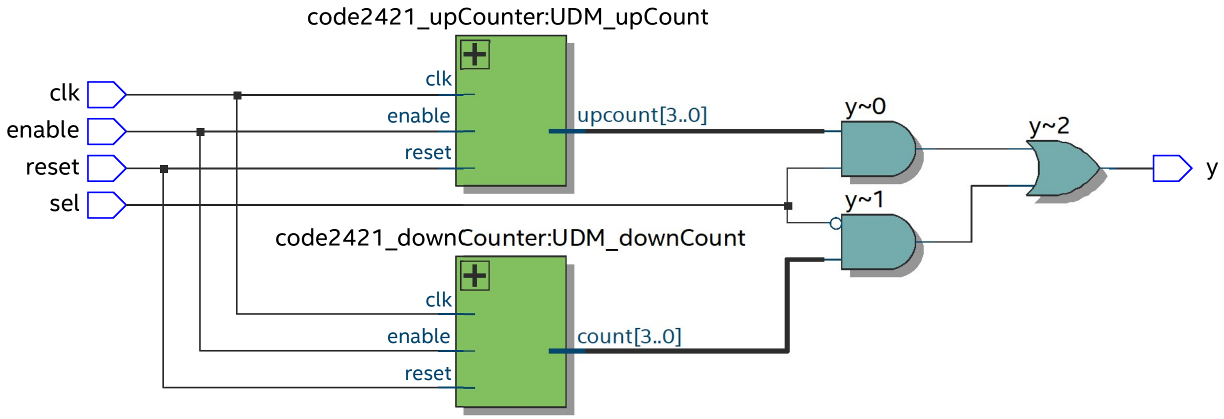
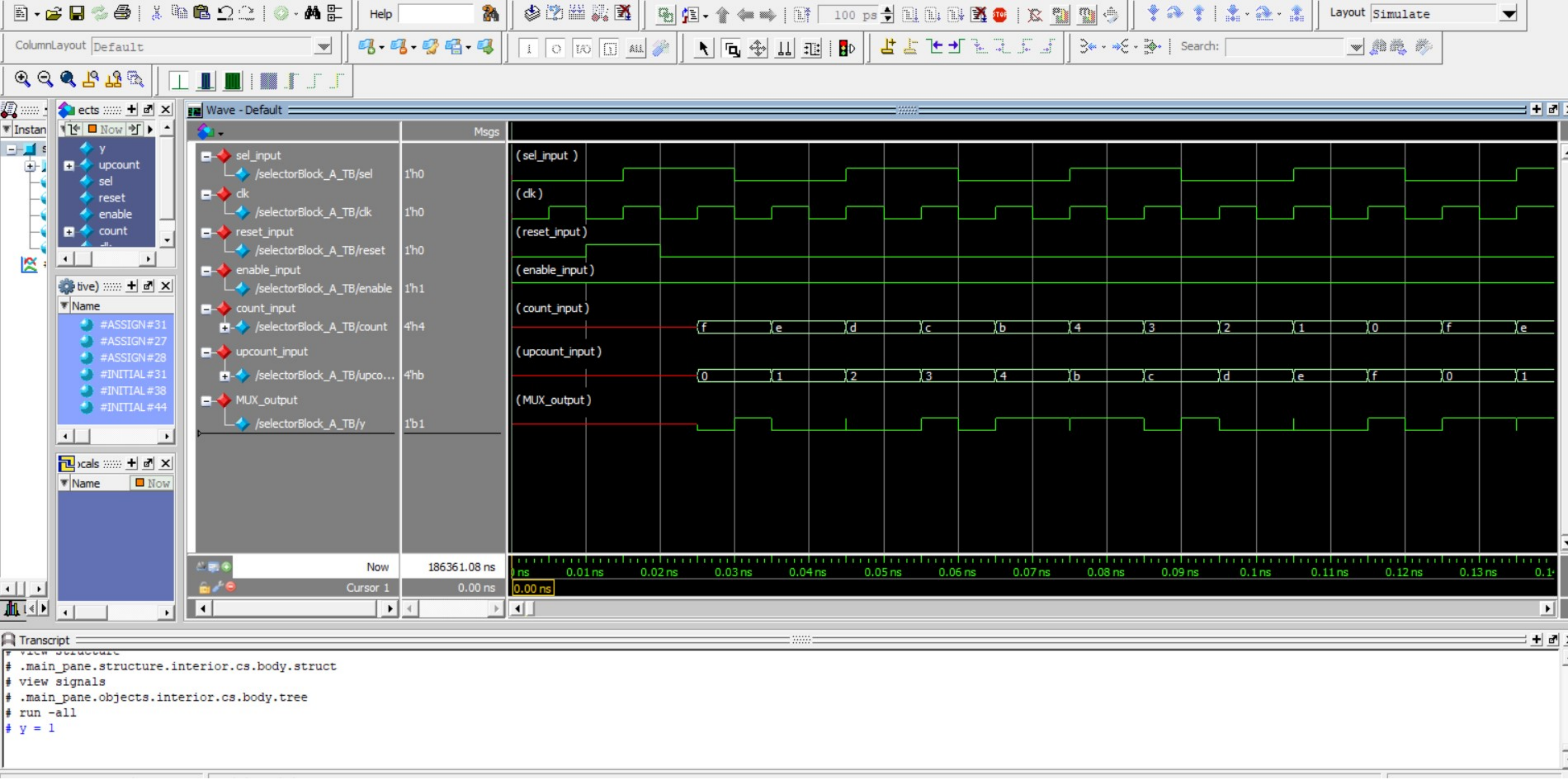
```
/*-----------------------------------------------------------------------------------
Name Lamin Jammeh
CLass: EE417 Summer 2024
Lesson 04 HW Question 4 Part 3
Group: Ron Kalin/ Lamin Jammeh
Project Description: This portion takes uses a 2-to-1 MUX to select between up and down
counter. when the sel bit is high the upcounter output is transfered to the next stage
else the downcounter output is transfered to the next state
-------------------------------------------------------------------------------------*/
//Step1: define the module and (port list)
   module selectorBlock_A (output wire y,
                           input sel,
                           input clk,
                           input reset,
                           input enable);

   wire [3:0] count;
   wire [3:0] upcount;

   //step 2 call the file for the code2421_downCounter
   code2421_downCounter UDM_downCount(
                                   .count(count),
                                   .clk(clk),
                                   .reset(reset),
                                   .enable(enable)
                                   );

   //call the file for the code2421_upCounter
   code2421_upCounter UDM_upCount(
                                   .upcount(upcount),
                                   .clk(clk),
                                   .reset(reset),
                                   .enable(enable)
                                   );
   //Instantiate the wire and logic gates
   //note when sel is high we count up and when low we count down
   //equation for output of mux Y = (In_1 & sel) | (In_2 & ~sel)
   assign y = (upcount & sel) | (count & ~sel);

   endmodule
```


code2421_upCounter:UDM_upCount
clk
enable
reset
sel
clk
enable
reset
upcount[3..0]
y~0
y~2
y
y~1
code2421_downCounter:UDM_downCount
clk
enable
reset
count[3..0]

```

module selectorBlock_A_TB ();

    // Inputs
    reg sel;
    reg clk;
    reg reset;
    reg enable;

    //declare internal probes as wire
    wire [3:0] count;
    wire [3:0] upcount;

    // Outputs
    wire y;

    // Instantiate the selectorBlock_A module
    selectorBlock_A UUT (
                         .y(y),
                         .sel(sel),
                         .clk(clk),
                         .reset(reset),
                         .enable(enable)
                        );

    //assign internal probe to count and upcount
   assign count = UUT.count;
   assign upcount = UUT.upcount;

    //step4 Initialize clock
initial
   begin
        clk = 0;
        forever #5 clk = ~clk; // 10ns period
    end

    //step5 initialize the sel bit for the test purpose
    initial
   begin
        sel = 0;
        forever #15 sel = ~sel; // 30ns period
    end
 // Initialize signals
initial
   begin
        clk = 0;
        reset = 0;
        enable = 1; // Enable the counter

        // Apply reset
        #10 reset = 1;
        #10 reset = 0;

        // Simulate some clock cycles
        #50;

        // Display output y
        $display("y = %b", y);
    end

endmodule

```



```
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# y = 1
```