Republic of the Philippines
BATANGAS STATE UNIVERSITY
The National Engineering University
Alangilan Campus
Golden Country Homes, Alangilan, Batangas City, Batangas, Philippines 4200
Tel Nos.: (+63 43) 425-0139 loc. 2121 / 2221
E-mail Address: ceafa@g.batstate-u.edu.ph | Website Address: http://www.batstate-u.edu.ph

College of Engineering – Department of Electrical Engineering

COMPUTER ENGINEERING PROGRAM
INTRODUCTION TO HDL
LABORATORY REPORT 5
COUNTERS

Name: Karylle Faye P. Capio **Section:** CpE - 3101
Sr-Code: 23-01342

1. **Activity's Intended Learning Outcomes (ILOs)**
   At the end of this activity, the student shall be able to:
   a. **Implement and simulate** a 3-bit UP asynchronous counter and a 3-bit DOWN synchronous counter in Verilog to observe their counting sequences under clock pulses.
   b. **Analyze and verify** counter behavior using output logs and timing diagrams, highlighting differences between asynchronous and synchronous operation.
2. **Objectives of the Activity**
   The activity aims to:
   a. To reinforce understanding of **sequential circuits** and how counters are realized using flip-flops.
   b. To practice **Verilog coding** for both asynchronous and synchronous counter designs.
   c. To use a **testbench and simulation tools** to observe counting sequences and wrap-around behavior.
   d. To analyze differences in timing between asynchronous and synchronous counters.

Sample Verilog Code:

FOR DESIGN

```
module ff_ff (
  input clk,
  input rst,
  input A,
  input B,
  output reg X
);
  always @(posedge clk or posedge rst) begin
     case ({A, B})
       //Conditions
     endcase
   end
  end
endmodule

module counter_des (
  input clk,
  input rst,
```

College of Engineering – Department of Electrical Engineering

```
  output [3:0] X
);
  //Conditions
endmodule
```

FOR TEST BENCH

```
module tb_counter;
  reg clk, rst;
  wire [3:0] X;

  counter_des uut (.clk(clk), .rst(rst), .Q(Q));

  initial begin
    $dumpfile("counter.vcd"); $dumpvars;
    $display("table");

//rst

    // Generate clock pulses
//Displaying
      end
    end

    $finish;
  end
endmodule
```

Republic of the Philippines
BATANGAS STATE UNIVERSITY
The National Engineering University
Alangilan Campus
Golden Country Homes, Alangilan, Batangas City, Batangas, Philippines 4200
Tel Nos.: (+63 43) 425-0139 loc. 2121 / 2221
E-mail Address: ceafa@g.batstate-u.edu.ph | Website Address: http://www.batstate-u.edu.ph

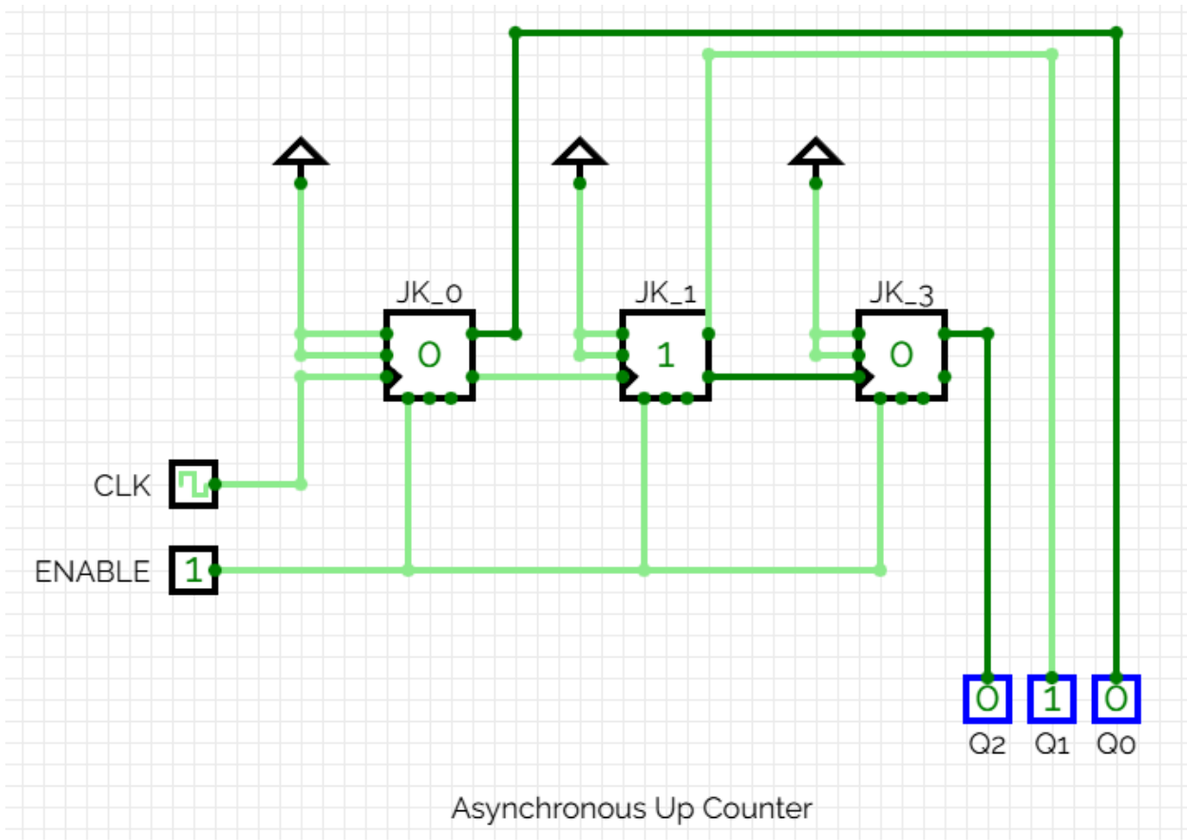
College of Engineering – Department of Electrical Engineering

----------*ASSESSMENT STARTS HERE*----------

ASSESSMENT 1: ***3-BIT UP ASYNCHRONOUS COUNTER***

A **3-bit UP asynchronous counter** that increments its count on each clock pulse, with outputs changing in a ripple fashion.

A. CIRCUIT DIAGRAM:



B. VERILOG CODE *(paste the actual code):*

Design Code:

```
module jk_ff (
  input clk,
  input rst,
  input J,
  input K,
  output reg Q
);
  always @(posedge clk or posedge rst) begin
    if (rst)
      Q <= 0;
```

College of Engineering – Department of Electrical Engineering

```
    else begin
      case ({J, K})
        2'b00: Q <= Q;      // No change
        2'b01: Q <= 0;      // Reset
        2'b10: Q <= 1;      // Set
        2'b11: Q <= ~Q;     // Toggle
      endcase
    end
  end
endmodule

module counter3_async_upripple (
  input clk,
  input rst,
  output [2:0] Q
);
  // JK inputs tied to 1 for toggle
  jk_ff jk0 (.clk(clk),   .rst(rst), .J(1'b1), .K(1'b1), .Q(Q[0]));
  jk_ff jk1 (.clk(~Q[0]), .rst(rst), .J(1'b1), .K(1'b1), .Q(Q[1]));
  jk_ff jk2 (.clk(~Q[1]), .rst(rst), .J(1'b1), .K(1'b1), .Q(Q[2]));
endmodule
```

Testbench Code:

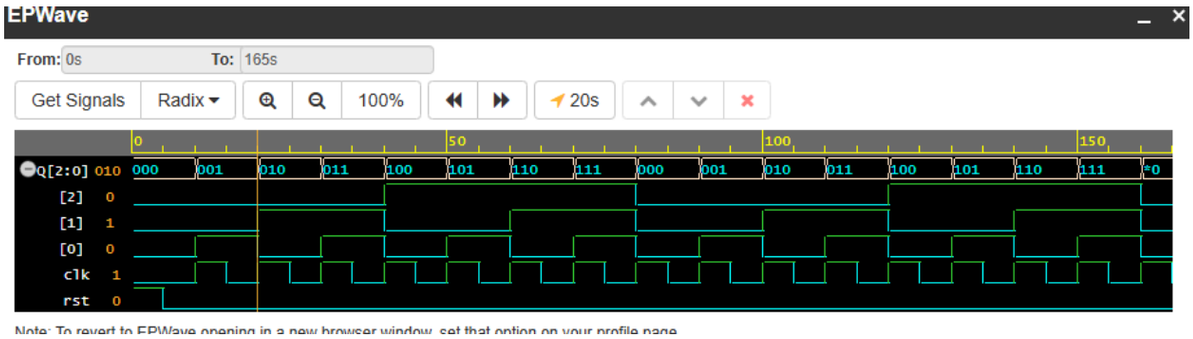
```
module tb_counter3_async_upripple;
  reg clk, rst;
  wire [2:0] Q;

  counter3_async_upripple uut (.clk(clk), .rst(rst), .Q(Q));

  initial begin
    $dumpfile("counter_3bit.vcd");
    $dumpvars;
    $display("Time | clk rst |  Q2 Q1 Q0");
    $display("-----+---------+-----------");

    clk = 0; rst = 1;
    #5 rst = 0; // release reset

    repeat (32) begin
      #5 clk = ~clk; // Toggle clock
      if (clk) begin
        $display("%4t |  %b   %b  |   %b  %b  %b",
          $time, clk, rst, Q[2], Q[1], Q[0]);
      end
    end

    $finish;
  end
```

```
endmodule
```

C. OUTPUT LOG (the output of your verilog code)

```
VCD warning: $dumpvars: Package ($unit) is not dumpable with VCD.
Time | clk rst |  Q2 Q1 Q0
-----+---------+-----------
  10 |  1   0  |   0  0  0
  20 |  1   0  |   0  0  1
  30 |  1   0  |   0  1  0
  40 |  1   0  |   0  1  1
  50 |  1   0  |   1  0  0
  60 |  1   0  |   1  0  1
  70 |  1   0  |   1  1  0
  80 |  1   0  |   1  1  1
```

D. EPWave



E. **Activity Conclusion/Takeaways** *(Discuss what happened to the circuit and the EPWave, and your conclusions)*

The EPWave output helps me visualize how the signals in the asynchronous counter behave over time. By looking at the waveform, I can see that the clock directly triggers only the first flip-flop, and the subsequent flip-flops toggle in sequence based on the output of the previous one. This ripple effect makes it clear why each bit changes at slightly different times, which is a defining characteristic of an asynchronous up-counter. The logs complement this observation by showing when each flip-flop updates, confirming the sequential toggling and ensuring that all parts of the circuit function as expected. The circuit diagram further explains the behavior seen in the waveform. It shows that only the least significant bit (LSB) flip-flop is connected to the main clock, while the other flip-flops are triggered by the outputs of the preceding flip-flops. This wiring structure directly causes the ripple timing observed in the EPWave output. By studying the waveform, logs, and circuit diagram together, I gain a complete understanding of the asynchronous counter's operation. The waveform illustrates the ripple effect in real time, the logs provide step-by-step verification of the toggling sequence, and the diagram reveals how the clock and flip-flops are connected to produce the counting pattern. This integrated analysis helps me clearly see how the asynchronous up-counter counts and why each bit toggles in the order it does.

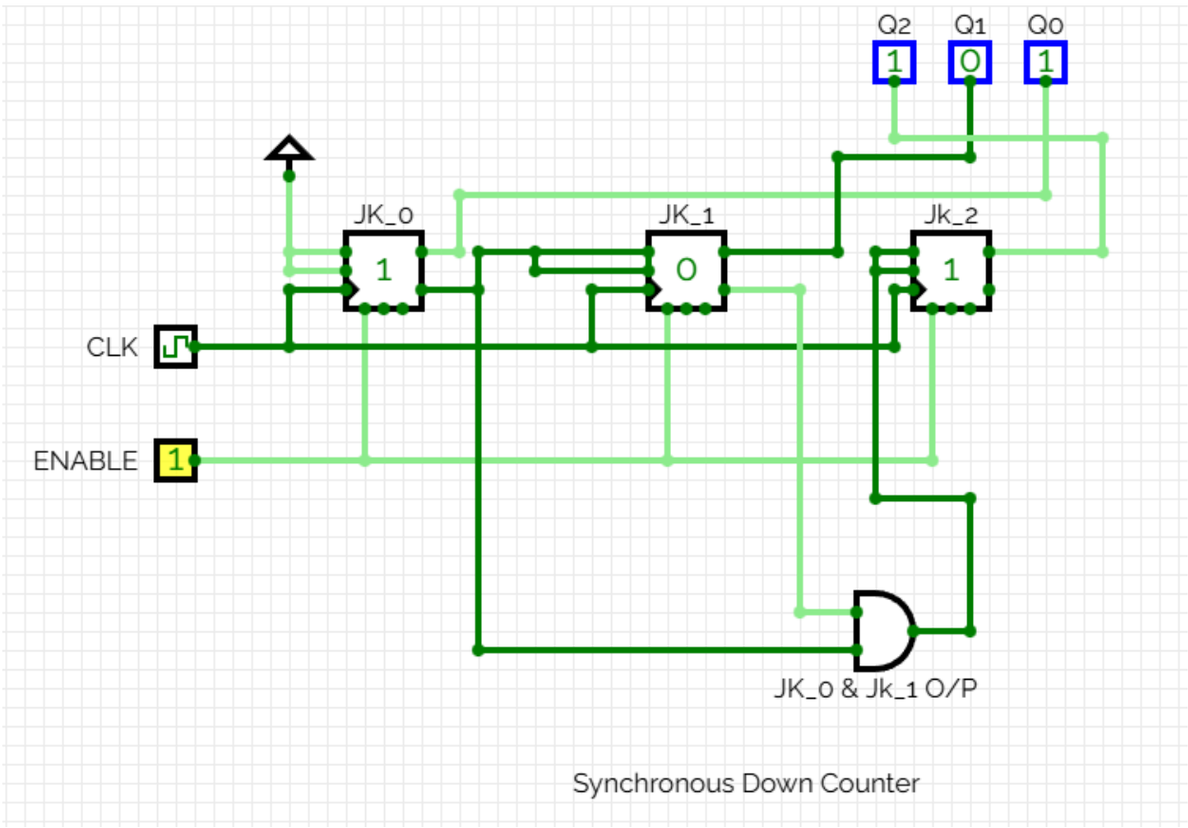
Republic of the Philippines
BATANGAS STATE UNIVERSITY
The National Engineering University
Alangilan Campus
Golden Country Homes, Alangilan, Batangas City, Batangas, Philippines 4200
Tel Nos.: (+63 43) 425-0139 loc. 2121 / 2221
E-mail Address: ceafa@g.batstate-u.edu.ph | Website Address: http://www.batstate-u.edu.ph

College of Engineering – Department of Electrical Engineering

ASSESSMENT 2: ***3-BIT DOWN SYNCHRONOUS COUNTER***

A **3-bit DOWN synchronous counter** that decrements its count on each clock pulse, with all flip-flops triggered simultaneously.

A. CIRCUIT DIAGRAM:



B. VERILOG CODE *(paste the actual code):*

Design Code:

```
module jk_ff (
  input clk, input rst,
  input J, input K,
  output reg Q
);
  always @(posedge clk or posedge rst) begin
    if (rst) Q <= 0;
    else begin
      case ({J,K})
        2'b00: Q <= Q;      // no change
        2'b01: Q <= 0;      // reset
        2'b10: Q <= 1;      // set
        2'b11: Q <= ~Q;     // toggle
```

College of Engineering – Department of Electrical Engineering

```
      endcase
    end
  end
endmodule

module sync3bit_downcnt (
  input clk, input rst,
  output [2:0] Q
);

  jk_ff ff0 (.clk(clk), .rst(rst), .J(1'b1), .K(1'b1), .Q(Q[0]));
  jk_ff ff1 (.clk(clk), .rst(rst), .J(~Q[0]), .K(~Q[0]), .Q(Q[1]));
  jk_ff ff2 (.clk(clk), .rst(rst), .J(~Q[0] & ~Q[1]), .K(~Q[0] & ~Q[1]), .Q(Q[2]));
endmodule
```

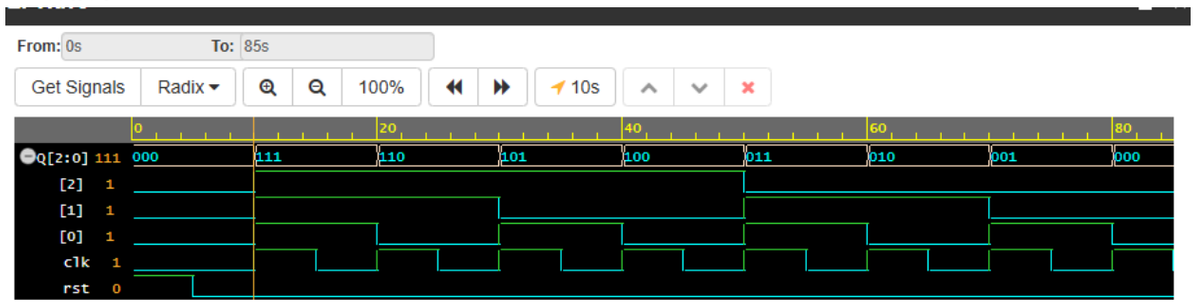
Testbench Code:

```
module tb_sync3bit_downcnt;
  reg clk, rst;
  wire [2:0] Q;

  sync3bit_downcnt uut (.clk(clk), .rst(rst), .Q(Q));

  initial begin
    $dumpfile("counter_3bit.vcd"); $dumpvars;

    $display("Time | clk rst | q2 q1 q0");
    $display("-----+---------+----------");

    clk = 0; rst = 1;
    #5 rst = 0; // release reset


    repeat (16) begin
      #5 clk = ~clk; // toggle clock
      if (clk) begin // only display at posedge

        $display("%4t |  %b   %b  |  %b  %b  %b",
          $time, clk, rst, Q[2], Q[1], Q[0]);
      end
    end

    $finish;
  end
endmodule
```

C. OUTPUT LOG (the output of your verilog code)

```
VCD warning: $dumpvars: Package ($unit) is not dumpable with VCD.
Time | clk rst | q2 q1 q0
-----+---------+----------
  10 |  1   0  |  0  0  0
  20 |  1   0  |  1  1  1
  30 |  1   0  |  1  1  0
  40 |  1   0  |  1  0  1
  50 |  1   0  |  1  0  0
  60 |  1   0  |  0  1  1
  70 |  1   0  |  0  1  0
  80 |  1   0  |  0  0  1
testbench.sv:27: $finish called at 85 (1s)
```

D. EPWave



Note: To revert to EPWave opening in a new browser window, set that option on your profile page.

E. Activity Conclusion/Takeaways *(Discuss what happened to the circuit and the EPWave, and your conclusions)*

The EPWave output helps me visualize how the signals in the 3-bit synchronous down counter behave over time. By looking at the waveform, I can see that the clock directly triggers all flip-flops simultaneously, rather than in a sequence. This synchronized behavior makes it clear why all bits change at the exact same instant, which is a defining characteristic of a synchronous down-counter. The logs complement this observation by showing the precise moment the value decrements, confirming the simultaneous updates and ensuring that all parts of the circuit function as expected. The circuit diagram further explains the behavior seen in the waveform. It shows that every flip-flop is connected to the main clock, while logic gates on the J and K inputs determine the toggling based on the state of the lower bits. This wiring structure directly produces the precise timing observed in the EPWave output. By studying the waveform, logs, and circuit diagram together, I gain a complete understanding of the synchronous counter's operation. The waveform illustrates the simultaneous transitions in real time, the logs provide step-by-step verification of the decrementing sequence, and the diagram reveals how the shared clock and combinational logic are connected to produce the counting pattern. This integrated analysis helps me clearly see how the synchronous down-counter counts and why all bits update in unison.

Republic of the Philippines
BATANGAS STATE UNIVERSITY
The National Engineering University
Alangilan Campus
Golden Country Homes, Alangilan, Batangas City, Batangas, Philippines 4200
Tel Nos.: (+63 43) 425-0139 loc. 2121 / 2221
E-mail Address: ceafa@g.batstate-u.edu.ph | Website Address: http://www.batstate-u.edu.ph

College of Engineering – Department of Electrical Engineering

CRITERIA	SCORE
Code	
The code is completely functional and responds correctly producing the correct outputs. **(5 Points)**	
The code is correct regarding syntax but required output is incorrect. **(3 Points)**	
The code has several syntax errors. **(1 Points)**	
Circuit Diagram	
Diagram is complete, clear, and accurately reflects counter logic and flip-flop connections. **(5 Points)**	
Diagram is mostly correct but has minor errors or missing labels. **(3 Points)**	
Diagram is unclear, incomplete, or incorrect. **(1 points)**	
Overall Work	
Student's work shows complete understanding of problem and all requirements. Able to explain entire program and design correctly as it is. **(10 Points)**	
Student's work shows understanding of problem and most requirements. Able to explain some program and design. **(7 Points)**	
Student's work shows slight understanding of problem and requirements. Unable to explain program and design. **(5 Points)**	
TOTAL SCORE:	
Percentage Score= (Total Score/20) * 100	
Other Comments:	