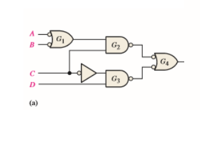
**Write down the Boolean expression of the following circuit .Give the output wave form of each gate in the diagram.**



* **BOOLEAN EXPRESSION:**

[(A’+B’) . C] + [C’ . D]

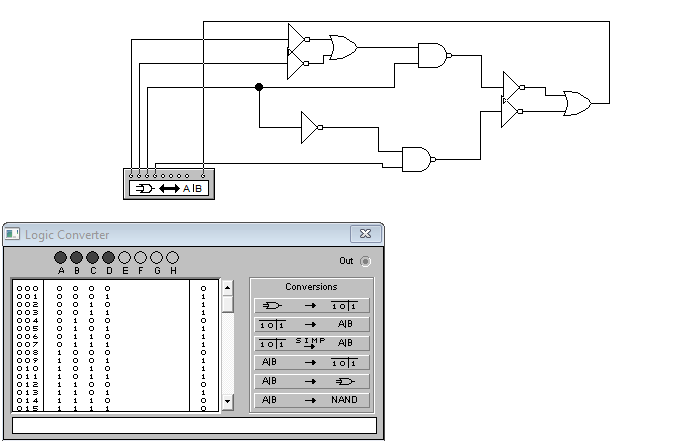
* **TRUTH TABLE :**

**LET ,**

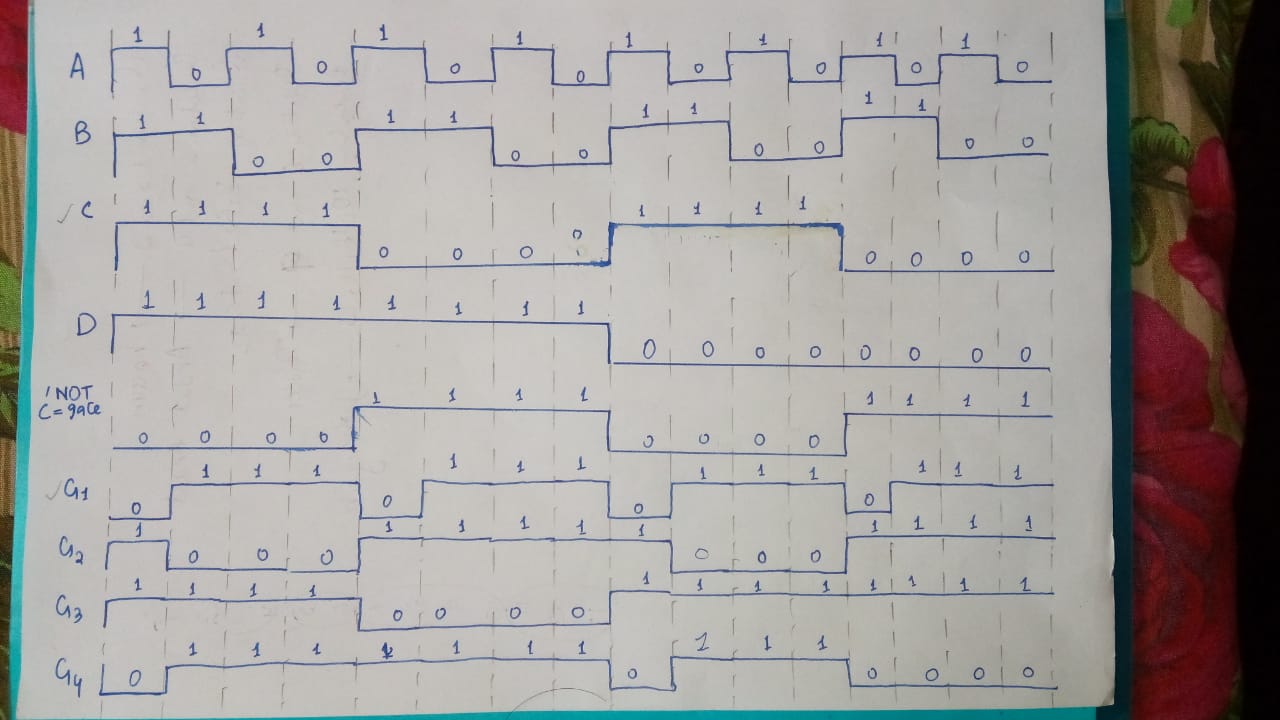
F=[(A’+B’) . C] + [C’ . D] , P=[(A’+B’) . C] , Q=[C’ . D]

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **A’** | **B’** | **C’** | **A’ + B’** | **P** | **Q** | **F** |
| **0** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** |
| **0** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | **1** |
| **0** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| **0** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **1** |
| **1** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| **1** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

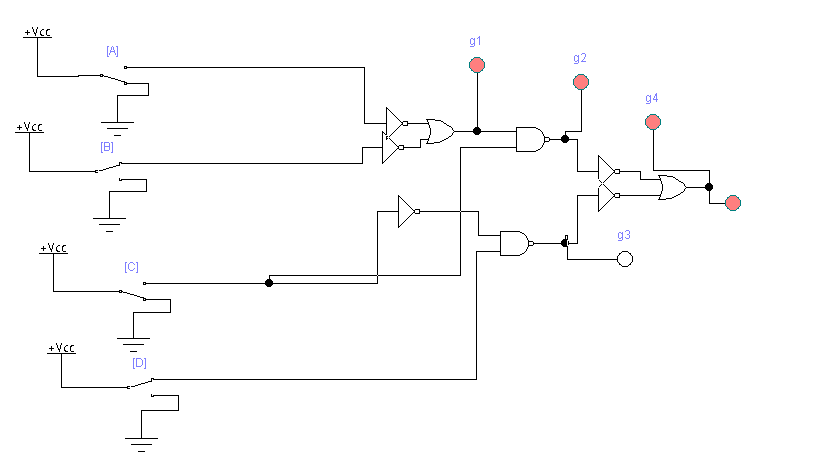
* **VERIFIACTION THROUGH LOGIC CONVERTER:**



* **OUTPUT WAVEFORM FOR EACH MENTIONED GATE :**

****

* **VERIFIACTION OF WAVEFORM THROUGH CIRCUIT SIMULATION:**



**QUESTION NO : 5 (b)**

**Explore a logic circuit with four input variables that will only produce a 1 output when**

**exactly three input variables are 1s. Also draw logic circuit diagram. (Attach Simulated**

**circuit diagram as well).**

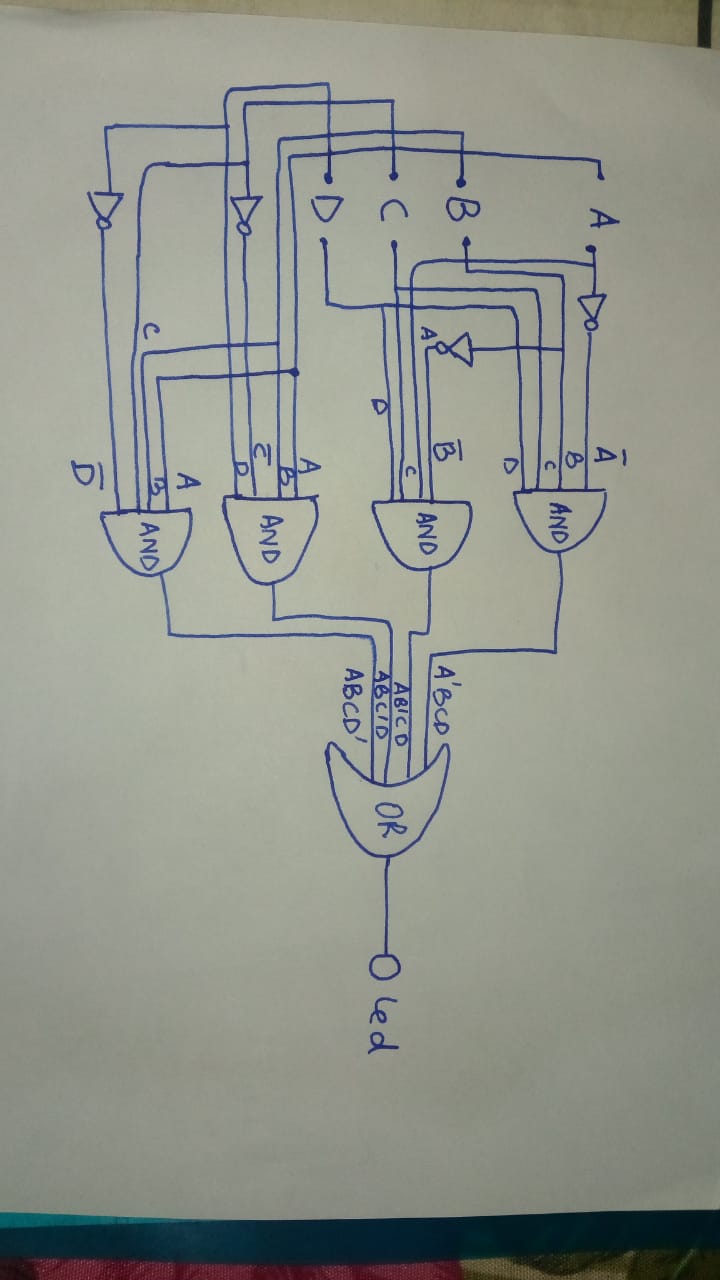
* **TRUTH TABLE :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **OUTPUT** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **0** |
| **0** | **0** | **1** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** |
| **0** | **1** | **1** | **0** | **0** |
| **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** | **0** |
| **1** | **1** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** | **0** |

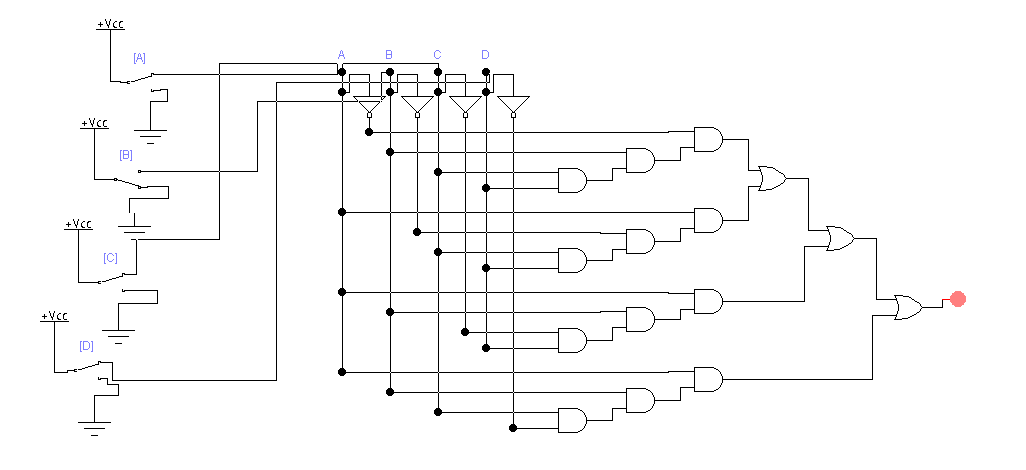
* **BOOLEAN EXPRESSION:**

A’BCD + AB’CD + ABC’D + ABCD’

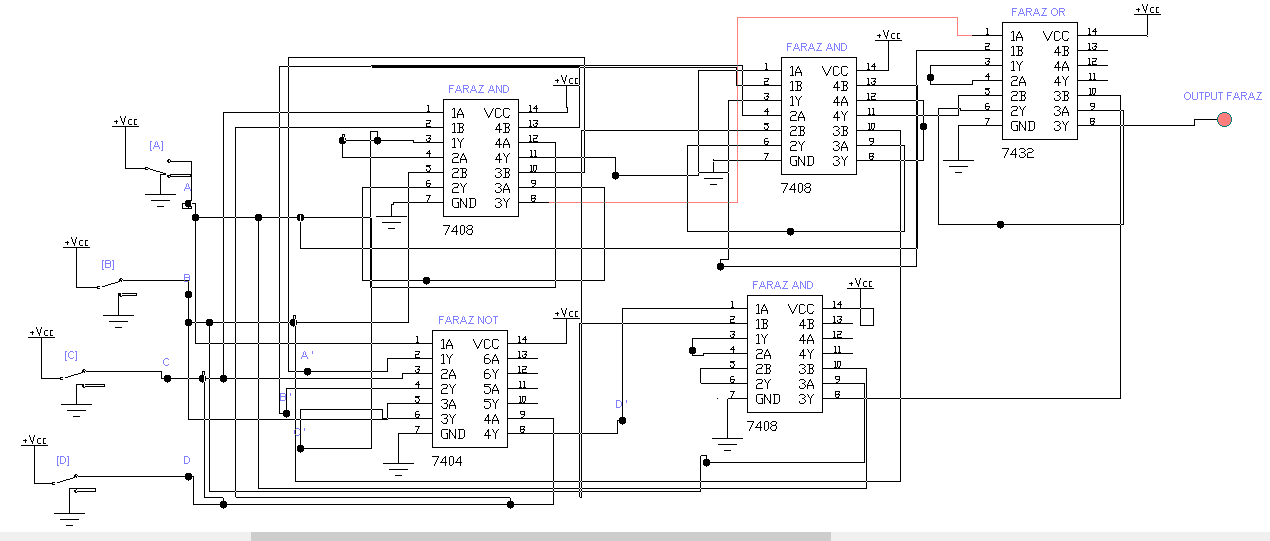
* **CIRCUIT DIAGRAM USING 4-INPUT GATES:**

****

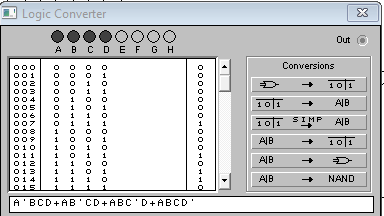
* **GATE-LEVEL SIMULATION CIRCUIT:**



* **IC-LEVEL SIMULATION CIRCUIT:**



* **VERIFICATION OF BOOLEAN EXPRESSION USING LOGIC CONVERTER:**



**QUESTION NO : 6 (a)**

Elaborate logic of CLC that takes 2-bit input value and produces the square of the value in

binary form with the help of truth table. Draw neat & clean logic diagram. (Attach Simulated

circuit diagram as well).

* **EXPECTED TRUTH TABLE FOR 2-bit CLC FOR SQUARE DETERMINATION:**
* The number of inputs should be **2 i-e A and B**
* The range of input will be **0-3**
* The highest square value will be **3­­2 = 9**
* The binary form of 9(max range) is **1001 i-e 4-bit output**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **P0** | **P1** | **P2** | **P3** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** | **1** |

* **BOOLEAN EQUATION FOR P0:**

**AB**

* **BOOLEAN EQUATION FOR P1:**

**AB’**

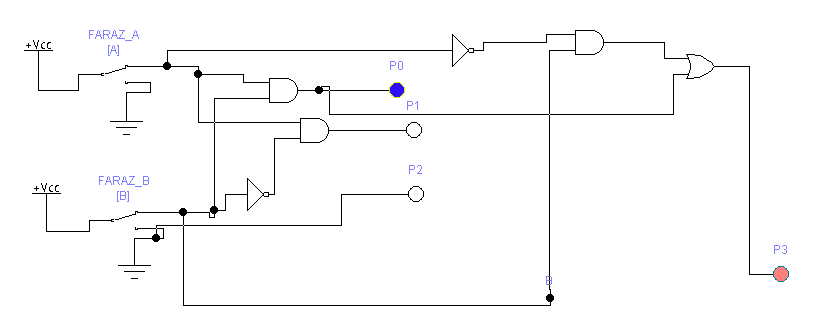
* **BOOLEAN EQUATION FOR P2:**

**0**

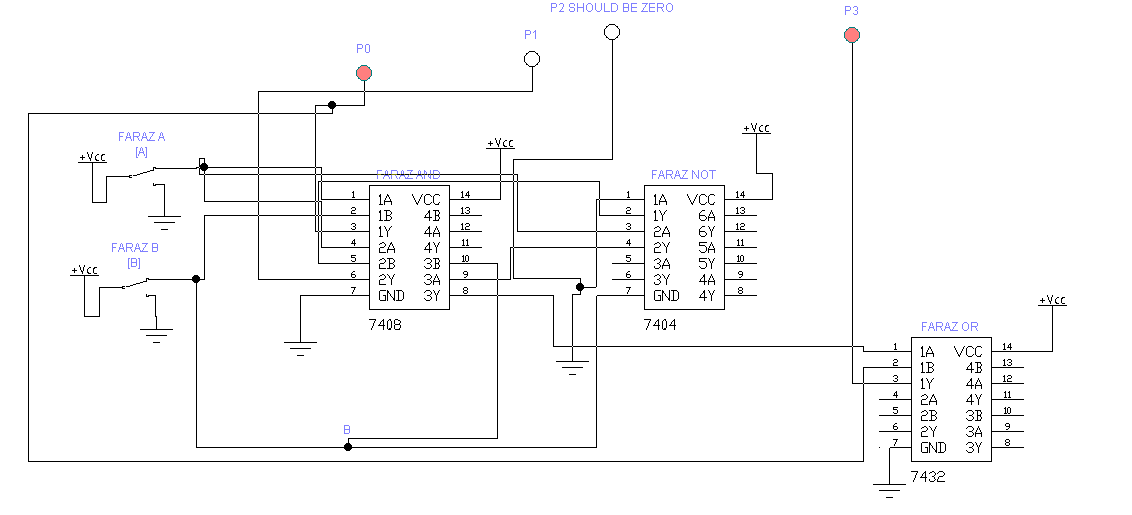
* **BOOLEAN EQUATION FOR P3:**

**A’B+AB**

* **GATE LEVEL DIAGRAM FOR CLC-SQUARE CIRCUIT:**



* **IC-LEVEL DIAGRAM FOR CLC-SQUARE CIRCUIT:**



* **EXPLAINATION:**
* First we have to identify that there are ‘2-bit’ input so that the numbers in range are 0-3.
* Then we have to take the square of highest possible number i-e ‘3’ which is ‘9’(1001) i-e maximum 4-bit output possible.
* After that we have to make the truth table and pick Boolean equation from each bit output and draw the circuit accordingly.

**QUESTION NO : 6 (b)**

1. **Cascade two 8 \*1 and one 2 \*1 multiplexers to implement 16 \*1 multiplexer logic.**

* **TRUTH TABLE OF EXPECTED 16\*1 MUX:**
* **Where S0 , S1 , S2 and S3  are selecting input lines and S3 being MSB.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S3** | **S2** | **S1** | **S0** | **OUTPUT FROM** |
| **0** | **0** | **0** | **0** | **I0** |
| **0** | **0** | **0** | **1** | **I1** |
| **0** | **0** | **1** | **0** | **I2** |
| **0** | **0** | **1** | **1** | **I3** |
| **0** | **1** | **0** | **0** | **I4** |
| **0** | **1** | **0** | **1** | **I5** |
| **0** | **1** | **1** | **0** | **I6** |
| **0** | **1** | **1** | **1** | **I7** |
| **1** | **0** | **0** | **0** | **I8** |
| **1** | **0** | **0** | **1** | **I9** |
| **1** | **0** | **1** | **0** | **I10** |
| **1** | **0** | **1** | **1** | **I11** |
| **1** | **1** | **0** | **0** | **I12** |
| **1** | **1** | **0** | **1** | **I13** |
| **1** | **1** | **1** | **0** | **I14** |
| **1** | **1** | **1** | **1** | **I15** |

* **ROUGH SKETCH FOR ABOVE CASCADING:**

**SELECT INPUT LINE(MSB)**

**SELECT INPUT LINES**

**SELECT INPUT LINES**

**16 INPUT LINES**

**1 OUTPUT LINE**

**FARAZ**

**2\*1 MUX**

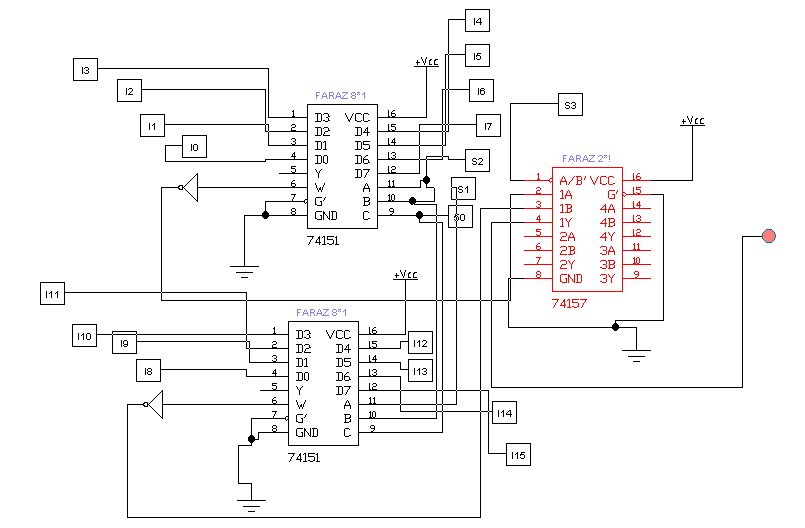
**FARAZ**

**8\*1 MUX**

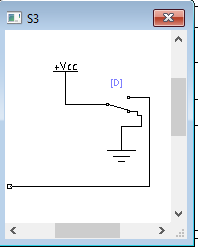
**FARAZ**

**8\*1 MUX**

* **IC-LEVEL IMPLEMENTATION FOR 16\*1 MUX CASCADING:**



* **SUB-CIRCUIT FOR EACH SWITCH:**
* **Where toggle key for each switch is different**

* **EXPLAINATION:**

In 16\*1 multiplexer , there are 16 input lines and one output line with 4 selecting input lines. The selecting input lines decides which input line transmit the signal to output.

To cascade the above from two 8\*1 and one 2\*1 mux, we have 3 select input lines in both 8\*1 mux. So connect the selecting inputs of both 8\*1 mux with eachother which are S0 , S1 , S2. The 2\*1 MUX has only one selecting input line which become the 4th one which is S3. In the circuit **S3  will be MSB.**

This S3 will be responsible to switch two 8\*1 mux to generate desired output.

For example, if the input is 1000 .Since in this example S3 is ‘1’ then output will be from second 8\*1 mux then later bits are ‘000’ hence output will be from ‘D0(from IC)’ which is ‘I8’.

**ii. Draw the logic diagram of a 2-to-4 line decoder using NOR gates only. Include an enable**

**input. (Attach Simulated circuit diagram as well).**

* **TRUTH TABLE FOR 2\*4 DECODER:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **E** | **A** | **B** | **P0** | **P1** | **P2** | **P3** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **0** | **0** | **1** |
| **0** | **DON’T CARE** | **DON’T CARE** | **0** | **0** | **0** | **0** |

* **BOOLAEN EQUATION FOR P0:**

**E A’ B’**

* **BOOLAEN EQUATION FOR P1:**

**E A’ B**

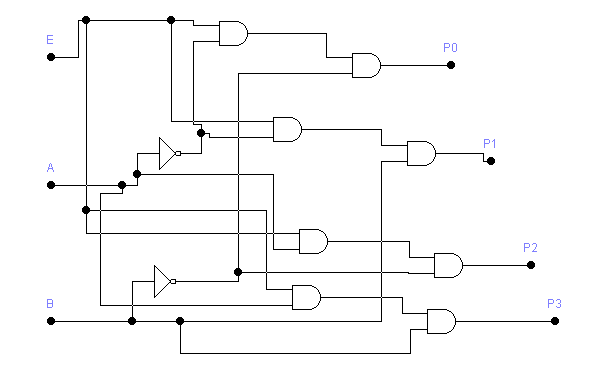
* **BOOLAEN EQUATION FOR P2:**

**E A B’**

* **BOOLAEN EQUATION FOR P3:**

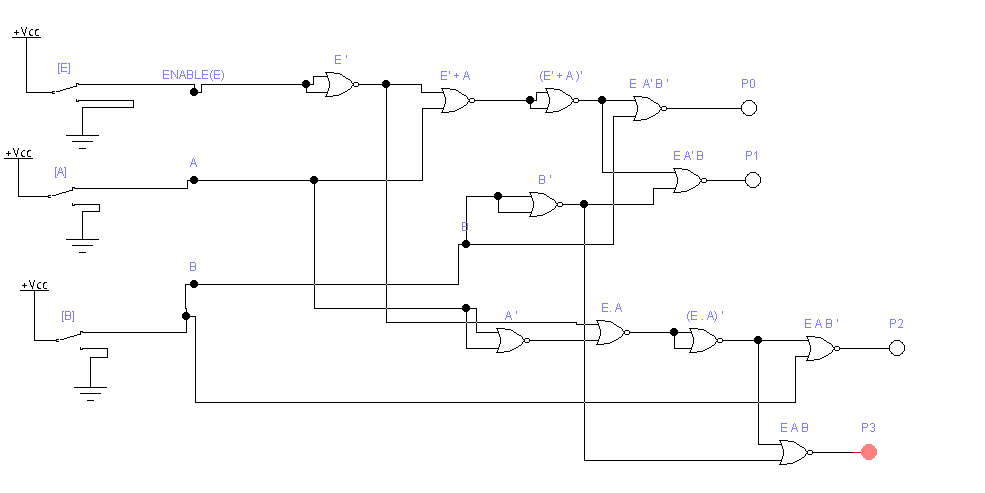
**E A B**

* **CIRCUIT FOR 2\*4 LINE DECODER WITHOUT USING NOR-GATES:**

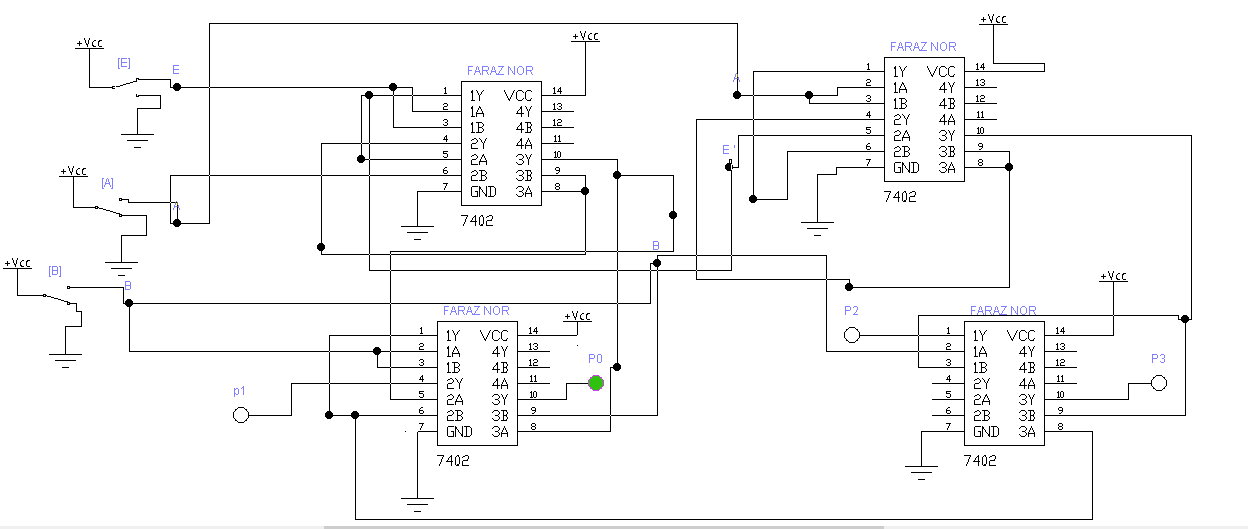


* **NOR – REALIZATION OF ABOVE CIRCUIT:**

By applying NOR-REALIZATION to above simple circuit, we got 2\*4 line decoder WITH only NOR-GATES.



* **IC-LEVEL IMPLEMENTATION FOR ABOVE DECODER:**



* **EXPLAINATION:**

From truth table of 2\*4 line decoder, we pick the Boolean equations for each output lines i-e P0 , P1 , P2 and P3.

Then we draw the simple circuit without using NOR GATES.

After all we applied NOR-REALIZATION METHOD to convert that circuit to NOR-GATES.

In NOR-REALIZATION we all know that AND gate can be replace by three NOR gates and NOT gate can be replace by one NOR gate.

**QUESTION NO: 7(a)**

**How well you understand Multiplexer, De- multiplexer, and Encoder? Explain with suitable applications**

* **MULTIPLEXER:**
* It is a device which accepts ‘2n’ number of signals and output only one signal of desired frequency at a time.
* The lines which decides whether which input line’s signal go towards the output are called selecting input lines which are ‘n’ in number.
* For example, consider 2\*1 MUX . It has 2 inputs i-e ‘21’ and 1 selecting input line and 1 output.

**2N INPUT LINES**

**MUX**

**1 OUTPUT**

**.**

**.**

**.**

**. . ….**

**N-SELECT LINES**

* **APPLICATIONS OF MULTIPLEXER:**
* **COMPUTER COMMUNICATION SYSTEMS:**

The purpose of communication systems is to receive the signals from transmitter(input) and select the desired frequency signal and move it towards the receiver(output). To get the things done, multiplexers are used.

* **COMPUTER MEMORY:**

Multiplexers are used in computer memory used to fetch desired data from memory. Our data whole data is connected to its input line and computer has just to give the selecting input to pick the desired data.

* **DE-MULTIPLEXER:**
* It is a device which accept ‘only 1’ signal as input and output ‘2n’ signals of desired frequency at a time. It has ‘n’ selecting inputs.
* For example, consider 1\*8 line decoder. It has 1 input and ‘23’ i-e ‘8’ outputs. And 3 select inputs.
* It is the reverse of multiplexer.
* It receive only one signal and output the signal at the desired stations(outputs).

**1 INPUT LINE**

**DE-MUX**

**2­N OUTPUT LINES**

* **. .**
  + - **.**

**N SELECT INPUTS**

* **APPLICATIONS OF DE-MULTIPLEXER:**
* **COMPUTER COMMUNICATION SYSTEMS:**

The purpose of communication systems is to receive the signals from transmitter(input) and select the desired frequency signal and move it towards the receiver(output). To get the things done, DE-multiplexers are also used.

* **ARITHMATIC AND LOGIC UNIT:**

The purpose of demultiplexers in ALU is to receive the arithmetic and logical operations output from ALU and move it to the required stations through multiple output lines.

* **ENCODER:**
* An encoder gives the opposite output as that of a decoder.
* Encoder has ‘**2n**’ input lines and ‘**n**’ output lines which is the reverse of decoder.
* For example, consider 4\*2 encoder. It has 4 inputs and 2 outputs. If the ‘I0’ of the decoder input will give signal i-e ‘on’ then the output will be its subscript code i-e ‘00’(0). And so on.

**2N INPUT LINES**

**ENCODER**

**N OUTPUT**

**.**

**.**

**.**

* **. .**

**.**

* **APPLICATIONS OF ENCODER:**
* Analog to digital converter
* robotics

**QUESTION NO: 7(b)**

Examine the logic of full adder circuit and implement with:

i. A decoder. Draw neat and clean diagram. (Attach Simulated circuit diagram as well).

* **TRUTH TABLE OF FULL ADDER:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A(MSB)** | **B** | **CIN** | **SUM** | **COUT** | **DECODER** |
| **0** | **0** | **0** | **0** | **0** | **Y0** |
| **0** | **0** | **1** | **1** | **0** | **Y1** |
| **0** | **1** | **0** | **1** | **0** | **Y2** |
| **0** | **1** | **1** | **0** | **1** | **Y3** |
| **1** | **0** | **0** | **1** | **0** | **Y4** |
| **1** | **0** | **1** | **0** | **1** | **Y5** |
| **1** | **1** | **0** | **0** | **1** | **Y6** |
| **1** | **1** | **1** | **1** | **1** | **Y7** |

* **BOOLEAN EQUATION FOR SUM:**

**A’B’CIN + AB’C’IN + A’BC’IN + ABCIN**

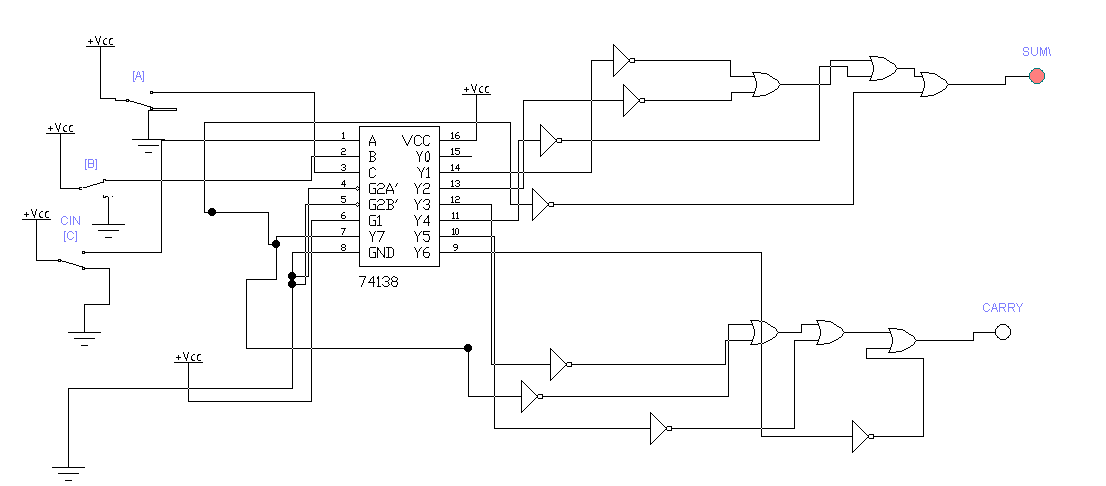
**SUM= Σ (Y1 , Y2 , Y4 , Y7)**

* **BOOLEAN EQUATION FOR CARRY:**

**AB’CIN + A’BCIN + ABC’IN + ABCIN**

**CARRY= Σ (Y3 , Y5 , Y6 , Y7)**

* **SIMULATED CIRCUIT:**
* C being MSB in simulated circuit
* Outputs are active low that’s why connecting not-gate



* **EXPLAINATION:**

In 3\*8 line Decoder, we have 3 input lines and 8 output lines. It converts the binary signal into decimal code from Y0 to y7 i-e 8 numbers. So according to the truth table, for sum, note the number of ones in the SUM-CLOUMN and sum(ORed) the related min-terms. similarly, for carry, note the number of ones in CARRY-COLUMN and sum(ORed) the related min-terms.

**ii. two 4 \*1 multiplexers. (Attach Simulated circuit diagram as well).**

* **TRUTH TABLE OF FULL ADDER:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A(MSB)** | **B** | **CIN** | **SUM** | **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** |

* **BOOLEAN EQUATION FOR SUM:**

**A’B’CIN + AB’C’IN + A’BC’IN + ABCIN**

**SUM= Σ (1,2,4,7)**

* **K-MAP FOR SUM:**

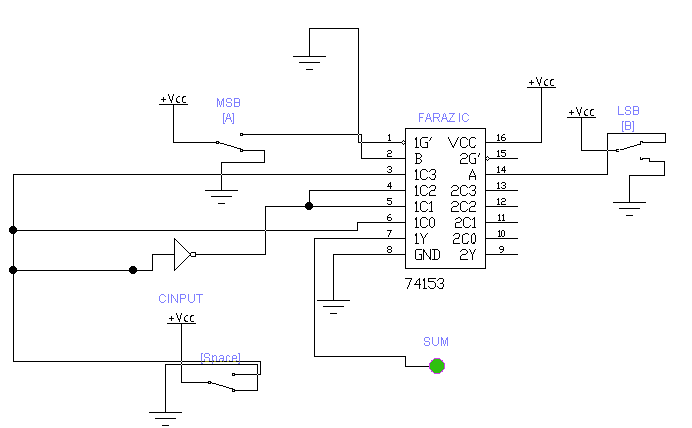
**AB CIN 0 1**

|  |  |  |
| --- | --- | --- |
| **I0**  **00** | **0** | **1** |
| **I1**  **01** | **1** | **0** |
| **I3**  **11** | **0** | **1** |
| **I2**  **10** | **1** | **0** |

* **EXPLAINATION:**

**From above K-MAP it is clear that in multiplexer I0 , I3 should be CIN and I1 , I2 should be C’IN(NOT).**

* **CIRCUIT FOR SUM USING 4\*1 MUX:**



* **EQUATION FOR CARRY:**

**CARRY=Σ(3,5,6,7)**

* **K-MAP FOR CARRY:**

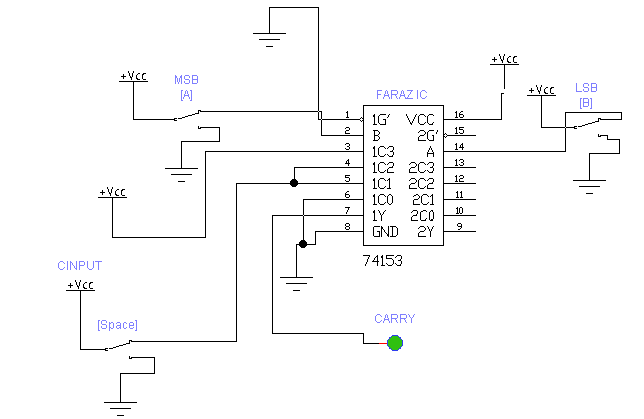
**AB CIN 0 1**

|  |  |  |
| --- | --- | --- |
| **I0**  **00** | **0** | **0** |
| **I1**  **01** | **0** | **1** |
| **I3**  **11** | **1** | **1** |
| **I2**  **10** | **0** | **1** |

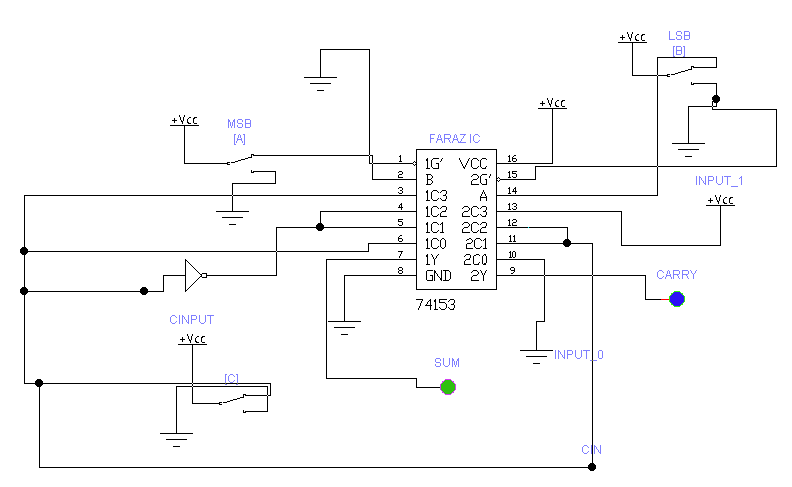
* **EXPLAINATION:**

From above K-MAP, it is clear that I0 will remain ‘0’ so it is connected to ground. Similarly, I3 will remain ‘1’ so It is connected to ‘vcc’ and I1 and I2 will remain which is CIN.

* **CIRCUIT FOR CARRY USING 4\*1 MUX:**



* **COMPLETE CIRCUIT OF FULL ADDER USING TWO 4\*1 MUX:**



* **REFRENCES:**
* **FOR QUESTION 5(a)**
* **Digital logic and computer design fundamentals by m . morris mano and Charles r rime**
* [**https://www.tutorialspoint.com/computer\_logical\_organization/logic\_gates.htm**](https://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm)
* **FOR QUESTION 5(a)**
* **Digital logic and computer design fundamentals by m . morris mano and Charles r rime**
* [**https://www.tutorialspoint.com/computer\_logical\_organization/logic\_gates.htm**](https://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm)
* **FOR QUESTION 6(a)**
* **Digital logic and computer design fundamentals by m . morris mano and Charles r rime ,chapter no 2**
* **FOR QUESTION 6(b)**
* **FOR QUESTION 7(a)**
* [**https://www.elprocus.com/multiplexer-and-demultiplexer/**](https://www.elprocus.com/multiplexer-and-demultiplexer/)
* [**https://www.watelectronics.com/different-types-encoder-decoder-applications/**](https://www.watelectronics.com/different-types-encoder-decoder-applications/)
* **Digital logic and computer design fundamentals by m . morris mano and Charles r rime ,chapter no 4**
* **FOR QUESTION 7(b)**
* **Digital logic and computer design fundamentals by m . morris mano and Charles r rime ,chapter no 4**