KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of the UGC Act, 1956

VLSI Design Lab
Spring 2020-2021

Debagnik Kar
Roll no.: 1804373, Section: ETC-06
Experiment: 02

Experiment No-2

Aim : (i) Introduction to behavioral modeling ;
(ii) Design of up counter and down counter (Synchronous and asynchronous), also design synchronous up-down counter.

Software used :- www.edaplayground.com

Theory :-

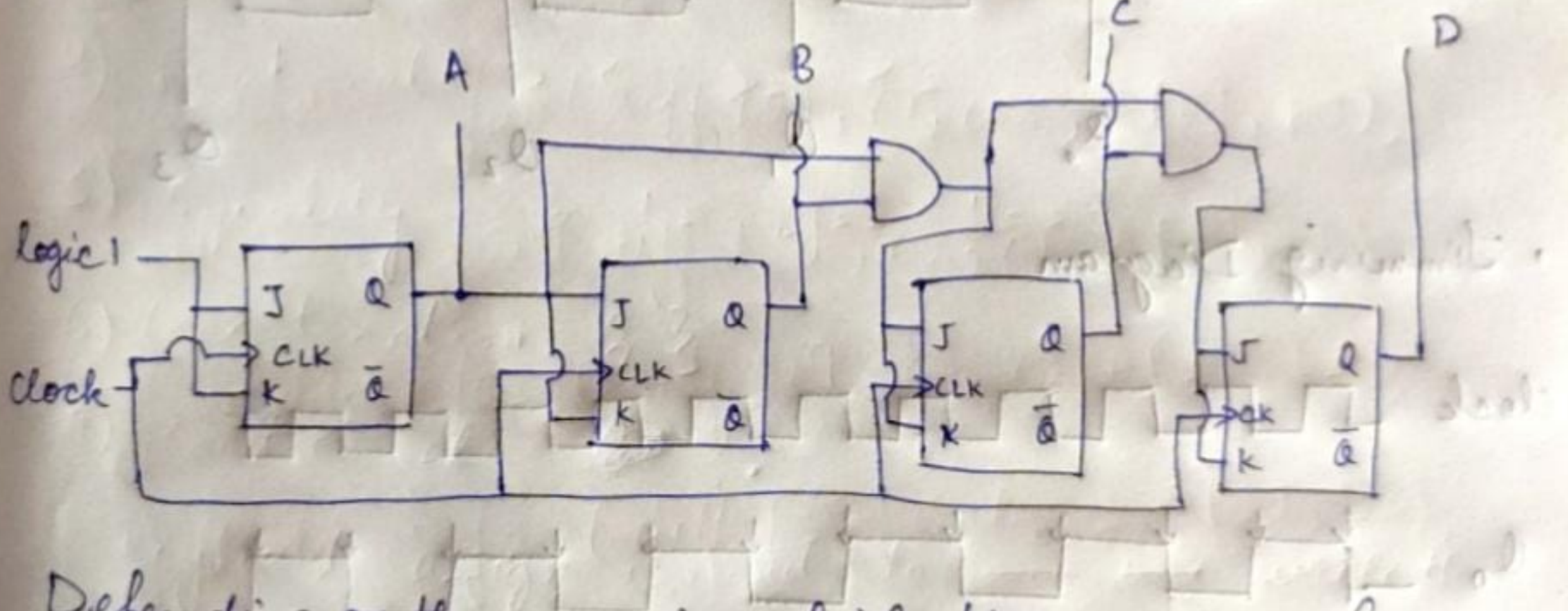
- Behavioral Modeling - It is used to describe complex circuits. In VHDL, behavioral modeling is done in the architecture block. Within the architecture block, processes are defined to model sequential circuits. Assignment statements are used. One needs to describe the value of outputs for various combinations of inputs. Thus, Behavioral modeling attempts to explain a decision and the model is then used to help predict future behavior.

Debagnik Kar,
1804373

- Counters - Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied. Counters are of two types.

(1) Synchronous counters

If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as Synchronous counter.

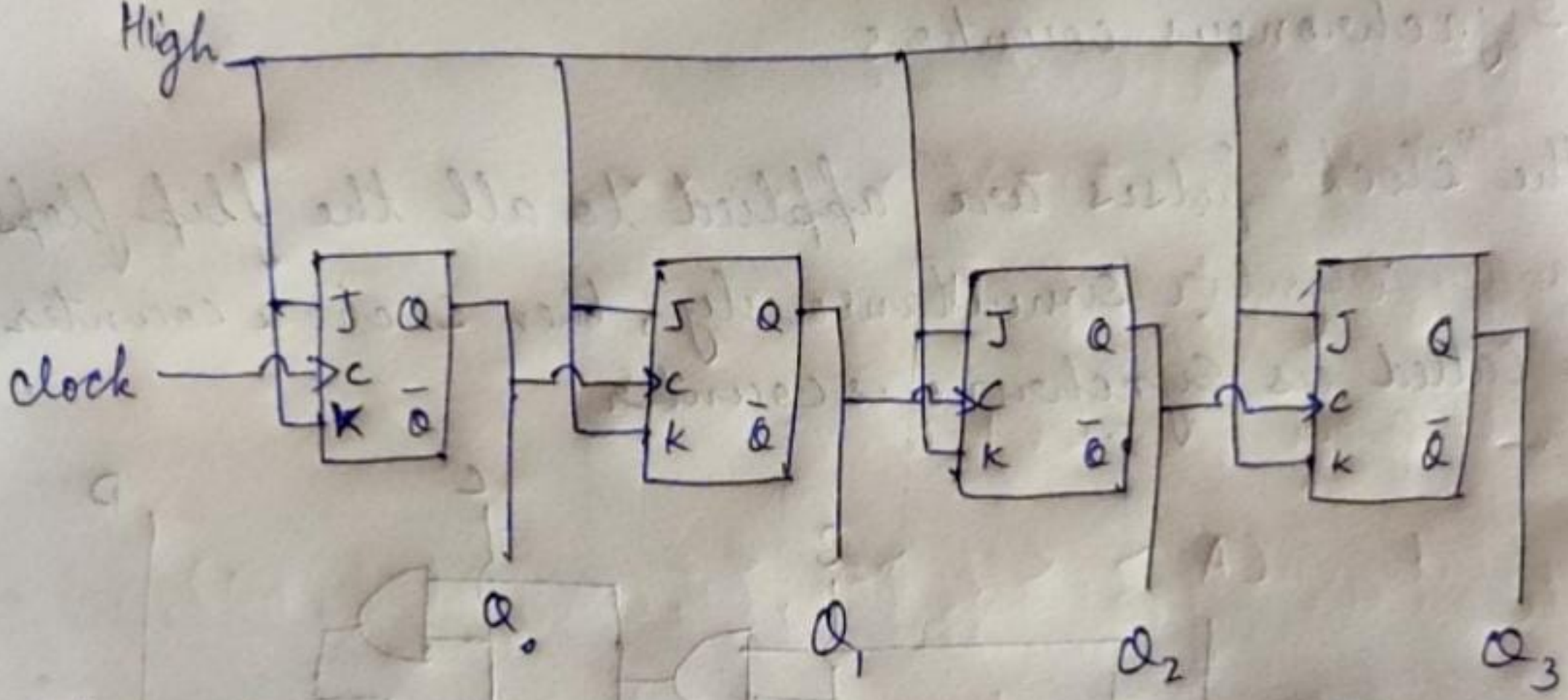


Depending on the way in which the counting progresses, the syncheronous or ~~asy~~ or asynchronous counters are classified as follows:-

1) Up counters
2) Down counters
3) Up/Down counters.

Debagnik Kar
1804373

(2) Asynchronous counters - In asynchronous counter we don't use universal clock, only first flip-flop is driven by main clock and the clock input of rest of the following flipflop is driven by output of previous flip flops.



• Timing Diagram

Debagnik Kar
18043773

Code:

UP Counter (Synchronous)

Design.sv

```
// Code your design here
module upcount(output[3:0]count,input clock,reset);
  reg [3:0]count;
  always @ (posedge clock) begin
    if (reset == 1)
      count = 0;
    else
      count = count + 1;
  end
endmodule
```

Testbench.sv

```
// Code your testbench here
// or browse Examples
module test_upcount;
  reg clock,reset;
  wire [3:0] count;
  upcount u1 (count, clock , reset);
  always #5 clock = ! clock;
  initial begin
     $dumpfile("dump.vcd"); $dumpvars;
    clock =0; reset= 1;
    #10 reset = 0;
  end
endmodule
```

Down Counter (Synchronous)

Design.sv

```
// Code your design here
module downcount(output[3:0]count,input clock,reset);
  reg [3:0]count;
  always @ (posedge clock) begin
    if (reset == 1)
      count = 0;
    else
      count = count - 1;
  end
endmodule
```

Testbench.sv

```
// Code your testbench here
// or browse Examples
module test_downcount;
  reg clock,reset;
  wire [3:0] count;
  downcount u1 (count, clock , reset);
  always #5 clock = ! clock;
  initial begin
     $dumpfile("dump.vcd");
     $dumpvars;
    clock =0; reset= 1;
    #10 reset = 0;
  end
endmodule
```

Up-Down Counter (Synchronous)

Design.sv

```
// Code your design here
module updown(output[3:0]count,input clock,reset,updn);
  reg [3:0]count;
  always @ (posedge clock) begin
    if (reset == 1)
      count = 0;
    else if(updn == 1)
      count = count + 1;
    else if(updn == 0)
      count = count - 1;
    else
      count=15;
  end
endmodule
```

Testbench.sv

```
// Code your testbench here
// or browse Examples
module test_updown;
  reg clock,reset, updn;
  wire [3:0] count;
  updown ud1 (count, clock , reset, updn);
  always #5 clock = ! clock;
  initial begin
     $dumpfile("dump.vcd"); $dumpvars;
    clock =0;updn = 1; reset= 1;
    #10 reset = 0;
    #210 updn = 0;
  end
endmodule
```

Observations:

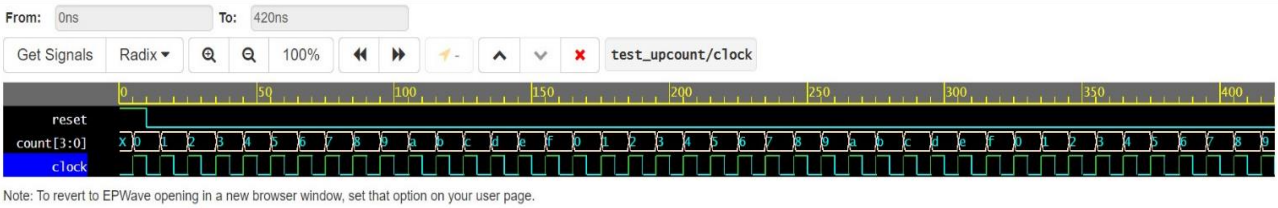


Fig 1: Up counter Waveform

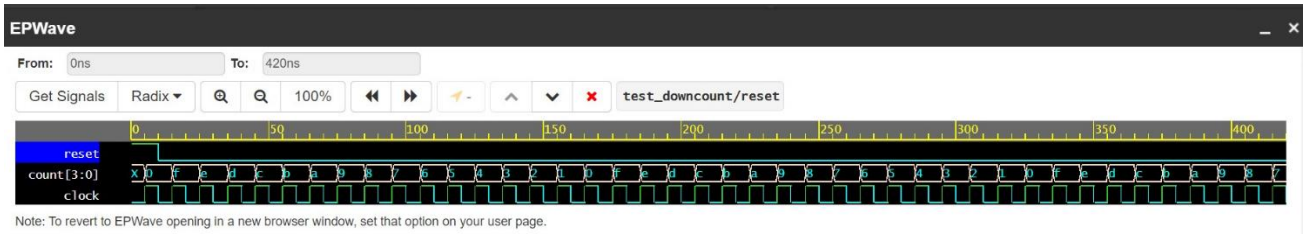


Fig 2: Down counter Waveform

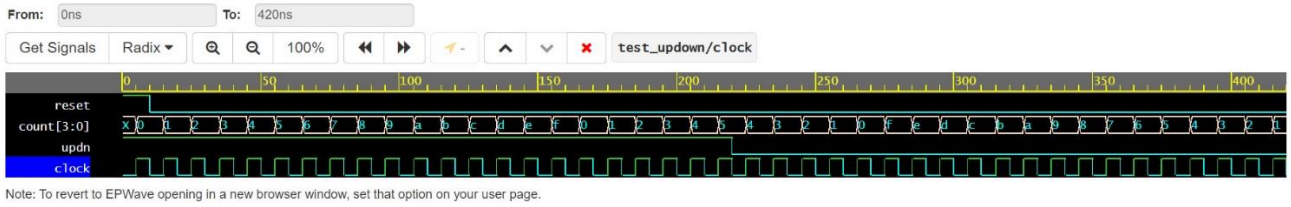


Fig 3: Up-Down Counter Waveform

Conclusion :- In this experiment we learned about Behavioral Modeling and we successfully designed Sychronous & Asynchronous up & down counters & also we designed synchronous up-down counter using edaplayground. We wrote their code, their respective test bench codes & synthesised their output waveforms that was same as we had learned in theory.

Debagnik Kar
1804373