Hardware Lab (CS 224)
Assignment 4

Group No. 18

Team Members:
Daksh Agarwal (230101123)
Aditya Shukla (230101115)
Kartik Maheshwari (230101116)
Ryan Aggarwal (230101117)

February 18, 2025

Contents

1 Objective

The objective of this lab is to implement and analyze 4 bit synchronous binary UP/DOWN counter on breadboard and verilog.

2 Overview

This counter has an **8-bit input** and a **4-bit output**, functioning as follows:

2.1 Inputs and Outputs

- **Clock (clk)**: Global clock input that drives the counter.
- **Up/Down (up_dn)**: Determines the counting direction:
  - `1` → Counts **up**.
  - `0` → Counts **down**.
- **Set (set)**: Loads the counter with a specified value (`val`).
  - **Takes priority** over the up/down control.
- **Reset (rst)**: Resets the counter to `0000` (or another predefined value).
  - **Has the highest priority.**
- **Value Input (val[3:0])**: 4-bit input used when the counter is set.
- **Counter Output (count[3:0])**: 4-bit output representing the current counter value.

2.2 Priority Order

The operations are executed in the following priority order (from highest to lowest):

1. **Reset (rst)**: Forces the counter to `0000`.
2. **Set (set)**: Loads `val` into the counter.
3. **Up/Down (up_dn)**: Determines whether the counter increments or decrements on each clock pulse.

2.3 Differences: Synchronous vs. Asynchronous Counters

- **Asynchronous Counters:** Also known as ripple counters, these counters trigger the first flip-flop with the external clock, and each subsequent flip-flop is triggered by the preceding one's output. This sequential triggering leads to cumulative delays, impacting precision and making them unsuitable for high-speed applications.
- **Synchronous Counters:** All flip-flops receive the clock signal simultaneously, controlled by combinational logic. This setup eliminates the ripple effect and reduces delay, making synchronous counters ideal for applications needing accurate and fast counting.

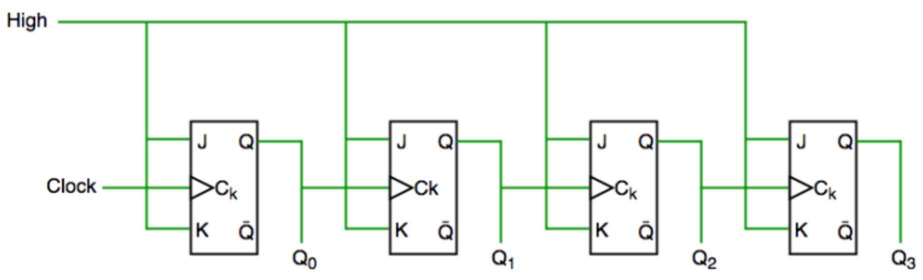


Figure 1: Asynchronous Counter

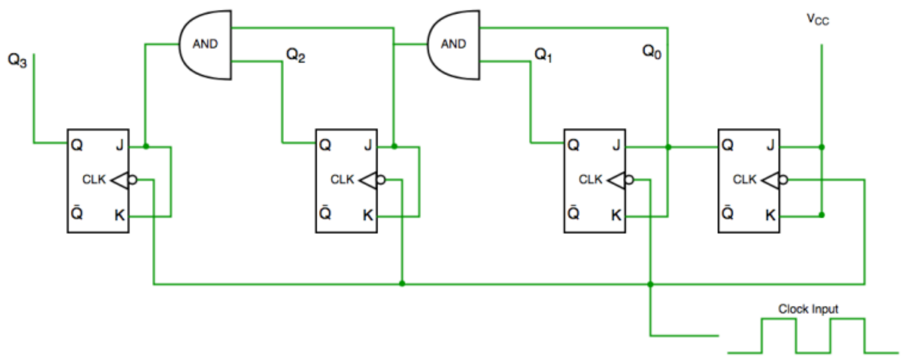


Figure 2: Synchronous Counter

3 IC's Used

The following ICs are used for implementation:

- HEX Inverter - 7404 (1)

- Quad 2 input OR- 7432 (1)
- Quad 2 input AND - 7408 (2)
- JK flip flop with Preset and Clear - 7476 (2)
- Quad 2:1 MUX - 74157 (3)

3.1 JK Flip Flop

JK flip-flops are versatile bistable devices where the output state follows the inputs J (set) and K (reset). It operates with a clock signal, changing states on clock edges based on the input conditions. This flip-flop can toggle its output when both J and K are high, making it suitable for complex sequential circuits.

J	**K**	**Q (Next State)**	**Operation**
0	0	Q (No Change)	Hold State
0	1	0	Reset
1	0	1	Set
1	1	$\bar{Q}$	Toggle

Table 1: Truth Table for JK Flip-Flop

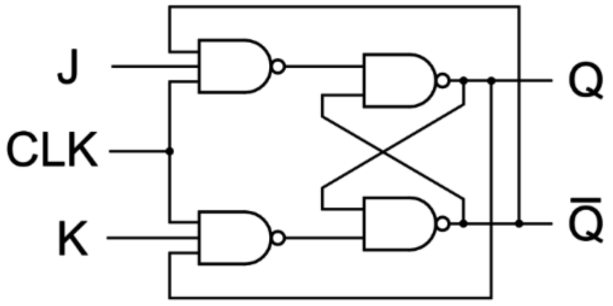


Figure 3: JK Flip-Flop

3.2 Quad 2:1 MUX

A Quad 2:1 Multiplexer (MUX) is a combinational logic circuit that selects one of two inputs (A or B) per output based on a common select signal (S). The 74157 IC contains four independent 2:1 multiplexers, making it useful for handling multiple data lines simultaneously.

Enable (G)	Select (S)	Input A	Input B	Output (Y)
0	0	A	B	A
0	1	A	B	B
1	X	X	X	High Impedance (Disabled)

Table 2: Truth Table for Quad 2:1 Multiplexer (MUX - 74157)

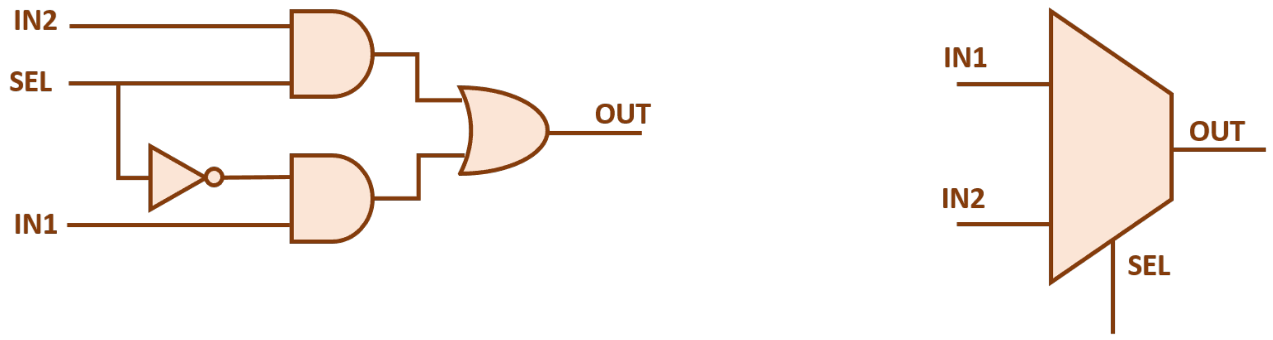
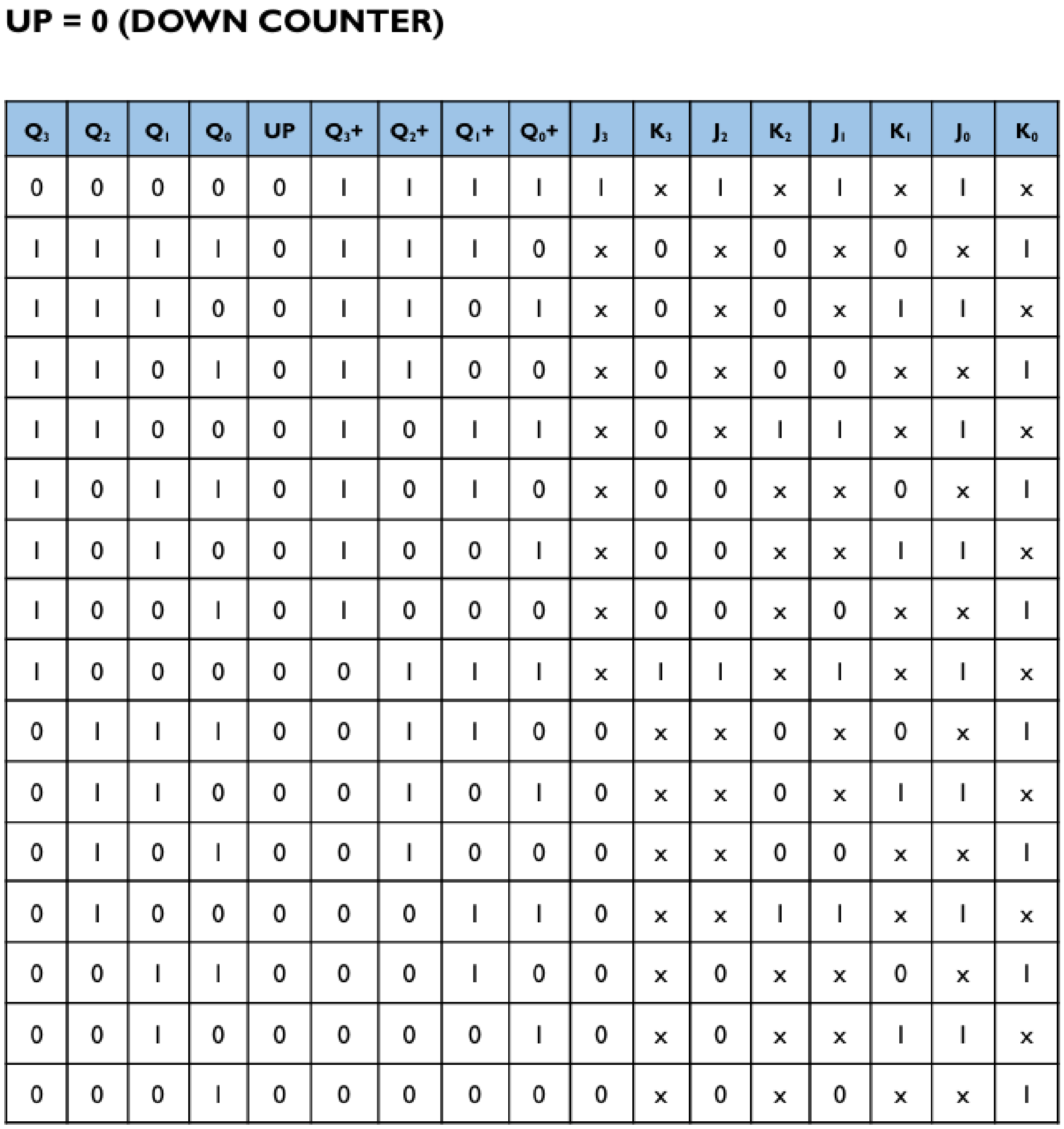


Figure 4: 2:1 Multiplexer

4 Truth Tables

4.1 Down Counter

The following is the truth table when Up=0: When the "Up" input signal of a counter is set to 0, it generally configures the counter to operate in a "down" counting mode. This means that instead of incrementing, the counter will decrement its count value with each clock cycle.


UP = 0 (DOWN COUNTER)

Q_3	Q_2	Q_1	Q_0	UP	Q_3+	Q_2+	Q_1+	Q_0+	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	1	1	1	1	x	1	x	1	x	1	x
1	1	1	1	0	1	1	1	0	x	0	x	0	x	0	x	1
1	1	1	0	0	1	1	0	1	x	0	x	0	x	1	1	x
1	1	0	1	0	1	1	0	0	x	0	x	0	0	x	x	1
1	1	0	0	0	1	0	1	1	x	0	x	1	1	x	1	x
1	0	1	1	0	1	0	1	0	x	0	0	x	x	0	x	1
1	0	1	0	0	1	0	0	1	x	0	0	x	x	1	1	x
1	0	0	1	0	1	0	0	0	x	0	0	x	0	x	x	1
1	0	0	0	0	0	1	1	1	x	1	1	x	1	x	1	x
0	1	1	1	0	0	1	1	0	0	x	x	0	x	0	x	1
0	1	1	0	0	0	1	0	1	0	x	x	0	x	1	1	x
0	1	0	1	0	0	1	0	0	0	x	x	0	0	x	x	1
0	1	0	0	0	0	0	1	1	0	x	x	1	1	x	1	x
0	0	1	1	0	0	0	1	0	0	x	0	x	x	0	x	1
0	0	1	0	0	0	0	0	1	0	x	0	x	x	1	1	x
0	0	0	1	0	0	0	0	0	0	x	0	x	0	x	x	1

4.2 Up Counter

The following is the truth table when Up=1: When the "Up" input is set to 1, the counter increments its value with each clock pulse. In a synchronous counter, all flip-flops toggle simultaneously based on combinational logic, eliminating propagation delays. The least significant bit toggles every clock cycle, while higher-order bits toggle only when all preceding bits are high. Logic gates manage state transitions and overflow handling, ensuring accurate and stable counting operation.

UP = 1 (UP COUNTER)

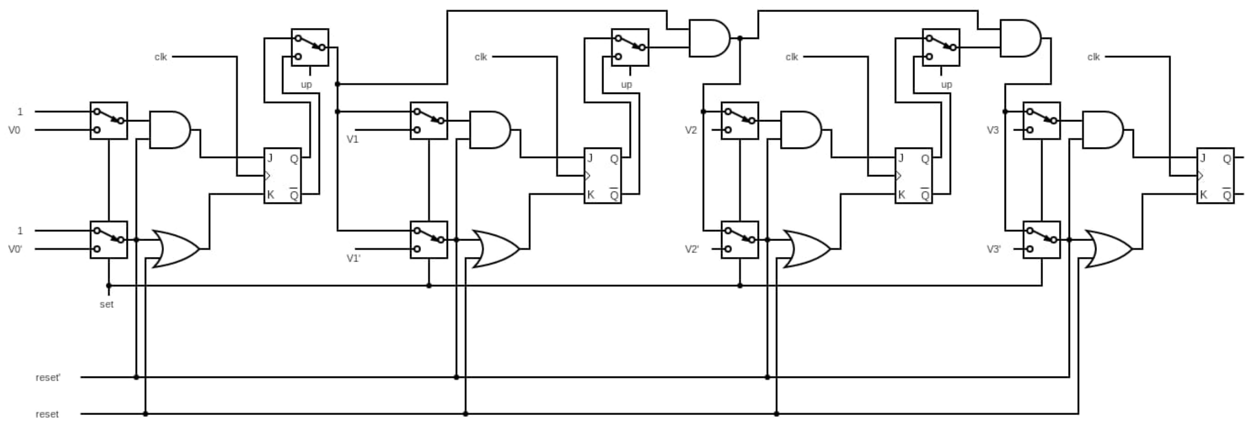
Q_3	Q_2	Q_1	Q_0	UP	Q_3+	Q_2+	Q_1+	Q_0+	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	1	0	0	0	1	0	x	0	x	0	x	1	x
1	1	1	1	1	0	0	1	0	0	x	0	x	1	x	x	1
1	1	1	0	1	0	0	1	1	0	x	0	x	x	0	1	x
1	1	0	1	1	0	1	0	0	0	x	1	x	x	1	x	1
1	1	0	0	1	0	1	0	1	0	x	x	0	0	x	1	x
1	0	1	1	1	0	1	1	0	0	x	x	0	1	x	x	1
1	0	1	0	1	0	1	1	1	0	x	x	0	x	0	1	x
1	0	0	1	1	1	0	0	0	1	x	x	1	x	1	x	1
1	0	0	0	1	1	0	0	1	x	0	0	x	0	x	1	x
0	1	1	1	1	1	0	1	0	x	0	0	x	1	x	x	1
0	1	1	0	1	1	0	1	1	x	0	0	x	x	0	1	x
0	1	0	1	1	1	1	0	0	x	0	1	x	x	1	x	1
0	1	0	0	1	1	1	0	1	x	0	x	0	0	x	1	x
0	0	1	1	1	1	1	1	0	x	0	x	0	1	x	x	1
0	0	1	0	1	1	1	1	1	x	0	x	0	x	0	1	x
0	0	0	1	1	0	0	0	0	x	1	x	1	x	1	x	1

5 Combinational Logic Expressions

We implemented the derived logic expressions using JK flip-flops. We connected the set and reset inputs to the appropriate JK inputs of the flip-flops, ensuring proper synchronization with the clock signal.

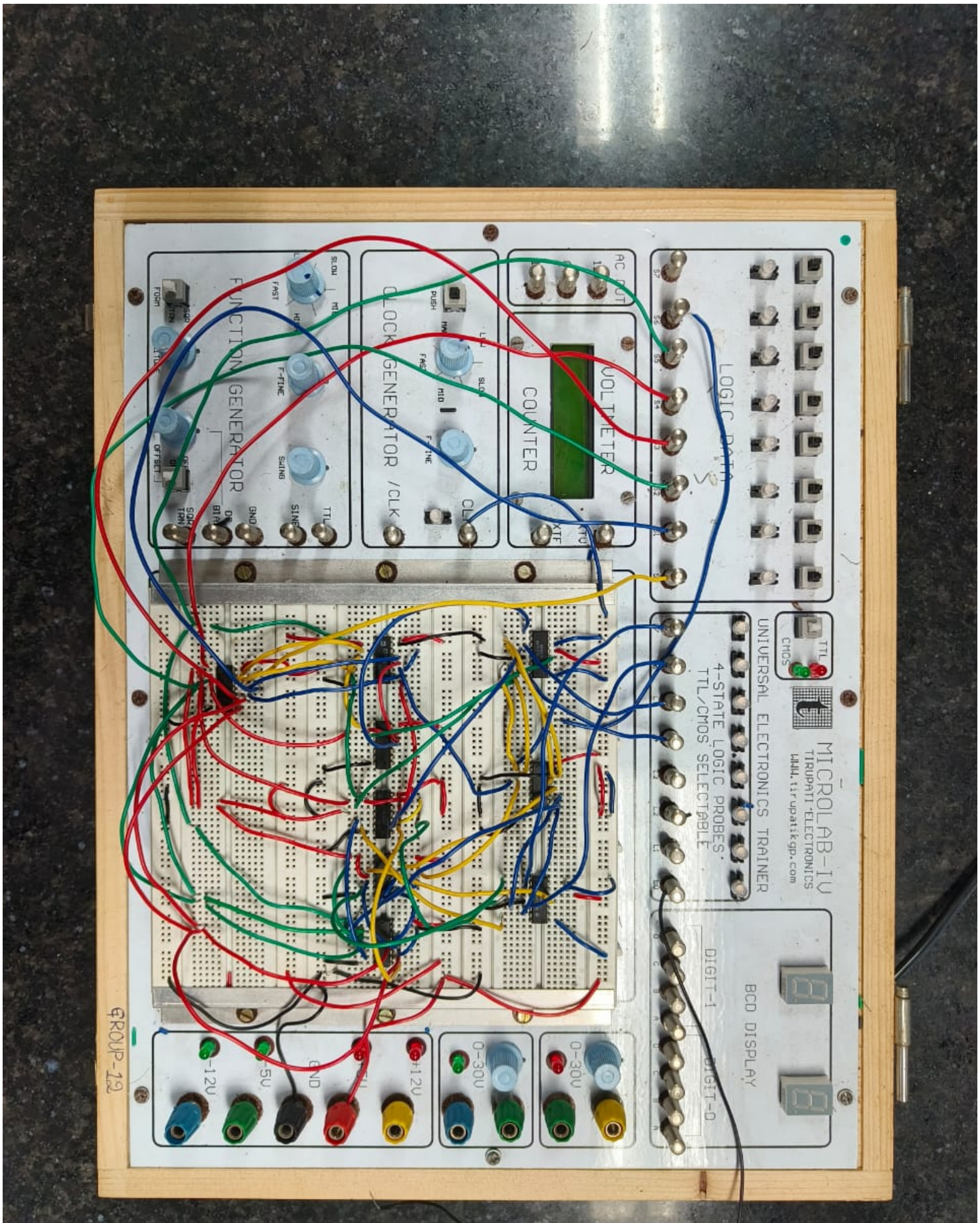
$$\begin{aligned}
J_0 &= 1 \\
J_1 &= Q_0(\text{up}) + Q_0'(\text{up})' \\
J_2 &= Q_1Q_0(\text{up}) + Q_1'Q_0'(\text{up})' \\
J_3 &= Q_2Q_1Q_0(\text{up}) + Q_2'Q_1'Q_0'(\text{up})' \\
K_0 &= 1 \\
K_1 &= Q_0(\text{up}) + Q_0'(\text{up})' \\
K_2 &= Q_1Q_0(\text{up}) + Q_1'Q_0'(\text{up})' \\
K_3 &= Q_2Q_1Q_0(\text{up}) + Q_2'Q_1'Q_0'(\text{up})'
\end{aligned}$$

6 Circuit Diagram



7 Circuit Photo

The following image shows the physical implementation of the designed circuit on a breadboard. The wiring connections, logic components, and display units are arranged as per the circuit schematic.


MICROLAB-IV
TIRUPATI ELECTRONICS
www.tirupatikgp.com
UNIVERSAL ELECTRONICS TRAINER
4-STATE LOGIC PROBES
TTL/CMOS SELECTABLE
LOGIC DATA
VOLTMETER
COUNTER
CLOCK GENERATOR /CLK
FUNCTION GENERATOR
BCD DISPLAY
DIGIT-1
DIGIT-0
AC OUT
-12V
-5V
GND
+12V
0-30V
0-30V
GROUP-12

8 Verilog code

8.1 Multiplexer (MUX)

- Implements a 2:1 multiplexer.
- Selects between two inputs ('a' and 'b') based on a 'select' signal.
- If 'select = 0', the output follows 'a'; otherwise, it follows 'b'.
- Used to control the selection of values in the counter logic.

8.2 JK Flip-Flop

- Implements a JK flip-flop with a positive edge-triggered clock.
- The output toggles when both inputs ('J' and 'K') are '1'.
- If 'J=1, K=0', it sets 'Q=1'; if 'J=0, K=1', it resets 'Q=0'.
- Provides both 'Q' and 'Q' outputs.
- Forms the core storage element in the counter.

8.3 Up/Down Counter (4-bit)

- A 4-bit synchronous up/down counter.
- Uses multiplexers and JK flip-flops for state transitions.
- Inputs:
  - 'clk': Clock signal to control counting.
  - 'reset': Resets the counter to '0'.
  - 'set': Loads a specified value into the counter.
  - 'up': Determines counting direction ('1' for up, '0' for down).
  - 'value[3:0]': The preset value for initialization.
- Outputs:
  - 'counter[3:0]': 4-bit counter output.
- Working Mechanism:
  - Multiplexers select between different input values ('set', 'up', 'reset').

  - JK flip-flops toggle based on control signals to maintain counter states.
  - Implements synchronous logic to ensure all flip-flops update simultaneously.

9 Test Bench and Simulation

9.1 Overview

- The testbench verifies the functionality of a 4-bit up/down counter.
- It simulates various operations, including counting up, counting down, resetting, and loading a preset value.

9.2 Signal Declarations

- 'clk': Clock signal toggling every 5 time units.
- 'reset': Active-high reset to initialize the counter to 0.
- 'set': Loads a specified 4-bit value into the counter.
- 'up': Determines counting direction ('1' for up, '0' for down).
- 'value[3:0]': The preset value loaded into the counter when 'set' is high.
- 'counter[3:0]': The 4-bit output of the counter.

9.3 Clock Generation

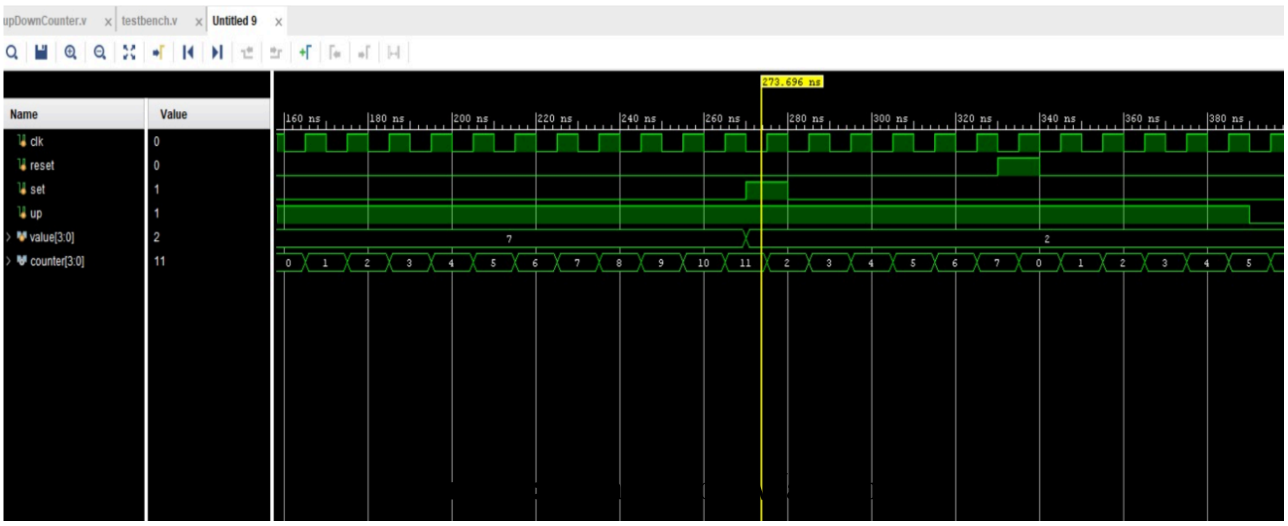
- The testbench generates a clock signal using 'always 5 clk = ˜clk;'.
- This creates a periodic clock pulse that drives the counter operation.

9.4 Test Sequence

- Initialization:
  - The clock starts at 0, reset is activated, and 'set' and 'up' are initialized.
- Resetting the Counter:
  - The reset signal is set high for 10 time units and then deactivated.

- Loading a Preset Value:
  - The testbench sets 'set = 1' and assigns a 4-bit value (e.g., '0111').
  - After 10 time units, 'set' is deactivated.
- Counting Operations:
  - The counter runs normally for 200 time units.
  - Another preset value (e.g., '0010') is loaded into the counter.
- Switching Counting Direction:
  - The 'up' signal is toggled to switch from up-counting to down-counting.
- Simulation End:
  - After completing all tests, the simulation terminates with '*finish*'.

9.5 Simulation Results


upDownCounter.v
testbench.v
Untitled 9
273.696 ns
Name
Value
clk 0
reset 0
set 1
up 1
value[3:0] 2
counter[3:0] 11
160 ns
180 ns
200 ns
220 ns
240 ns
260 ns
280 ns
300 ns
320 ns
340 ns
360 ns
380 ns
7
2
0 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 0 1 2 3 4 5