KIIT, Deemed to be University
School of Electronics Engineering
Digital System Design Laboratory [EC 29005]

EXPERIMENT - 6

Aim:

Design and simulation of modulo counter using Verilog behavioral modeling.
Design modulo counter using JK flip-flops.

Component/Software Used:

Component/Software	Specification
ICs	7476, 7408, 7490, 7447
Bread Board, Power supply, LEDs, Resistors,Switches, Connecting wires	As per requirement
Software(s) Used	Vivado 2016.1

Theory:

Flip-flop:

A counter is one of the most beneficial and adaptable components of a digital system . The number of clock cycles can be counted using a counter that is powered by a clock. The function of a counter is to count the number of clock pulses. Since the clock pulses occur at known intervals, the counter can be used as an instrument for measuring time and, therefore, period or frequency. It can be used to keep track of the number of events that have occurred, such as the number of times a button has been pressed or the number of pulses received from a sensor. Flip-flops are the basic building blocks of a counter. There are basically two different types of counters synchronous and asynchronous.

An asynchronous binary counter is also called a serial or ripple counter. The ripple counter is simple and straightforward in operation and its construction requires a minimum amount of hardware. It does, however, have a speed limit. Each flip-flop is triggered by the previous flip-flop, and thus the counter has a cumulative settling time.

A synchronous binary counter is also called a parallel counter. An increase in the speed of operation can be achieved by use of a parallel or synchronous counter. Here, every flip-flop is triggered by the clock (synchronism), and thus the settling time is simply equal to the delay time of a single flip-flop. Usually, the additional hardware is required to get the speed improvement.

Counters can also be designed to count up or down, and the number of bits used to represent the count will determine the maximum number of counts that can be recorded. For example, a 4-bit counter can count from 0 to 15, while an 8-bit counter can count from 0 to 255.

A digital counter is designed using a set of flip-flops (FFs) whose states change in response to pulses applied at the input to the counter. A counter can be used as a frequency divider to obtain waveform with frequencies that are specific fractions of the clock frequency. They are also used to perform the timing function as in digital watches, to create time delays, to produce non-sequential binary counts, to generate pulse trains, and to act as frequency counters, etc.

The number of states or counting sequences through which a particular counter advances before returning once again back to its original first state is called the modulus (MOD). Modulus Counters, or simply MOD counters, are defined based on the number of states that the counter will sequence through before returning back to its original value. For example, a counter that counts from 00_2 to 11_2 in binary, that is 0 to 3 in decimal, has a modulus value of 4 ($00 \rightarrow 1 \rightarrow 10 \rightarrow 11$, and return back to 00) so would therefore be called a modulo-4, or mod-4, counter. Note also that it has taken four clock pulses to get from 00 to 11.

Design Procedure of a synchronous Counter:

1. Find the number of FFs (flip-flop) required.
The number of Flip-flops required can be determined by using the equation:
$M \leq 2^N$ where, M is the MOD number and N is the number of required flip-flops.
2. Draw a state diagram for a given sequence.
3. Draw the FF transition table (excitation table).
4. Use K-map to derive the logic expressions of inputs of FFs.
5. Implemention using FFs and required combinational logic(gates).

Design of Mod-6 Counter using JK flip-flops:

To design the Mod-6 synchronous counter, contain six counter states (i.e from 0 to 5). No of FFs required = 3 given by Q_A,Q_B and Q_C where Q_C is the MSB and Q_A is LSB to design this counter.

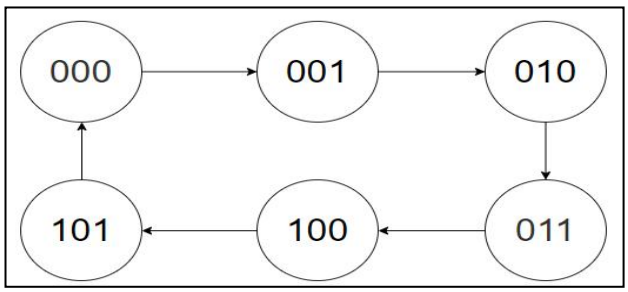


Figure 6.1: State Diagram of MOD-6 synchronous counter

Sl. No.	Clock pulse	Q_C	Q_B	Q_A
1	0	0	0	0
2	1	0	0	1
3	2	0	1	0
4	3	0	1	1
5	4	1	0	0
6	5	1	0	1

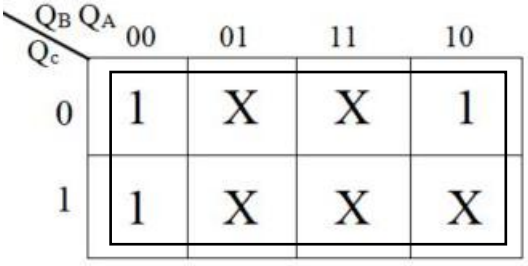
Table 6.1: Count table of MOD-6 counter

Present state			Next State			Inputs of the FFs					
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

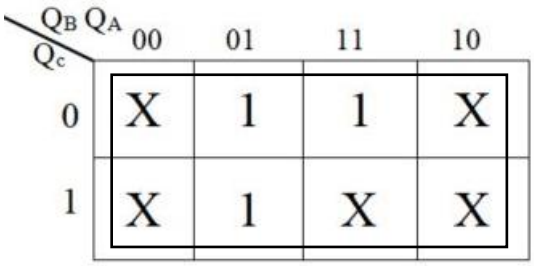
Table 6.2: Excitation table of MOD-6 counter

K-map and logic expressions of inputs of FFs.

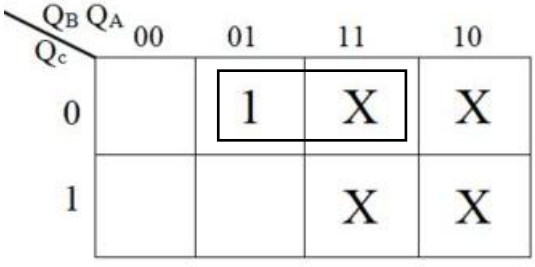
Note: The counts "110" and "111" need to be consider as Don't care (X) in all K-maps.



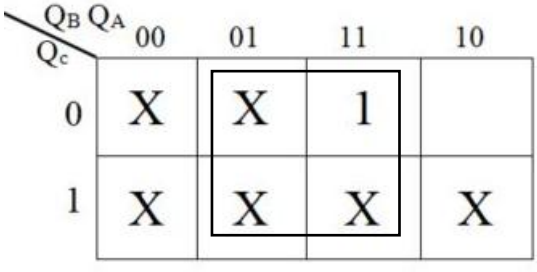
$$J_A(Q_C, Q_B, Q_A) = \sum (0,2,4) + d(1,3,5,6,7) = 1$$



$$K_A(Q_C, Q_B, Q_A) = \sum (0,2,4) + d(1,3,5,6,7) = 1$$



$$J_B(Q_C, Q_B, Q_A) = \sum (1) + d(2,3,6,7)$$
$$= \overline{Q}_C\, Q_A$$



$$K_B(Q_C, Q_B, Q_A) = \sum (3) + d(0,1,4,5,6,7)$$
$$= Q_A$$

Q_C \ $Q_B Q_A$	00	01	11	10
0			1	
1	X	X	X	X

$$J_C(Q_C, Q_B, Q_A) = \sum(3) + d(4,5,6,7) = Q_B Q_A$$

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	X	X
1		1	X	X

$$K_C(Q_C, Q_B, Q_A) = \sum(5) + d(0,1,2,3,6,7) = Q_A$$

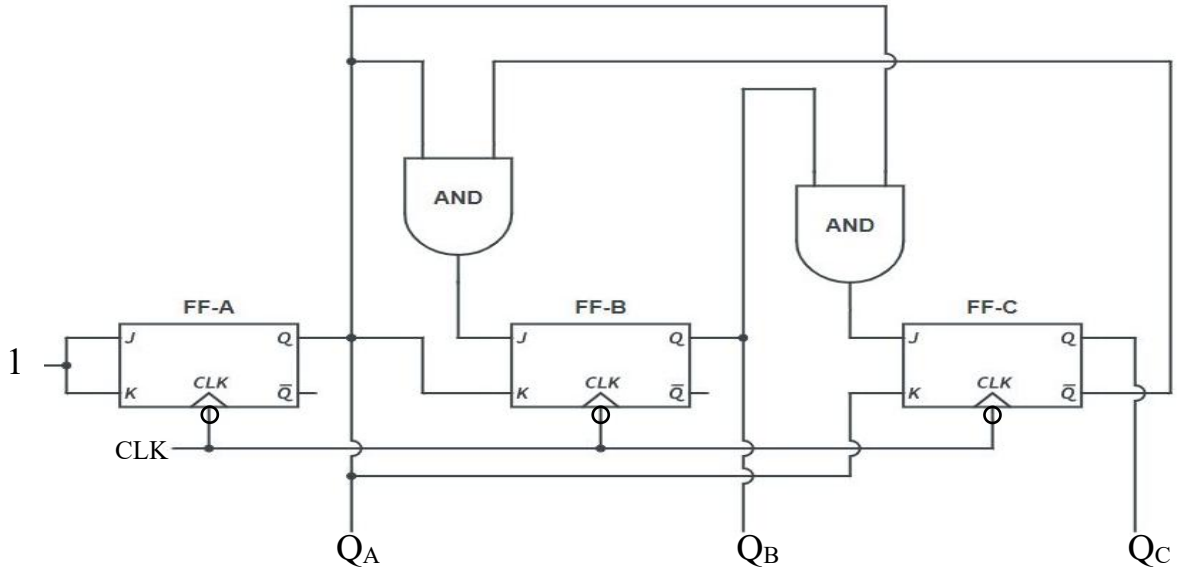


Figure 6.2 Logic Diagram of synchronous binary MOD-6 counter

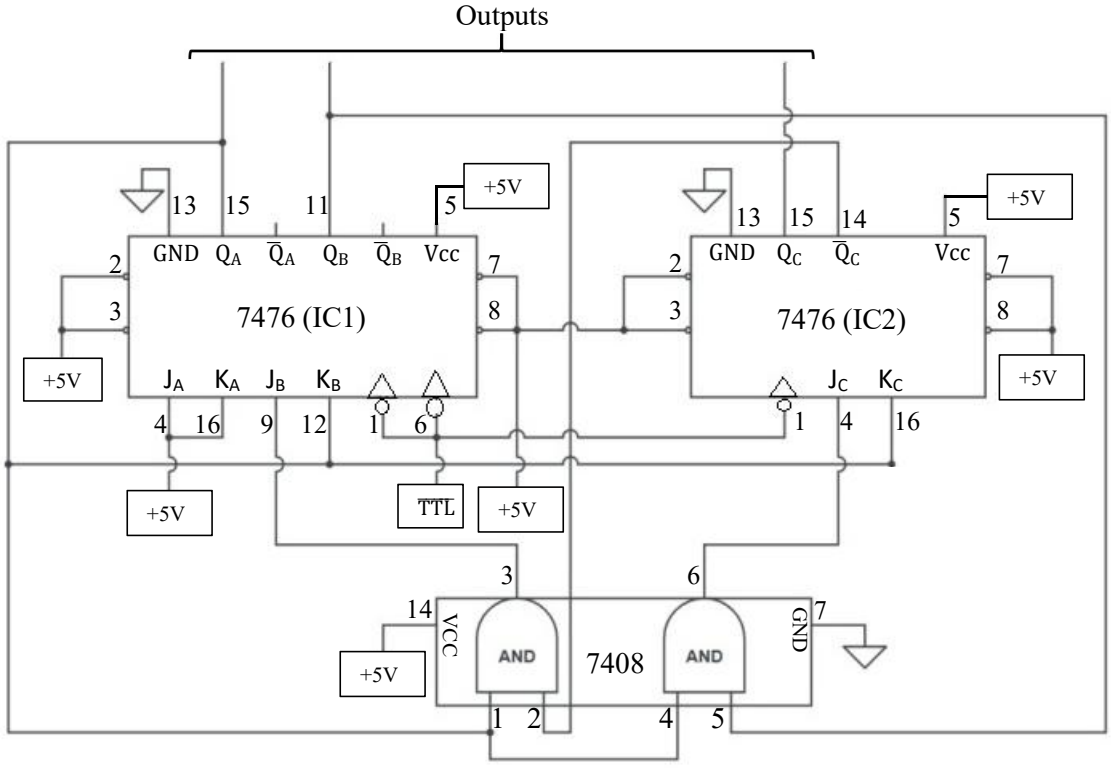


Figure 6.3: Logic Diagram of synchronous binary MOD-6 counter using IC 7476 and IC 7408

Procedure

For Software Simulation:

a) Create a module with required number of variables and mention it's input/output.
b) Write the description of given Boolean function using operators or by using the built in primitive gates.
c) Synthesize to create RTL Schematic.
d) Create another module referred as test bench to verify the functionality and to obtain the waveform of input and output.
e) Follow the steps required to simulate the design and compare the obtained output with the corresponding truth table.
f) Take the screenshots of the RTL schematic and simulated waveforms.

Note: Students need to write the Verilog HDL code by their own for which they can refer Appendix - A if required.

For Hardware implementation:

a) Turn off the power of the Trainer Kit before constructing any circuit.
b) Connect **power supply (+ 5 V DC)** pin and **ground** pin to the respective pins of the trainer kit.
c) Place the ICs properly on the bread board in the Trainer Kit.
d) Connect VCC and GND pins of each chip to the power supply and ground bus strips on the bread board.
e) Connect the input and output pins of chips to the input switches and output LEDs respectively in the Trainer Kit.
f) Check the connections before you turn on the power.
g) Apply various combinations of inputs according to truth tables and observe outputs of LEDs.

Observation:

To be written by students

Design Problem

Design and Simulation of decade counter using Verilog HDL.

Hardware implementation of a decade counter to display the digits 0 to 9 using IC 7490 (decade counter), IC 7447 (BCD to decimal decoder) and seven segment display.

Solution:

IC 7490 : The IC 7490 is 4-bit ripple type Decade Counter. It consists of four master/slave flip-flops which are internally connected to provide a divide-by-two section and a divide-by-five section. Each section has a separate clock input which initiates state changes of the counter on the HIGH-to-LOW clock transition. State changes of the Q outputs do not occur simultaneously because of internal ripple delays. Therefore, decoded output signals are subject to decoding spikes and should not be used for clocks or strobes.

A gated NAND asynchronous Master Reset (MR_1, MR_2) is provided which overrides clocks and resets (clears) all the flip-flops. A gated NAND asynchronous Master Set (MS_1, MS_2) .

Since the output from the divide-by-two section is not internally connected to the succeeding stages, the devices may be operated in various counting modes.

BCD Decade (8421) Counter : The $\overline{CLKB}$ input must be externally connected to the Q_A output. The $\overline{CLKA}$ input receives the incoming count producing a BCD count sequence.

Symmetrical Biquinary Divide-By-Ten Counter: The Q_C output must be externally connected to the $\overline{CLKA}$ input. The input count is then applied to the $\overline{CLKA}$ input and a divide-by ten square wave is obtained at output Q0.

Divide-By-Two and Divide-By-Five Counter: No external interconnections are required. The first flip-flop is used as a binary element for the divide-by-two function ($\overline{CLKA}$ as the input and Q_A as the output). The $\overline{CLKB}$ input is used to obtain binary divide-by-five operation at the Q_C output.

The pin diagram and the logic symbol are given in Figure 6.4 and Figure 6.5 respectively. The count table is shown in Table 6.3. In Figure 6.5, Q_D (MSB) , Q_C,$Q_B$andQ_A (LSB) are the outputs of the Decade Counter.

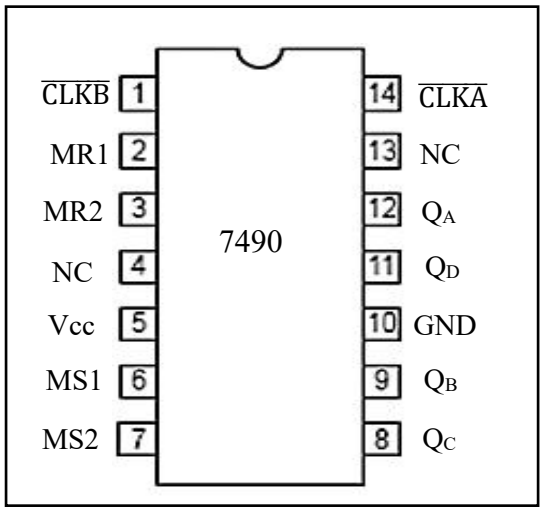


Figure 6.4: Pin diagram of IC 7490

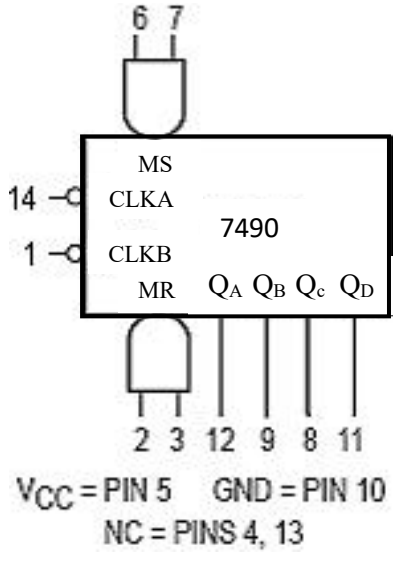


Figure 6.5: Logic symbol of IC 7490

Reset/Set inputs				Outputs			
MR1	MR2	MS1	MS2	Q_D	Q_C	Q_B	Q_A
1	1	0	X	0	0	0	0
1	1	X	0	0	0	0	0
X	X	1	1	1	0	0	1
X	0	X	1	Count			
0	X	0	X	Count			
0	X	X	0	Count			
X	0	0	X	Count			

	Outputs			
Count	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Table 6.3: Mode selection and Count table of 7490 (Decade counter)

IC 7447 :

The IC 7447 is BCD-to-Seven segment decoder. The IC is stand-alone and requires no external components other than the LED current limiting resistors. The output of the IC has complete ripple blanking and requires no external driver transistors. It incorporates automatic leading/or trailing-edge, zero blanking control ripple-blanking input and ripple-blanking output ($\overline{RBI}$ and $\overline{RBO}$). A Lamp test of these devices may be performed at any time when the blanking input / ripple-blanking output ($\overline{BI}/\overline{RBO}$) mode is at a high logic level.

The 7447 decodes the input data in the pattern indicated in the Table 6.4 and the segment identification illustration. If the input data is decimal zero, a LOW signal applied to the $\overline{RBI}$ blanks the display and causes a multi digit display. For example, by grounding the $\overline{RBI}$ of the highest order decoder and connecting its $\overline{BI}/\overline{RBO}$ to $\overline{RBI}$ of the next lowest order decoder, etc., leading zeros will be suppressed. Similarly, by grounding RBI of the lowest order decoder and connecting its $\overline{BI}/\overline{RBO}$ to $\overline{RBI}$ of the next highest order decoder, etc., trailing zeros will be suppressed. Leading and trailing zeros can be suppressed simultaneously by using external gates, i.e.: by driving $\overline{RBI}$ of a intermediate decoder from an OR gate whose inputs are $\overline{BI}/\overline{RBO}$ of the next highest and lowest order decoders. $\overline{BI}/\overline{RBO}$ also serves as an unconditional blanking input. The internal NAND gate that generates the $\overline{RBO}$ signal has a resistive pull-up, as opposed to a totem pole, and thus $\overline{BI}/\overline{RBO}$ can be forced LOW by external means, using wired collector logic. A LOW signal thus applied to $\overline{BI}/\overline{RBO}$ turns off all segment outputs. This blanking feature can be used to control display intensity by varying the duty cycle of the blanking signal. A LOW signal applied to $\overline{LT}$ turns on all segment outputs, provided that $\overline{BI}/\overline{RBO}$ is not forced LOW

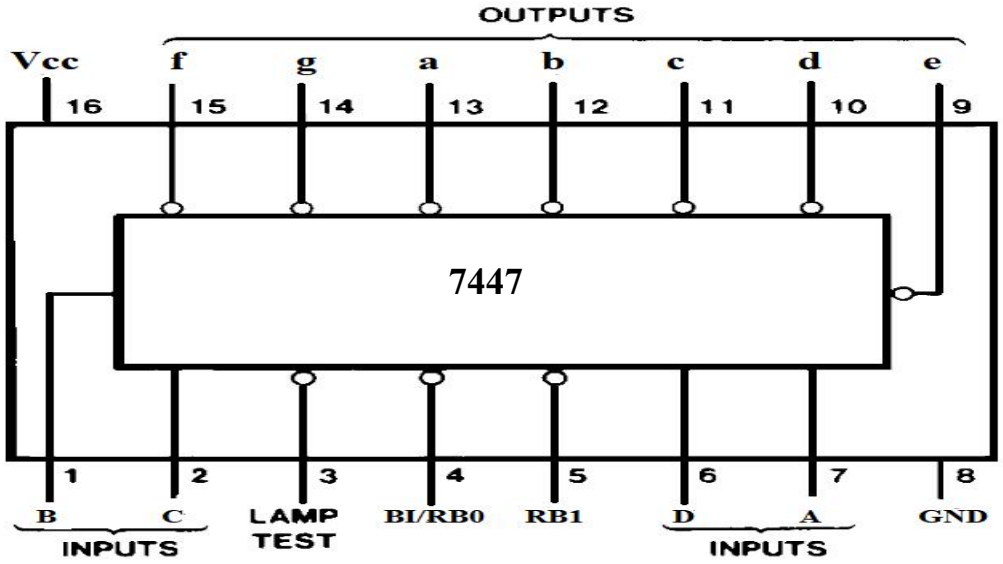


Figure 6.6: Pin diagram of IC 7447

Decimal or function	Inputs						$\overline{BI}$/ $\overline{RBO}$	Outputs							Note
	$\overline{LT}$	$\overline{RBI}$	D	C	B	A		$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$	
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	
1	1	X	0	0	0	1	1	1	0	0	1	1	1	1	
2	1	X	0	0	1	0	1	0	0	1	0	0	1	0	
3	1	X	0	0	1	1	1	0	0	0	0	1	1	0	
4	1	X	0	1	0	0	1	1	0	0	1	1	0	0	
5	1	X	0	1	0	1	1	0	1	0	0	1	0	0	
6	1	X	0	1	1	0	1	1	1	0	0	0	0	0	
7	1	X	0	1	1	1	1	0	0	0	1	1	1	1	
8	1	X	1	0	0	0	1	0	0	0	0	0	0	0	
9	1	X	1	0	0	1	1	0	0	0	1	1	0	0	(1)
10	1	X	1	0	1	0	1	1	1	1	0	0	1	0	
11	1	X	1	0	1	1	1	1	1	0	0	1	1	0	
12	1	X	1	1	0	0	1	1	0	1	1	1	0	0	
13	1	X	1	1	0	1	1	0	1	1	0	1	0	0	
14	1	X	1	1	1	0	1	1	1	1	0	0	0	0	
15	1	X	1	1	1	1	1	1	1	1	1	1	1	1	
$\overline{BI}$	X	X	X	X	X	X	0	1	1	1	1	1	1	1	(2)
$\overline{RBI}$	1	0	0	0	0	0	0	1	1	1	1	1	1	1	(3)
$\overline{LT}$	0	X	X	X	X	X	1	0	0	0	1	0	0	0	(4)

Table 6.4: Function table for IC 7447.

Note (1): The blanking input ($\overline{BI}$) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ($\overline{RBI}$) must be open or high if blanking of a decimal zero is not desired.

Note (2): When a low logic level is applied directly to the blanking input ($\overline{BI}$), all segment outputs are high regardless of the level of any other input.

Note (3): When ripple-blanking input ($\overline{RBI}$) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go high and the ripple-blanking output ($\overline{RBO}$) goes to a low level (response condition).

Note (4): When the blanking-output ($\overline{BI}/\overline{RBO}$) is open or held high and a low is applied to the lamp-test input, all segment outputs are low. Input ($\overline{BI}/\overline{RBO}$) is a wire-AND logic serving as a blanking input ($\overline{BI}$) and /or ripple blanking output ($\overline{RBO}$).

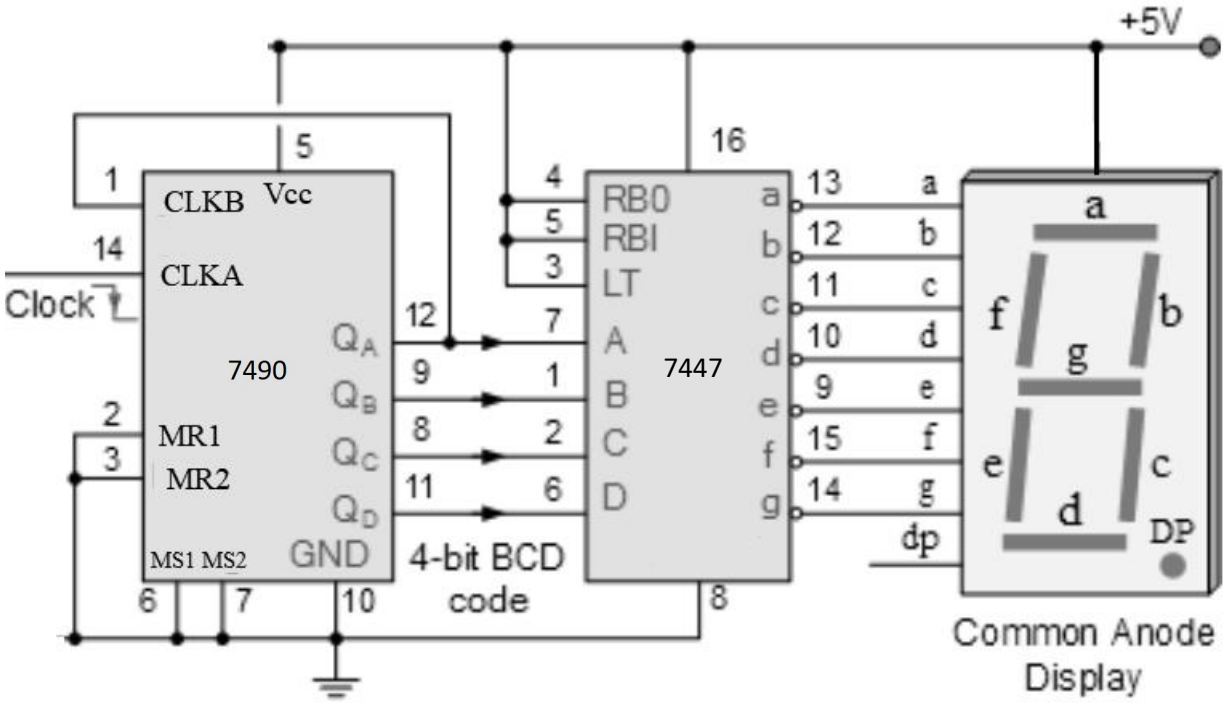


Figure 6.7: Logic diagram of Decade counter using IC 7490 and IC 7447

Conclusion:

To be written by students.

Sample viva-voice questions

1. What is a counter?
2. What are the applications of counters?
3. What do you mean by presetting the counter?
4. What is the difference between asynchronous and synchronous counter?
5. What is the modulus of a counter?
6. What is the maximum modulus of a counter with 5 nos. of flip-flop?
7. In mod-20 counter, how many flip-flops are there?
8. What is a BCD counter?