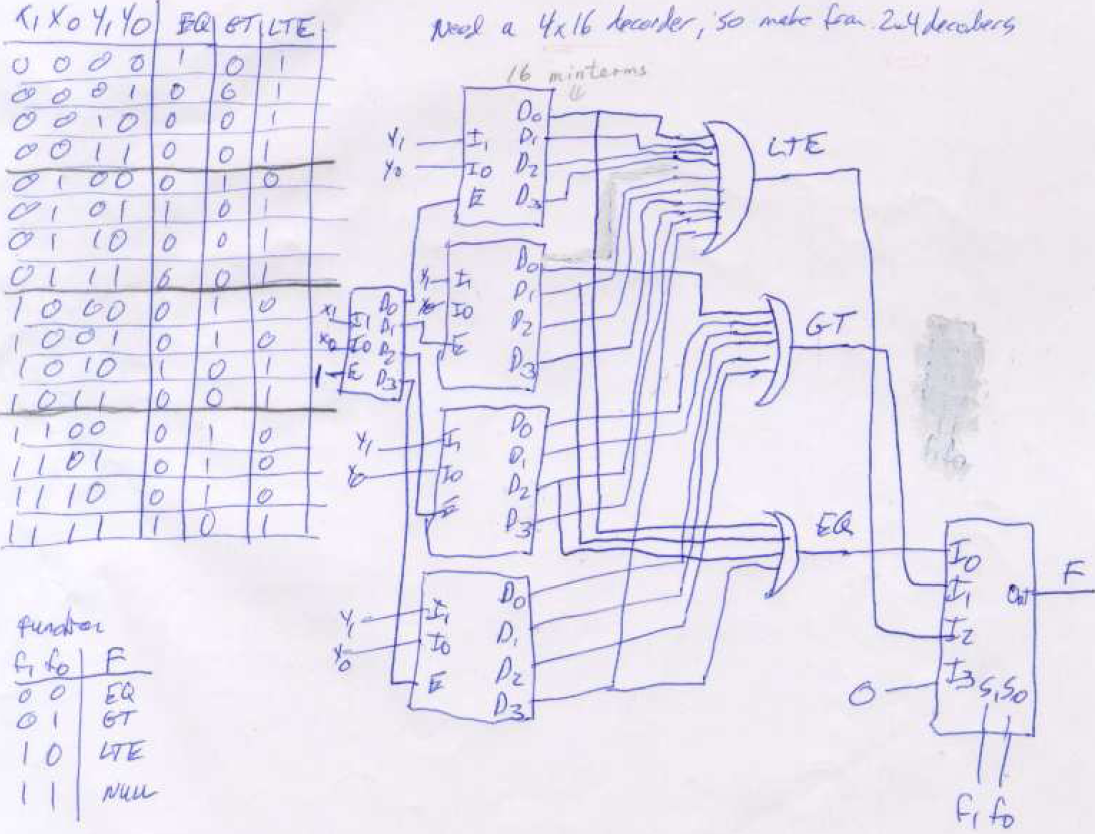
****

****

1. **Using 2x4 decoders and one 4x1 mux, design a circuit with the following properties. It has 2-bit binary inputs [x1 x0] [y1 y0] and**
   1. **It has four modes to select the function of the circuit**
      1. **EQ outputs a 1 if two inputs are equal**
      2. **GT outputs a 1 if X is larger than Y**
      3. **LTE outputs a 1 if the X is smaller or equal to Y**
      4. **NULL outputs a 0**

**You must use a minimum number of 2x1 decoders with enables (i.e. if the enable is 0 all the outputs of the decoder are 0, otherwise, the output is 1 for the minterm), only one 4x1 mux and a minimum number of additional gates.**

**Show the circuit diagram with all the pins to the decoders and mux labeled properly. State any assumptions you are making in your design**



1. Design and Compare 8-bit adder/subtractor circuit using two’s and one’s complement for subtraction. Show your design for deducing the correct result (positive or negative) on subtraction.

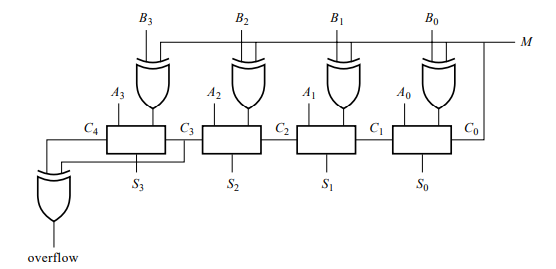
Using above 8-bit adder find the average of 4 8-bit inputs.

***Hint:*** *a division by 2 is equivalent to a right shift by one place.*

**Key – Adder/Subtractor Circuit – 6 marks (3 for circuit design and 3 for deducing the sign of the result)**

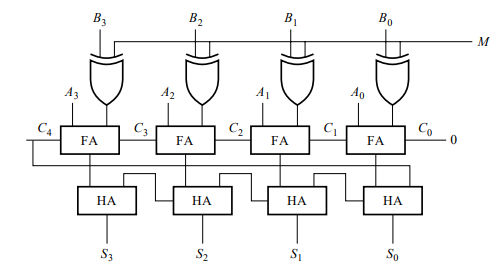
**Average of 4 8-bit numbers – 5 marks (3 for additions and 2 for finding average using shifter)**

**Two’s complement adder/subtractor circuit**



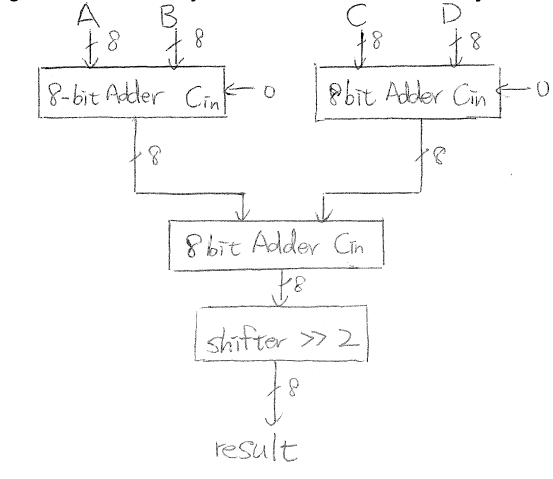
Sign of the result depends on End Carry – EC= 1 => positive result otherwise negative and 2’s complement of the result gives correct magnitude. (logic for circuit design need to be shown)

**One’s complement adder/subtractor circuit**



Sign of the result depends on End Carry – EC= 1 => add 1 to result and it is positive otherwise negative and 1’s complement of the result gives correct magnitude. (logic for circuit design need to be shown)

**Average of 4 8-bit numbers**



1. Calculate and Compare the Total Propagation Time of 4-bit Ripple Carry Binary Parallel Adder (BPA) and 4-bit BPA with Look-Ahead Carry Generator for the following timing specifications:
2. XOR - 40ns
3. AND – 25ns
4. OR – 15ns

**Key: 2 marks for referring the diagram from the notes / 3 marks for applying the given delays and finding the total propagation time comparatively**

4-bit RC BPA = 40 + 4\*40 = 200ns

4-bit BPA + Look Ahead Carry Generator = 40 + 25 + 15 + 40 = 120ns

1. Design a code converter circuit to convert the 4x3 keypad outputs (4 rows X 3 columns = 7 outputs) to a 4 bit binary code as below:

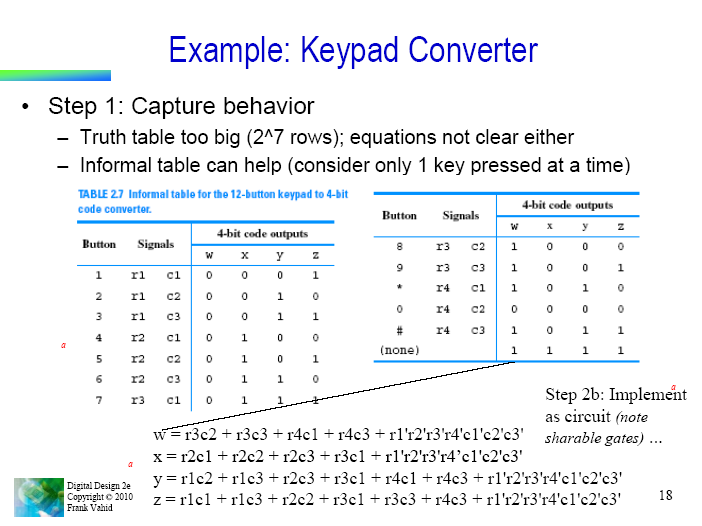
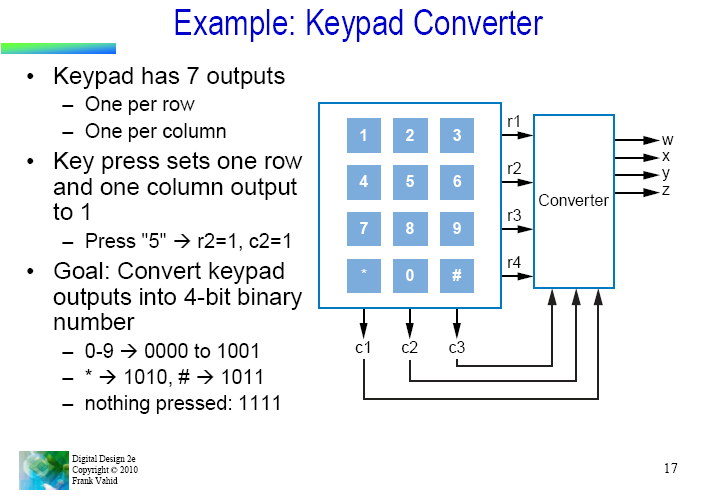
0 – 9 -> 0000 to 1001

\*->1010

#->1011

Nothing pressed -> 1111

**Key: 3 marks for realizing the rows and cols of 4x3 keypad / 3 marks for encoding the rows and cols in binary / 3 marks for truth table / 3 marks for expressions**



1. Design a circuit which executes the following code:

***If (A+B > 14)***

***Then S = C + D + 1***

***Else***

***S = C + D***

A, B, C and D are 4-bit binary numbers

You can use the following components:

• Adder

• Comparator

• Subtractor

• Multiplexer

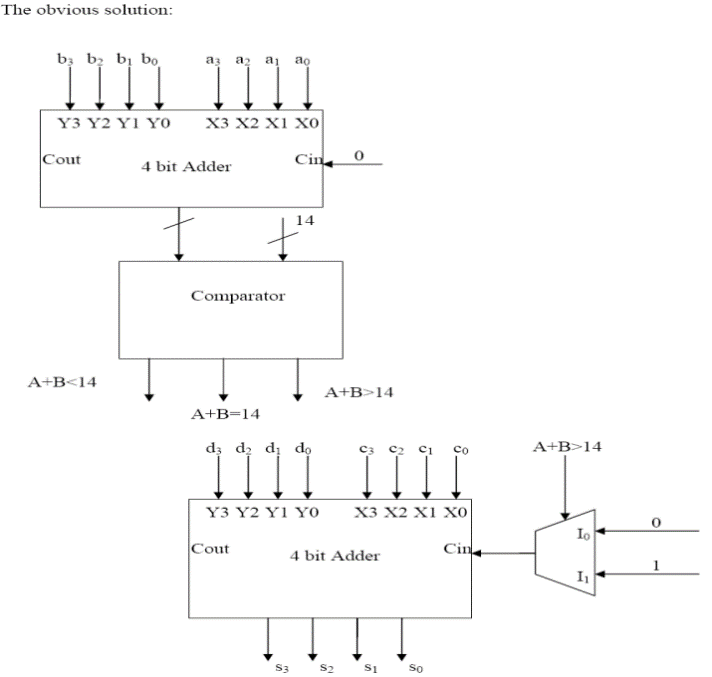
Suggest how to redesign the above circuit using only two 4-bit adders.

***Hint:*** There is a tricky solution to this problem which uses just adders.

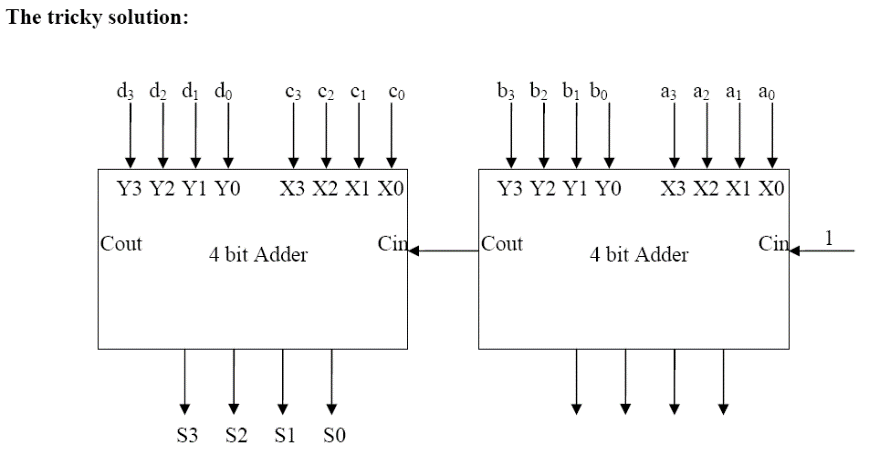
**Key: 5 marks for using right components (MSI) and circuit connections**

**6 marks for finding the logic to do just using adders alone**

**Using the components**



**Using just adders**



12,13,14 and 15 answer