Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate’s answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate’s understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No. | Sub Q. N. | Answer | Marking Scheme
--- | --- | --- | ---
Q.1 | Attempt any FIVE of the following: | 10-Total Marks

a) Convert \((D8F)_{16}\) into binary and octal.

Ans:

```
Step 1:  D  F  8  16 \\
1101 1000 1111  \to Binary

(D8F)_{16} = (110110001111)_{2}

Step 2:  1101 1000 1111  \to Octal

\frac{1101}{8} \frac{1000}{8} \frac{1111}{8} \to Octal

(D8F)_{16} = (6617)_{8}
```

2M

 Ans:

b) Draw symbol, Truth table and logic equation of Ex-OR gate.

Ans:

```
Logic Equation = \overline{A}B + \overline{A}B

Truth Table:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

1M 1M 1M
### c) State the DeMorgan’s Theorems.

**Ans:**

De Morgan’s 1<sup>st</sup> Theorem: Complement of sum is equal to product of their individual complements.

\[ \overline{A + B} = \overline{A} \cdot \overline{B} \]

De Morgan’s 2<sup>nd</sup> Theorem: Complement of product is equal to sum of their individual complements.

\[ \overline{A \cdot B} = \overline{A} + \overline{B} \]

### d) Convert the following expression into standard SOP form.

**Y = AB + A\overline{C} + BC**

**Ans:**

\[
Y = AB + A\overline{C} + BC \\
\text{Total variable ABC} \\
1^{\text{st}} \text{Product term} = AB \ (C \text{ is missing}) \\
2^{\text{nd}} \text{Product term} = A\overline{C} \ (B \text{ is missing}) \\
3^{\text{rd}} \text{Product term} = BC \ (A \text{ is missing}) \\
Y = AB \cdot 1 + A\overline{C} \cdot 1 + BC \cdot 1 \\
Y = AB(C + \overline{C}) \cdot (B + \overline{B}) + BC(A + \overline{A}) \\
Y = ABC + AB\overline{C} + A\overline{B}\overline{C} + ABC + \overline{A}BC \\
Y = ABC + AB\overline{C} + A\overline{B}\overline{C} + \overline{A}BC \text{ Standard SOP Form}
\]

### e) Draw symbol and write truth table of D and T Flip Flop.

**Ans:**

(Note: Symbol with other triggering method also can be consider)

#### Symbol

![Symbol](image)

#### Truth Table

<table>
<thead>
<tr>
<th>D</th>
<th>Q&lt;sub&gt;n&lt;/sub&gt;</th>
<th>Q&lt;sub&gt;n+1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### f) Write down number of flip flops are required to count 16 clock pulses.

**Ans:**

No of states = no. of clock pulses = 16
2^n = m  
n = no.of flip flops required  
m= no.of states  
2^n = 16  
n = 4  
4 flip flops are required to count 16 clock pulse.

g) **List the types of DAC**  

**Ans:**  
1) Binary weighted DAC  
2) R–2R ladder network DAC

<table>
<thead>
<tr>
<th>Q.2</th>
<th>Attempt any <strong>THREE</strong> of the following:</th>
<th>12-Total Marks</th>
</tr>
</thead>
</table>
| a)  | Perform the subtraction using 2’S Complement methods.  
(52)\textsubscript{10} – (65)\textsubscript{10} | 4M |

**Ans:**

\[
\begin{array}{c|c|c}
\hline
2 & 5 & 2 & 0 \ (LSB) \\
2 & 2 & 6 & 0 \\
2 & 1 & 3 & 1 \\
2 & 6 & 0 & 1 \ (MSB) \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c}
\hline
2 & 6 & 5 & 1 \ (LSB) \\
2 & 3 & 2 & 0 \\
2 & 1 & 6 & 0 \\
2 & 8 & 1 & 0 \\
2 & 2 & 1 & 0 \ (MSB) \\
\hline
\end{array}
\]

\[
(55)\textsubscript{10} = (110100)\textsubscript{2} \\
(65)\textsubscript{10} = (100001)\textsubscript{2} \\
\]

\[
\begin{array}{c|c}
1000001 & \frac{1}{2} \\
1111110 & + 1 \\
\hline
\end{array}
\]

\[
\begin{array}{c|c}
1111111 & \frac{1}{2} \\
0110100 & + 0111111 \\
\hline
11110011 & \text{As final carry is not generated, result is negative and its 2's complement form is to get final answer take 2's of result} \\
\end{array}
\]

\[
\frac{1}{2} \times (0110100) = \frac{1}{2} \times (1101)\textsubscript{2} \\
\]

\[
\frac{1}{2} \times (0110100) = \frac{1}{2} \times (1101)\textsubscript{2} \\
\]

b) **Simplify the following Boolean Expression and Implement using logic gate.**  

\[ABCD + AB\overline{C}D + ABC\overline{D} + ABCD\]  

4M
c) Minimize the four variable logic function using K map.  
\[ F(A,B,C,D) = \Sigma m(0,1,2,3,5,7,8,9,11,14) \]  
**Ans:**  
Kmap with place value - 1M  
Pair - 1M  
Answer - 2M

d) Implement the following function using demultiplexer.  
\[ f_1 = \Sigma m(0,2,4,6) \]  
\[ f_2 = \Sigma m(1,3,5) \]  
**Ans:**
Q.3

Attempt any THREE of the following:

12-TOTAL MARKS

a)

Realize the following logic expression using only NAND gates.

(i) OR
(ii) AND
(iii) NOT

Ans:

(i) OR

OR gate from NAND gates

INPUT A

INPUT B

OUTPUT

(ii) AND

AND gate

Input A

Input B

NAND gate

NOT gate

Output

(iii) NOT

(out put A bar)

Ans:

4M
b) Draw binary to gray converter and write its truth table.

Ans:

<table>
<thead>
<tr>
<th>Binary Input</th>
<th>Gray Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3</td>
<td>B2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

K-MAP FOR G3:

- G3 = B3

K-MAP FOR G2:

- G2 = \( \overline{B3}B2 + \overline{B2}B3 \)
- = B3 XOR B2

K-MAP FOR G1:
\[ G_0 = \overline{B_1}B_0 + B_1\overline{B_0} \]
\[ = B_1 \text{ XOR } B_0 \]

(Note: Realization of output equation can be done Basic or Universal)

c) Describe the working of JK flip flop with truth table and logic Diagram.

**Ans:**

Logic Diagram:

```
\[ \begin{array}{c}
  J \\
  C_{LK} \\
  K \\
  Q \\
  \overline{Q}
\end{array} \]
```
Working:
The JK flip flop is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”. Due to this additional clocked input, a JK flip-flop has four possible input combinations, “logic 1”, “logic 0”, “no change” and “toggle”. Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: J = S and K = R. The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q. This cross coupling of the SR flip-flop allows the previously invalid condition of S = “1” and R = “1” state to be used to produce a “toggle action” as the two inputs are now interlocked. If the circuit is now “SET” the J input is inhibited by the “0” status of Q through the lower NAND gate. If the circuit is “RESET” the K input is inhibited by the “0” status of Q through the upper NAND gate. As Q and Q are always different we can use them to control the input. When both inputs J and K are equal to logic “1”, the JK flip flop toggles.

### d) Describe the working of 4 bit SISO (serial in serial out) shift register with diagram and waveform if input is 01101.

**Ans:**
Diagram:(use SR or JK or D type flip flop)

![Diagram of 4 bit SISO shift register](image)

**Working:**
The DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register** or SISO.
The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk). The logic circuit diagram below shows a generalized serial-in serial-out shift register, Output of FFA is Q₁,FFB Q₂,FFC Q₃ and FFD is Q₄.
Waveform: (Input is 01101)

Q.4

Attempt any THREE of the following:

a) Design a full Adder using Truth Table and K-map.

Ans:
A full adder is a combinational logic circuit that performs addition between three bits, the two input bits A and B, and carry C from the previous bit.

```
A
B
C - IN
```

```
```

Truth Table:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Truth table 1½ M
b) Describe the working of ring counter using D flip flop with diagram and waveforms.

Ans:

Diagram:

Waveforms:
Working:
The ring counter is a cascaded connection of flip flops, in which the output of last flip flop is connected to input of first flip flop. In ring counter if the output of any stage is 1, then its reminder is 0. The Ring counters transfer the same output throughout the circuit. That means if the output of the first flip flop is 1, then this is transferred to its next stage i.e. 2nd flip flop. By transferring the output to its next stage, the output of first flip flop becomes 0. And this process continues for all the stages of a ring counter. If we use n flip flops in the ring counter, the ‘1’ is circulated for every n clock cycles.

**Explanation:**

1 M

<table>
<thead>
<tr>
<th>Waveform</th>
</tr>
</thead>
<tbody>
<tr>
<td>:1½ M</td>
</tr>
</tbody>
</table>

**c)** *Draw block diagram of programmable logic Array.*

**Ans:**

![Diagram](image)

**d)** *Compare the following:*

(i) Volatile with Non Volatile.
(ii) EPROM with EEPROM.

**Ans:**

<table>
<thead>
<tr>
<th>Volatile with Non Volatile.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>definition</td>
</tr>
<tr>
<td>classification</td>
</tr>
<tr>
<td>Effect of power</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>applications</td>
</tr>
</tbody>
</table>

ii) EPROM with EEPROM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EPROM</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stands for</td>
<td>Erasable Programable Read-Only Memory.</td>
<td>Electrically Erasable Programable Read-Only Memory.</td>
</tr>
<tr>
<td>Basic</td>
<td>Ultraviolet Light is used to erase the content of EPROM.</td>
<td>EEPROM contents are erased using electrical signal.</td>
</tr>
<tr>
<td>Appearance</td>
<td>EPROM has a transparent quartz crystal window at the top.</td>
<td>EEPROM are totally encased in an opaque plastic case.</td>
</tr>
<tr>
<td>Technology</td>
<td>EPROM is modern version of PROM.</td>
<td>EEPROM is the modern version of EPROM.</td>
</tr>
</tbody>
</table>

e) Describe the working principal of successive approximation ADC.

Ans: Note: Other relevant diagram and explanation also can be considered.

Diagram:

Working:
The successive approximation A/D converter is as shown in fig. An analog voltage (Va) is constantly compared with voltage Vi, using a comparator. The output produced by comparator (Vo) is applied to an electronic Programmer. If Va=Vi, then Vo=0 & then no conversion is required. The programmer displays the value of Vi in the form of digital O/P. But if Va Vi, then the O/P is changed by the programmer. If Va> Vi, then value of Vi is increased by 50% of earlier value. But if Va< Vi, then value of Vi is decreased by 50% of earlier value.

This new value is converted into analog form, by D/A converter so as to compare it with Va again. This procedure is repeated till we get Va=Vi. As the value of Vi is changed successively, this method is called as successive-approximation A/D converter.
When the starts signal goes low the successive approximation register SAR is cleared and output voltage of DAC will be 0v. When start goes high the conversion starts. After starts, during first clock pulse the control circuit set MSB bit so SAR output will be 1000 0000. This is connected as input to DAC so output of DAC is compared with Vin input voltage. If $V_{DAC}$ is more than Vin the comparator output –Vsat, if $V_{DAC}$ is less than Vin, the comparator output is +Vsat.

If output of DAC i.e. $V_{DAC}$ is +Vsat (i.e. unknown analog input voltage $V_{in}> V_{DAC}$) then MSB bit is kept set, otherwise it is reset.

Consider MSB is set so SAR will contain 1000 0000. The next clock pulse will set next bit i.e. D6 bit is kept as it is, but if it –Vsat the D6 bit reset. The process of checking and taking decision to keep bit set or to reset is continued upto D0. Then the DAC input will be digital data equal to analog input.

When the conversion is finished the control circuits sends out an end of conversion signal and data is locked in buffer register.

Q.5 Attempt any TWO of the following:

12-M

(a) Convert the following binary number $(11001101)_2$ into Gray Code and Excess-3 Code.

2M
(ii) Perform the BCD Addition.

\[(17)_{10} + (57)_{10}\]

Ans:

\[
\begin{align*}
(17)_{10} & \quad 0001 \quad 0111 \\
(57)_{10} & \quad + \quad 0101 \quad 0111 \\
& \quad \hline \\
& \quad 0110 \quad 1110 \\
& \quad \text{Valid} \quad \text{Invalid} \\
& \quad \text{BCD} \quad \text{BCD} \\
\end{align*}
\]

\[\frac{1}{2} \text{ Each step}\]

ADD 0110 TO Invalid BCD

\[
\begin{align*}
1 & \quad 11 \\
0110 & \quad 1110 \\
+ & \quad 0000 \quad 0110 \\
01110100 & \quad \hline \\
7 & \quad 4 \\
& \quad (74)_{10} \\
\end{align*}
\]

\[\frac{1}{2} \text{ Each step}\]

(iii) Perform the binary addition.

\[(10110 \cdot 110)_{2} + (1001 \cdot 10)_{2}\]

Ans:

\[
\begin{align*}
10110.110 & - (1001.10)_{2} = (100000.010)_{2} \\
11111 & - (10110.110) \\
+ & \quad 1001.10 \\
100000.010 & \quad \hline \\
\end{align*}
\]

(b) Design a 4bit ripple counter using JK flip flop, with truth table and waveforms.

Ans:

Circuit Diagram:
Truth Table:

<table>
<thead>
<tr>
<th>State</th>
<th>( Q_a )</th>
<th>( Q_b )</th>
<th>( Q_c )</th>
<th>( Q_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Timing Diagram / Waveforms:

(c) Calculate the analog output for 4 bit weighted register type DAC for inputs
(i) 1011
(ii) 1001
Assume (\( V_{fs} \)) full scale range of voltage is 5V

Ans:

Given:
\[ VR = V_{fs} = 5V \]

Formula Used:
\[ Vo = - VR \left( B1.2^{-1} + B2.2^{-2} + B3.2^{-3} + B4.2^{-4} \right) \]

1. 1011
\[ Vo = - VR \left( B1.2^{-1} + B2.2^{-2} + B3.2^{-3} + B4.2^{-4} \right) \]
\[ V_o = \frac{-5}{1} \left( \frac{1}{2} + 0 + \frac{1}{2^3} + \frac{1}{2^4} \right) \]
\[ V_o = \frac{-5}{1} \left( \frac{1}{2} + \frac{1}{16} \right) \]
\[ V_o = \frac{-5}{1} \left( 0.5 + 0.0625 \right) = 3.4375V \]

2. \[ V_o = -V_R \left( B_{1.2}^{-1} + B_{2.2}^{-2} + B_{3.2}^{-3} + B_{4.2}^{-4} \right) \]
\[ V_o = -10 \left( \frac{1}{2} + 0 + 0 + \frac{1}{2^4} \right) \]
\[ V_o = -10 \left( 0.5 + 0.0625 \right) = 2.8125V \]

Q.6

Attempt any TWO of the following:

(a)

Compare TTL, CMOS and ECL logic family on the following points.

(i) Basic Gates
(ii) Propogation delay
(iii) Fan out
(iv) Power Dissipation
(v) Noise immunity
(vi) Speed power product

Ans:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>TTL</th>
<th>CMOS</th>
<th>ECL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic gates</td>
<td>NAND</td>
<td>NOR/NAND</td>
<td>OR/NOR</td>
</tr>
<tr>
<td>Propagation delay</td>
<td>10</td>
<td>70-105</td>
<td>2</td>
</tr>
<tr>
<td>Fan out</td>
<td>10</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>Power Dissipation</td>
<td>10mW</td>
<td>1.01mW</td>
<td>40-55mW</td>
</tr>
<tr>
<td>Noise Immunity</td>
<td>0.2V</td>
<td>5V</td>
<td>0.25V</td>
</tr>
<tr>
<td>Speed Power Product</td>
<td>100</td>
<td>0.7</td>
<td>100</td>
</tr>
</tbody>
</table>

(b) Design a BCD adder using IC 7483.

Ans: (Note: Labeled combinational circuit can be drawn using universal gate also)

1) To implement BCD adder we require:
   • 4-bit binary adder for initial addition
   • Logic circuit to detect sum greater than 9
   • One more 4-bit adder to add 0110201102 in the sum if sum is greater than 9 or carry is 1
2) The logic circuit to detect sum greater than 9 can be determined by simplifying the
Boolean expression of given truth Table.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_3)</td>
<td>(S_2)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ Y = S_3 \cdot S_2 + S_3 \cdot S_1 \]

3) \(Y=1\) indicates sum is greater than 9. We can put one more term, \(C_{\text{out}}\) in the above expression to check whether carry is one.
4) If any one condition is satisfied we add 6(0110) in the sum.
5) With this design information we can draw the block diagram of BCD adder, as shown in figure below.
Design a 3 bit synchronous counter using JK Flip Flop.

Ans:

1) Step 1:
Construct JK state table with corresponding excitation table:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next state</th>
<th>Flip-flop inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2 Q1 Q0</td>
<td>Q2 Q1 Q0</td>
<td>J2 K2 J1 K1 J0 K0</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 1 0</td>
<td>0 X 1 X X 1 X 1</td>
</tr>
<tr>
<td>0 0 1</td>
<td>1 0 0</td>
<td>1 X 1 X X 0 X 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 0</td>
<td>X 1 X X 0 X 1 X</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 0 0</td>
<td>1 X X 1 X X 1 X 1</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 0 0</td>
<td>X 0 0 X 1 X X 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 0 1</td>
<td>X 0 0 X 1 X X 1</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 0 0</td>
<td>X 0 X 0 1 X X 1</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0 0</td>
<td>X 1 X X 1 X X 1</td>
</tr>
</tbody>
</table>

State Table and Corresponding Excitation Table (d=don't care)
Step 2:
Build Karnaugh Map or Kmap for each JK inputs:

\[
\begin{array}{ccc}
\text{Q0} & \text{Q1} & \text{J0} \\
00 & 0 & 0 \\
01 & 1 & X \\
11 & X & X \\
10 & X & X \\
\end{array}
\quad
\begin{array}{ccc}
\text{Q0} & \text{Q1} & \text{J1} \\
00 & 0 & 1 \\
01 & X & X \\
11 & X & X \\
10 & 0 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
\text{Q0} & \text{Q1} & \text{J2} \\
00 & 0 & 1 \\
01 & X & X \\
11 & X & X \\
10 & 0 & 1 \\
\end{array}
\]

\(J2 = Q1.Q0\)
\(J1 = Q0\)
\(J0 = 1\)

\(K2 = Q1.Q0\)
\(K1 = Q0\)
\(K = 1\)

Step 3:
Draw the complete design as below:

Note: It can also be designed using any other Flip Flop.