END SEMESTER ASSESSMENT (ESA)
B.TECH. (CSE)
III SEMESTER

UE18CS206 – DIGITAL DESIGN & COMPUTER ORGANIZATION LABORATORY

PROJECT REPORT

ON

"4 BIT UP DOWN COUNTER USING JK FLIP FLOPS"

SUBMITTED BY

NAME	SRN
Student1:SUJAL S	PES2UG21CS548
Student2:SUDHANVA SAMAGA	PES2UG21CS544
Student3:SRINIVAASAN N S	PES2UG21CS542
Student4:SHREY	PES2UG21CS500

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

ABSTRACT OF THE PROJECT:

This project takes us on a tour in which we design a 4bit
Synchronous up down counter using Verilog.

The counter is implemented by the use of 4 jk flip flops since it is a 4 bit counter.
Jk flip flops are different from SR flip flops in the way the they toggle while inputs are both 1.

Since it is a synchronous counter there is a single clock to all the 4 flip flops.We have implemented a switch which allows us to toggle between the counters(up and down).

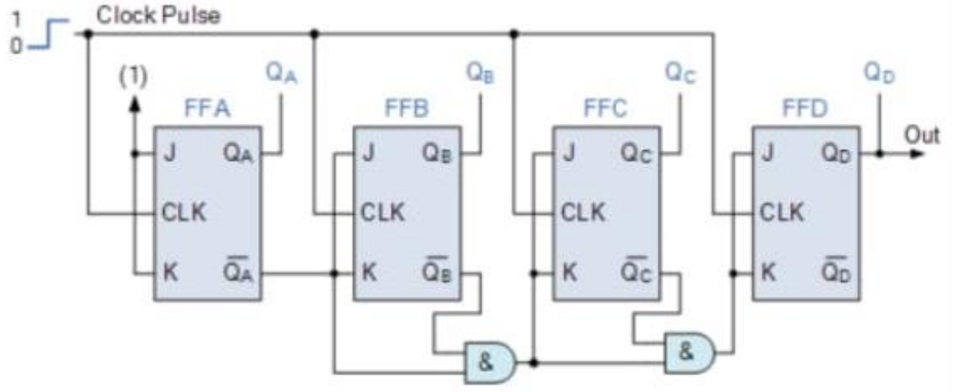
The input being x,output is incremented by 1 in the up counter and is decremented by 1 in the down counter.

The Gtkwaveform changes with respect to clock input every 5 units of time.

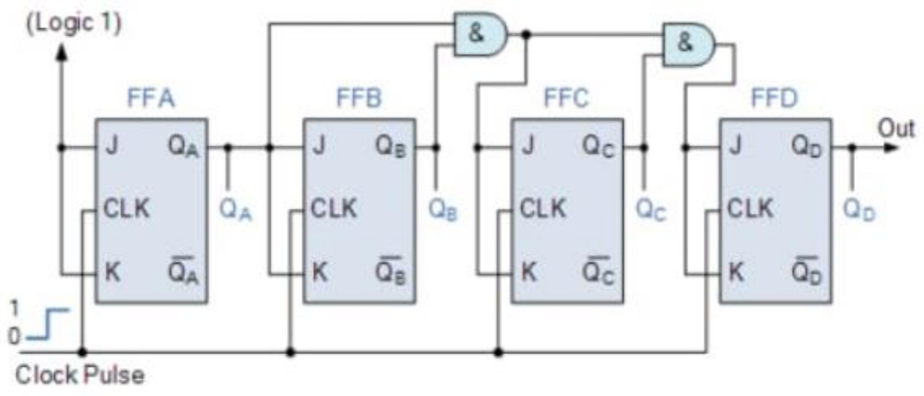
The Output in the up counter is Q whereas in the down counter it is the negation which is Qbar.

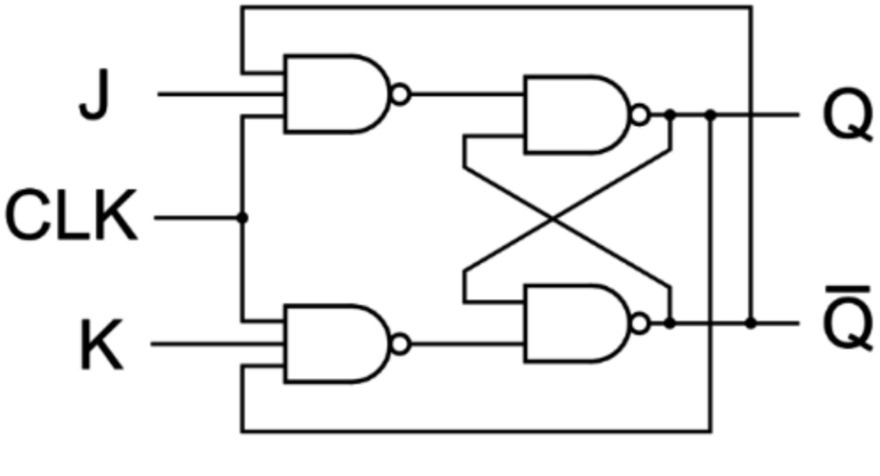
CIRCUIT DIAGRAM:

Binary 4-bit Synchronous Down Counter



Binary 4-bit Synchronous Up Counter





MAIN VERILOG CODE:

```
`timescale 1ns / 1ps

module JKFlipFlop(Q,Q_Bar,J,K,Clock,Reset);       // Implementation Of JK Flipflop ..

input J,K;
input Clock,Reset;
output reg Q;
output reg Q_Bar;

always @ (posedge Clock)
begin

  if (Reset == 1'b1)
  begin
     Q <= 1'b0;
  end

  else
  begin
    case({J,K})
      2'b00 :  Q <= Q;                    // Both J And K == 0 then Memory state..
      2'b01 :  Q <= 1'b0;                  // If J = 0; K = 1 then reset state
      2'b10 :  Q <= 1'b1;                  // If J = 1; K = 0 then Set state
      2'b11 :  Q <= ~Q;                    // Both J And K == 1 then Toggle state..
    endcase
  end

  Q_Bar <= ~(Q);
```

```
end
endmodule

module UpDownCounter(Switch,Q_out,Q_barout,J,K,Clock,Reset);

input wire Switch;
input wire [3:0]J;
input wire [3:0]K;
input wire Clock;
input wire Reset;

output wire [3:0]Q_out;
output wire [3:0]Q_barout;


if(Switch <= 1'b1)          //Upcounter
begin
  assign J[0] = 1'b1;
  assign K[0] = 1'b1;

  assign J[1] = Q_out[0];
  assign K[1] = Q_out[0];

  assign J[2] = Q_out[0] & Q_out[1];
  assign K[2] = Q_out[0] & Q_out[1];

  assign J[3] = Q_out[0] & Q_out[1] & Q_out[2];
  assign K[3] = Q_out[0] & Q_out[1] & Q_out[2];

  JKFlipFlop index0(Q_out[0],Q_barout[0],J[0], K[0], Clock,Reset); //LSB
  JKFlipFlop index1(Q_out[1],Q_barout[1],J[1], K[1], Clock,Reset);
  JKFlipFlop index2(Q_out[2],Q_barout[2],J[2], K[2], Clock,Reset);
  JKFlipFlop index3(Q_out[3],Q_barout[3],J[3], K[3], Clock,Reset); //MSB
end

else    // Down Counter
begin
  assign J[0] = 1'b1;
  assign K[0] = 1'b1;

  assign J[1] = Q_barout[0];
  assign K[1] = Q_barout[0];

  assign J[2] = Q_barout[0] & Q_barout[1];
  assign K[2] = Q_barout[0] & Q_barout[1];

  assign J[3] = Q_barout[0] & Q_barout[1] & Q_barout[2];
  assign K[3] = Q_barout[0] & Q_barout[1] & Q_barout[2];

  JKFlipFlop index5(Q_out[0],Q_barout[0],J[0] ,K[0], Clock,Reset);
  JKFlipFlop index6(Q_out[1],Q_barout[1],J[1], K[1], Clock,Reset);
  JKFlipFlop index7(Q_out[2],Q_barout[2],J[2], K[2], Clock,Reset);
  JKFlipFlop index8(Q_out[3],Q_barout[3],J[3], K[3], Clock,Reset);
end
endmodule
```

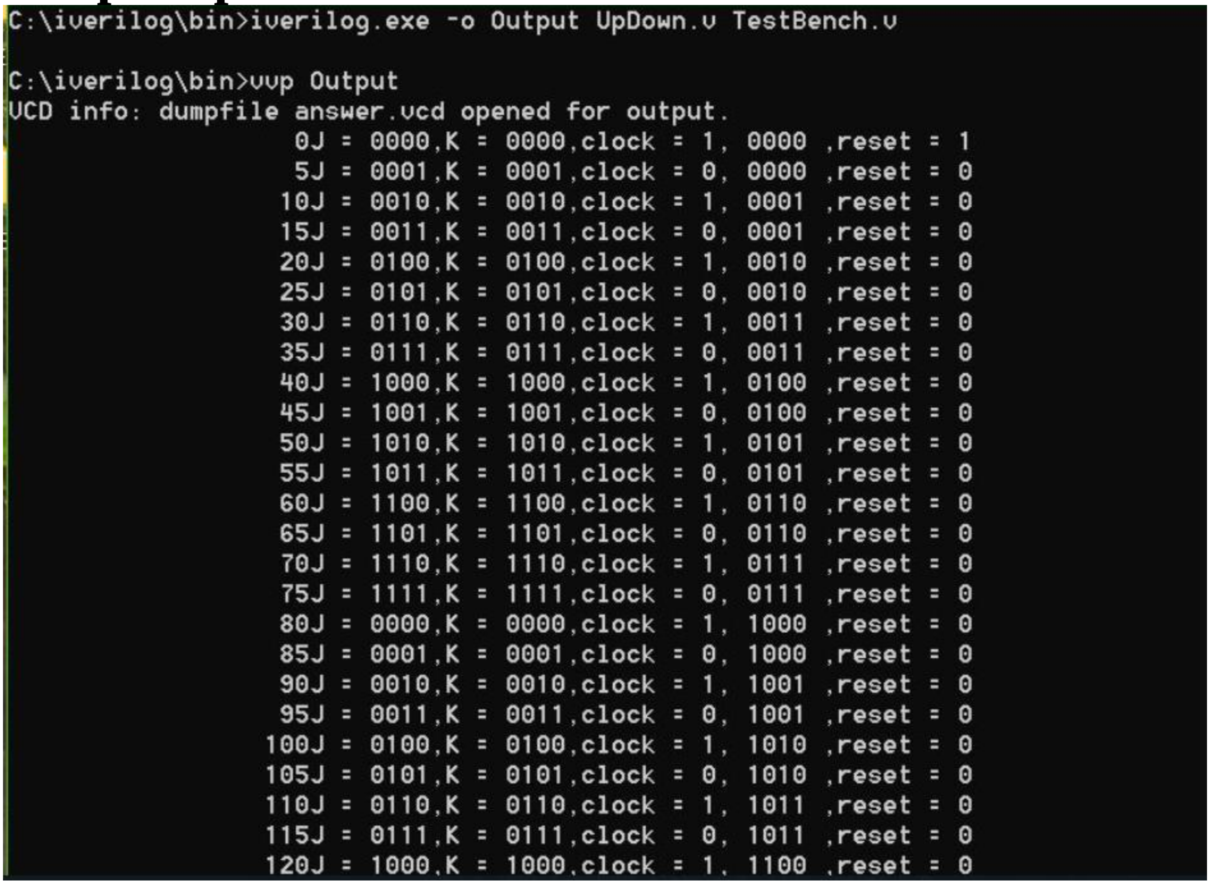
TEST BENCH FILE:

```
timescale 1ns / 1ps

module tb;

reg  clock,reset;                 // reg == input
reg  [3:0]J;
reg  [3:0]K;

reg Switch;

wire [3:0]out;                    // wire == output
wire [3:0]outBAR;

UpDownCounter answer(Switch,out,outBAR,J,K,clock,reset);

always @ (*)
begin

#5
   clock <= ~clock;
   J <=  J + 1'b1;
   K <=  K + 1'b1;

end

initial
begin

   Switch = 1'b1;            // indicating to do Upcounter
reset = 1'b1;
   clock = 1'b1;
   J = 4'b0000;
   K = 4'b0000;

#5 reset = 1'b0;

end

initial
begin

   Switch = 1'b0;         // indicating to do Downcounter
   reset = 1'b1;
   clock = 1'b1;
   J = 4'b0000;
   K = 4'b0000;
```

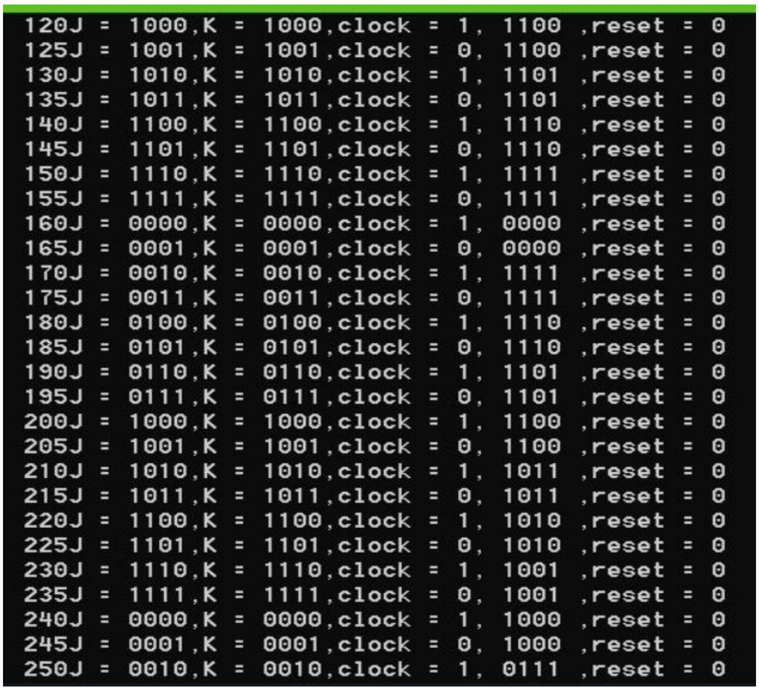
```
#170 reset = 1'b0;

end

initial
begin
   $monitor($time,"J = %b,K = %b,clock = %b, %b%b%b%b ,reset =
%b ",J,K,clock,out[3],out[2],out[1],out[0],reset);
   #160;
            $monitor($time,"J = %b,K = %b,clock = %b, %b%b%b%b
,reset = %b
",J,K,clock,outBAR[3],outBAR[2],outBAR[1],outBAR[0],reset);
            #160 $finish;
end


initial
begin
$dumpfile ("answer.vcd");
 $dumpvars (0, tb);
end


endmodule
```

SCREEN SHOT OF THE OUTPUT:

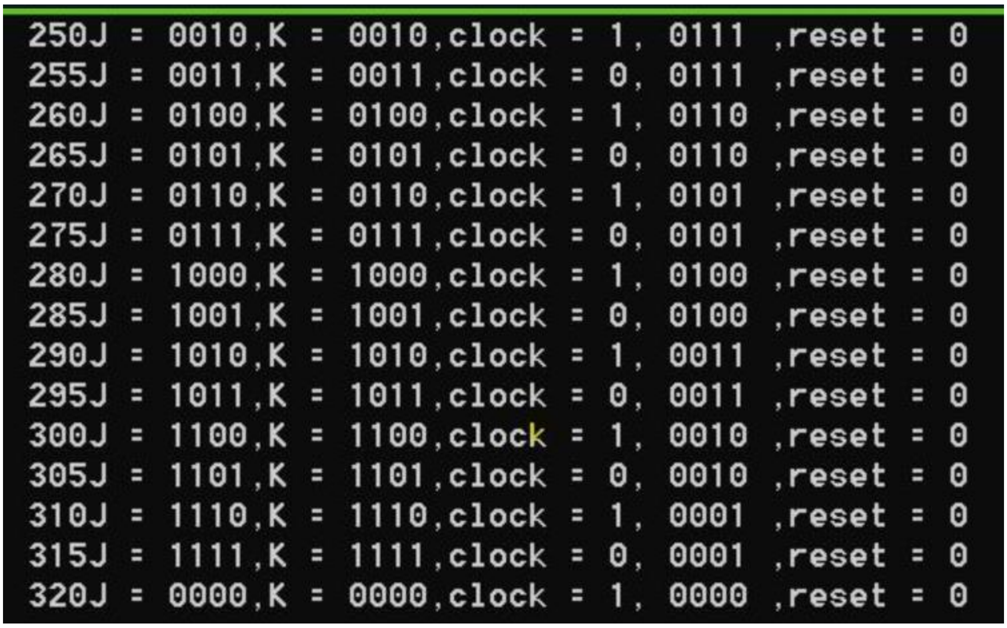
1.vvp output



```
C:\iverilog\bin>iverilog.exe -o Output UpDown.v TestBench.v

C:\iverilog\bin>vvp Output
VCD info: dumpfile answer.vcd opened for output.
                   0J = 0000,K = 0000,clock = 1, 0000 ,reset = 1
                   5J = 0001,K = 0001,clock = 0, 0000 ,reset = 0
                  10J = 0010,K = 0010,clock = 1, 0001 ,reset = 0
                  15J = 0011,K = 0011,clock = 0, 0001 ,reset = 0
                  20J = 0100,K = 0100,clock = 1, 0010 ,reset = 0
                  25J = 0101,K = 0101,clock = 0, 0010 ,reset = 0
                  30J = 0110,K = 0110,clock = 1, 0011 ,reset = 0
                  35J = 0111,K = 0111,clock = 0, 0011 ,reset = 0
                  40J = 1000,K = 1000,clock = 1, 0100 ,reset = 0
                  45J = 1001,K = 1001,clock = 0, 0100 ,reset = 0
                  50J = 1010,K = 1010,clock = 1, 0101 ,reset = 0
                  55J = 1011,K = 1011,clock = 0, 0101 ,reset = 0
                  60J = 1100,K = 1100,clock = 1, 0110 ,reset = 0
                  65J = 1101,K = 1101,clock = 0, 0110 ,reset = 0
                  70J = 1110,K = 1110,clock = 1, 0111 ,reset = 0
                  75J = 1111,K = 1111,clock = 0, 0111 ,reset = 0
                  80J = 0000,K = 0000,clock = 1, 1000 ,reset = 0
                  85J = 0001,K = 0001,clock = 0, 1000 ,reset = 0
                  90J = 0010,K = 0010,clock = 1, 1001 ,reset = 0
                  95J = 0011,K = 0011,clock = 0, 1001 ,reset = 0
                 100J = 0100,K = 0100,clock = 1, 1010 ,reset = 0
                 105J = 0101,K = 0101,clock = 0, 1010 ,reset = 0
                 110J = 0110,K = 0110,clock = 1, 1011 ,reset = 0
                 115J = 0111,K = 0111,clock = 0, 1011 ,reset = 0
                 120J = 1000,K = 1000,clock = 1, 1100 ,reset = 0
```



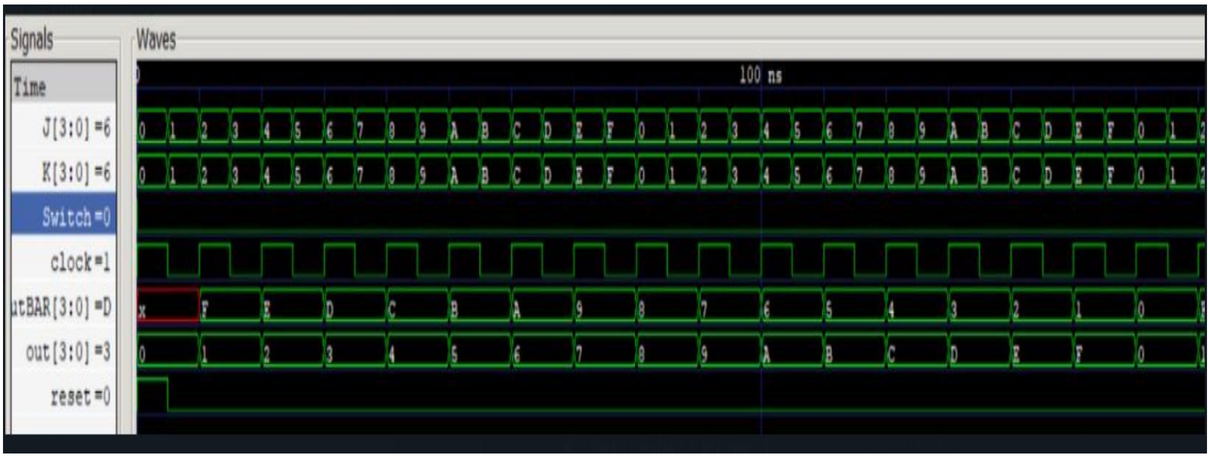
```
 120J = 1000,K = 1000,clock = 1, 1100 ,reset = 0
 125J = 1001,K = 1001,clock = 0, 1100 ,reset = 0
 130J = 1010,K = 1010,clock = 1, 1101 ,reset = 0
 135J = 1011,K = 1011,clock = 0, 1101 ,reset = 0
 140J = 1100,K = 1100,clock = 1, 1110 ,reset = 0
 145J = 1101,K = 1101,clock = 0, 1110 ,reset = 0
 150J = 1110,K = 1110,clock = 1, 1111 ,reset = 0
 155J = 1111,K = 1111,clock = 0, 1111 ,reset = 0
 160J = 0000,K = 0000,clock = 1, 0000 ,reset = 0
 165J = 0001,K = 0001,clock = 0, 0000 ,reset = 0
 170J = 0010,K = 0010,clock = 1, 1111 ,reset = 0
 175J = 0011,K = 0011,clock = 0, 1111 ,reset = 0
 180J = 0100,K = 0100,clock = 1, 1110 ,reset = 0
 185J = 0101,K = 0101,clock = 0, 1110 ,reset = 0
 190J = 0110,K = 0110,clock = 1, 1101 ,reset = 0
 195J = 0111,K = 0111,clock = 0, 1101 ,reset = 0
 200J = 1000,K = 1000,clock = 1, 1100 ,reset = 0
 205J = 1001,K = 1001,clock = 0, 1100 ,reset = 0
 210J = 1010,K = 1010,clock = 1, 1011 ,reset = 0
 215J = 1011,K = 1011,clock = 0, 1011 ,reset = 0
 220J = 1100,K = 1100,clock = 1, 1010 ,reset = 0
 225J = 1101,K = 1101,clock = 0, 1010 ,reset = 0
 230J = 1110,K = 1110,clock = 1, 1001 ,reset = 0
 235J = 1111,K = 1111,clock = 0, 1001 ,reset = 0
 240J = 0000,K = 0000,clock = 1, 1000 ,reset = 0
 245J = 0001,K = 0001,clock = 0, 1000 ,reset = 0
 250J = 0010,K = 0010,clock = 1, 0111 ,reset = 0
```



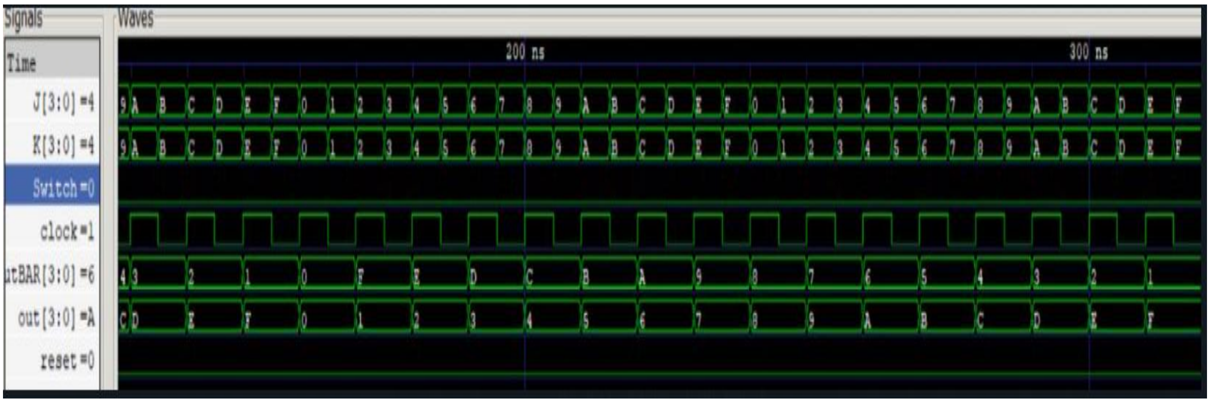
```
250J = 0010,K = 0010,clock = 1, 0111 ,reset = 0
255J = 0011,K = 0011,clock = 0, 0111 ,reset = 0
260J = 0100,K = 0100,clock = 1, 0110 ,reset = 0
265J = 0101,K = 0101,clock = 0, 0110 ,reset = 0
270J = 0110,K = 0110,clock = 1, 0101 ,reset = 0
275J = 0111,K = 0111,clock = 0, 0101 ,reset = 0
280J = 1000,K = 1000,clock = 1, 0100 ,reset = 0
285J = 1001,K = 1001,clock = 0, 0100 ,reset = 0
290J = 1010,K = 1010,clock = 1, 0011 ,reset = 0
295J = 1011,K = 1011,clock = 0, 0011 ,reset = 0
300J = 1100,K = 1100,clock = 1, 0010 ,reset = 0
305J = 1101,K = 1101,clock = 0, 0010 ,reset = 0
310J = 1110,K = 1110,clock = 1, 0001 ,reset = 0
315J = 1111,K = 1111,clock = 0, 0001 ,reset = 0
320J = 0000,K = 0000,clock = 1, 0000 ,reset = 0
```

2.Gtkwave outputs

1.up counter=out



2.Down counter=outBAR


Signals
Waves
Time
200 ns
300 ns
J[3:0] =4
K[3:0] =4
Switch =0
clock =1
utBAR[3:0] =6
out[3:0] =A
reset =0