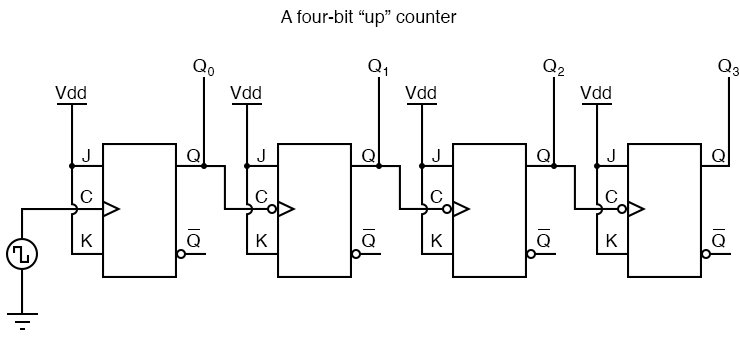
**LAB EXPERIMENT 8**

**Aim:** To study the outputs of different counters in electronics. Design the up counter, down counter and up-down counter.

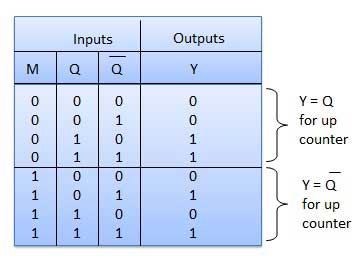
**Theory:**

1. Up Counter: The Up Counter models a generic up counter with between 2 and 32 output bits. On each clock edge, the D0 ... Dn outputs can increase one count or hold the last count, depending on the EN pin. The clock-edge trigger can be set with the Trigger Condition parameter to be either rising edge (0\_TO\_1) or falling edge (1\_TO\_0).The counter may also be reset to a value with the RST pin. When the RST pin goes active, the counter changes to the value specified by the Reset parameter. The Reset Type parameter controls whether the reset event is asynchronous or synchronous to the CLK edge.

Circuit Diagram:

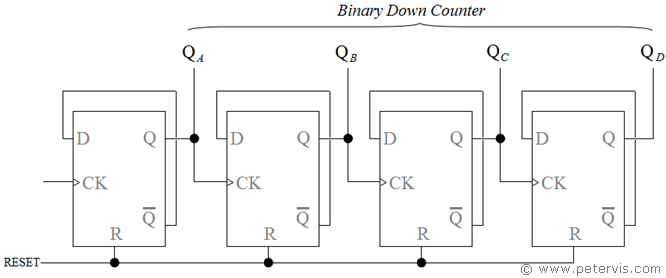


Truth Table:

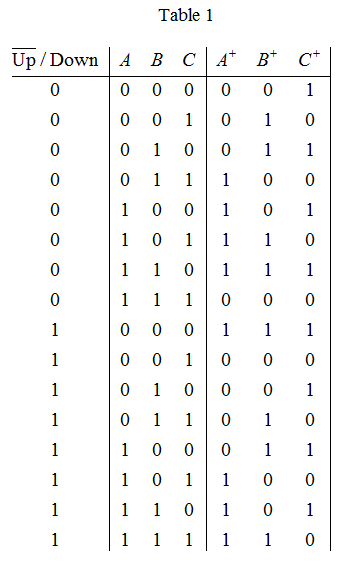


1. Down Counter: The Down Counter models a generic down counter with between 2 and 32 output bits.On each clock edge, the D0 ... Dn outputs can decrease one count or hold the last count, depending on the EN pin.The clock-edge trigger can be set with the Trigger Condition parameter to be either rising edge (0\_TO\_1) or falling edge (1\_TO\_0).The counter may also be set to a value with the SET pin. When the Set pin goes active, the counter changes to the value specified by the Set  parameter. The  Set Type parameter controls whether the set event is asynchronous or synchronous to the CLK edge.

Circuit Diagram:

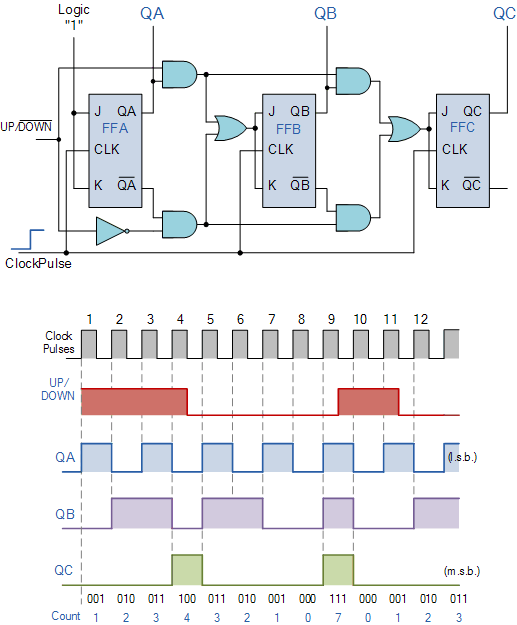


Truth Table:



1. UP/DOWN Counter: Up/Down counters, are capable of counting in either direction through any given count sequence and they can be reversed at any point within their count sequence by using an additional control input.

Circuit Diagram and Outputs:



**Verilog Code of the Program and screenshots:**

1. **Up Counter**
2. **Verilog Code of the Program:**

module UpCounter\_Rahil\_062(

    input clk,

    input rst,

    output reg[3:0] counter

    );

always@(posedge clk or posedge rst)

begin

if(rst)

counter<=4'b0;

else

counter <= counter+1'b1;

end

endmodule

1. **Screenshot of the Program and Outputs:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

1. **RTL Schematic:**

A picture containing text, indoor, wall, screenshot

Description automatically generated

A picture containing text, indoor, screenshot

Description automatically generated

* 1. **Down Counter:**
     1. **Verilog code of the Program:**

module DownCounter\_Rahil\_062(

    input clk,

    input reset,

    output reg[3:0] count

    );

always@(posedge clk or posedge reset)

begin

if(reset)

count<=4'b0;

else

count<=count-1'b1;

end

endmodule

* + 1. **Screenshots of the Program and Outputs:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

* + 1. **RTL Schematics:**

A picture containing text, indoor, screenshot, display

Description automatically generated

A picture containing text, indoor, screenshot

Description automatically generated

* 1. **Up Down Counter**
     1. **Verilog Code of the Program:**

module UpDown\_Rahil\_062(

    input clk,

    input reset,

    input up\_down,

    output reg[3:0] count

    );

always@(posedge clk or posedge reset)

begin

if(reset)

count<=4'b0;

else if(up\_down)

count<=count+1'b1;

else

count<=count-1'b1;

end

endmodule

* + 1. **Screenshot of Program Code and Output**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* + 1. **RTL Scehmatic:**

A picture containing text, indoor, screenshot

Description automatically generated

A picture containing text, indoor, screenshot

Description automatically generated