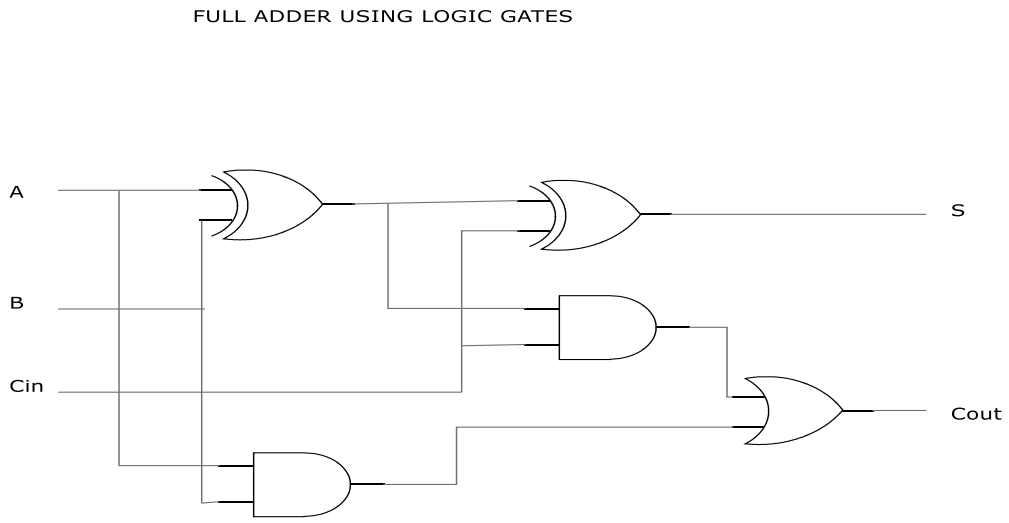
Qn1. Show the implementation of a full adder circuit with digital logic gates. Construct an eight-bit adder by using this full adder.

Ans:

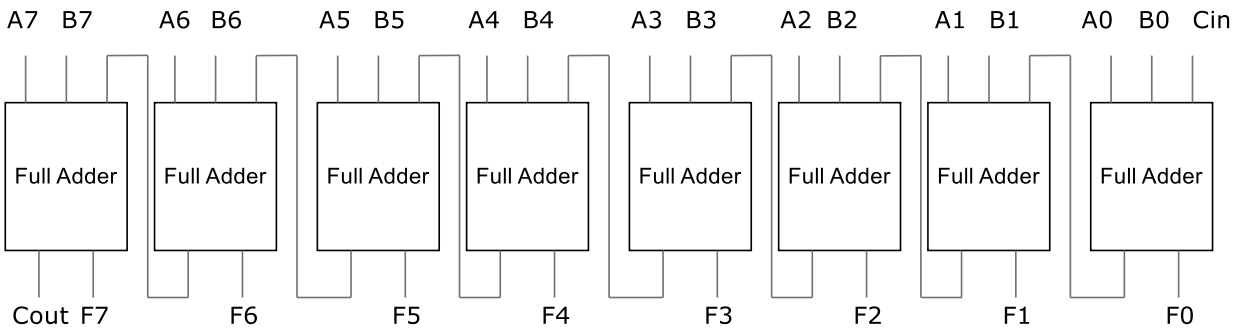
Full Adder: A full adder circuit adds three one bit numbers and produces two one bit binary numbers as sum and carry. In the following circuit, which represents a full adder, we have three inputs as A, B, C_{in} and two outputs as sum S and carry C_{out}.



The Truth table of the full adder is shown below

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

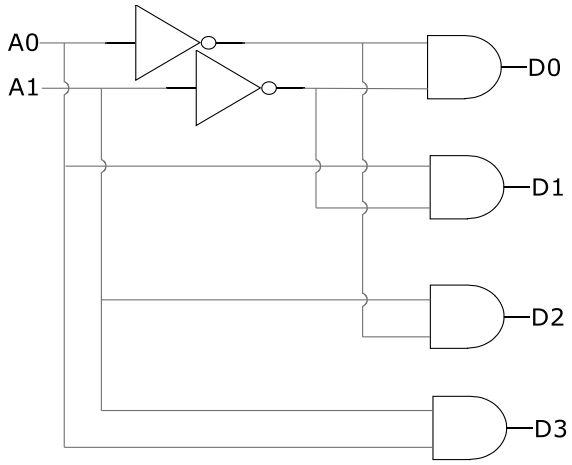
8 Bit Adder designed using full adder circuits is shown below.



Qn2.Explain the working principle of decorer, multiplexer and priority encoder

Ans:

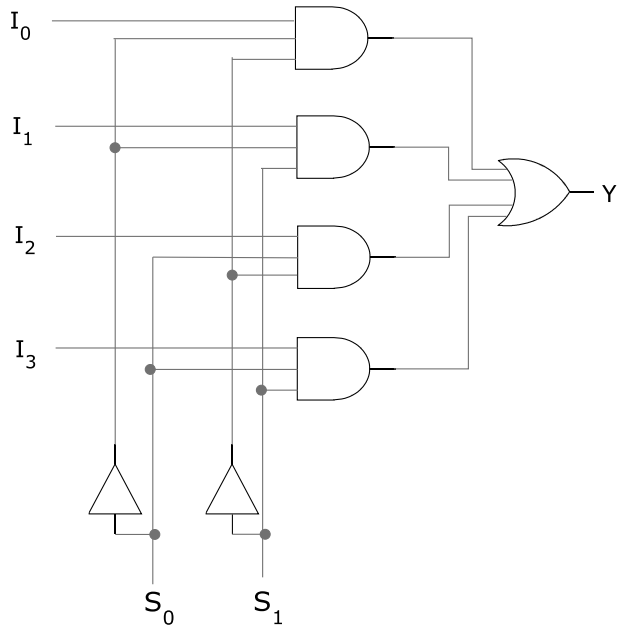
Decoder: A decoder is a combinational circuit that converts binary information from *n* input line to maximum 2^n unique output lines. We explain the concept using a 2 to 4 decoder. Here we have two inputs A0, A1 and four outputs D0, D1, D2, D3, D4.



Based on the value of the inputs, only one output line has the value 1 and the others are 0. For example, of A0=0 and A1=1, first output line D0 as the value 1 and the other outputs are 0. The Truth table of the 2 to 4 decoder is as follows.

A0	A1	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Multiplexer: A multiplexer is a combinational circuit that selects one of many input lines and directs it to a single output line based on the value of the select lines. It has 2^n input lines and n selection lines. We explain the concept using a 4-1 multiplexer.

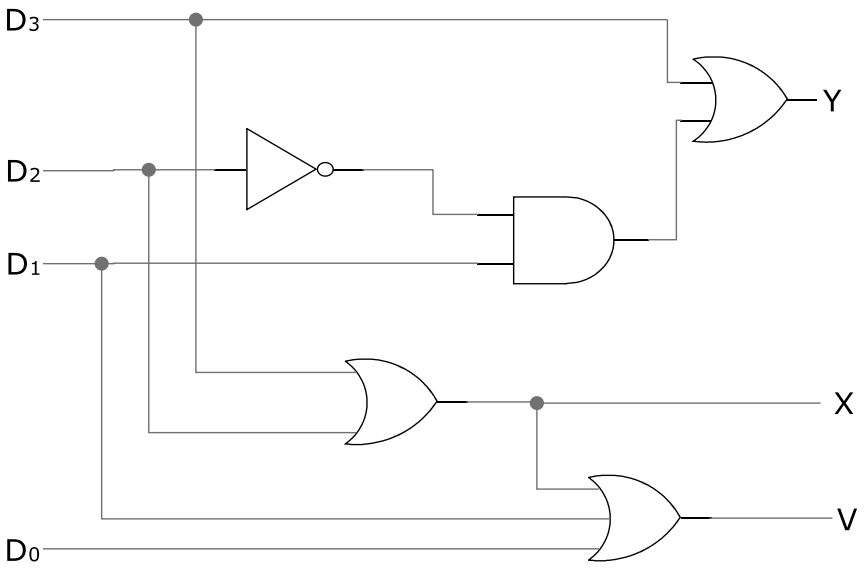


For example, is select line values are S_0=1 and S_1=1, last input line I_3 is directed to the output. The Truth table of 4-1 multiplexer is as follows.

S_0	S_1	Y
0	0	Y_0
0	1	Y_1
1	0	Y_2
1	1	Y_3

Priority encoder: It is an encoder circuit that implements a priority function. The operation is that if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence. In other words, a priority encoder encodes in binary the input of highest priority that is 1 irrespective of other inputs being 1 simultaneously.

We will explain the concept using an example given below. Here the priority encoder has D_0, D_1 ,D_2, D_3 as inputs and Y,X,V as the outputs.



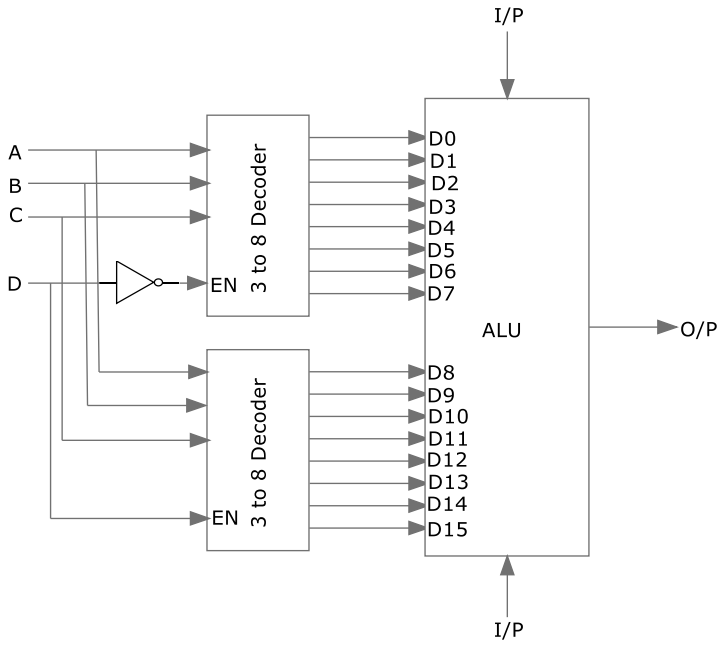
Here the order of priority is $D_3 > D_2 > D_1 > D_0$. Outputs Y and X are used to encode the input of the highest priority that is 1. There is another additional output V, which is 1 if at least one input is 1 (i.e., input is valid) and is 0 if all inputs are 0 (i.e., input is invalid). If input, say D_2 is 1 and D_3 is 0 then, the input of highest priority that is 1 is D_2. So, irrespective of the other inputs of lower priority than D_2 (i.e., D_1, D_0) we encode D_2 as X=1 and Y=0 that corresponds to D_2.

The truth table of the priority encoder is as follows:

INPUTS				OUTPUT		
D_0	D_1	D_2	D_3	X	Y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Qn3.There are 16 operations in an ALU. We need to select a particular operation at some point of time. Show the implementation with decoder.
Ans:

Here we are using two 3 to 8 decoders. A,B,C,D are the inputs of decoders. D_0, D_1,……. D_{15} are the inputs of the ALU which corresponds to the 16 operations to be selected. So, at a time only one of the inputs D_0, D_1,……. D_{15} is 1 that corresponds to the instruction to be executed.
The connection between the decoders and the ALU is shown below. For example if A=0,B=0,C=0,D=1, then the lower decoder is enabled because D=1. As A=0,B=0,C=0, the first line i.e., D8 of the lower decoder is 1; so task corresponding to D8 is selected in the ALU.

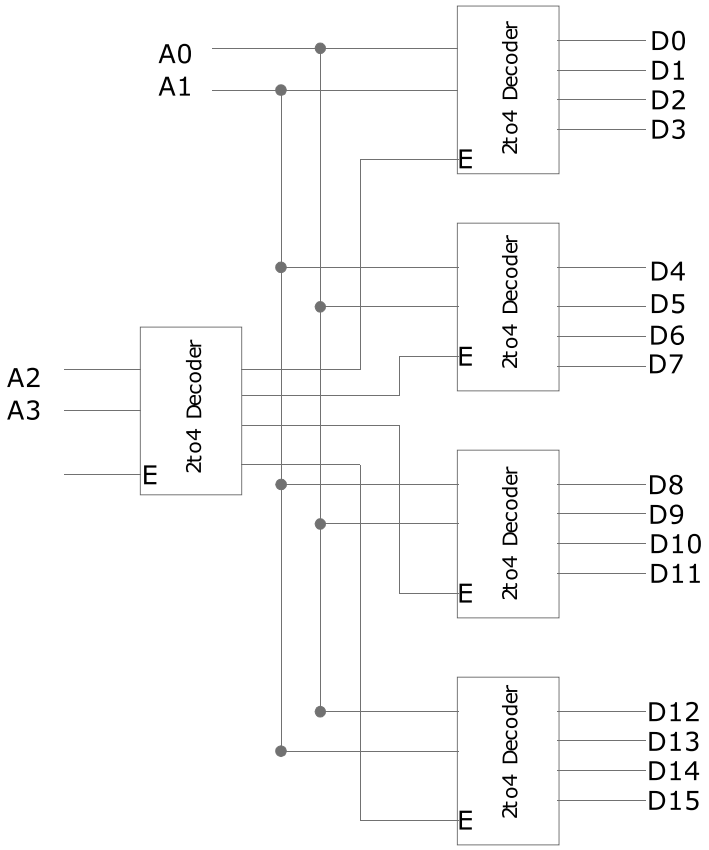


Qn4.We have 2 x 4 decoder with "CHIP SELECT" line. Show the construction of a 4 x 16 decoder.

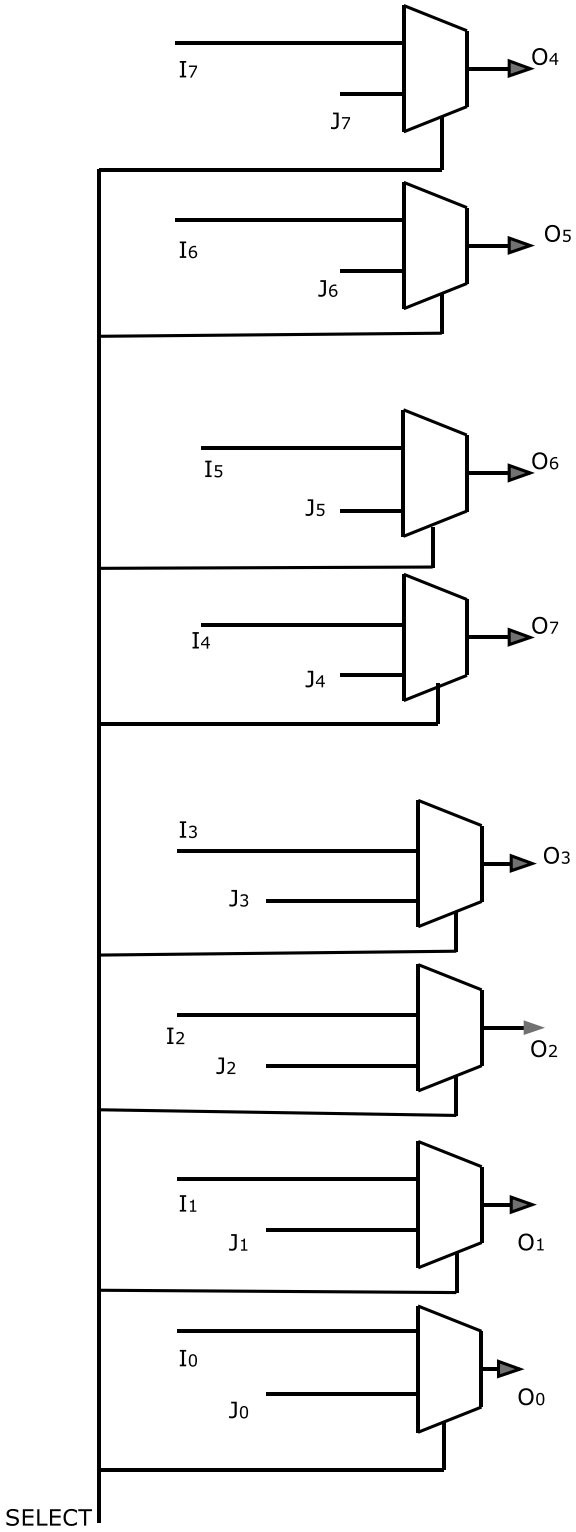
Ans:

A 4x16 decoder is constructed with five 2x4 decoders as shown below. Here A_0, A_1, A_2, A_3 are the inputs. D_0, D_1,…. D_{15} are outputs. E is enable input for each decoder.

We explain the working with an example. The enable for the decoder of the first column is always 1. Now if A_0=1, A_1=1, A_2=1, A_3=1 , then the bottommost decoder of the second column is enabled (by the fact that A_2=1, A_3=1). Now as A_0=1, A_1=1, so D15 is selected.

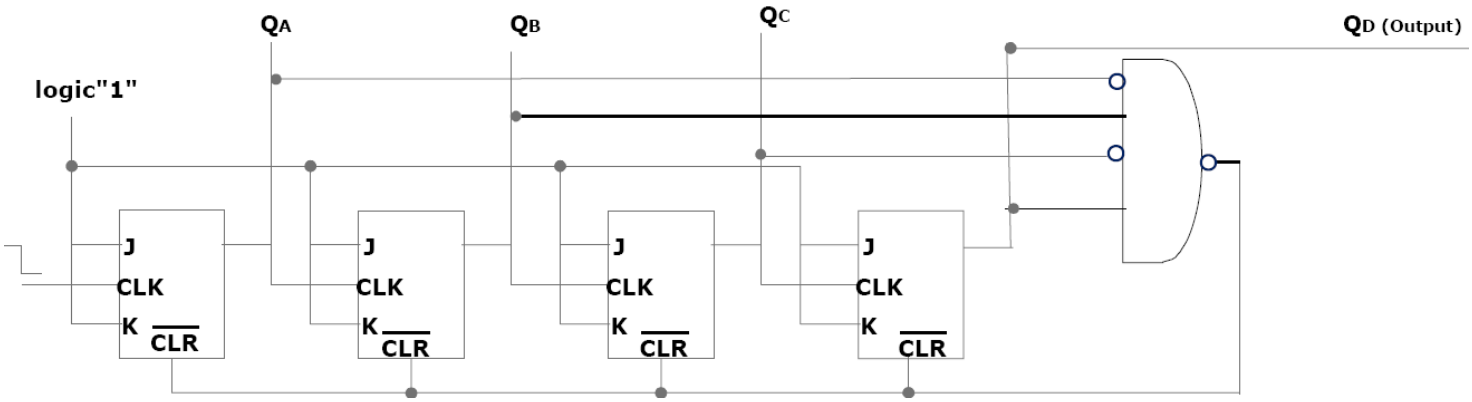


Here we have eight 2x1 multiplexers for handling two sets of 8-bit of data. $I_{0,}$ $I_{1....}I_7$ is the first set of data and $J_{0,}$ $J_{1....}J_7$ is the second set. $O_{0,}$ $O_{1,}$…. O_7 are outputs of the 8, 2-1 multiplexers. We have one SELECT line. The multiplexer based circuit is shown below.



Qn6. We need to count the seconds of a minute. Show the construction of this counter using two decade counter.

The basic schematic of an asynchronous decade counter is shown below.



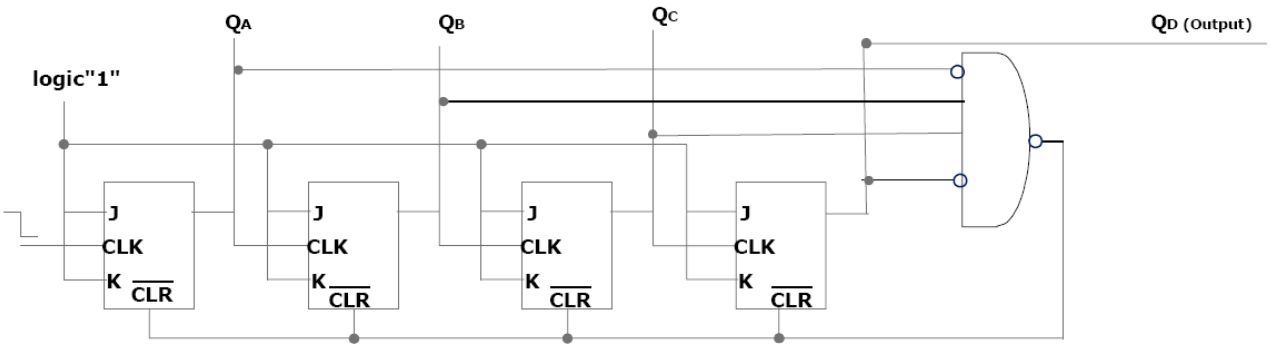
The truth table of the counter is as follows.

Clock count	Output bit pattern				Decimal value
	Q_D	Q_C	Q_B	Q_A	
CLR to all flops	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	Counter resets its outputs back to zero

It may be noted that once Q_D=1,Q_C=0, Q_B=1,Q_A=0, the NAND gate clears the flip-flops

The decade counter counts from 0 to 9. If we consider Q_D as the output and clock as the input, Q_D makes a toggle after every 10^{th} toggle of clock. In other words, the decade counter is a divide by 10 counter.

In a similar way we can implement a divide by 6 counter, whose details are given below:



Clock count	Output bit pattern				Decimal value
	Q_D	Q_C	Q_B	Q_A	
CLR to all flops	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	Counter resets its outputs back to zero

Now if we cascade these two counters by connecting the clock of the divide by 6 counter with the output of the divide by 10 counter we get a divide by 60 counter i.e., counts from 0 to 59. This circuit can count the seconds of a minute. The following diagram illustrates the idea.